

**UNIVERSIDADE ESTADUAL DE CAMPINAS**  
**FACULDADE DE ENGENHARIA ELÉTRICA**  
**E DE COMPUTAÇÃO**  
**DEPARTAMENTO DE COMUNICAÇÕES**

**ESTUDOS PARA OBTENÇÃO DO**  
**HARDWARE DE UM SISTEMA DE**  
**AQUISIÇÃO E VISUALIZAÇÃO DE**  
**VÍDEO DIGITAL**

Este exemplar corresponde à redação final da tese  
defendida por Ricardo Massahiro Nishihara  
e aprovada pela Comissão  
Julgadora em 30 / Out / 1996

Yuzo Iano  
Orientador

Autor : Ricardo Massahiro Nishihara  
Orientador: Prof. Dr. Yuzo Iano

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da  
Universidade Estadual de Campinas - UNICAMP- como parte dos requisitos exigidos para  
obtenção do título de Mestre em Engenharia Elétrica.

Outubro de 1996



UNIDADE	BC
N.º CHAMADA	UNICAMP
	N.º 633e
V.	Ex.
TOMBO BIC	30566
PROC.	281197
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00
DATA	24/03/97
N.º CPD	

CM.00098146-8

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

N633e Nishihara, Ricardo Massahiro  
Estudos para obtenção do hardware de um sistema de aquisição e visualização de vídeo digital / Ricardo Massahiro Nishihara.--Campinas, SP: [s.n.], 1996.

Orientador: Yuzo Yano.

Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Televisão digital. 2. Aquisição de dados. 3. Hardware. 4. Processamento de sinais - Técnicas digitais. I. Iano, Yuzo. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

# *Agradecimentos*

*Agradeço a Deus.*

*Agradeço à FEEC-UNICAMP pelos recursos fornecidos para elaboração deste trabalho.*

*Agradeço também ao CPqD-Telebrás pelo apoio e incentivo dados para a conclusão desta dissertação.*

*Agradeço também à FAPESP (Fundação de Apoio à Pesquisa do Estado de São Paulo), ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) e à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).*

*Agradeço a meu orientador Prof. Dr. Yuzo Iano pelo apoio, incentivo e sobretudo paciência.*

*Agradeço a meus pais que sem os quais não chegaria até aqui, a meus queridos irmãos Marcelo e Andréia pelo apoio e incentivo, e a todos aqueles que de alguma forma contribuíram para a realização deste trabalho.*

# SUMÁRIO

<b>CAPÍTULO 1: INTRODUÇÃO</b>	<b>1</b>
1.1 <i>Considerações Iniciais</i>	1
1.2 <i>Formatos de Vídeo Digital</i>	3
1.2.1 <i>A Recomendação CCIR 601</i>	4
1.3 <i>Problemas com as Soluções de Placa para PC</i>	8
1.4 <i>Solução via Projeto de um Sistema de Aquisição e Visualização Dedicado</i>	11
1.5 <i>Bibliografia</i>	11
<b>CAPÍTULO 2: VISÃO GERAL DE UM SISTEMA DE AQUISIÇÃO E VIZUALIZAÇÃO DEDICADO</b>	<b>12</b>
2.1 <i>Características Gerais do Sistema</i>	12
2.2 <i>Diagrama de Blocos Funcional</i>	13
2.3 <i>Barramento de E/S de Vídeo</i>	15
2.3.1 <i>Sinais do Barramento de E/S de Vídeo</i>	15
2.3.2 <i>Protocolos dos ciclos "Entrada de Vídeo" e "Saída de Vídeo"</i>	16
2.4 <i>Barramento VME</i>	17
2.4.1 <i>Um Breve Histórico</i>	17
2.4.2 <i>Características Mecânicas</i>	18
2.4.3 <i>Definições Lógicas do Barramento VME</i>	21
2.4.4 <i>Data Transfer Bus - DTB</i>	22
2.4.5 <i>Arbitration Bus</i>	23
2.4.6 <i>Priority Interrupt Bus</i>	30
2.4.7 <i>Utility Bus</i>	32
2.5 <i>Bibliografia</i>	32

## **CAPÍTULO 3: MÓDULO DIGITALIZADOR DE VÍDEO 33**

- 3.1 *Conversores A/D e D/A 33*
  - 3.1.1 *Conversores A/D do tipo paralelo ou "flash" 33*
  - 3.1.2 *Conversores D/A 35*
  - 3.1.3 *Parâmetros de Desempenho de Conversores A/D e D/A 37*
  - 3.1.4 *Conversores A/D e D/A Comerciais 40*
- 3.2 *Filtros Anti-Aliasing de Pré e Pós Processamento 43*
- 3.3 *PLL (Phased Locked Loop) e Sintetizadores de Freqüências 45*
  - 3.3.1 *Princípio de Operação de um Sistema PLL 45*
  - 3.3.2 *O transiente do Processo de Captura 46*
  - 3.3.3 *O Modelo Linear do PLL 48*
  - 3.3.4 *Efeitos do Filtro Passa-Baixas 49*
  - 3.3.5 *O CI PLL NE564 54*
  - 3.3.6 *Síntese de Freqüência usando PLL 55*
- 3.4 *Amplificadores e Processadores de Sincronismo de Vídeo 55*
  - 3.4.1 *O Separador de Sincronismo LM1881 55*
  - 3.4.2 *O Gerador de Sincronismo LM1882 58*
  - 3.4.3 *MC14577 - Amplificador de Vídeo 60*
- 3.5 *Projeto do Digitalizador de Vídeo 60*
  - 3.5.1 *Diagrama de Blocos do Subsistema Digitalizador de Vídeo 60*
  - 3.5.2 *Unidade de Entrada de Vídeo 61*
  - 3.5.3 *Unidade de Saída de Vídeo 61*
  - 3.5.4 *Processadores de Sincronismo/Sintetizadores de Freqüência 64*
- 3.6 *Bibliografia 66*

## **CAPÍTULO 4: ARMAZENADOR DE QUADROS BASEADO EM MEMÓRIA SEMICONDUTORA 67**

- 4.1 *Memórias Dinâmicas - DRAM 68*
  - 4.1.1 *Ciclo de Leitura Convencional 69*
  - 4.1.2 *Ciclo de Escrita Convencional 74*
  - 4.1.3 *Modo de Operação Página 75*

- 4.1.4 *Modo de Operação Nibble* 75
- 4.1.5 *Ciclos de Reforço* 78
- 4.1.6 *Características de chip's e módulos DRAM comerciais* 78
  
- 4.2 *Memórias Estáticas - SRAM* 81
  - 4.2.1 *Características de uma memória estática típica* 81
  
- 4.3 *VideoRAM* 84
  - 4.3.1 *Operação da Porta SAM* 85
  - 4.3.2 *Operação de Transferência entre a SAM e a DRAM Interna* 87
  
- 4.4 *Field Memory* 91
  - 4.4.1 *Operação de Escrita* 92
  - 4.4.2 *Operação de Leitura* 93
  
- 4.5 *Questões a serem respondidas quando se projeta um armazenador de quadros* 94
  - 4.5.1 *Quantidade de Memória Requerida* 95
  - 4.5.2 *Escolha do Dispositivo de Memória* 95
  - 4.5.3 *Arranjos de Multiplexagem/Demultiplexagem* 96
  - 4.5.4 *Características Mecânicas do Módulo Armazenador* 96
  
- 4.6 *Organização do Armazenador de Quadros* 98
  
- 4.7 *Bibliografia* 109

## **CAPÍTULO 5: INTERFACE DE COMUNICAÇÃO COM O COMPUTADOR HOSPEDEIRO** 110

- 5.1 *Por que usar a interface SCSI* 110
  
- 5.2 *A Interface SCSI* 110
  - 5.2.1 *Características Mecânicas* 111
  - 5.2.2 *Especificações Elétricas* 116
  - 5.2.3 *Definições Lógicas Básicas* 120
  - 5.2.4 *Sinais SCSI* 123
  - 5.2.5 *Fases SCSI* 126
  - 5.2.6 *Condições do barramento SCSI* 134
  - 5.2.6 *Ponteiros SCSI* 135
  - 5.2.7 *Sistema de Mensagens* 136

5.2.8	<i>Comandos SCSI</i>	140
5.2.9	<i>Exemplo de Execução de um Comando SCSI</i>	142
5.3	<i>Considerações a cerca da implementação da interface SCSI</i>	144
5.3.1	<i>Módulo MVME162 - Controlador Master VME com Interface SCSI</i>	145
5.4	<i>Bibliografia</i>	146

## **CAPÍTULO 6: CONCLUSÕES** 147

### **APÊNDICES**

- A: Exemplo de implementação SCSI usando o PI/T 68230**
- B: Chip Controlador SCSI - Am53C94/Am53C96**

# CAPÍTULO 1

## INTRODUÇÃO

### 1.1 Considerações Iniciais

Um grupo de pesquisa com interesses em compressão digital de vídeo certamente se deparará com o seguinte problema: como conseguir seqüências de quadros de vídeo digitalizado em condições apropriadas à realização de experimentos envolvendo simulação em computador de protótipos-softwares de codificadores de vídeo ? Por seqüências em condições apropriadas entenda-se:

1. A seqüência deve ter comprimento suficiente para que seja possível a realização de experimentos envolvendo movimento.
2. A seqüência deve ser digitalizada em formatos de entrada que possam ser facilmente adaptados àqueles que costumam ser utilizados por codecs já padronizados como o MPEG1 e MPEG2.
3. E o que é mais importante a seqüência não pode sofrer nenhum tipo de pré-processamento envolvendo compressão.

Um outra questão a ser abordada é a que se refere à validação de algoritmos e sistemas de codificação digital de vídeo. De um modo geral, os sistemas de vídeo costumam ser avaliados tanto em termos objetivos quanto em termos subjetivos. Avaliação objetiva é aquela feita através da medição de parâmetros extraídos da forma de onda de saída do sistema ou do sinal obtido a partir da diferença entre a sua entrada e a sua saída. A avaliação subjetiva, por sua vez, baseia-se em experimentos de exibição de um conjunto de seqüências de imagens processadas pelo sistema sob teste a um certo número de observadores, os quais devem então fornecer notas subjetivas a cerca da qualidade da imagem processada pelo sistema.

Os métodos assim como as condições logísticas e ambientais sob as quais devem ser realizados experimentos de avaliação subjetiva encontram-se especificados na recomendação R.500-5 do ITU-R. Nesta recomendação são considerados de maneira relativamente bem detalhada metodologias de teste subjetivos tanto de sistemas analógicos quanto de sistemas codificadores digitais. A estrutura dos ensaios para os casos analógico e digital é basicamente a mesma, sendo que um critério mais cuidadoso deve ser utilizado na escolha das seqüências de teste usadas nas avaliações dos sistemas digitais.

Em se tratando de avaliação objetiva, a situação é bem diferente, visto que grande parte do conhecimento adquirido na caracterização dos sistemas analógicos não pode ser diretamente aplicado para o caso digital. É sabido que a caracterização ou avaliação objetiva dos sistemas de vídeo analógicos pode ser feita através do uso de sinais de teste e de medidas de desempenho extraídas a partir da forma de onda do sinal de saída do sistema. Exemplos de medidas como estas são o ganho e a fase diferenciais, o fator K, ruído sobre o

elemento barra do padrão de teste, etc. De um modo geral estas medidas procuram levantar características, como resposta em frequência, introdução de ruído, e não-linearidades no processamento do sinal que se manifestarão como interferências entre as componentes de luminância e crominância do sinal de vídeo. Estes sinais de teste e os parâmetros a eles associados, no caso dos sistemas analógicos, são bem definidos (vide Rec 473 e 567 do ITU-R) e já bem conhecidos, existindo inclusive dados que ilustram a relação existente entre alguns destes parâmetros e resultados obtidos por meio de avaliações subjetivos (vide Rec. 654). Quando nos referimos aos sistemas de codificação digital de vídeo, entretanto, dificuldades passam a surgir, dado a não-existência de uma metodologia de avaliação objetiva padronizada eficiente e bem definida que permita validações significativas do sistema codificador digital de vídeo, sem que seja necessário uso intensivo de avaliações subjetivas. Um das razões para a dificuldade em se definir tal metodologia, reside no fato de que a abordagem via sinais e padrões de teste estáticos muito utilizada nos sistemas analógicos não é aplicável aos codecs digitais de vídeo mais sofisticados. Um codec digital é capaz de alterar o seu modo de operação de acordo com o conteúdo da imagem de entrada, alocando, por exemplo, para uma área com mais detalhes e menos bits para uma outra região com menos detalhes. Assim se utilizarmos um sinal teste estático estaremos caracterizando apenas um dos possíveis modos de operação do codec. Portanto, a avaliação objetiva de codecs deve ser feita utilizando-se um conjunto de cenas naturais. Uma medida objetiva para avaliar a qualidade da imagem de protótipos-software de codecs de vídeo, que costuma ser bastante utilizada por desenvolvedores é a relação sinal-ruído de pico,  $SNR_{pico}$ , ou o erro quadrático médio,  $EQM$ . Note que esta medida não se correlaciona bem com a percepção visual humana. Por exemplo, se uma imagem for altamente degradada em uma região espacialmente pequena mas bastante notada por um observador humano (a face de um apresentador de TV por exemplo), ela pode apresentar um valor  $SNR_{pico}$  baixo e contraditoriamente uma qualidade subjetiva baixa. Ou seja a avaliação objetivas de codecs digitais é ainda uma em aberto.

Pelas razões expostas acima, conclui-se que a validação de um codificador digital de vídeo, seja ele um protótipo de software ou de hardware, requer necessariamente o uso de experimentos de avaliação subjetiva bem elaborados. Se consideramos que os sistemas gráficos dos PC's atuais apresentam limitações que impedem a exibição de cenas foto realísticas nas condições necessárias à realização deste tipo de experimentos, temos então mais um problema a ser abordado.

Neste trabalho procuramos encontrar soluções em hardware para as duas questões levantadas acima, ou seja aquisição e visualização de vídeo em condições particulares à atividade de pesquisa em algoritmos de compressão digital de vídeo. Em princípio poderíamos viabilizar o hardware para um sistema de aquisição e visualização como este, através de soluções de placa para PC, ou através do projeto de um sistema aquisição dedicado. As placas captura de vídeo para PC entretanto, como veremos mais adiante, apresentam limitações que dificultam e em alguns casos inviabilizam esta solução para a aplicação em questão. Tal situação torna a opção pelo projeto de um sistema dedicado às necessidades particulares de nosso grupo de pesquisa uma alternativa bastante atraente. Assim sendo neste trabalho fizemos a opção por esta última alternativa, e apresentaremos uma proposta para um sistema dedicado para aquisição e visualização de vídeo digital que dentro do possível atenda as necessidades acima consideradas.

# 1.2 Formatos de Vídeo Digital

Conforme já mencionado, para a aplicação em questão, o sinal de vídeo deve ser capturado em formatos compatíveis ou conversíveis àqueles correntemente utilizados como sinais fonte por codecs de vídeo já padronizados. A tabela 1.1 a seguir, lista alguns destes codecs com os seus respectivos formatos fonte.

TABELA 1.1 - Alguns formatos fontes usados por codecs de vídeo

CODECS DE VÍDEO (APLICAÇÃO)	FORMATO DO SINAL FONTE
MPEG1 (APLICAÇÕES MULTIMÍDIA)	SIF
H261-H263 (VÍDEO CONFERÊNCIA)	CIF
CCIR. 721 (TV DIGITAL)	CCIR 601: 4:2:2
MPEG2 (TV DIGITAL/HDTV)	CCIR 601: 4:2:0, 4:2:2 e 4:4:4

Note que os codecs aplicados a vídeo ou televisão digital utilizam como sinal-fonte o formato definido pela recomendação CCIR 601.

Além disso os formatos SIF e CIF utilizados pelo codec multimídia MPEG1 e pelo codec videoconferência H 261, podem ser obtidos facilmente a partir do membro 4:2:2 da família de formatos R. 601, através de uma dizimação de ordem 2, conforme mostram as figuras 1.1 e 1.2. Portanto, os formatos mais adequados para a aquisição de vídeo destinados a nossa aplicação são aqueles definidos pela recomendação CCIR 601.

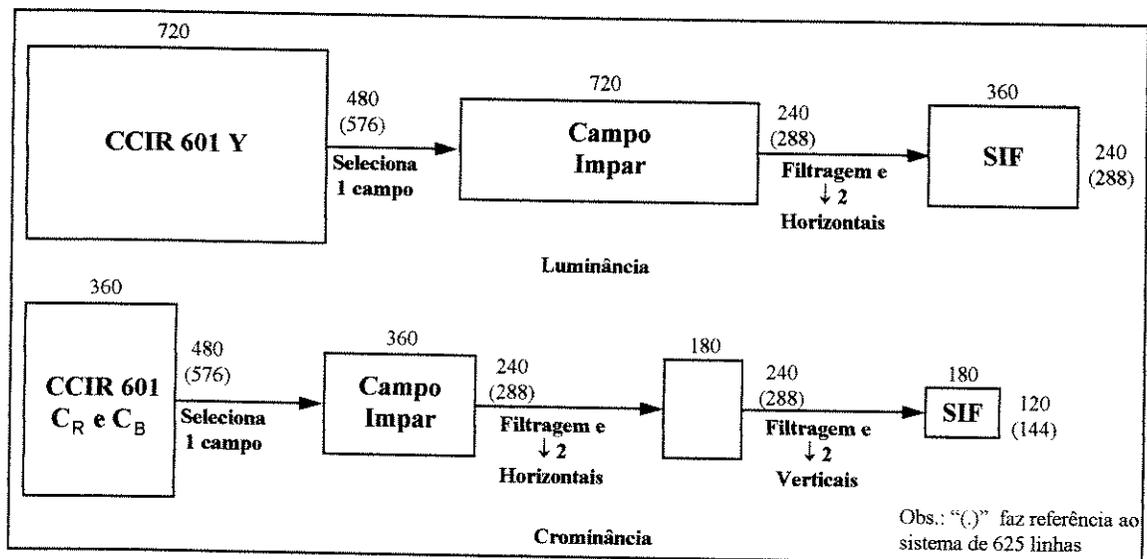


Figura 1.1 Diagrama simplificado de um conversor 601 → SIF.

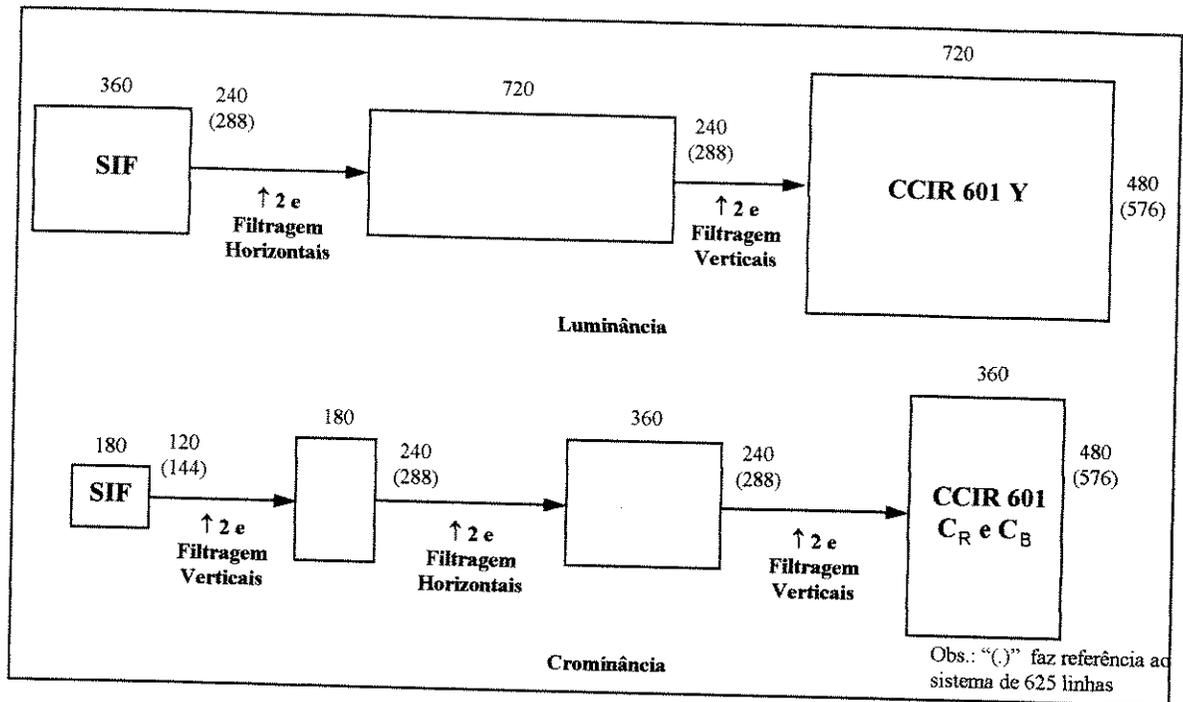


Figura 1.2 Diagrama simplificado de um conversor SIF → 601.

## 1.2.1 A Recomendação CCIR 601

O aumento no uso de sinais de vídeo no formato em componentes em equipamentos digitais, juntamente com a possibilidade de obter-se uma área substancial de compatibilidade entre os sistemas de 525 e 625 linhas, levaram o CCIR a considerar a possibilidade de estabelecer um padrão para a digitalização de vídeo na forma de componentes. Esta padronização veio através da Rec. 601 proposta 1981, que realiza uma especificação dos diversos parâmetros envolvidos no processo de digitalização PCM dos sinais de vídeo em componentes.

A seguir faz-se a citação das considerações iniciais apresentadas no texto da Recomendação 601[1]:

“ O CCIR,

*Considerando*

- que são claras as vantagens para difusores de televisão e produtores de programas em que se tenha padrões digitais de estúdio com um grande número de parâmetros significativos em comum para sistemas de 525-linhas e 625-linhas;*
- que uma abordagem digital que seja compatível a nível mundial permite o desenvolvimento de equipamentos com muitas características em comum, possibilita economia na operação, e facilita a troca internacional de programas;*
- que uma família extensível de padrões de codificação digital compatíveis é desejável. Membros desta família poderiam corresponder a diferentes níveis de qualidade, o que facilitaria a integração de processamento adicional às atuais técnicas de produção, e proveria futuras necessidades;*
- que um sistema baseado em codificação de componentes é capaz de atender alguns e talvez todos os objetivos desejados;*

- e) *que o co-posicionamento de amostras representando sinais de luminância e diferença de cor (ou, se utilizado, os sinais vermelho, verde e azul) facilita o processamento dos sinais em componentes digitais requerido pelas atuais técnicas de produção*

*Unanimemente recomenda*

*que o seguinte seja usado como base para padrões de codificação digital para estúdios de televisão em países usando o sistema de 525-linhas assim como aqueles usando o sistema de 625-linhas: ....”*

A consideração a) aponta para a questão da compatibilidade dos padrões com relação a sua aplicação aos sistemas de 525 e 625 linhas. A chave para esta compatibilidade reside no fato do tempo de varredura horizontal dos dois sistemas ser praticamente o mesmo. Assim é possível escolhendo-se a mesma frequência de amostragem para os dois sistemas de varredura ter o mesmo número de amostras ativas por linha. Em e) é feita a referência ao fato de escolher-se valores para as frequências de amostragem de luminância e crominância que sejam múltiplos inteiros de 2.25 MHz (mínimo múltiplo comum das frequências de linha nos sistemas 525/60 e 625/50). Desta forma garante-se um número inteiro de amostras por linha, o que permite a obtenção amostras de luminância e crominância co-situadas formando um mosaico ortogonal temporalmente estático para ambos os sistemas.

Após muitos testes e investigações comparando-se diferentes frequências de amostragem para luminância e crominância escolheu-se uma estrutura de amostragem usando-se 13.5 MHz para captura de Y, 6.75 MHz para  $C_B$ <sup>1</sup>, e 6.75 MHz para  $C_R$ <sup>2</sup>. Esta razão de frequências é normalmente designada por 4:2:2<sup>3</sup> o que faz referência a uma família extensível de padrões. Conforme destaca a consideração c), sempre foi objetivo da Rec. 601 que esta família acomodasse diferentes requisitos de qualidade centradas em torno do membro 4:2:2 como o padrão básico para estúdio. Assim, por exemplo, em equipamentos usados para armazenamento de material de notícias (*gathering news*) em que existe um compromisso entre qualidade e portabilidade, o membro 2:1:1 da família com frequências de amostragem de 6.75 ; 3.75 e 3.75 MHz poderia ser utilizado. Para captura em componentes RGB poderia ser utilizado o membro 4:4:4, e para aplicações em HDTV poderia se utilizar um múltiplo do membro 4:2:2.

As tabelas 1.1 e 1.2 a seguir mostram parâmetros de codificação do membro 4:2:2 e 4:4:4 respectivamente. As características mais importantes a notar-se, além das frequências de amostragem utilizadas, são a escolha do PCM de 8bits por amostra para cada um dos três sinais e do uso de 720 amostras para Y e 360 (membro 4:2:2) ou 720 amostras (membro 4:4:4) para os sinais diferença de cor, na parte ativa da linha. A estrutura de amostragem ortogonal é mostrado na figura 1.3. Na figura 1.4 é mostrada a relação entre as amostras e a sinalização de sincronismo horizontal de vídeo, ela ilustra como os tempos de sincronismo para os sistemas de 625 e 525 linhas foram ajustados para que se obtivesse uma estrutura de amostragem comum aos dois sistemas durante a linha ativa.

<sup>1</sup> Sinal B - Y a menos da constante 0.713, ou seja,  $C_B = 0.713(B - Y)$

<sup>2</sup> Sinal R - Y a menos da constante 0.564, ou seja,  $C_R = 0.713(R - Y)$

<sup>3</sup> Com o “4” sendo associado à frequência de 13.5 Mhz

TABELA 1.1 membro 4:2:2 da família CCIR 601

Parâmetros	Sistemas de 525-linhas, 60 campos/s	Sistemas de 625-linhas, 50 campos/s
1. Sinais Codificados: $Y, C_R, C_B$	Sinais obtidos a partir dos sinais pré corrigidos gamma: $E_Y, E_R - E_Y, e E_B - E_Y$	
2. Número de amostras por linha total: - sinal de luminância (Y) - sinais diferença de cor ( $C_R, C_B$ )	858 429	864 432
3. Estrutura de amostragem	Ortogonal e repetida ao longo das linhas, campos e quadro. Amostras de $C_R$ e $C_B$ co-posicionadas com as amostras ímpares (primeira, terceira, quinta, etc.) de Y em cada linha	
4. Freqüência de Amostragem: - sinal de luminância - sinal diferença de cor	13.5 MHz 6.75 MHz	
5. Forma de codificação	PCM com quantização uniforme, 8 (opcionalmente 10) bits por amostra, para o sinal de luminância e cada sinal diferença de cor.	
6. Número de amostras por linha ativa: - sinal de luminância (Y) - sinais diferença de cor ( $C_R, C_B$ )	720 360	
7. Relação de temporização horizontal entre os caso analógico e digital: - do final da linha ativa digital à $0_H$ .	16 períodos de relógio de luminância	12 períodos de relógio de luminância
8. Níveis de quantização: - escala - sinal de luminância - sinais diferença de cor	<ul style="list-style-type: none"> <li>◆ 0 a 255</li> <li>◆ 220 níveis de quantização com o nível preto associado ao 16 e pico de branco ao 235.</li> <li>◆ 225 níveis de quantização na parte central da escala de quantização com o sinal zero correspondendo ao nível 128.</li> </ul>	
9. Palavras código reservadas	Palavras código correspondendo aos níveis de 0 a 255 são usadas exclusivamente para sincronização. Os níveis 1 a 154 estão disponíveis para vídeo.	

TABELA 1.2- Parâmetros do membro 4:4:4 da família CCIR 601

Parâmetros	Sistemas de 525-linhas, 60 campos/s	Sistemas de 625-linhas, 50 campos/s
1. Sinais Codificados: Y, C <sub>R</sub> , C <sub>B</sub> ou R, G, B	Sinais obtidos a partir dos sinais pré corrigidos gamma: E <sub>Y</sub> , E <sub>R</sub> - E <sub>Y</sub> , e E <sub>B</sub> - E <sub>Y</sub> ou E <sub>R</sub> , E <sub>G</sub> , e E <sub>B</sub> .	
2. Número de amostras por linha total: - sinal de luminância (Y)	858	864
3. Estrutura de amostragem	Ortogonal e repetida ao longo das linhas, campos e quadro. As três estruturas de amostragem devem ser coincidentes entre si e coincidentes também com a estrutura de amostragem da luminância do membro 4:4:2.	
4. Frequência de Amostragem: - sinal de luminância	13.5 MHz	
5. Forma de codificação	PCM com quantização uniforme, 8 (opcionalmente 10) bits por amostra	
6. Número de amostras por linha ativa: - sinal de luminância (Y)	720	
7. Níveis de quantização: - escala - sinal de luminância ou sinais R, G, B - sinais diferença de cor	<ul style="list-style-type: none"> <li>◆ 0 a 255</li> <li>◆ 220 níveis de quantização com o nível preto associado ao 16 e pico de branco ao 235.</li> <li>◆ 225 níveis de quantização na parte central da escala de quantização com o sinal zero correspondendo ao nível 128.</li> </ul>	

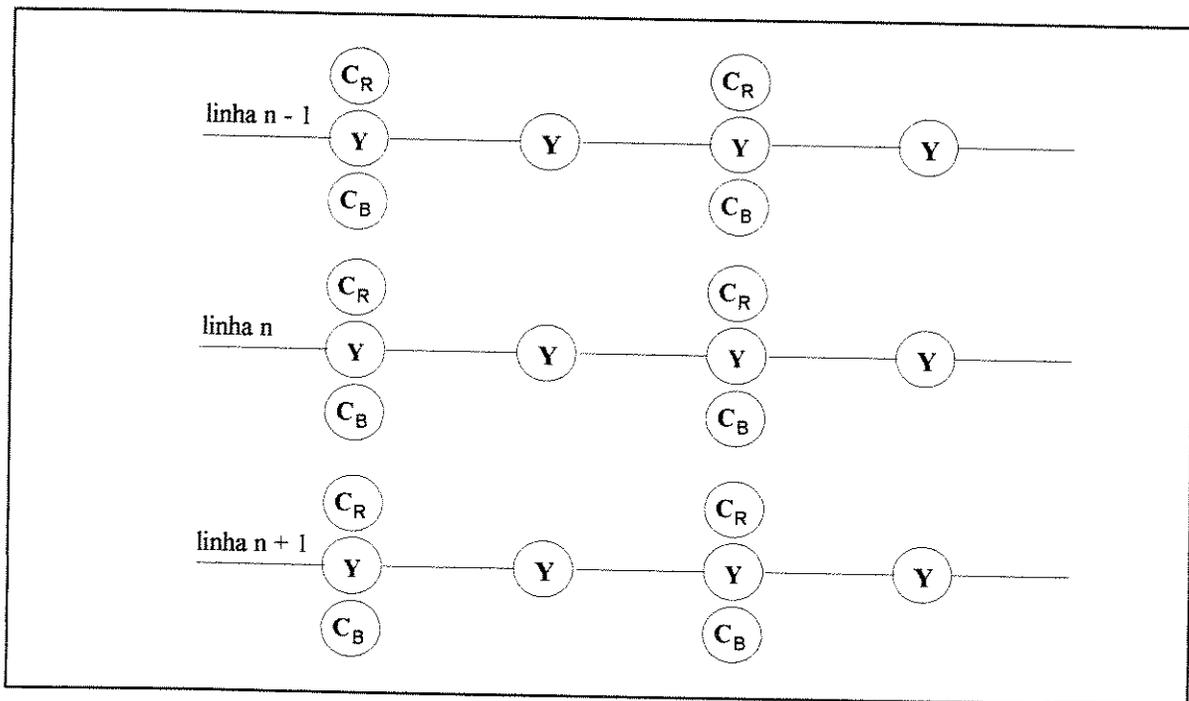


Figura 1.3 Estrutura de amostragem ortogonal para o membro 4:2:2

Sistemas 525-linhas 60 campos/s	122T	720T	16T	
	$0_H$	Período de linha ativa digital	$0_H$	Próxima linha
Sistemas 625-linhas 50 campos/s	132T	720T	12T	

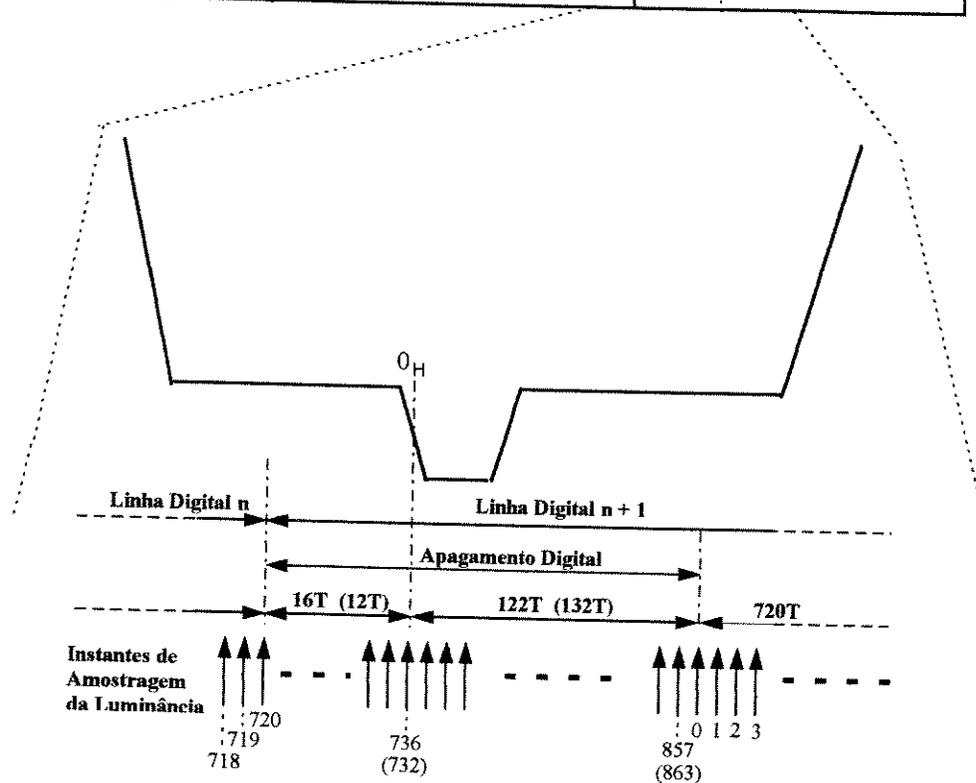


Figura 1. 4 Relação entre as amostras de vídeo sincronização de linha analógica

### 1.3 Problemas com as Soluções de Placas para PC

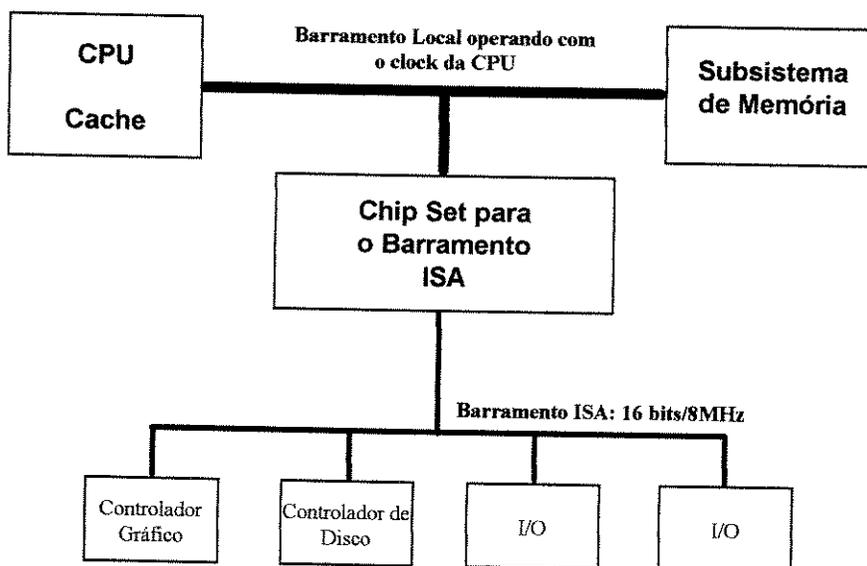
Embora existam no mercado inúmeras soluções de placa para PC que realizem a captura de quadros de vídeo, estas em geral não atendem os requisitos especificados no item 1.1, em especial a questão da não utilização de compressão durante o processo de captura. A maioria destas placas, principalmente as de custo mais baixo, são voltadas para o desenvolvimento de aplicações multimídia e ou de edição digital de vídeo com qualidade semi-profissional. Assim, o uso de compressão para aplicações como estas, desde que seja utilizado um esquema de compressão eficiente e adequadamente configurado, não representa problema algum, mas sim, uma solução que permite ao usuário conseguir capturar um número maior de quadros com menos recursos de hardware (como memória, disco, etc). Fora esta questão da otimização de recursos, há ainda uma outra razão que talvez explique o fato das placas de captura de quadros de vídeo que não fazem uso de compressão (também conhecidas como frame-grabbers) estarem fora do fluxo principal de mercado. A arquitetura tradicional dos PC's AT baseada no barramento ISA, para a qual é projetada a maioria das placas de captura de vídeo de baixo custo, não prove

recursos que permitam transferências eficientes de fluxos de vídeo digitalizado não-comprimido entre os periféricos ligados aos slots do barramento ISA.

O barramento ISA (“Industry Standart Architecture”) nada mais é que a padronização do barramento do PC AT. Originalmente, este barramento estava sincronizado com a CPU, mas, quando o relógio da CPU excedeu os 10 MHz, os periféricos (que então operavam a até 8 MHz) não puderam acompanhá-la. Por isso, o barramento ISA permaneceu operando a 8 MHz, usando um barramento de dados de 16 bits, o que dá uma velocidade máxima de comunicação de dados teórica de 16 MB/s. Na prática devido a questões de overhead esta velocidade deve estar em torno de 2.5 MB/s [2], portanto longe de atender às altas taxas de dados requeridas para a transferência de vídeo digitalizado sem compressão conforme mostra a tabela 1.3. Na figura 1.4 mostra o diagrama de blocos simplificado de uma configuração ISA típica.

**TABELA 1.3 - Taxas de Dados de Pico para os Esquemas da CCIR 601**

	Número de Bytes/Quadro	Taxa de Quadros	Taxa de Amostragem	Taxa de dados de pico (bytes/seg)
CCIR 601 422 525 linhas	691200	29.97	13.5 MHz	27.0 MBps
CCIR 601 422 625 linhas	829440	25.00	13.5 MHz	27.0 MBps
CCIR 601 444 525 linhas	1036800	29.97	13.5 MHz	40.5 MBps
CCIR 601 444 625 linhas	1244160	25.00	13.5 MHz	40.5 MBps



**Figura 1.4** Configuração típica de um sistema PC com barramento ISA

Portanto, a utilização de compressão nas placas de captura de vídeo para PC, que visam atender o mercado de aplicações em edição semi-profissional e multimídia, constitui-se também em um mecanismo para se vencer o gargalo existente no barramento de expansão do PC. Na figura 1.5, ilustramos este mecanismo,

mostrando um diagrama de blocos simplificado de uma placa típica de captura de vídeo para aplicações multimídia [3] [4]. Diferentes esquemas tem sido utilizados na implementação do módulo compressão de vídeo destas placas, citando-se como exemplo, Motion JPEG (adaptação do esquema JPEG para imagens em movimento implementada através de chips dedicados como CL550 da C-Cube Microsystems), Indeo (esquema de codificação de propriedade da Intel, implementada no chip Intel i750), e MPEG. Um grande número de exemplos de placas deste tipo são discutidos nas referências [5] e [6].

Um outro tipo de solução para realizar captura de vídeo, que poderia ser utilizada, são as placas dedicadas a processamento digital de imagem. Estas placas são baseadas em DSP's e utilizam uma maneira diferente para enfrentar o problema da baixa velocidade do barramento expansão do hospedeiro: possuem localmente (ou seja, na própria placa de aquisição) a quantidade de memória necessária para armazenar uma seqüência de vídeo de inteira ainda que de comprimento limitado. A seqüência poderia ser processada localmente usando-se o DSP, ou transferida para o hospedeiro através do barramento de expansão. Como exemplo deste tipo de placa cita-se um produto da LSI (Loughborough Sound Images): a placa MDC40IC TIM-40 Image Processing Module baseada que tem memória local suficiente para armazenar um seqüência de 16 quadros com resolução de até 512x512. Embora não utilize compressão, esta placa não suporta formatos de vídeo digital como o CCIR 601. Um outro problema é o comprimento relativamente pequeno das seqüências capturadas e impossibilidade de expandir a capacidade de armazenamento deste sistema. Além estas placas por não possuírem o volume de escala das placas concebidas para aplicações multimídia possuem também um custo relativamente elevado.

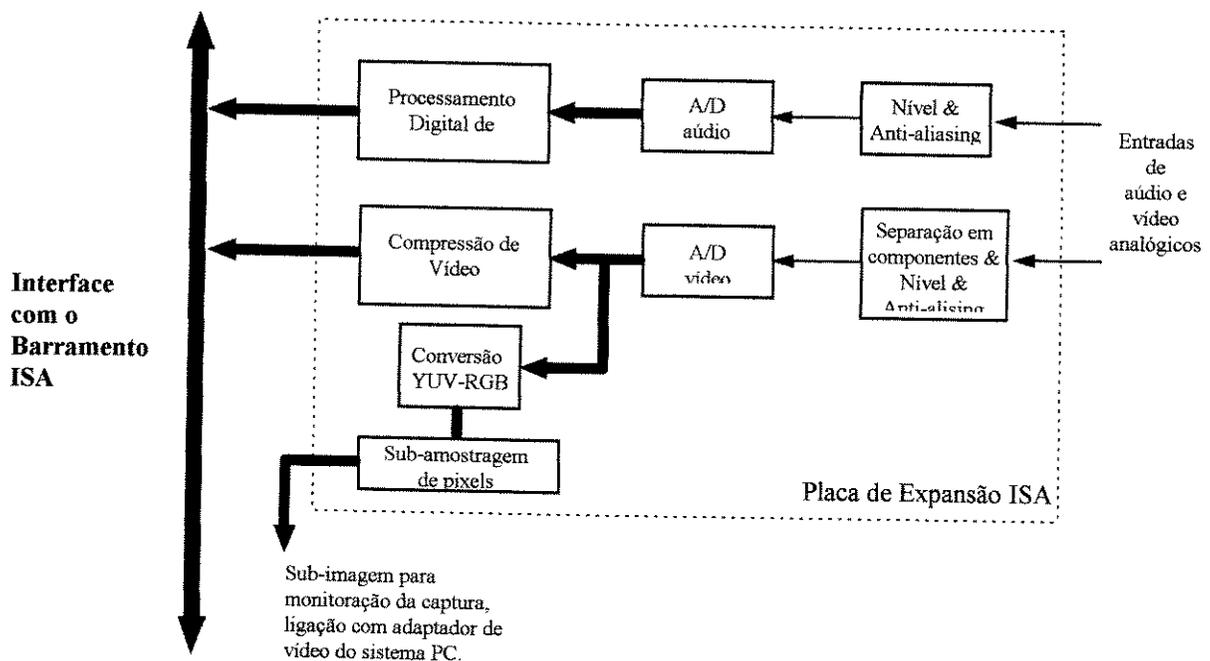


Figura 1.5 Diagrama de Blocos Simplificado de uma Placa de Captura com Compressão

## 1.4 Solução via Projeto de um Sistema de Aquisição e Visualização Dedicado

Em vista dos problemas citados na seção anterior, o projeto de um sistema dedicado justifica-se como uma alternativa bastante atraente. A estratégia geral a ser utilizada é bem similar a da placa dedicada a processamento digital de imagem mencionada na seção anterior, ou seja digitaliza-se a seqüência de quadros, armazena-se a seqüência inteira em um buffer de memória com largura de banda suficiente para suportar as altas taxas resultantes do processo de digitalização, e posteriormente transfere-se os dados a uma velocidade mais lenta ao computador hospedeiro através de interfaces de comunicação padrão. A captura deve ser feita no formato 4:4:4 da família CCIR 601 a fim de que se possa obter a maioria dos formatos fontes utilizados pelos codecs digitais.

## 1.5 Bibliografia

- [1] *CCIR Rec. 601, "Encoding Parameters of Digital Television for Studios", Recommendations of the CCIR - Broadcasting Service (Television), Vol XI - part I, ITU Geneva 1990*
- [2] R. Zelenovsky & A. Mendonça, "PC - Um Guia Prático de Hardware e Interfaceamento", Editora Interciência 1996.
- [3] E. J. Kroeker, "Challenges in Full-Motion Video/Audio for Personal Computers" *SMPTE J*: 24 - 31, Jan 1993.
- [4] D. Hodgson, "The Design Challenges of Full-Motion Video for PC-Compatible Computer Systems" *SMPTE J*: 207 - 216, Mar 1993.
- [5] J. Ozer, "Digital Video Shot by Shot" *PC Magazine*: 104-120, Apr 11 1995.
- [6] C. Johnson & J. Ozer, "Video para Windows" *Windows Computing ( Edição Brasileira)*: 26-62, Jan 1994.

# CAPÍTULO 2

## VISÃO GERAL DE UM SISTEMA DE AQUISIÇÃO E VISUALIZAÇÃO DEDICADO

### 2.1 Características Gerais do Sistema

A seguir listamos as características gerais que um sistema de aquisição e visualização de vídeo digital deve possuir para atender as necessidades recorrentes da pesquisa em codificação de vídeo:

1. Formato de Captura 444 RGB ou YUV com palavras segundo a R601.
2. Captura sem compressão.
3. Capacidade para pelo menos 30 quadros a fim de permitir experimentos envolvendo movimento.
4. Capacidade de visualização fotorealística o que possibilitar a realização de experimentos subjetivos.

Quanto a escolha do formato de captura optou-se pelo membro CCIR 601 de maior resolução, visto que a partir dele se poderia chegar a qualquer um dos demais formatos sem nenhuma perda de qualidade.

Quanto a característica 4), as razões por não se optar pela visualização dos quadros capturados utilizando o sistema de vídeo de um PC, são limitações no quanto a resolução de cor e incompatibilidades quanto ao formato do pixel. Por exemplo os padrão VGA, utilizado pelos PC's de arquiteturas ISA ou EISA, usam mapeamento de cores de oito bits por pixel. Este esquema funciona bem para a maioria dos gráficos utilizados nestes computadores, mas não é adequado à aplicações foto-realísticas como exibição de vídeo em movimento. Mesmo em computadores Macintosh e estações RISC, que utilizam uma resolução de cores de 24 bits (o que estaria em conformidade com a recomendação 601), tem-se problemas em razão destas máquinas utilizarem formatos de imagem baseados em pixel quadrado, que não é utilizado nos padrões de vídeo digital. Como já discutimos na seção anterior, a Rec. 601 define o uso de 720 amostras ativas por linha. este número decorre da frequência de amostragem de 13.5 MHz, escolhida pelo fato de garantir um estrutura de amostragem ortogonal, por garantir a qualidade dos processos de conversão A/D e D/A, e por possibilidade uma certa compatibilidade entre os sistemas de 525 e 625 linhas. Um sistema de TV de 525 linhas (padrão M), possui cerca de 480 linhas ativas, assim se utilizássemos um pixel de formato quadrado, teríamos 640 amostras por linha ao invés de 720. Dessa forma se tentássemos visualizar os quadros de vídeo capturados segundo um padrão de vídeo digital, sem processamentos de conversão de formato de pixel, observaríamos um "alargamento" da imagem na direção horizontal.

## 2.2 Diagrama de Blocos Funcional

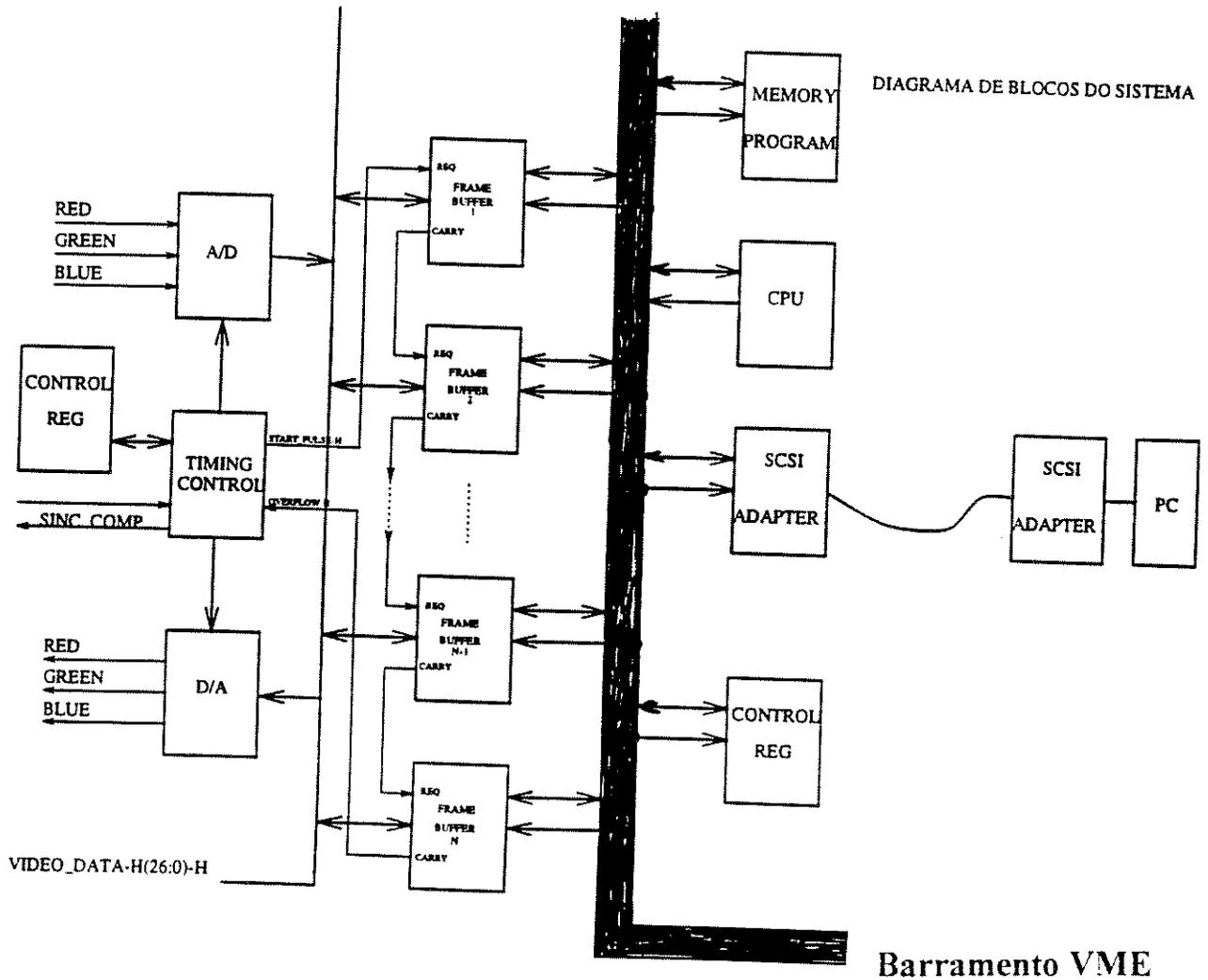


Figura 2.1 - Diagrama de blocos de um sistema dedicado

O diagrama de blocos do sistema de aquisição e visualização é mostrado na figura 2.1. Note que a estratégia utilizada é bastante similar a da placa de processamento digital de imagem mencionada no capítulo anterior, ou

seja, para contornar o gargalo do barramento de expansão, digitaliza-se a seqüência de imagem, armazena-se a seqüência inteira em um buffer de memória com capacidade e largura de banda suficientes para tanto, e posteriormente transfere-se os dados para o computador hospedeiro a uma velocidade mais lenta, compatível às interfaces de comunicação padrão.

Conforme ilustrado na figura 2.1 o sistema consiste basicamente de três módulos funcionais:

- **Módulo Digitalizador de Vídeo:** Este módulo é o responsável pela entrada e saída de vídeo analógico ao sistema. As conversões A/D e D/A do sinal de vídeo são realizadas neste subsistema, que é também responsável pelos pré e pós processamento requeridos por estas operações de conversão de domínio. Assim, questões como filtragem passa-baixas para se tratar o aliasing de componentes do espectro do sinal de vídeo devem ser implementadas neste módulo. Além disso, o digitalizador deve tratar a informação de sincronismo de vídeo, a fim de que se possa realizar o descarte de amostras não ativa. O módulo digitalizador é tratado no capítulo 3.
- **Módulo Armazenador de Quadros:** O Armazenador de quadros é o mecanismo utilizado pelo sistema para vencer o gargalo existente no barramento de expansão do computador hospedeiro. Ele recebe o fluxo de dados a altas taxas gerado pelo módulo digitalizador durante o processo de captura da seqüência de imagens e o armazena para a fim de torná-lo disponível ao hospedeiro a velocidades compatíveis às interfaces de comunicação disponíveis. Deve portanto além de largura de banda para receber e gerar este fluxo de dados ter memória suficiente para armazenar a seqüência de quadros inteira. No capítulo 4, abordamos em mais detalhe a questão do armazenador.
- **Interface de Comunicação com o Hospedeiro:** Feita a digitalização e a armazenagem dos dados de vídeo digital, o passo seguinte é torna estes dados disponíveis ao computador hospedeiro. O módulo de Interface de Comunicação com o Hospedeiro realiza a transferência de dados entre o computador Hospedeiro (no caso um PC ou uma estação de trabalho). Aqui a melhor opção é a utilização de uma interface padrão por questões de flexibilidade (por um sistema que possuísse uma porta de acesso SCSI poderia em princípio comunicar-se com qualquer computador que suportasse esta interface padrão). Dentre as interfaces padrão, a interface SCSI é a mais conveniente dado a sua velocidade, e ao fato de estar disponível tanto em computadores PC (através de placas adaptadoras) quanto estações SUN, que são os tipos de máquinas utilizadas pelo nosso grupo de pesquisa. A interface SCSI é discutida no capítulo 5.

Conforme mostra ainda a figura 2.1, o sistema possui uma arquitetura baseada em dois barramentos: um deles dedicado a operações de entrada e saída de vídeo digital e o outro, um barramento VME, de uso geral a fim de permitir a implementação da interface padrão com o Hospedeiro e operações de gerenciamento do sistema.

## 2.3 O Barramento de E/S de Vídeo

Nesta questão consideremos a proposta de um barramento dedicado a operações de entrada e saída de vídeo realizadas entre o módulo digitalizador e o módulo armazenador do sistema. Nesta discussão feitas apenas considerações a respeito de como seriam as características lógicas de um barramento como este.

### 2.3.1 Sinais do Barramento de E/S de Vídeo

O barramento E/S de vídeo proposto possui os seguintes sinais:

- **CLK (13.5 MHz):** Sinal de relógio de 13.5 Mhz sobre o qual se baseada toda a temporização das transferências de dados realizadas sobre o barramento. A borda de subida deste sinal pode indica (quando o sinal *Active\_Line* está ativado) a validade de uma amostra de vídeo no barramento.
- **START\_PULSE-H<sup>1</sup>:** Pulso ativado pelo módulo digitalizador requerendo o início de um ciclo de entrada de vídeo.
- **BUSY-L<sup>2</sup>:** Sinal ativado pelo módulo armazenador quando não estiver disponível para a transferência de E/S de vídeo. Para que o digitalizador entenda que o armazenador está ocupado, este sinal deve estar ativado na borda de descida do pulso *Sync\_Field*.
- **ACTIVE\_LINE-H:** Sinal acionado pelo Digitalizador para indicar o período de linha ativa digital, ou seja, o período em que o barramento contem as amostras da linha ativa de vídeo.
- **SYNC\_FIELD-H:** Sinal acionado pelo Digitalizador para indicar o início de um campo. Este sinal será utilizado pelo Armazenador para determinar se o número de campos armazenado excedeu sua capacidade de memória.
- **OVERFLOW-H:** Quando o número de campos armazenados exceder a capacidade do armazenador, este sinal é ativado pelo Armazenador. Este pulso ao ser percebido pelo Digitalizador interrompe o ciclo de E/S. Como ocorre logo após o pulso de ativação de *Sync\_Field*, ou seja de maneira similar ao *Start\_Pulse*, este sinal pode ser usado para disparar um outro módulo Armazenador.
- **IN-L/OUT-H:** Este sinal indica a direção da transferência de E/S, assumindo nível 0 para um ciclo de entrada de vídeo e nível 1 para um ciclo de saída.
- **DATA[R(7:0); G(7:0); B(7:0)]:** Linhas de dados de 24 bits que transportam amostras de vídeo digitalizado.

---

<sup>1</sup>A notação *START\_PULSE-H* indica que este sinal é ativo alto.

<sup>2</sup>A notação *BUSY-L* indica que este sinal é ativo baixo.

### 2.3.2 Protocolos dos Ciclos “Entrada de Vídeo” e “Saída de Vídeo”

Na figura 2.2 são consideradas as operações realizadas durante um ciclo de entrada e saída de vídeo.

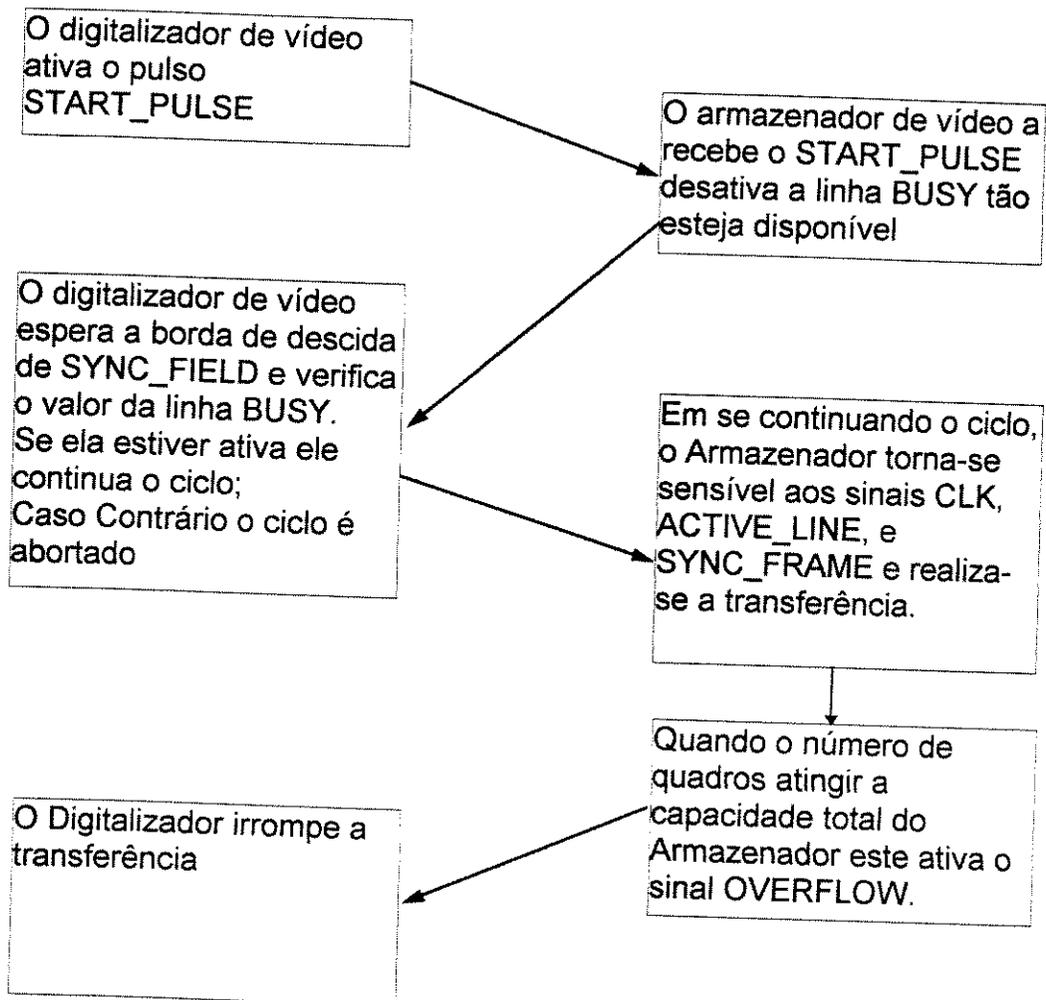


Figura 2.2 - Diagrama de fluxo para um ciclo de entrada e saída de vídeo.

## 2.4 O Barramento VME (Versa Module Europe Bus)

O uso do barramento VME para as operações de gerenciamento e interfaceamento com o hospedeiro, justifica-se pelas seguintes razões:

- Por ser um barramento padronizado seu uso permite o acesso a um grande número de opções de módulos prontos o que permite otimizações de custo e desempenho.
- O VME possui características mecânicas, ou melhor, dimensões maiores, o que torna mais fácil a implementação das placas do sistema, que outros barramentos padronizados como o próprio ISA utilizado no PC.

### 2.4.1 Um Breve Histórico

O padrão VMEbus originou-se a partir do barramento VERSAbus desenvolvido pela Motorola em 1979. O VERSAbus foi utilizado como backplane no desenvolvimento de sistemas EXORmacs baseados nos microprocessadores 68000. Em 1981, o Grupo de Micro Sistemas Europeu da Motorola desenvolveu um novo barramento chamado de “Versa Module Europe Bus”, atualmente conhecido como VMEbus. A principal diferença entre o VMEbus e o VERSAbus é que o primeiro permite o uso de módulos projetados sobre placas do tipo “Eurocard”, que são fisicamente muito menores que as placas do VERSAbus. Um grande número de companhias americanas e europeias passaram então a utilizar o barramento VME, e ele acabou tornando-se um padrão informal para sistemas baseados na família de microprocessadores 68000. O padrão tornou-se oficial quando o VMEbus foi reconhecido e aprovado pelo IEEE (Institute of Electrical and Electronics Engineers, Inc.), sendo conhecido como IEEE P1014 nos Estados Unidos e como IEC 821 (International Electrotechnical Commission) na Europa. A especificação do VMEbus tem três objetivos primários:

- Permitir a comunicação entre dispositivos sobre o VMEbus sem perturbar as atividades internas de outros dispositivos ligados ao barramento.
- Especificar características elétricas e mecânicas necessárias para se projetar dispositivos que irão comunicar-se confiavelmente e sem ambigüidades com outros dispositivos ligados ao barramento.
- Especificar protocolos que definam precisamente a interação entre o barramento VME e os dispositivos ligados a ele.

Embora tenha sido concebido inicialmente para sistemas baseados na família 68000, atualmente existem placas para o VMEbus contendo processadores poderosos como: DEC Alpha, MIPS, i960, vários chips de DSP, AM29000 (RISC), PowerPC e 80486. Segundo a “home page” sobre VME da Internet existem cerca de 200 fabricantes de produtos para o padrão VME, sendo a maioria deles originários dos Estados Unidos, Alemanha e Canadá. O objetivo desta seção não é fornecer uma descrição detalhada a respeito do barramento VME, mas sim

fazer uma breve introdução ao barramento VME de 32 bits<sup>3</sup>. As informações contidas nesta seção visam apenas auxiliar e completar a exposição da organização de nosso sistema de aquisição e visualização de vídeo, bem como do projeto das interfaces de alguns módulos que serão mostrados mais adiante no texto. Informações mais detalhadas podem ser encontradas na referência [1].

### 2.4.2 Características Mecânicas do VME

A figura 2.3 mostra um chassi típico VME que suporta os dois tipos possíveis de placas VME, ou seja as placas do tipo:

- “Single Eurocard” : de altura 3U (132.5mm) e profundidade de 160mm, e
- “Double Eurocard” : de altura 6U (265.9mm) e profundidade 160mm.

Outra característica marcante do barramento VME são seus dois conectores. Eles são do tipo “two-piece” e contem três fileiras de 32 pinos (96 no total), seguindo o padrão DIN41612, conforme mostra a figura 2.4. Segundo a terminologia VME, quando faz parte da placa, o conector é referenciado como P1 (ou P2), e a parte que está no backplane é referenciada como J1 (ou J2). O conector P1, chamado de conector primário, contem todas as funções necessárias para uma implementação básica do VMEbus. O conector P2, chamado de

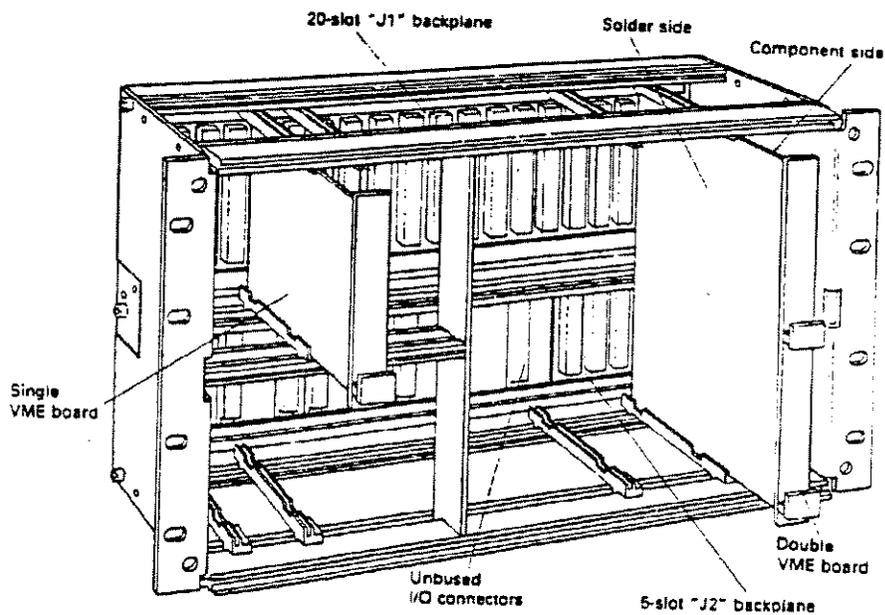


Figura 2.3 - Chassi típico do VMEbus

<sup>3</sup> Já existe a proposta para um novo barramento VME de 64 bits.

## Visão Geral de um Sistema de Aquisição e Visualiz. Dedicado 19

conector secundário, fornece facilidades de expansão, permitindo que as linhas de endereço sejam expandidas de 16 para 32, e que os bits de dados passem de 16 para 32 bits. As tabelas 2.1 e 2.2 mostram as pinagens dos conectores P1 e P2 respectivamente.

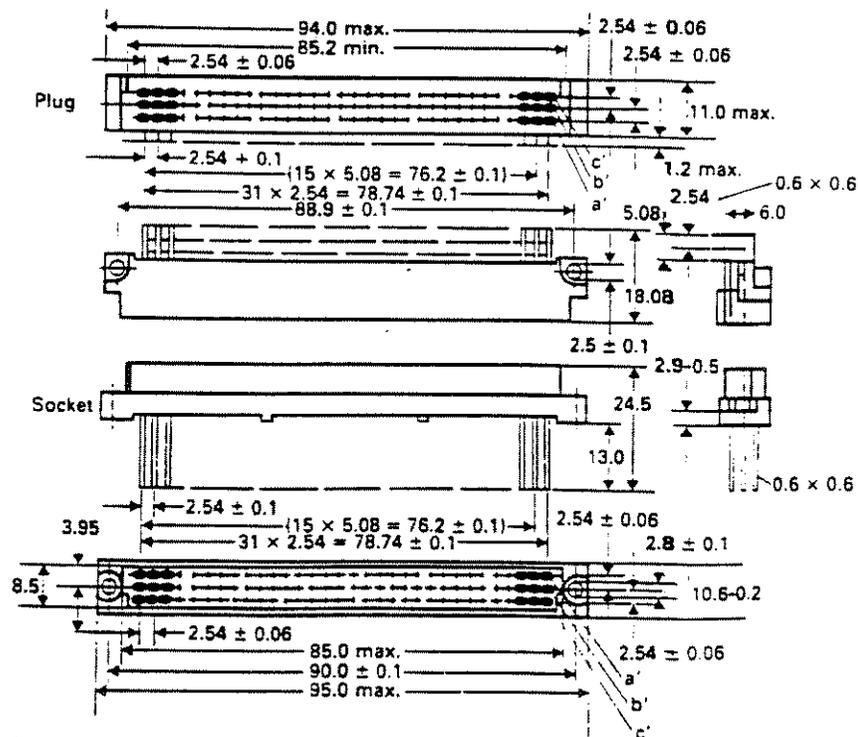
**TABELA 2.1 - Pinagem do conector P1**

Pino	Sinal Associado		
	row A	row B	row C
01	D 00	BBSY* <sup>4</sup>	D 08
02	D 01	BCLR*	D 09
03	D 02	ACFAIL*	D 10
04	D 03	BG0IN*	D 11
05	D 04	BG0OUT*	D 12
06	D 05	BG1IN*	D 13
07	D 06	BG1OUT*	D 14
08	D 07	BG2IN*	D 15
09	GND	BG2OUT*	GND
10	SYSCLK	BG3IN*	SYSFAIL*
11	GND	BG3OUT*	BERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*
14	WRITE*	BR2*	AM5
15	GND	BR3*	A 23
16	DTACK*	AM0	A 22
17	GND	AM1	A 21
18	AS*	AM2	A 20
19	GND	AM3	A 19
20	IACK*	GND	A 18
21	IACKIN*	SERCLK	A 17
22	IACKOUT*	SERDAT*	A 16
23	AM4	GND	A 15
24	A 07	IRQ7*	A 14
25	A 06	IRQ6*	A 13
26	A 05	IRQ5*	A 12
27	A 04	IRQ4*	A 11
28	A 03	IRQ3*	A 10
29	A 02	IRQ2*	A 09
30	A 01	IRQ1*	A 08
31	-12 V	+5 V	+12 V
32	+5 V	+5 V	+5 V

<sup>4</sup>A notação SINAL\* indica que este sinal é ativo-baixo.

TABELA 2.2 - Pinagem do conector P2

Pino	Sinais Associados		
	row a	row b	row c
01	User I/O	+5 V	User I/O
02	User I/O	GND	User I/O
03	User I/O	Reserved	User I/O
04	User I/O	A 24	User I/O
05	User I/O	A 25	User I/O
06	User I/O	A 26	User I/O
07	User I/O	A 27	User I/O
08	User I/O	A 28	User I/O
09	User I/O	A 29	User I/O
10	User I/O	A 30	User I/O
11	User I/O	A 31	User I/O
12	User I/O	GND	User I/O
13	User I/O	+5 V	User I/O
14	User I/O	D 16	User I/O
15	User I/O	D 17	User I/O
16	User I/O	D 18	User I/O
17	User I/O	D 19	User I/O
18	User I/O	D 20	User I/O
19	User I/O	D 21	User I/O
20	User I/O	D 22	User I/O
21	User I/O	D 23	User I/O
22	User I/O	GND	User I/O
23	User I/O	D 24	User I/O
24	User I/O	D 25	User I/O
25	User I/O	D 26	User I/O
26	User I/O	D 27	User I/O
27	User I/O	D 28	User I/O
28	User I/O	D 29	User I/O
29	User I/O	D 30	User I/O
30	User I/O	D 31	User I/O
31	User I/O	GND	User I/O
32	User I/O	+5 V	User I/O



All dimensions in millimeters.

Figura 2.4 - Conector "two-piece" DIN41612 utilizado no VMEbus

### 2.4.3 Definições Lógicas do barramento VME

Na terminologia VME, as posições ao longo do barramento VME nos quais os cartões são plugados são chamados de slots. Um módulo é uma coleção de componentes eletrônicos com um propósito único. Assim sendo pode existir mais de um módulo em um único cartão.

O barramento VME consiste de quatro grupos de sinais, conforme ilustrado na figura 2.5. Ainda observando esta figura, podemos verificar que a definição de um sistema com barramento VME especifica um número de módulos que formam a interface entre o barramento backplane VME e os vários módulos-usuários que compõe o sistema. Os módulos funcionais especificados pelo VME são:

- **Mestre (Master):** Um Mestre é um módulo capaz de iniciar transferências no barramento.
- **Escravo (Slave):** Um Escravo é um módulo capaz de responder a uma transferência de dados iniciada por um Mestre. Um dispositivo Escravo típico é um módulo de memória.
- **Requester:** Um Requester é um módulo que se encontra na mesma placa de um módulo Mestre ou e que é capaz de requerer controle do barramento sempre que seu Mestre necessitar.
- **Interrupter:** Um Interrupter é um módulo funcional capaz de requerer serviço de um subsistema Mestre pela geração de uma requisição de interrupção. O Interrupter fornece também informação de status quando esta for requerida.

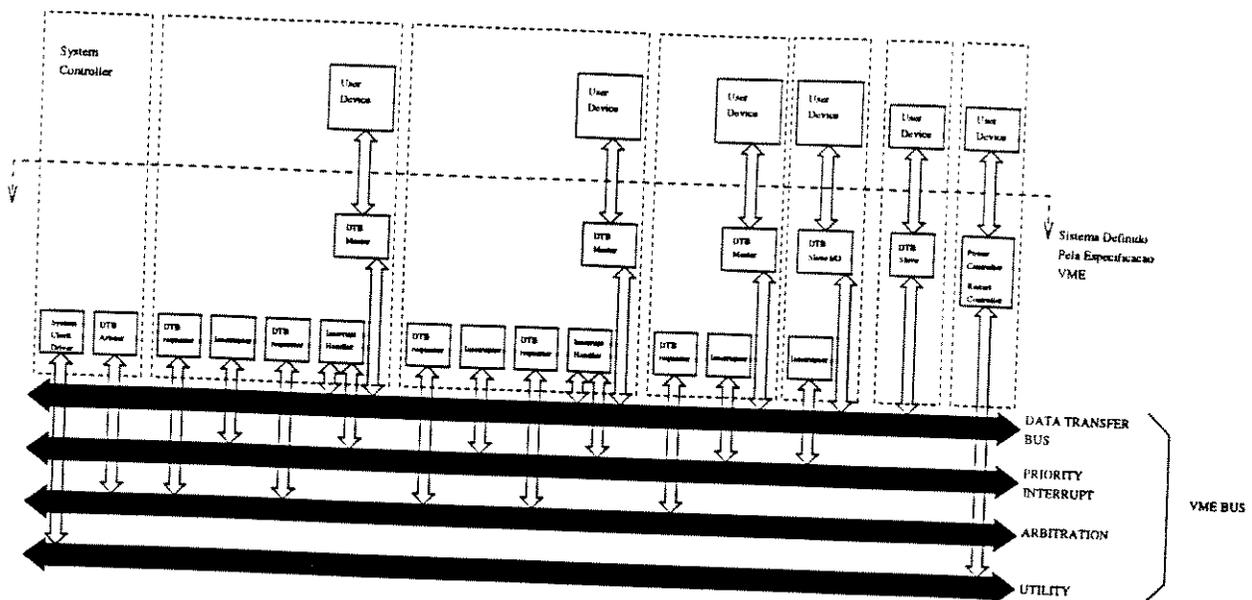


Figura 2.5 - Módulos funcionais e sub-barramentos do VMEbus

- **Arbiter:** Um árbitro é um módulo funcional que recebe requisições pelo controle do barramento de transferência de dados (DTB) por parte de outros módulos, e segundo uma prioridade, passa o controle do DTB para o Requester apropriado.

Definidos alguns dos módulos que fazem parte da especificação do barramento VME, passemos aos quatro grupos de sinais que o formam:

- **Data Transfer Bus - DTB:** O DTB é formado pelas linhas de dados, de endereço, e pelos sinais de controle associados. Ele suporta as transferências de dados entre os módulos Mestres e Escravos do sistema.
- **Arbitration Bus:** Um sistema baseado no barramento VME pode ser configurado para possuir mais de um Mestre. O Arbitration Bus implementa o mecanismo que realiza a passagem de controle do barramento, garantindo que em um dado instante, somente um dentre os vários Mestres da configuração tenha o controle do barramento.
- **Priority Interrupt Bus:** Este grupo de sinais, juntamente com os módulos associados agregam ao barramento VME a capacidade de tratamento de interrupções.
- **Utilities Bus:** Funções de relógio, detecção de falhas, e de inicialização estão agrupadas neste sub-barramento.

### 2.4.4 Data Transfer Bus - DTB

Conforme já mencionamos, os módulos funcionais Mestre e Escravo usam o DTB para transferir dados entre si. A seguir são apresentados detalhes a respeito dos sinais que compõe o DTB e do protocolo que rege a transferência de dados sobre este barramento.

#### Sinais do DTB

- **A01-A23:** Barramento de endereços - Linhas acionadas por saídas 3-State que especificam um endereço de memória.
- **A24-A31:** Barramento de endereços - Linhas acionadas por saídas 3-State que especificam um endereço de memória.
- **AS\* - ADDRESS STROBE:** Sinal acionado por saída 3-State que indica um endereço válido nas linhas de endereço.
- **AM0-AM5 ADDRESS MODIFIER:** Linhas acionadas por saídas 3-State que fornecem informações adicionais sobre o barramento de endereços, tais como: largura, tipo de ciclo, identificação do Mestre, etc.
- **D00-D15:** Barramento de dados - Linhas bidirecionais acionadas por saídas 3-State que servem de meio de comunicação entre o Mestre e o Escravo.
- **D16-D31:** Barramento de dados - Linhas bidimensionais acionadas por saídas 3-State para expansão do barramento de dados.

- **DS0\*** - **Data Strobe 0**: Sinal acionado por saída 3-State que indica que a transferência ocorrerá nas linhas D00-D07 do barramento de dados.
- **DS1\*** - **Data Strobe 1**: Sinal acionado por saída 3-State que indica que a transferência ocorrerá nas linhas D08-D15 do barramento de dados.
- **LWORD\***: Sinal acionado via driver 3-State que indica uma transferência de 32 bits.
- **DTACK\*** - **Data Transfer Acknowledge**: Sinal acionado por saída Open Collector pelo módulo Escravo. A borda de descida deste sinal indica que um dado válido está disponível no barramento de dados durante um ciclo de leitura, ou que um dado foi aceito do barramento de dados em um ciclo de escrita.
- **WRITE\***: Sinal acionado por saída 3-State que especifica a direção da transferência de dados. Um nível alto indica um ciclo de leitura e um nível baixo um ciclo de escrita.

### Transferência de Dados sobre o Barramento VME

Quando um módulo deseja transferir dados para um módulo Escravo, ele deve primeiro adquirir o controle do barramento através de seu módulo Requester, conforme será descrito mais adiante. No caso por exemplo de um ciclo de leitura sobre o DTB, quando o módulo em questão tornar-se o Mestre do barramento, ele procederá conforme o ilustrado na figura 2.6 . Um diagrama de temporização desta transferência simplificado é mostrado na figura 2.7 .

### 2.4.5 Arbitration Bus

Por arbitragem do barramento entenda-se o mecanismo através do qual se realiza a passagem do controle do barramento para um Mestre dentre vários Mestres requerentes de uma configuração do sistema.

O sub-barramento de arbitragem possui 14 linhas, as quais podem ser arranjadas em dois grupos. Um deles formado pelas linhas acionadas pelos módulos Requesters dos Mestres e o outro pelas linhas acionadas pelo árbitro do sistema, que está fisicamente localizado no slot 1. Na tabela 3 são mostrados os nomes com os respectivos mnemônicos, e os módulos acionadores das linhas de arbitragem do VMEbus.

**TABELA 2.3 - Linhas de arbitragem do VMEbus**

Mnemônico do Sinal	Nome do Sinal	Acionado pelo Módulo
BR0*	Bus request 0	Requester
BR1*	Bus request 1	Requester
BR2*	Bus request 2	Requester
BR3*	Bus request 3	Requester
BGOUT0*	Bus grant out 0	Requester
BGOUT1*	Bus grant out 1	Requester
BGOUT2*	Bus grant out 2	Requester
BGOUT3*	Bus grant out 3	Requester
BBSY*	Bus busy	Requester
BCLR*	Bus clear	Requester
BG0IN*	bus grant in 0	Árbitro
BG1IN*	bus grant in 1	Árbitro
BG2IN*	bus grant in 2	Árbitro
BG3IN*	bus grant in 3	Árbitro

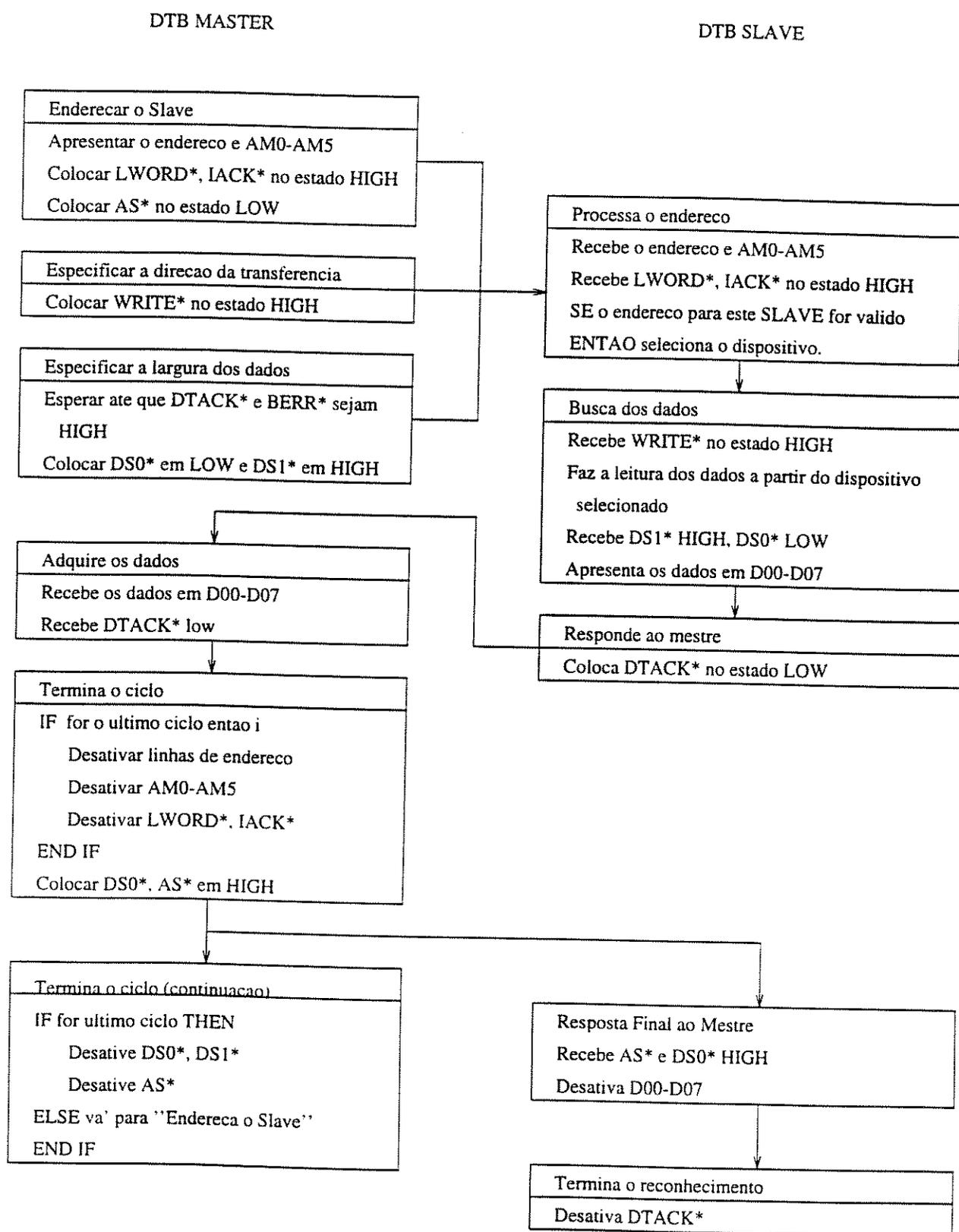


Figura 2.6 - Diagrama de fluxo de um ciclo de leitura do VMEbus

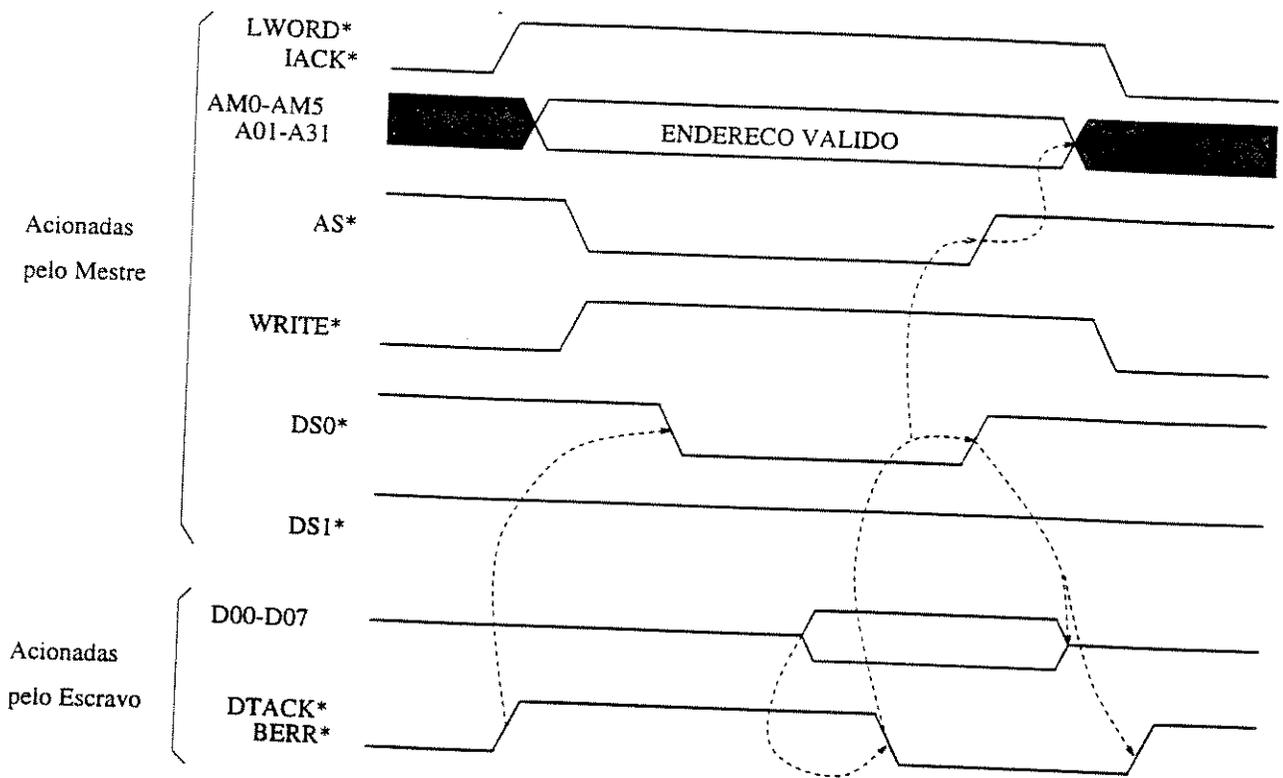


Figura 2.7 - Ciclo de leitura envolvendo um byte sobre o DTB do VMEbus

Dois tipos de módulos participam do processo de arbitragem: os Requesters associados aos respectivos Mestres, e o árbitro que se torna o controlador do sistema e atua globalmente sobre o barramento.

Sempre que ocorre uma situação onde um número de entidades compete por recursos limitados, um algoritmo deve ser usado para tratar a distribuição destes recursos. Na terminologia VME, as estratégias adotadas para tratarem este problema são chamadas de “opções de arbitragem”, e as três opções disponíveis são:

- **Opção RRS (Round Robin Select):** que associa prioridades aos Mestres de uma forma “rotativa”. Ou seja os níveis de requisição de barramento ( BR0\* até BR3\*) no que se refere as suas prioridades são tratados como se formassem um ciclo, isto é: BR0\* é seguido de BR3\*, que é seguido de BR2\*, que é seguido de BR1\*, que é seguido de BR0\* que novamente é seguido de BR3\*, e assim por diante. Em um instante qualquer, um dos quatro níveis é assumido como sendo o de maior prioridade ( BR2\*, por exemplo ), de maneira que este nível pode assumir o controle do barramento. Se o nível assumido como sendo de maior prioridade não quiser acessar o barramento, então o próximo nível na escala de prioridade (ou seja, BR1\* no nosso exemplo) será testado, e assim sucessivamente.
- **Opção PRI (Prioritized):** que associa um nível de prioridade fixo a cada uma das linhas de requisição, sendo BR3\* associada ao nível de maior prioridade e BR0\* ao nível de menor prioridade.
- **Opção SGL (Single Level):** Nesta opção trata-se apenas as requisições ligadas a linha BR3\*. A prioridade é então determinada por “daisy chaining”, ou seja o módulo mais próximo do módulo árbitro (localizado no

slot 1) tem a maior prioridade, e a prioridade de um módulo é tanto menor quanto mais distante ele estiver do árbitro do sistema.

### **Linhas de Arbitragem**

O arranjo físico das linhas de arbitragem é mostrado na figura 2.8 e o modo como elas se relacionam com os módulos árbitro e Requester é ilustrado na figura 2.9. Note que na figura 2.8, as linhas bus-grant-in e bus-grant-out são quebradas e vão somente de um slot para outro. Uma linha bus-grant-in proveniente de um módulo “mais a esquerda” ( e portanto de um módulo de maior prioridade) é passada ao módulo a sua direita como um sinal bus-grant-out.

O arranjo da figura 2.8 é chamado de “daisy-chain”. Uma linha de barramento contínua permite a passagem de um sinal em duas direções para todos os dispositivos conectados a ela. Uma linha em “daisy chain” é unidirecional, passando um sinal de uma terminação específica para outra. Além disso cada módulo conectado a um arranjo como este, pode passar o sinal recebido ao módulo seguinte sem alterações, ou injetar um sinal “próprio” em sua linha.

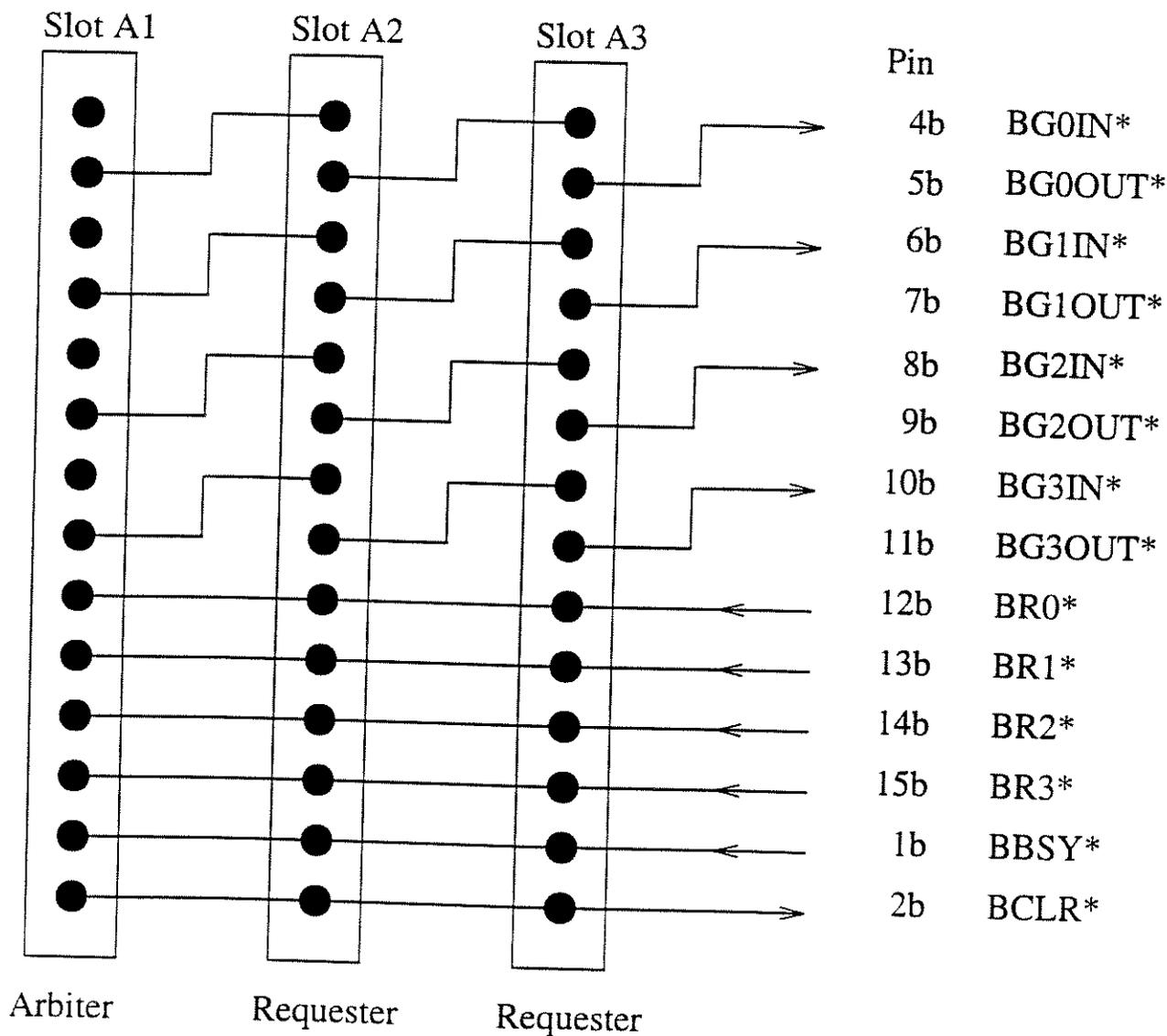


Figura 2.8 - Disposição das linhas de arbitragem do barramento VME

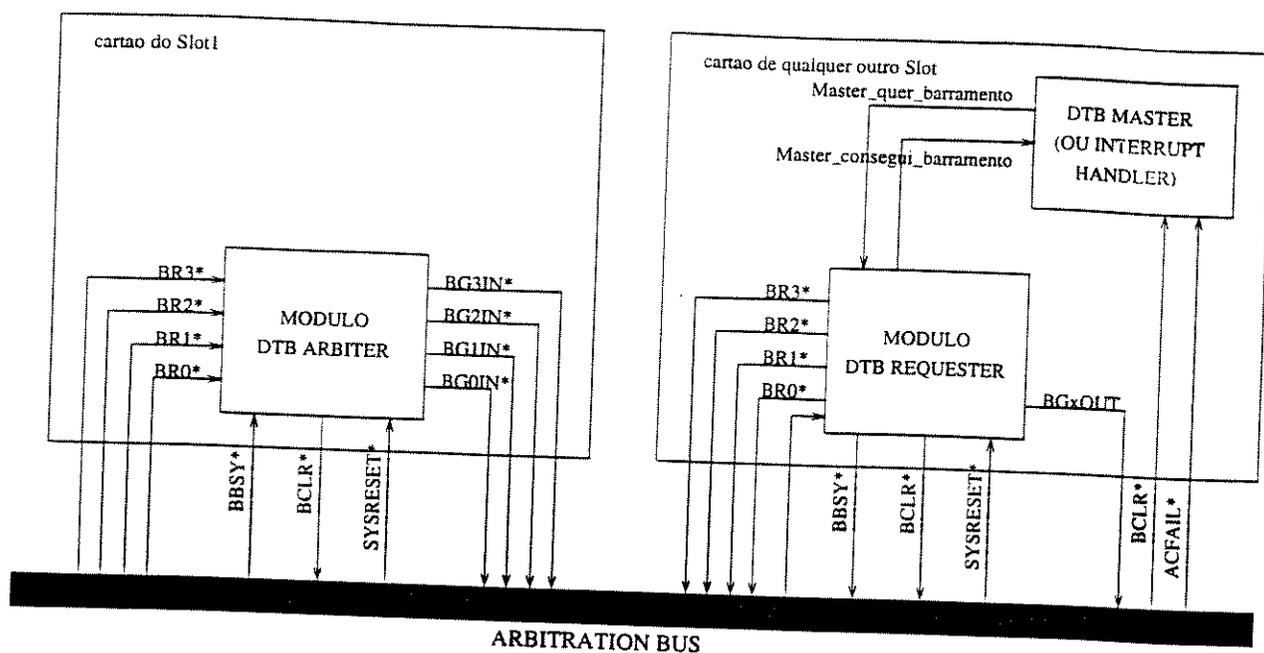


Figura 2.9 - Disposição dos módulos árbitro e requester

### O processo de arbitragem

Um módulo Requester faz uma requisição pelo controle do barramento ativando uma das linhas de requisição (BR0\* até BR3\*). Note que o Requester ativa somente uma linha de requisição que é aquela que determina a prioridade do Requester.

O árbitro, ao receber a requisição pelo barramento, pode (dependendo da opção de arbitragem escolhida) ativar uma de suas linhas “bus-grant-out” (BG0OUT\* até BG3OUT\*). O sinal “bus-grant-out” propaga-se então ao longo do “daisy-chain”. Se um módulo “x” não requereu acesso ao barramento, ele passa a frente o sinal “bus grant”, ativando sua linha BGxOUT. Se entretanto, o módulo “x” havia requerido o controle do barramento, ele não ativa o seu sinal BGxOUT.

O arranjo “daisy-chain” das linhas de arbitragem fornecem uma priorização automática, um vez que os Requester mais a frente na linha não recebem um sinal “bus grant” se um Requester de maior prioridade (mais próximo do árbitro) já tiver aceitado o controle do barramento.

Ao conseguir o controle do barramento, o Requester ativa a linha BBSY\* (“bus busy”) indicando a posse do barramento, e o controle não ser transferido para outro módulo enquanto BBSY\* permanecer ativado.

A linha BCLR\* (“bus clear”) é ativada pelo árbitro do sistema e informa ao Mestre correntemente de posse do barramento, que um outro Mestre com prioridade maior deseja acessar o barramento. Tipicamente, embora não seja obrigatório, o Mestre de posse do DTB terminará a sua tarefa e desativará a linha BBSY\*. A linha BCLR\* só é ativada quando se implementa a opção de arbitragem PRI.

### Um exemplo de arbitragem

A figura 2.10 mostra uma possível seqüência de eventos que ocorreriam durante a arbitragem de duas

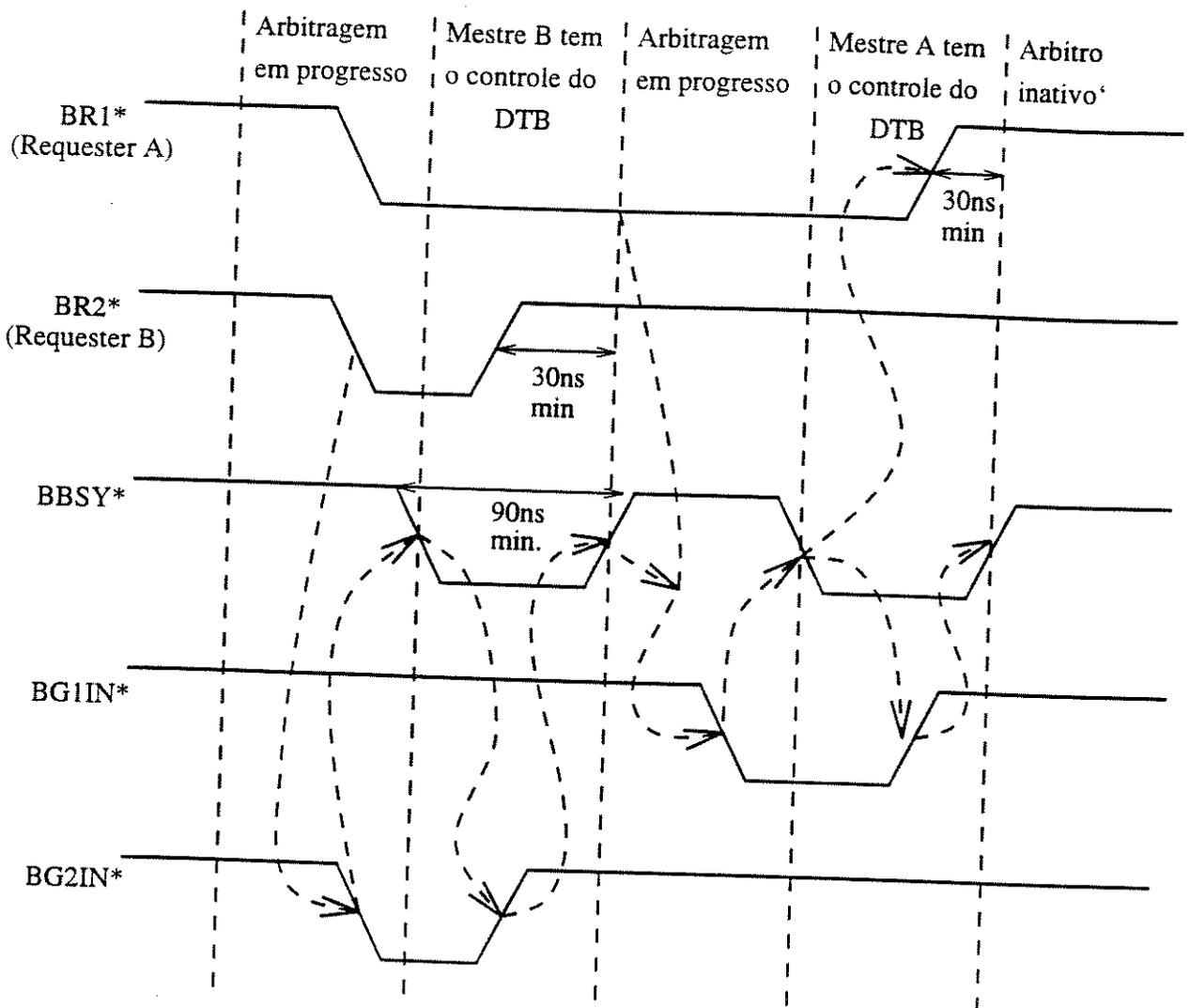


Figura 2.11 - Exemplo de Arbitragem entre duas requisições de níveis diferentes

requisições com níveis de prioridade diferentes.

A seqüência de eventos tem início com os Requesters A e B ativando as suas linhas de requisição, BR1\* e BR2\* respectivamente. Assumindo que o árbitro detecta a ativação destes dois sinais de forma simultânea, ele ativará somente a linha BG2IN\*, uma vez que a requisição BR2\* tem maior prioridade (assumindo-se que a opção de arbitragem é a PRI). Quando o sinal BG2IN\* propagando-se ao longo do "daisy-chain" atinge o Requester B, ele responderá ativando a linha BBSY\*. O Requester B então desativa BR2\* e informa ao seu Mestre que o barramento de dados está disponível.

Após detectar que BBSY\* foi ativado, o árbitro desativa BG2IN\*, e quando o Mestre B completa a sua transferência de dados, o Requester B desativa BBSY\*. Note que BBSY\* só é desativado quando BG2IN\* está

desativado e já se passaram ao menos 30 ns da desativação de BR2\*. Estes 30 ns são a garantia de que o árbitro não interpretará o antigo valor ativo-baixo de BR2\* como sendo uma nova requisição.

O árbitro interpreta a desativação de BBSY\* como um sinal para arbitrar o barramento mais uma vez. Dado que BR1\* está ativado e é a única linha de requisição atualmente neste estado, o árbitro ativa a linha BGIN\*. O Requester A responde colocando BBSY\* no estado ativo-baixo e 30 ns depois desativa BR1\* informando ao Mestre B que ele detem o controle do barramento. Neste exemplo, uma vez que nenhuma de requisição está ativada quando o Requester A desativa BBSY\*, o árbitro permanece inativo até que nova requisição seja feita.

### 2.4.6 Priority Interrupt Bus

O Priority Interrupt Bus habilita os módulos conectados ao barramento VME a requisitarem serviço de um Mestre do sistema. Dois tipos de módulos estão associados ao barramento de interrupção:

- O Interrupter, que é o modulo a requisitar o serviço , e;
- O Interrupt Handler, que recebe esta requisição.

#### Linhas do barramento de interrupção

As dez linhas do barramento de interrupção são dadas na tabela 2.4. Sete destas linhas, IRQ1\* até IRQ2\*, são dedicadas à requisição de interrupção, sendo acionadas pelos módulos Interrupters. As linhas IACK\*, IACKIN\* e IACKOUT\*, estão associadas à sequência de reconhecimento da interrupção que é iniciada pelo Interrupt Handler.

TABELA 2.4 - Linhas de interrupção do VMEbus

Mnemônico do sinal	Nome do sinal
IRQ7*	Interrupt request 7
IRQ6*	Interrupt request 6
IRQ5*	Interrupt request 5
IRQ4*	Interrupt request 4
IRQ3*	Interrupt request 3
IRQ2*	Interrupt request 2
IRQ1*	Interrupt request 1
IACK*	Interrupt acknowlegd
IACKIN*	Interrupt acknowlegd input
IACKOUT*	Interrupt acknowlegd output

A figura 2.11 mostra como é a estrutura “daisy chain” das linhas de interrupção IACKIN\* e IACKOUT\*. A linha IACK\* é ativada pelo Interrupt Handler, e é uma linha “continua” do barramento que está conectada ao pino IACKIN\* do slot 1. De modo que quando IACK\* for ativada, inicia-se a propagação do sinal de reconhecimento da interrupção através das linhas em “daisy chain”, IACKIN\* e IACKOUT\*.

Cada uma das linhas de requisição de interrupção pode ser compartilhada por dois ou mais Interrupters. De modo que, quando uma destas linhas for ativada, o módulo tratador de interrupção não poderia em principio identificar qual é o módulo que está requisitando a interrupção. Entretanto, este problema é

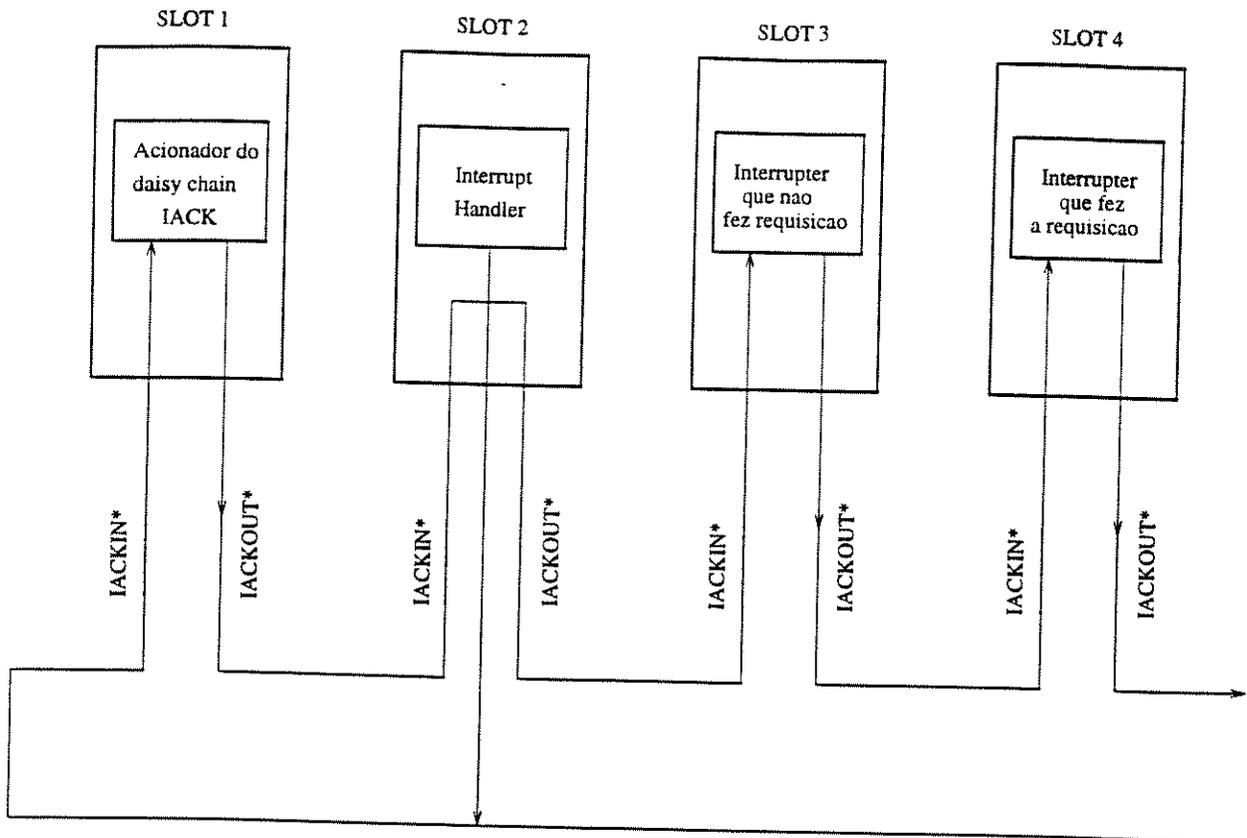


Figura 2.11 - Linhas de interrupção IACK\*, IACKIN\* e IACKOUT\*

resolvido pelo arranjo em “daisy chain” das linhas IACKIN\* e IACKOUT\*, que assegura o reconhecimento de um único módulo Interrupter.

Vejamos então como funcionam estas linhas de interrupção em “daisy-chain”. Quando uma interrupção é reconhecida, a linha IACKIN\* do slot 1 é ativada, e cada módulo que está ativando uma das linhas de requisição fica aguardando a chegada de um nível baixo a sua linha IACKIN\*, a fim de poder aceitar o reconhecimento da interrupção. O módulo requisitante mais próximo do slot 1 aceita então o reconhecimento e não passa a frente o nível baixo em sua saída IACKOUT\*, encerrando assim, a propagação do sinal de aceitação ao longo das linhas de reconhecimento de interrupção.

### Funções do módulo Interrupter Handler

O módulo Interrupter Handler é o responsável pelo tratamento das requisições de interrupções e executa basicamente quatro funções:

1. Prioriza as requisições, associando a linha IRQ7\* a maior prioridade e a IRQ1\* a menor prioridade.
2. Usa o Requester associado (ou seja que está na mesma placa) para acessar o barramento de transferência de dados, e ao conseguir o acesso, reconhece a interrupção ativando a linha IACK\*.
3. Lê o byte de status (chamado de byte ID na terminologia VME) colocado no DTB pelo Interrupter que foi reconhecido.
4. Com base na informação obtida no byte de status, ele inicia a rotina de atendimento da interrupção.

### Funções do módulo Interrupter

Os Interrupters são a fonte de requisições de interrupção e são usados para estabelecer as seguintes tarefas:

- Requisitar uma interrupção ao Interrupt Handler ativando
- uma linha IRQx\*.
- Fornecer o byte de status ao Interrupt Handler quando sua requisição de interrupção é atendida.
- Passar a frente o sinal de reconhecimento (colocando sua saída IACKOUT\* no nível baixo) se não requisitou a interrupção.

### 2.4.7 Utility Bus

As funções de relógio, detecção de falhas, e de inicialização estão reunidas neste grupo de sinais. Os principais sinais são:

- [SYSCLK]: Sinal relógio de 16MHz que serve de base de temporização para todos os módulos do barramento VME. Não é necessário que este relógio tenha a mesma frequência ou fase do relógio do processador.
- [SYSRESET\*]: Sinal Open-Collector que quando em nível baixo provoca inicialização de todo o sistema.
- [SYSFAIL\*]: Sinal Open-Collector que ao ser acionado indica falha no sistema. Este sinal pode ser gerado por qualquer módulo do barramento.
- [ACFAIL\*]: Sinal Open-Collector que ao ser acionado indica que a tensão de entrada (AC) da fonte de alimentação caiu a níveis de falha.

## 2.5 Bibliografia

[1] VITA. VME Bus Specifications Manual. PRINTEX Publishing, 1985.

[2] Clements A. "Microprocessor System Design: 68000 Hardware, Software and Interfacing" PWS-KENT Publishing Company, 1987.

# CAPÍTULO 3

## MÓDULO DIGITALIZADOR DE VÍDEO

Neste capítulo passaremos a detalhar nossa proposta para um sistema de aquisição e visualização de vídeo digital, considerando inicialmente o módulo digitalizador de vídeo. Conforme discutido no capítulo anterior, este módulo tem como funções básicas realizar as conversões A/D e D/A do sinal de vídeo. Além disso, o módulo deve tratar, durante uma operação de captura, a informação de sincronismo para que, em se detectando os intervalos vertical e horizontal, se possa descartar as amostras de vídeo não-ativo. Outra tarefa realizada pelo módulo é a geração da informação de sincronismo, para que o sinal de vídeo analógico resultante da conversão D/A possa ser exibido adequadamente em um monitor de vídeo analógico.

A apresentação considera inicialmente os diversos componentes básicos utilizados no módulo digitalizador. Assim na seção 3.1 são discutidos os conversores A/D e D/A aplicáveis à digitalização do sinal de vídeo. Os filtros passa-baixas anti-aliasing de pré e pós-processamento, são considerados na seção 3.2. Na seção 3.3 abordamos os circuitos PLL e a sua aplicação à implementação de sintetizadores de frequência, necessários para geração do relógio de amostragem do digitalizador. Outros circuitos para aplicações específicos em vídeo como geradores e separadores de sincronismo, e amplificadores de vídeo são discutidos na seção 3.4.

Por fim, na seção 3.5 mostramos a aplicação dos componentes discutidos ao longo deste capítulo ao projeto de um digitalizador.

### 3.1 Conversores A/D e D/A

#### 3.1.1 Conversores A/D do tipo paralelo ou “flash”

As frequências de amostragem utilizadas para digitalização do sinal de vídeo em geral assumem valores relativamente altos, como 13.5MHz ou 14.3 MHz ( $4f_{sc}$ ). Por outro lado, com relação aos níveis de quantização necessários, os requisitos são bem menos rigorosos, e uma quantização de 8 bits produz bons resultados. Em situações como esta, ou seja, alta velocidade de operação e um número de níveis de quantização moderado, o mais adequado é utilizar-se conversores A/D do tipo paralelo.

Os conversores A/D do tipo paralelo são os mais rápidos e mais simples do ponto de vista conceitual. Neste tipo de conversor utiliza-se um comparador analógico com referência fixa para cada nível de quantização representado pela palavra digital. As saídas destes comparadores são então conectadas de maneira

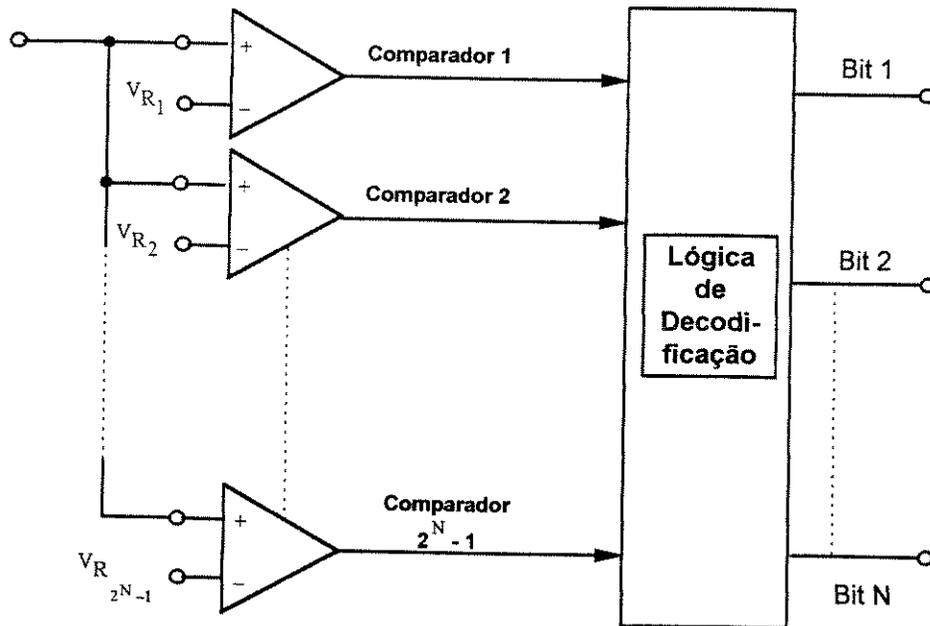


Figura 3.1 - Diagrama de blocos conceitual de um conversor A/D paralelo

apropriada a uma lógica de decodificação, produzindo então a palavra digital. A figura 3.1 mostra um diagrama de bloco conceitual de um conversor A/D paralelo. Note que para uma resolução de  $N$  bits, são necessários  $2^N - 1$  comparadores e níveis de referência em separado. Ou seja, a complexidade do dispositivo conversor aumenta muito rápido a medida que cresce o número de bits em paralelo utilizado. Nos conversores paralelos todos os bits são processados simultaneamente, e dessa forma, é possível realizar a conversão em um único ciclo de relógio. Em razão desta última propriedade, os conversores paralelos são também chamados de conversores do tipo flash.

A figura 3.2 mostra a arquitetura básica de um conversor A/D paralelo de  $N$  bits em mais detalhe. Os  $2^N - 1$  níveis de decisão são gerados por um arranjo de resistores que polarizam os  $2^N - 1$  comparadores. As entradas dos vários comparadores são conectadas a uma entrada analógica comum, e acionadas de maneira simultânea pela entrada analógica  $V_A$ . Portanto, para qualquer nível de tensão assumido pela entrada analógica, as saídas dos comparadores com tensões de referência abaixo de  $V_A$  assumirão o nível lógico "1", ao passo que as saídas daqueles cujas referências estiverem acima de  $V_A$  assumirão o nível lógico "0". A determinação do segmento resistor no qual se encontra  $V_A$  é feita por um arranjo de portas AND's, como o ilustrado na figura 3.3. Esse nível é detectado pela comparação da saída lógica de cada comparador com a dos comparadores que estão imediatamente acima e abaixo. A informação resultante pode ser então decodificada através de lógica discreta combinacional ou através de um arranjo lógico programável (PLA). Embora o processo de conversão possa ser, ao menos ponto de vista conceitual, realizado em um único ciclo, na prática são utilizados dois ciclos de relógio, sendo o primeiro deles destinado à amostragem-armazenagem do nível da entrada analógica pelos comparadores, e o segundo para completar a operação de decodificação.

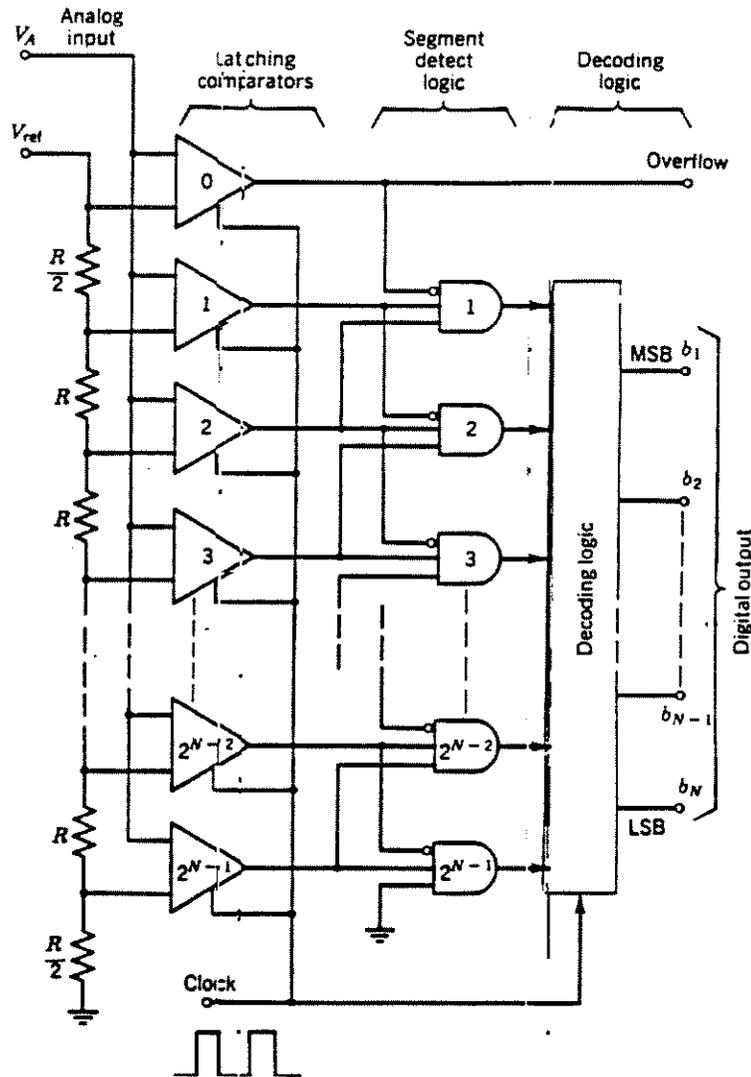


Figura 3.2 - Arquitetura básica de um conversor A/D paralelo.

### 3.1.2 Conversores D/A

Embora exista um grande número de técnicas para implementação de circuitos conversores D/A, estes costumam ser agrupados, segundo os seus princípios de funcionamento, em uma das três seguintes categorias: (1) circuitos baseados em escalonamento de corrente, (2) circuitos baseados em escalonamento de tensão e, (3) circuitos baseados em escalonamento de carga. Entre estes, o primeiro grupo de circuitos é mais comum em projetos de dispositivos utilizando a tecnologia bipolar, enquanto os circuitos baseados em escalonamento de tensão e carga são mais adequados a projetos de componentes MOS.

Os conversores D/A utilizados em aplicações envolvendo vídeo, costumam ser implementados utilizando-se tecnologia bipolar em conjunção com circuitos baseados em escalonamento de corrente. Este arranjo é o adequado para a implementação de redes de resistores de precisão e chaves de corrente de alta velocidade, o que permite obtenção de dispositivos mais rápidos. Neste item portanto, discutiremos os princípios básicos apenas do grupo (1) de circuitos. Detalhes a respeito das outras duas técnicas mencionadas podem ser encontrados na ref [1].

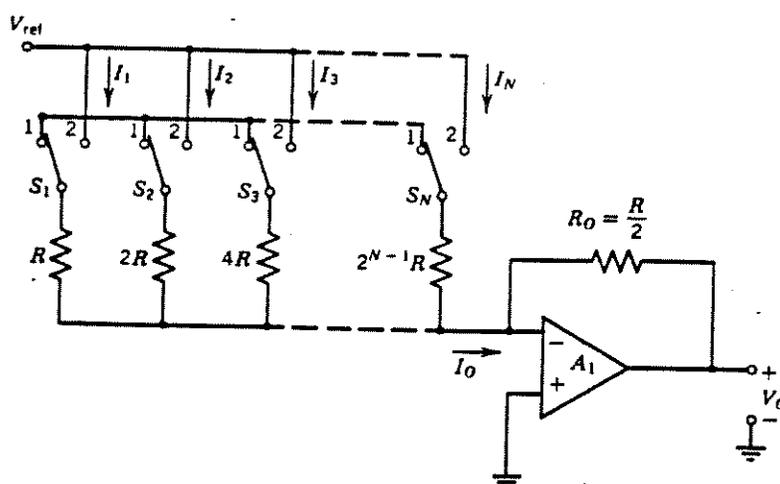


Figura 3.3 - Conversor D/A baseado em escalonamento de corrente.

Em conversores D/A operando sob o princípio de escalonamento de corrente, a conversão é realizada através da geração de um conjunto de correntes binariamente ponderadas, as quais são então somadas de maneira seletiva, para que se obtenha, durante a operação de captura, uma saída analógica. A figura 3.3 mostra uma configuração básica para geração e soma de um conjunto de correntes binariamente ponderadas  $I_1, I_2, \dots, I_N$ . As correntes são geradas conectando-se uma rede de resistores com valores binariamente ponderados a uma tensão de referência  $V_{ref}$ . A corrente de saída  $I_O$  é então soma das correntes  $I_i$  no ponto terra virtual, isto é, na entrada inversora do amplificador operacional  $A_1$ , e a tensão de saída do conversor será dada por:

$$V_O = -I_O R_O = -V_{ref} (b_1 2^{-1} + b_2 2^{-2} + \dots + b_N 2^{-N}) \quad (3.1)$$

As posições das chaves  $S_1, S_2, \dots, S_N$  simulam os valores dos coeficientes  $b_1, b_2, \dots, b_N$  na eq. (3.1). Os coeficientes  $b_i$  assumem valores "0" ou "1", se a chave  $S_i$  correspondente estiver na posição 1 ou 2 respectivamente.

No conversor D/A mostrado na figura 3.3, a soma ponderada de correntes da equação (3.1) é atingida usando-se uma rede de N braços paralelos independentes, cujas impedâncias tem os valores respectivos de  $R, 2R, 4R, 8R$ , etc. Assim, nesse tipo de configuração, a faixa requerida de valores dos resistores usados na rede aumenta muito rapidamente a medida que cresce número de bits utilizada pelo conversor. Por exemplo, para resolução de 8 bits requer-se um conjunto de resistores de precisão cobrindo os valores de  $R$  até  $128R$ , o que dificulta a implementação deste conversor na forma monolítica.

Uma configuração alternativa da rede de resistores, que elimina a necessidade de uma grande faixa de valores de resistência para implementação da rede, é a chamada rede R2R mostrada na figura 3.4. Neste tipo de rede, a divisão binária das correntes é atingido através do particionamento sucessivo de corrente entre os braços paralelo ( $2R$ ) e série ( $R$ ). Dessa forma, as correntes  $I_i$ 's dos braços da rede mantêm a relação binária:

$$I_1 = 2I_2 = 4I_3 = \dots = 2^{N-1} I_{N-1} \quad (3.2)$$

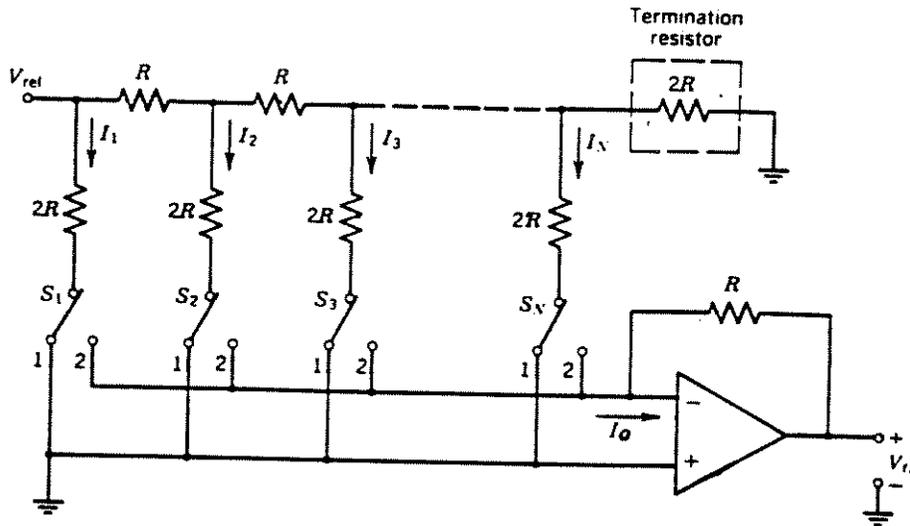


Figura 3.4 - Conversor D/A usando uma rede de resistores R2R

utilizando-se apenas resistores com os valores R e 2R. A desvantagem em relação a configuração de rede anterior é que um número duas vezes maior de resistores é utilizado na confecção da rede.

Nos conversores D/A monolíticos, em geral a escada R-2R é a configuração mais indicada na implementação de conversores com resolução maiores que 6 bits, em razão deste arranjo requerer uma menor faixa de valores dos componentes da rede de resistores. Embora, a configuração de rede de resistores ponderados que é a mais adequada para seções conversores de 4 bits, também possa ser utilizada na implementação de conversores mais complexos através do casteamento de seções.

### 3.1.3 Parâmetros DC de Desempenho dos Conversores A/D e D/A

A seguir discutiremos alguns parâmetros que costuma ser usados para se quantificar o desempenho DC de conversores.

#### Erro de Offset

Para os conversores A/D, este erro é definido como a distância entre o ponto da linha central de código real que cruza o eixo correspondente ao código de menor magnitude e a origem das características de transferência, conforme mostra a figura 3.6. Para os conversores D/A, o erro de offset é o desvio da saída real com relação à saída de um conversor ideal quando a saída ideal deveria estar fixada em zero.

#### Erro de Ganho

Para conversores A/D, erro de ganho é definida como a diferença na inclinação das linhas centrais de código real e ideal, conforme mostra a figura 3.6. Para os conversores D/A, erro de ganho é definido como o desvio relativo da saída analógica, devido a imprecisão do fator de escala ou da tensão de referência.

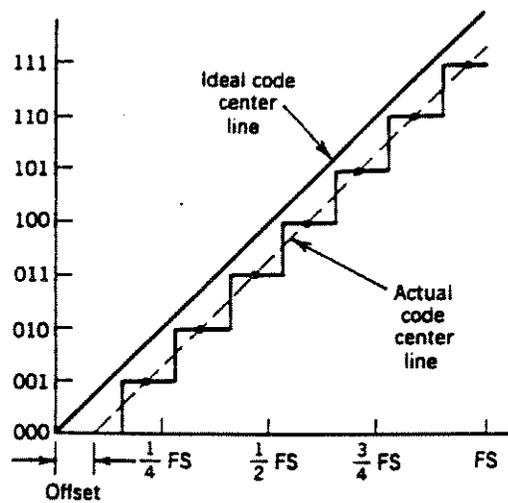


Figura 3.5 - Erro de offset

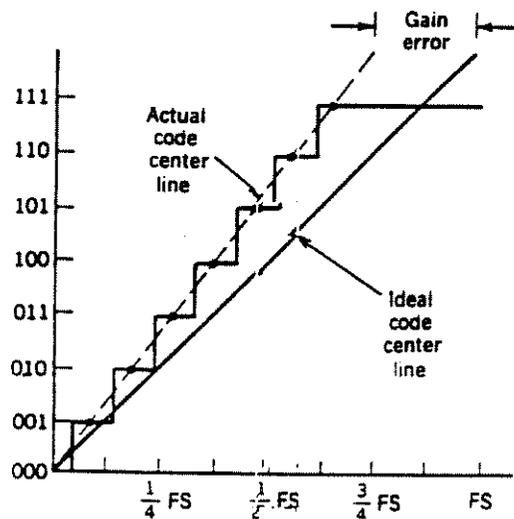


Figura 3.6 - Erro de ganho

### Não-Linearidade Integral

Nos conversores A/D, indica a curvatura da linha central de código real e a linha reta que liga os pontos limites das características de transferência do conversor A/D (ou seja, reta que une o ponto origem ao ponto de leitura da tensão de fundo de escala  $V_{FS}$ ). Este parâmetro é reportado como a diferença entre as duas curvas no pior caso, e costuma ser expresso como uma fração da tensão  $V_{FS}$  ou da tensão correspondente ao LSB (low significative bit) (ver figura 3.7).

Para os conversores D/A, este parâmetro também é uma medida do desvio da função de transferência do conversor a partir de uma linha reta ideal. Ele é definido como o desvio no pior caso entre a característica de transferência do conversor D/A e a linha reta que une a origem ao ponto de fundo de escala do conversor. Para ambos conversores assume-se que os erros de ganho e offset já tenham sido previamente compensados.

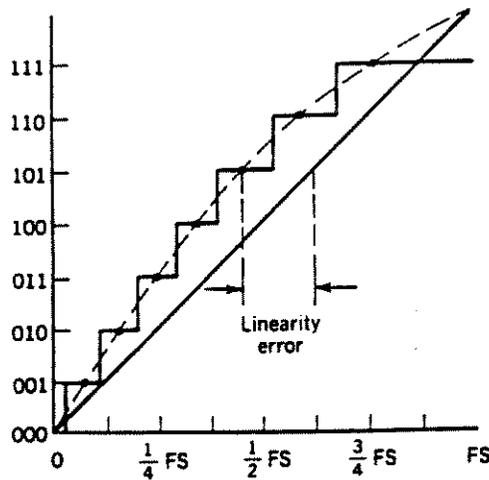


Figura 3.7 - Não linearidade integral

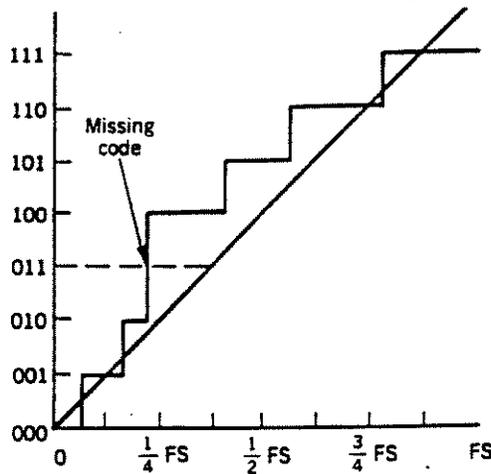


Figura 3.8 - Não linearidade diferencial

**Não-Linearidade Diferencial**

A *não-linearidade diferencial* (ver figura 3.8) avalia o quão uniforme são as distâncias entre transições adjacentes da característica de transferência de um conversor A/D ou D/A. Note que idealmente, o tamanho do passo entre transições equivale a 1 LSB. A *não-linearidade diferencial*  $E_{LD}$ , é portanto, definida como o desvio diferencial do tamanho de cada passo em relação ao seu valor ideal, ou seja:

$$E_{LD} = (\Delta V - \Delta V_{LSB}) / \Delta V_{LSB} \tag{3.3}$$

onde  $\Delta V$  é o tamanho medido em volts de um passo entre transições adjacentes do conversor, e  $\Delta V_{LSB}$  é valor em volts correspondente a 1 LSB .

Por exemplo, se um conversor D/A ou A/D é especificado como tendo uma não-linearidade diferencial máxima de  $\pm \frac{1}{2}$  LSB, então as distâncias máxima e mínima entre transições adjacentes serão  $\frac{1}{2}$  LSB e  $1 \frac{1}{2}$  LSB, respectivamente.

**Monotonicidade**

Um conversor A/D ou D/A possui monotonicidade , se a sua saída aumentar, sempre que sua entrada aumentar. Matematicamente, podemos dizer que uma função contínua é monotônica se a sua derivada primeira for maior ou

igual zero. Para o caso discreto, o mesmo se aplica a diferença primeira, ou seja diferença dos valores assumidos pela saída do conversor entre duas transições sucessivas. Vale ressaltar que a não-monotonicidade de conversor também pode ser interpretada como um resultado de uma não linearidade diferencial excessiva, e que um erro de não-linearidade diferencial menor do que  $\pm 1\text{LSB}$  já garante a monotonicidade do conversor.

### Precisão Relativa

Embora a precisão de um conversor A/D ou D/A possa ser especificada em termos absolutos, desde que se utilize um padrão rastreável por um órgão de padronização nacional ou internacional, em testes de conversores, a precisão dada em termos relativos costuma ser de maior interesse.

A precisão relativa é definida como o erro no pior caso entre as saídas dos conversores real e a ideal, podendo ser expressa como uma percentagem da tensão de fundo de escala, como o número de bits, ou como uma fração do LSB. Assim, uma precisão relativa de N bits significa que o erro possível  $\Delta V_e$  de

$$\Delta V_e \leq \frac{V_{FS}}{2^N} \quad (3.4)$$

Similarmente, para um conversor de N bits, uma precisão relativa de  $\frac{1}{2}$  LSB significa que:

$$\Delta V_e \leq V_{FS} \frac{1}{2} \frac{1}{2^N} = V_{FS} \frac{1}{2^{N+1}} \quad (3.5)$$

## 3.1.4 Conversores A/D e D/A Comerciais

### MC44251- Conversor A/D Triplo de 8 bits para Vídeo

Este CI de fabricação da Motorola, contém 3 conversores A/D do tipo paralelo independentes colocados em um único encapsulamento. Na figura 3.9 é mostrado um diagrama esquemático simplificado deste componente. Cada conversor A/D constitui-se de 256 comparadores/registadores e um circuito codificador. O MC44251 pode ser usado para converter tanto sinais de vídeo em componentes RGB, quanto em componentes YUV. Para enriquecimento do processo de conversão, o MC44251 oferece um gerador de dithering para cada um dos conversores A/D. O gerador de dithering reduz degradações de natureza granular causadas por erros de quantização durante o processo de digitalização. Cada conversor A/D é acionado por um sinal de relógio comum e recebem informação de sincronismo a partir das entradas HZ e VTN. O pino VTN também controla a função de dithering que será desabilitada se colocarmos esta entrada no nível lógico zero. A técnica de dithering utilizada consiste em se adicionar um pequeno nível DC ao sinal de entrada, que é alternado de linha para linha de metade de um LSB.

Este componente pode operar tanto com acoplamento DC quanto com acoplamento AC. Quando o sinal de entrada é acoplado com nível DC, o dispositivo rastreará o nível DC médio da forma de onda de entrada. Para acoplamento ac, um circuito de restauração de nível DC amostra e ajusta o nível dc médio do sinal de entrada. O MC44250 oferece três níveis de clamping para acoplamento ac, selecionados através do pino MODE. No modo RGB, os níveis de clamping são setados para a tensão referente ao nível de decisão 16/256.

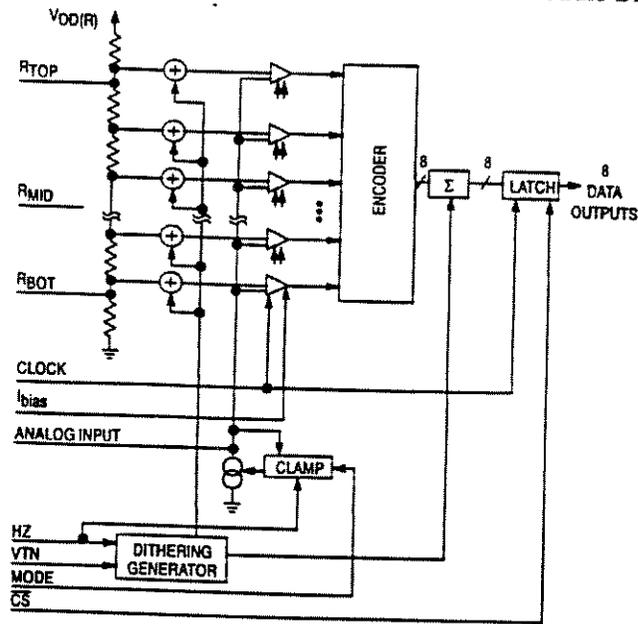


Figura 3.9 - Diagrama esquemático simplificado do ADC MC44251

No modo YUV, para as entradas UV os níveis de clamping são setados para a tensão correspondente ao nível 128/256, enquanto para a entrada Y o clamping é setado para o nível 16/256 ou 64/256.

Na tabela 3.1 são mostrados parâmetros dos conversores A/D do MC44250.

TABELA 3.1 - Parâmetros de Desempenho DC do conversor A/D MC44250

Parâmetro	Max	Unidade
Não-Linearidade Diferencial	±1.0	LSB
Não-Linearidade Integral	±2.0	LSB
Erro de Ganho	±1.0	%
Erro de Offset	±4.0	LSB

Os parâmetros de temporização do MC44251 são mostrados na figura 3.10 e listados na tabela 3.2.

TABELA 3.2 - Parâmetros de Temporização do conversor MC44250

Parâmetro	Máx	Mín
td (atraso na saída analógica)	70ns	-
Frequência de CLK	50MHz	-

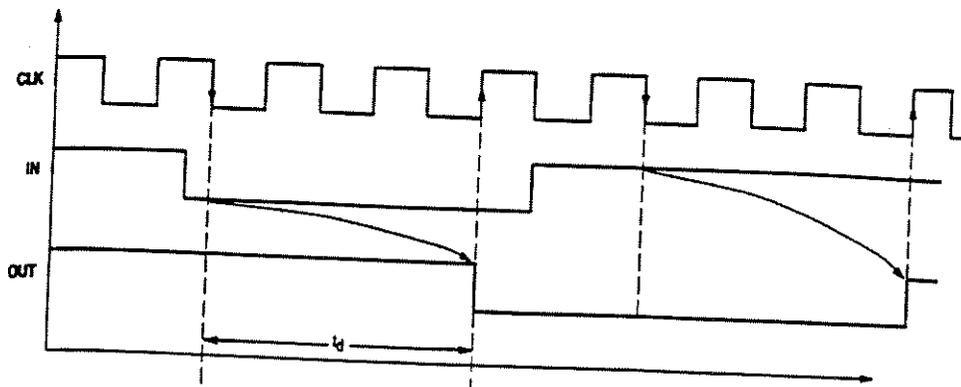


Figura 3.10 - Temporização da operação de conversão

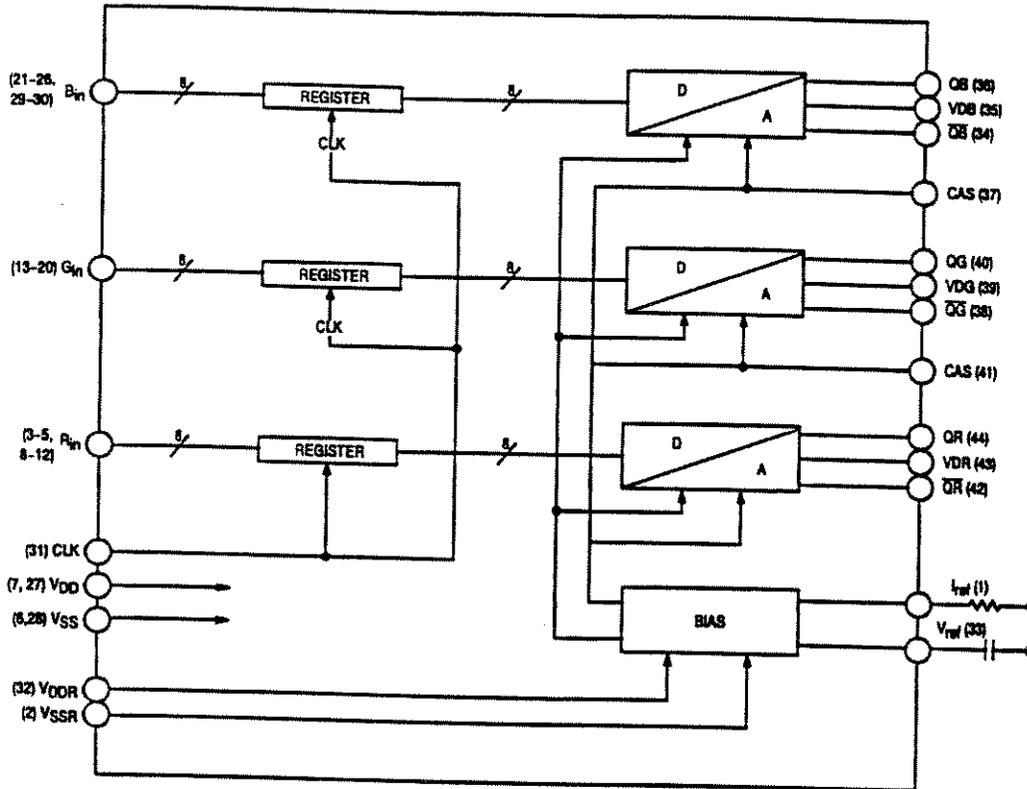


Figura 3.11 - Diagrama de blocos do MC44200

### MC44200 - Conversor D/A Triplo de 8bits para Vídeo

O MCM44200 possui três conversores D/A, acionados por uma entrada de relógio comum. Ele possui uma fonte de referência, mas pode também operar com uma tensão de referência externa utilizando-se a entrada  $V_{ref}$  (pino 33). O diagrama de blocos simplificado do MC44200 é mostrado na figura 3.11. Cada uma das palavras digitais de oito bits de entrada é armazenada em um registrador interno na borda de subida do sinal de relógio e é convertida para um valor de tensão analógico através de uma operação de escalonamento de corrente.

A temporização entre a entrada digital, o sinal de relógio CLK, e a saída analógica é mostrada na figura 3.12 e os valores destes parâmetros de temporização é mostrada na tabela 3.3. Na tabela 3.4 são mostrados os parâmetros DC do conversor D/A.

TABELA 3.3 - Parâmetros de Temporização do conversor MC44200

Parâmetro	Máx	Mín
$t_d$ (atraso na saída analógica)	70ns	-
Frequência de CLK	50MHz	-

TABELA 3.4 - Parâmetros DC do conversor MC44200

Parâmetro	Máx	Mín
Não Linearidade Diferencial	0.5 LSB	-
Não Linearidade Integral	1.0 LSB	-

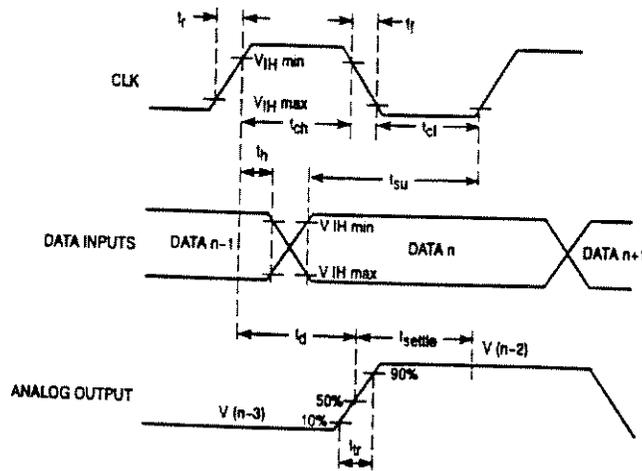


Figura 3.12 - Temporização da saída analógica

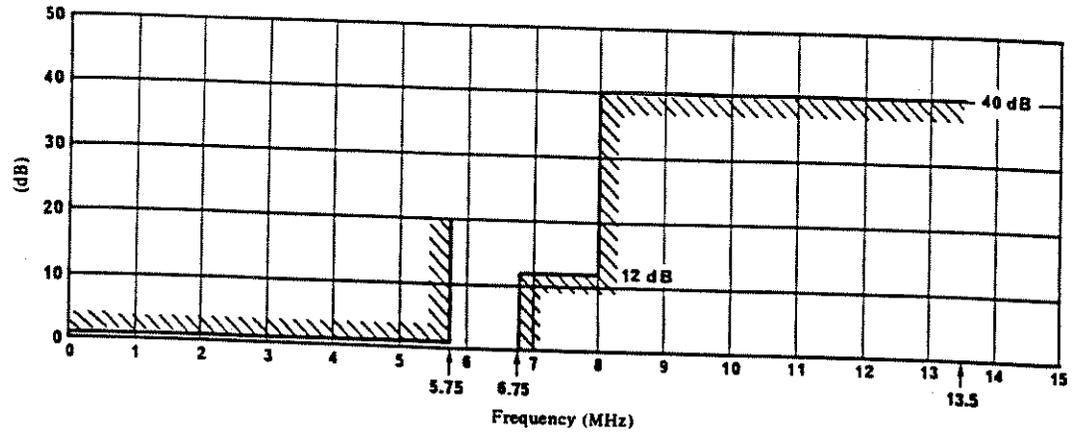
## 3.2 Filtros Anti-Aliasing de Pré e Pós-Processamento

Para uma faixa de passagem de vídeo variando de 0 a 5.5 MHz, uma frequência de amostragem de 13.5 MHz como recomendada pela CCIR 601 esta a cerca de 22% acima do limite de Nyquist . Portanto, filtros analógicos com frequências de corte bem abruptas devem ser utilizados antes e depois dos processos de conversão A/D e D/A para que não ocorram degradações no sinal de vídeo devido a aliasing de componentes espectrais do sinal de vídeo. A importância das características dos filtros, para que se atinja uma alta qualidade no processo de conversão A/D e D/A tem sido reconhecida pelo CCIR através da inclusão na Recomendação 601 de especificações com tolerâncias bem rigorosas para os filtros a serem utilizados no processo de digitalização dos sinais de vídeo em componentes YUV. As especificações CCIR 601 para o filtro a ser utilizado sobre o sinal de luminância são mostradas na figura 3.13.

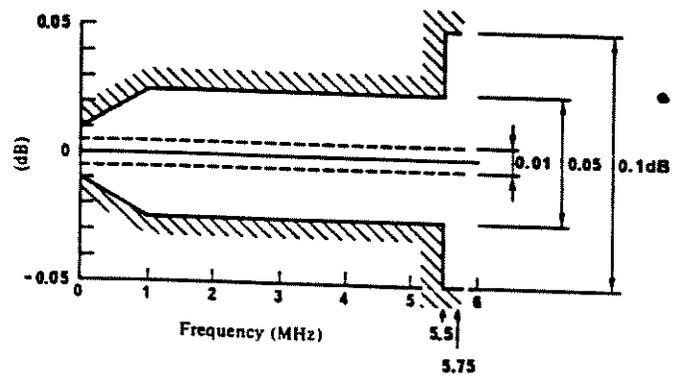
Os problemas práticos de projeto e implementação de filtros aumentam de maneira significativa quando se deseja atingir equalização de atraso de grupo juntamente com frequências de corte elevadas. Uma outra questão a ser considerada é a equalização de amplitude  $\frac{\sin x}{x}$  necessária após o processo de conversão D/A, devido ao fato da reconstrução do sinal não ser feita com pulsos impulsivos ideais, mas sim através de pulsos com uma dada largura  $\tau$ .

Filtros comerciais que atendam as especificações requeridas pela CCIR 601 podem ser encontrados em catálogos de empresas especializadas na confecção de filtros como a Matthey Electronics. A tabela 3.5 a seguir mostra os parâmetros da série LC422 que atende as especificações da Rec. 601 e já tratam as questões de equalização de atraso de grupo e de equalização de amplitude.

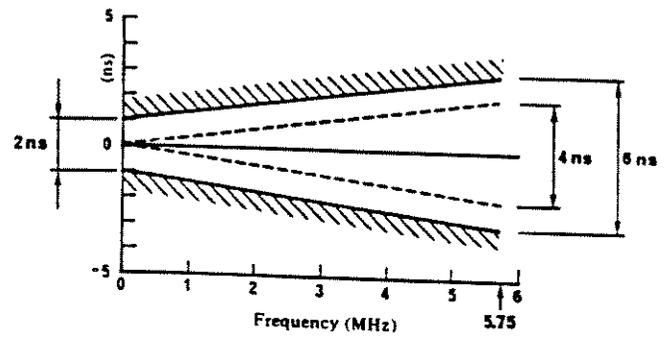
Specification for a luminance or RGB signal filter used when sampling at 13.5 MHz



a) Template for insertion loss/frequency characteristic



b) Passband ripple tolerance



c) Passband group-delay tolerance

Figura 3.13 - Máscara do filtro passa-baixas para Y especificado pela Rec. 601.

TABELA 3.5a - Filtros Matthey - LC422Y e LC422YS  
(Luminância)

	LC422Y	LC422YS
Banda de passagem	5.75MHz	5.75MHz
Ripple na banda de passagem	<0.3dB	<0.3dB
Frequência de atenuação 12dB	6.75MHz	6.75MHz
Início da banda de rejeição	7.6MHz	7.6MHz
Atraso de grupo	<20ns em 5.20MHz	<25ns em 5.20MHz
Perda em 100KHz	<2.0dB	<6.0dB

TABELA 3.5b - Filtros Matthey - LC422Y e LC422YS  
(Crominância)

	LC422U	LC422US
Banda de passagem	2.75MHz	2.75MHz
Ripple na banda de passagem	<0.3dB	<0.3dB
Frequência de atenuação 12dB	3.375MHz	3.375MHz
Início da banda de rejeição	4.0MHz	4.0MHz
Atraso de grupo	<20ns em 2.65MHz	<25ns em 2.65MHz
Perda em 100KHz	<1.5dB	<4.5dB

## 3.3 PLL (Phase-Locked Loop) e Sintetizadores de Frequência

### 3.3.1 Princípio de operação de um Sistema PLL

Um sistema PLL (Phase-Locked Loop) é composto basicamente de três blocos: um detetor de fase, um filtro de malha, e um oscilador controlado à tensão (VCO). Estes três blocos são conectados de maneira a formar um sistema realimentado, como mostra a figura 3.14. O detetor de fase compara a fase do sinal de entrada  $V_s(t)$  com a frequência de saída do VCO e gera uma tensão erro  $V_d(t)$ . Este sinal erro é então filtrado pelo filtro de malha e aplicado ao terminal de controle do VCO na forma da tensão erro  $V_e(t)$ , a fim de controlar sua frequência de operação.

Quando nenhum sinal é aplicado ao PLL, a tensão erro  $V_e(t)$  é igual a zero e o VCO opera em uma frequência estacionária  $\omega_0 = 2\pi f_0$ , conhecida como frequência *free running* do VCO. Se um sinal periódico, com frequência  $\omega_s = 2\pi f_s$  suficientemente próxima da frequência *free running*, é aplicado a entrada do sistema, a realimentação do PLL faz com uma tensão erro seja gerada, forçando a sincronização da frequência de saída do VCO com a frequência de entrada  $f_s$ . Esta situação é conhecida como condição de *locking* do PLL.

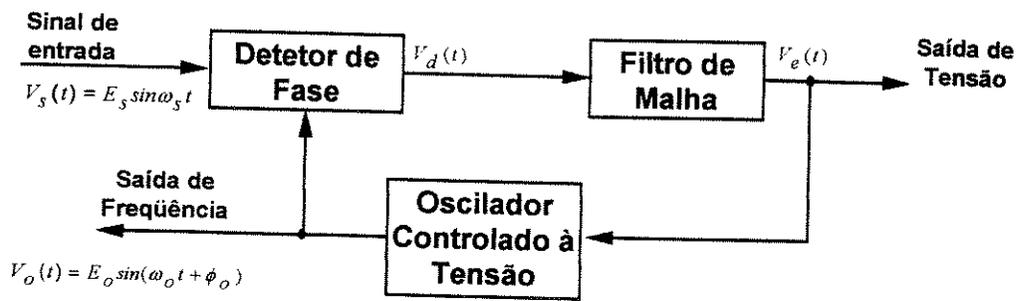


Figura 3.14 - Diagrama de blocos de um sistema PLL

Em condição de locking, a frequência do VCO é idêntica a frequência da entrada  $f_s$ , a menos de uma pequena diferença de fase  $\phi_0$ . Este erro de fase é necessário para gerar a tensão de erro  $V_e(t)$  que desloca a frequência do VCO de  $f_0$  para  $f_s$ , mantendo desta forma o locking do PLL. Se a frequência de entrada varia lentamente, o PLL é capaz de manter-se em locking rasterando o sinal de entrada através da geração de um erro de fase adicional, o qual produzirá a mudança em  $V_e(t)$  necessária para correção da frequência do VCO. A faixa de frequência sobre a qual o sistema PLL pode manter o locking com o sinal de entrada é definida como o *faixa de locking* do sistema. Ressalta-se que esta difere da faixa de frequências sobre a qual o PLL pode adquirir a condição de locking com o sinal de entrada, esta última conhecida como *faixa de captura* do sistema. Por razões que serão discutidas mais adiante, a faixa de captura é sempre menor que a faixa de locking, e a esta diferença entre as duas faixas esta relacionada com as características do filtro da malha PLL.

A saída de um sistema PLL pode ser obtida tanto a partir do sinal de tensão erro filtrada  $V_e(t)$ , quanto do sinal de frequência produzido no terminal de saída do oscilador VCO. A saída via tensão de erro é utilizada em aplicações de discriminação de frequência, como ocorre em demoduladores FM, ao passo que a saída via VCO é utilizada em aplicações como condicionamento de sinal, recuperação de relógio, e síntese de frequência. Esta última que veremos em detalhe mais adiante.

### 3.3.2 O Transiente do Processo de Captura

A captura, ou seja o processo pelo qual a frequência do VCO adquire o sincronismo com a frequência de entrada, possui intrinsecamente uma natureza não-linear. Tal fato não permite analisar o transiente de captura através de um tratamento matemático simples. Entretanto, uma descrição qualitativa do processo pode ser dada como se segue.

Consideremos, inicialmente, que a malha do PLL está aberta entre a saída do filtro e a entrada de controle do VCO. O que faz a tensão-erro de saída reduzir-se artificialmente a zero, e o VCO operando em sua frequência free-running  $f_0$ . Assumamos, então, que um sinal com frequência  $f_s$  muito próxima a  $f_0$  seja aplicado à entrada do PLL. Como o detetor de fase normalmente funciona como um multiplicador, teremos em sua saída duas componentes de frequência: a componente-soma, ou seja,

$$f_{soma} = f_0 + f_s \quad (3.6)$$

e a componente-diferença,

$$\Delta f = |f_0 - f_s| \quad (3.7)$$

Normalmente, a componente-soma é eliminada pelo filtro passa-baixa do PLL, enquanto a componente diferença, se  $f_s$  for suficientemente próxima de  $f_0$ , aparecerá na saída de tensão do PLL como um batimento senoidal, conforme mostra a porção mais à esquerda da figura 3.15.

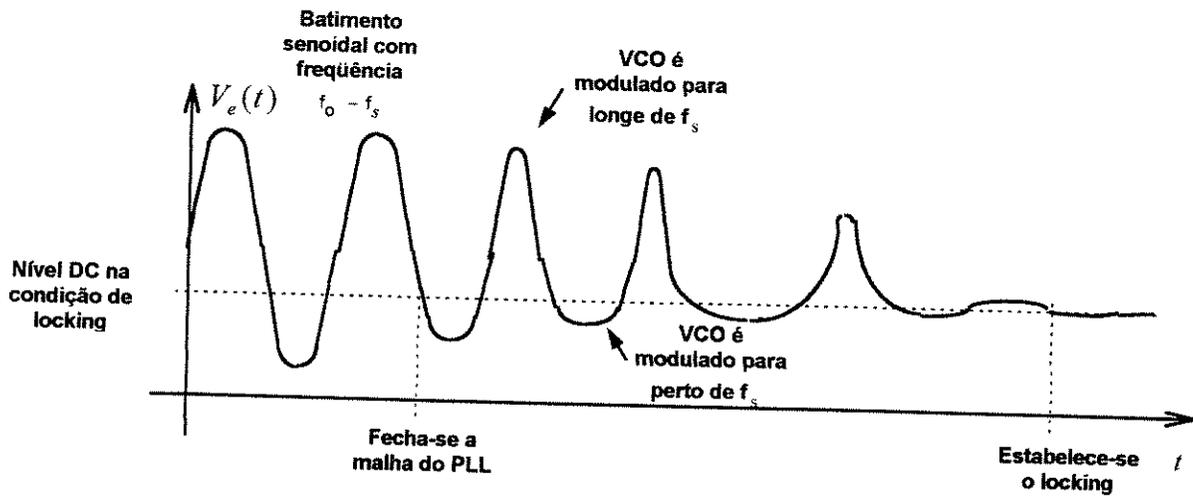


Figura 3.15 - Transitório da tensão erro durante a captura

Fechemos então a malha do PLL, conectando-se a saída de seu filtro passa-baixas à entrada de controle de seu oscilador VCO. Isto fará com que a frequência do VCO seja agora modulada pelo sinal diferença de frequência  $\Delta f$ . Quando isto ocorre o batimento  $\Delta f$  torna-se uma função do tempo. E dado que a frequência do VCO é agora variante com o tempo, esta alternadamente se moverá para mais perto e mais longe da frequência de entrada  $f_s$ . Assim, como a tensão de erro filtrada  $V_e(t)$  é proporcional à diferença das frequências do VCO e do sinal de entrada, esta também assumirá alternadamente meio-ciclos “positivos” e “negativos” a fim de aumentar e diminuir a frequência do VCO. Conforme mostra a porção central da figura 3.15, esta série de ciclos da tensão  $V_e(t)$  não se parece com o batimento senoidal observado anteriormente, mas sim uma forma de onda de ciclos assimétricos, onde os meio-ciclos que modulam o VCO em direção à frequência de entrada possuem uma aparência mais suave, enquanto os meio-ciclos que levam o VCO para longe da frequência de entrada apresentam picos mais acentuados. Em razão da assimetria, esta forma de onda contém uma componente DC finita que força a frequência do VCO a estabilizar-se com a frequência de entrada a menos de um pequeno erro de fase, estabelecendo-se desta forma, a condição de locking.

O tempo total gasto pelo PLL para estabelecer a condição de locking é conhecido como *pull-in time*. O pull-in time depende das diferenças de fase e frequência iniciais dos dois sinais, bem como do ganho de malha aberta e da banda de passagem do filtro passa-baixas. Sob certas condições, o pull-in time pode ser menor que um período do batimento e o PLL pode atingir a condição de locking sem o transiente oscilante mostrado na figura 3.15.

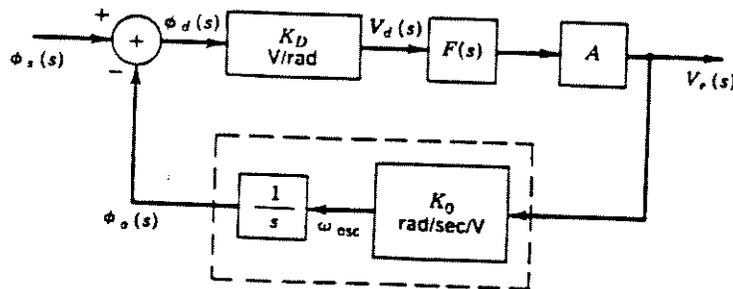


Figura 3.16 - Modelo linear de PLL

### 3.3.3 O Modelo Linear do PLL

Em condição de locking, os transientes não lineares do processo de captura não estão mais presentes, e o PLL pode ser analisado como um sistema servo linear através do uso de técnicas da teoria clássica de controle.

A figura 3.16 mostra o diagrama de blocos do sistema PLL como um sistema linear realimentado no domínio  $s (= \sigma + j\omega)$  de frequência complexa. O detector de fase fornece uma tensão de saída proporcional a diferença de fase  $\phi_d$  entre o sinal de entrada e a saída do VCO, com um ganho de conversão de  $K_D$  V/rad. A função de transferência do filtro é dada por  $F(s)$ , e  $A$  é o ganho do amplificador (incluído aqui para manter o modelo mais geral). O ganho de conversão tensão-frequência do VCO é dado por  $K_0$  rad/(sec.V). A frequência do VCO é proporcional à tensão de erro  $V_e$  e é dada por:

$$\omega_{osc} = \omega_0 + K_0 V_e \quad (3.8)$$

onde  $\omega_0$  é a frequência free-running do VCO. Entretanto, o detector de fase é sensível a diferenças de fase, e a fim de se manter a coerência da análise, a saída do VCO deve ser em termos de fase ao invés de frequência. Como a fase é definida como a integral da frequência, ou seja:

$$\phi_o(t) = \phi_o(t=0) + \int_0^t \omega_o(t) dt, \quad (3.9)$$

o bloco VCO atua como um integrador e sua função de transferência pode ser expressa como:

$$\phi_o(s) = \frac{K_0 V_e}{s} \quad (3.10)$$

Assim, a função de transferência  $H(s)$  do sistema servo será dada por:

$$H(s) = \frac{V_e(s)}{\phi_s(s)} = \frac{sK_D A F(s)}{s + K_D K_0 A F(s)} \quad (3.11)$$

onde  $\phi_s$  é a fase do sinal de entrada. Se combinarmos todas as constantes de ganho definindo a constante de ganho total em malha aberta  $K_{MA}$  como,  $K_{MA} = K_D K_0 A$ , podemos expressar a função de transferência como:

$$H(s) = \frac{V_e(s)}{\phi_s(s)} = \frac{K_{MA}}{K_0} \frac{sF(s)}{s + K_{MA}F(s)} \quad (3.12)$$

Normalmente também é interessante analisar a resposta do PLL a variações da frequência do sinal de entrada ao invés de sua fase. Assim, como a frequência é a derivada temporal da fase, o que no domínio  $s$  pode ser expresso como:

$$\Delta\omega_s(s) = s\phi_s(s) \quad (3.13)$$

Teremos então, em termos de variação de frequência, a função de transferência do PLL reescrita como<sup>1</sup>:

$$\frac{V_e(s)}{\Delta\omega_s(s)} = \frac{1}{s} \frac{V_e(s)}{\phi_s(s)} = \frac{K_{MA}}{K_0} \frac{F(s)}{s + K_{MA}F(s)} \quad (3.14)$$

### 3.3.4 Efeitos do Filtro Passa-Baixas

Valendo-se da terminologia da teoria de análise linear de sistemas realimentados podemos definir a função transferência em malha aberta do PLL,  $A(s)$ , como se segue:

$$A(s) = \frac{K_{MA}F(s)}{s} \quad (3.15)$$

Sabemos que as características em malha fechada de um sistema realimentado estão relacionadas à performance em malha aberta, dado que a função de transferência  $H(s)$  relaciona-se com  $A(s)$  como se segue:

$$H(s) = \frac{1}{K_0} \frac{A(s)}{1 + A(s)} \quad (3.16)$$

e que as raízes do polinômio característico  $1 + A(s)$ , as quais determinam o transitório e o regime da operação do PLL, são prontamente determinadas através de técnicas baseadas no lugar das raízes (LR).

#### Filtro de ordem zero

O caso mais simples de operação de um PLL é aquele em que o filtro passa-baixas é removido totalmente, ou seja  $F(s) = 1.0$ . Neste caso, a função de transferência em malha fechada do sistema PLL reduz a:

$$H(s) = \frac{1}{K_0} \frac{1}{1 + s/K_L} \quad (3.17)$$

e a malha do PLL constitui um sistema de primeira ordem.

A figura 3.17 mostra o lugar das raízes e a resposta em frequência de um PLL com filtro de ordem zero. O pólo de malha aberta na origem do LR deve-se à integral executada pelo VCO. Nesta situação, o sistema

<sup>1</sup> Vale ressaltar entretanto, que na análise que se segue trataremos apenas a resposta do PLL a variações de fase do sinal de entrada.

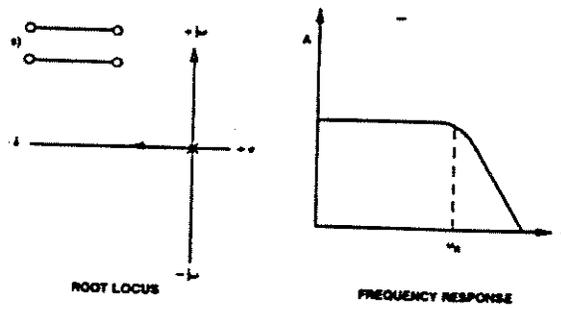


Figura 3.17 - LR de um PLL com filtro de ordem zero

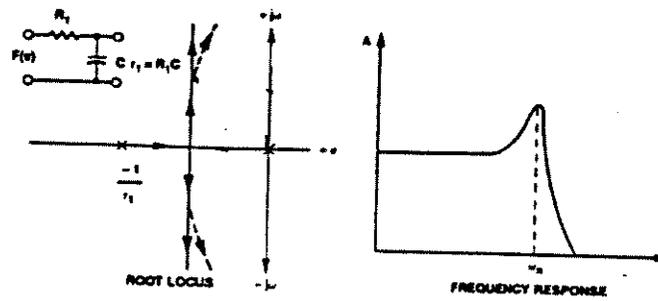


Figura 3.18 - LR de um PLL com filtro de ordem 1

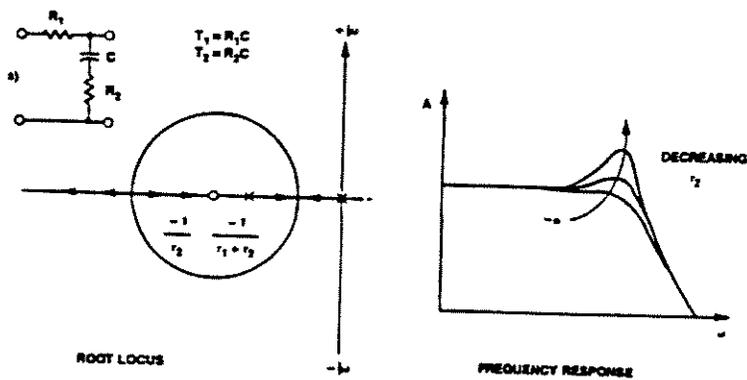


Figura 3.19 - LR de um PLL com filtro avanço-atraso

comporta-se como um filtro passa-baixas de pólo único com ganho DC de  $1/K_L$  e banda de passagem 3dB,  $K_L$ .

Esta configuração tem seu uso bastante limitada devido a sua baixa capacidade de rejeição à interferências. Sem o filtro passa-baixas na malha, todas as componentes de alta frequência presentes na saída do detector de fase aparecem na saída do PLL. Dentre estas componentes incluem-se tanto àquelas devido a ruído quanto àquelas devido a sinais indesejáveis na entrada do PLL com frequência próximas à frequência do sinal de interesse. Ou seja, as características de seletividade do PLL são bastante degradadas.

### Filtro de Ordem 1

Com a adição de um filtro passa-baixas de pólo único com  $F(s)$  dada por:

$$F(s) = \frac{1}{1 + s/\omega_1} \quad (3.18)$$

onde  $\omega_1 = 1/R_1 C_1$ , o PLL torna-se um sistema de segunda ordem com função de transferência  $H(s)$  como se segue:

$$H(s) = \frac{1}{K_0} \frac{1}{s^2/\omega_n + (2\zeta/\omega_n)s + 1} \quad (3.19)$$

onde  $\omega_n = \sqrt{K_L \omega_1}$  e  $\zeta = \frac{1}{2} \sqrt{\frac{\omega_1}{K_L}}$  são respectivamente a frequência natural e o fator de amortecimento do sistema.

O lugar das raízes e a resposta em frequência do PLL de segunda ordem são mostrados na figura 3.18. Novamente, um pólo de malha aberta é colocado na origem do LR devido ao integrador do modelo VCO, e um outro pólo de malha aberta é colocado em  $\omega_1$  devido à ação do filtro passa-baixas. Algumas observações podem ser feitas a partir deste LR:

- A medida que o ganho  $K_{MA}$  aumenta, a parte imaginária dos pólos de malha fechada aumenta, e o sistema além de tornar-se mais sub-amortecido ( $\zeta < 1$ ), também sofre um aumento em sua frequência natural.
- Se a frequência de corte do filtro  $\omega_1$  diminui, a parte real dos pólos de malha fechada diminui, o que leva a uma redução no fator de amortecimento do sistema.

Como em qualquer outro sistema realimentado real, deslocamentos excessivos ou pólos não-dominantes associados a outros blocos do PLL podem fazer com que o lugar das raízes mova-se em direção ao semi-plano direito como mostra a linha pontilhada da figura 3.18, o que levaria o sistema à instabilidade caso valores muito altos do ganho de malha  $K_{MA}$  ou valores muito pequenos da frequência de corte do filtro passa-baixas  $\omega_1$  fossem utilizados.

### Filtro Avanço-Atraso

O problema da estabilidade pode ser eliminado através do uso de um filtro do tipo avanço-atraso como o indicado na figura 3.19. Este filtro tem a função de transferência dada por :

$$F(s) = \frac{1 + s/\omega_2}{1 + s/\omega_1} \quad (3.20)$$

onde  $\omega_1 = 1/(R_1 + R_2)C_1$  e  $\omega_2 = 1/(R_2 C_1)$ . Escolhendo-se  $R_2$  de maneira adequada, este tipo de filtro confina o lugar das raízes no semi-plano esquerdo assegurando assim a estabilidade do sistema. O filtro avanço-atraso permite sistemas com uma resposta intermediária àquelas obtidas pelos sistemas de primeira e segunda ordem, além de fornecer um controle adicional sobre a resposta transiente do sistema. Assim com  $R_2 = 0$ , o sistema comporta-se como um sistema de segunda ordem e quando  $R_2 \rightarrow \infty$ , o sistema se comporta com um sistema de primeira ordem devido ao cancelamento de um pólo por um zero.

### Cálculo das faixas de locking e de captura

Como já mencionado, define-se faixa de locking do PLL, como a faixa de frequências, centrada em  $f_0$ , sobre a qual o PLL pode rastrear a frequência do sinal de entrada, dado que já se encontra em condição de locking.

A faixa de locking de um PLL,  $\pm\Delta\omega_L$ , é em princípio determinada pela quantidade de tensão de erro  $V_e$  que pode ser gerada dentro do PLL, ou seja:

$$\pm\Delta\omega_L = \pm(V_e)_{max} K_0 \quad (3.21)$$

A tensão de erro máxima  $(V_e)_{max}$ , por sua vez, depende do ganho do bloco detector de fase  $K_D$ , do ganho do estágio de amplificação  $A$ , e do erro de fase máximo  $(\phi_d)_{max}$  que o detector de fase é capaz de fornecer, ou seja:

$$\pm(V_e)_{max} = \pm(\phi_d)_{max} K_D A \quad (3.22)$$

Como o erro de fase máximo detectado pelos circuitos detectores de fase mais comumente usados em PLL monolíticos é :

$$(\phi_d)_{max} = \pm\frac{\pi}{2} rad \quad (3.23)$$

Pode-se, então expressar-se a faixa de locking do PLL dada em rad/s como:

$$\Delta\omega_L = \frac{\pi}{2} K_D A K_0 = \frac{\pi}{2} K_L \quad (3.24)$$

Em termos de frequência dada em Hertz, simplifica-se a expressão para:

$$\Delta f_L = \frac{\Delta\omega_L}{2\pi} = \frac{K_L}{4} \quad (3.25)$$

Note que na determinação da faixa de locking, assumiu-se o PLL em regime com um sinal de erro DC dentro da malha. Portanto, as características do filtro passa-baixas  $F(s)$  não entram no equacionamento, dado que sob estas condições pode-se que o filtro passa-baixas possui ganho unitário para as frequências muito baixas próximas a componente DC.

A faixa de captura por sua vez, é definida como a faixa de frequências de entrada centrada em  $f_0$  sobre a qual o PLL pode adquirir a condição de locking. Vale lembrar que a faixa de captura  $\Delta f_L$  é sempre menor que a faixa de locking se o filtro passa-baixa for utilizado. O que faz o PLL apresentar uma característica de histerese, de modo que o PLL é capaz de manter-se em locking com o sinal de entrada sobre uma faixa mais larga do que àquela em que pode vir a capturá-lo. As razões qualitativas para estes fenômeno serão discutidas a seguir.

Se assumirmos inicialmente que a malha é aberta na saída do estágio amplificador, e que um sinal de frequência  $f_s$  é aplicado à entrada do PLL, teremos na saída do PLL um batimento senoidal dado por:

$$V_e(t) = \frac{\pi}{2} K_D A |F(j\Delta\omega_i)| \cos(\Delta\omega_i t) \quad (3.26)$$

onde  $\Delta\omega_i = |\omega_s - \omega_0|$  é a frequência do batimento, e  $|F(j\Delta\omega_i)|$  é a magnitude da resposta do filtro na frequência do batimento. A amplitude desta tensão erro filtrada  $V_e$  é portanto:

$$V_e|_{pico} = \frac{\pi}{2} K_D A |F(j\Delta\omega_i)| \quad (3.27)$$

Para que ocorra a captura é necessário, aplicar-se ao terminal de controle do VCO uma de tensão de magnitude  $|V_{osc}|$  suficiente para deslocar sua frequência de oscilação de  $\Delta\omega_i$  em relação a  $f_0$ , ou seja:

$$|V_{osc}| = \frac{\Delta\omega_i}{K_0} = \frac{|\omega_s - \omega_0|}{K_0} \quad (3.28)$$

Como uma estimativa aproximada, podemos considerar que a captura ocorre somente se o valor de pico da tensão de erro  $V_e|_{pico}$  for igual ou um pouco maior que a tensão  $|V_{osc}|$  requerida para o deslocamento do frequência do VCO. Portanto a condição de captura pode ser aproximada por:

$$\frac{\Delta\omega_i}{K_0} \approx \frac{\pi}{2} K_D A |F(j\Delta\omega_i)| \quad (3.29)$$

Entretanto, na frequência em que ocorre a captura,  $\Delta\omega_i = \Delta\omega_c$ , dada a definição de faixa de captura. E assim sendo podemos estimar a faixa de captura do PLL como:

$$\Delta\omega_c \approx \frac{\pi}{2} K_L |F(j\Delta\omega_c)| \quad (3.30)$$

ou equivalente em termos da faixa de locking :

$$\Delta\omega_c \approx \Delta\omega_L |F(j\Delta\omega_c)| \quad (3.31)$$

Uma vez que a magnitude da resposta em frequência do filtro é sempre menor que 1, note que a faixa de captura estimada é sempre menor que a faixa de locking do PLL, que esta diferença torna-se mais significativa a medida que a banda de passagem do filtro é reduzida.

### 3.3.5 O CI PLL NE 564

O NE 564 é um PLL monolítico capaz de operar com frequências de até 50 Mhz, sendo portanto atraente para aplicações em vídeo. Como mostra o diagrama de blocos da figura 5.20, ele é composto por um oscilador VCO, um comparador de fase, um limitador de entrada, e um processador de pós-deteção.

Parâmetros-chave do PLL, como o ganho de malha, a frequência free-running do VCO, a faixa de locking, a faixa de captura, o fator de amortecimento, e a banda de passagem em malha fechada, são definidos através de componentes externos a fim de que se possa otimizar a performance do circuito para uma aplicação específica.

Assim, o ganho de malha pode ser ajustado através da corrente  $I_2$  injetada no pino 2 do NE564, que altera a constante de conversão  $K_D$  do detector de fase, sendo portanto possível ajustar a faixa de locking do PLL através da corrente  $I_2$ .

Quanto a frequência free-running do VCO, ela é determinada pelo capacitor externo  $C_1$  colocado entre os pinos 12 e 13, podendo ser aproximada por:

$$f_0 \approx \frac{1}{22R_C C_1} \quad (3.32)$$

onde  $R_C = 100\Omega$  é um valor definido internamente.

O filtro passa-baixas do PLL pode ser implementado através da adição de capacitores e resistores externos entre pinos 4 e 5 do CI, conforme mostra a figura 3.21. Um filtro de primeira ordem, por exemplo, pode ser implementado, adicionando-se um par de capacitores  $C_3$  e a função de transferência  $F(s)$  será dada por:

$$F(s) = \frac{1}{1 + sRC_3} \quad (3.33)$$

onde  $R = 1.3 \text{ K}\Omega$  também é definido internamente.

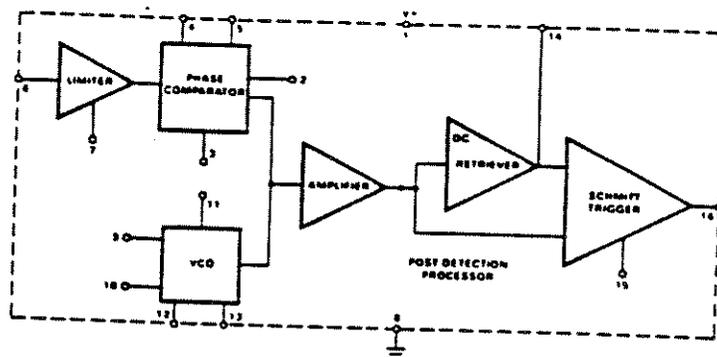


Figura 3.20 - Diagrama de Blocos do CI PLL NE564

### 3.3.6 Síntese de Frequência usando PLL

Inserindo-se um circuito divisor de frequência em sua malha de realimentação, um PLL pode funcionar como um multiplicador de frequência. Um diagrama de blocos desta configuração é mostrada na figura 3.22. Aqui, a malha é quebrada entre o terminal de saída do VCO e o detector de fase para que um contador de módulo N seja inserido. Neste caso,  $f_s$  é a frequência na saída do divisor que entrará em locking com a frequência de entrada  $f_s$ , enquanto o oscilador VCO operará na realidade com uma frequência que é N vezes maior que esta mesma frequência de entrada  $f_s$ . Uma aplicação prática desta propriedade de multiplicação de frequência do PLL, e na qual estamos aqui particularmente interessados é o uso do PLL para obtenção dos chamados sintetizadores de frequência. Em nosso caso em particular, estamos interessados em obter um sintetizador que gere um relógio de amostragem de 13.5 MHz perfeitamente sincronizado com a frequência horizontal do sinal de vídeo.

O NE564 pode ser utilizado na implementação de um sintetizador como mostra a figura 3.21.

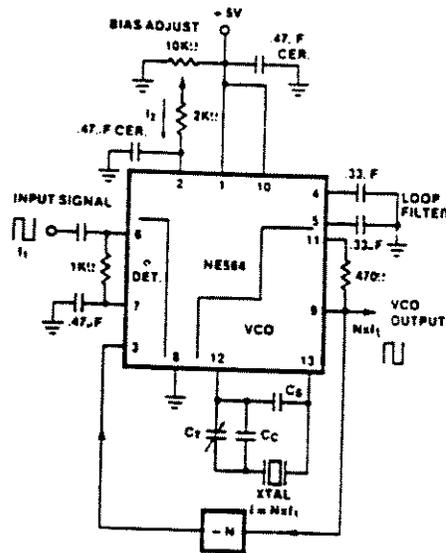


Figura 3.21 - Sintetizador de frequência baseado no PLL NE564

## 3.4 Amplificadores e Processadores de Sincronismo de Vídeo

### 3.4.1 O Separador de Sincronismo LM1881

LM1881 é um CI de fabricação da National Semiconductors que extrai informações de sincronismo de uma fonte de vídeo composto analógico. O LM1881 fornece esta informação de sincronismo por meio de quatro sinais principais: um sinal de sincronismo composto contendo informação de temporização horizontal e vertical; um sinal de sincronismo vertical; um pulso *porta de salva* ocorrendo na janela de tempo em deveria estar a salva de cor do sinal de vídeo composto; e uma saída *campo impar/par* que indica se o campo corrente é um campo-impar ou um campo-par. Na figura 3.22 é mostrado diagrama esquemático simplificado do LM1881. As formas de ondas dos sinais de sincronismo searados pelo LM1881 são mostrados na figura 3.23.

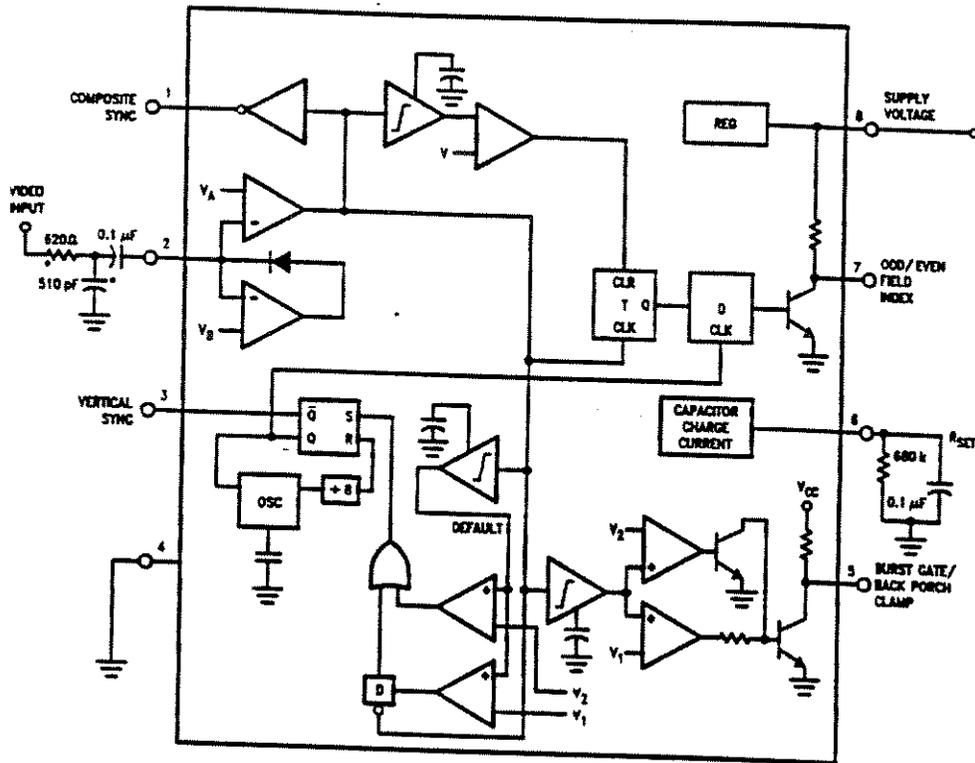


Figura 3.22 - Diagrama de blocos simplificado do LM1881

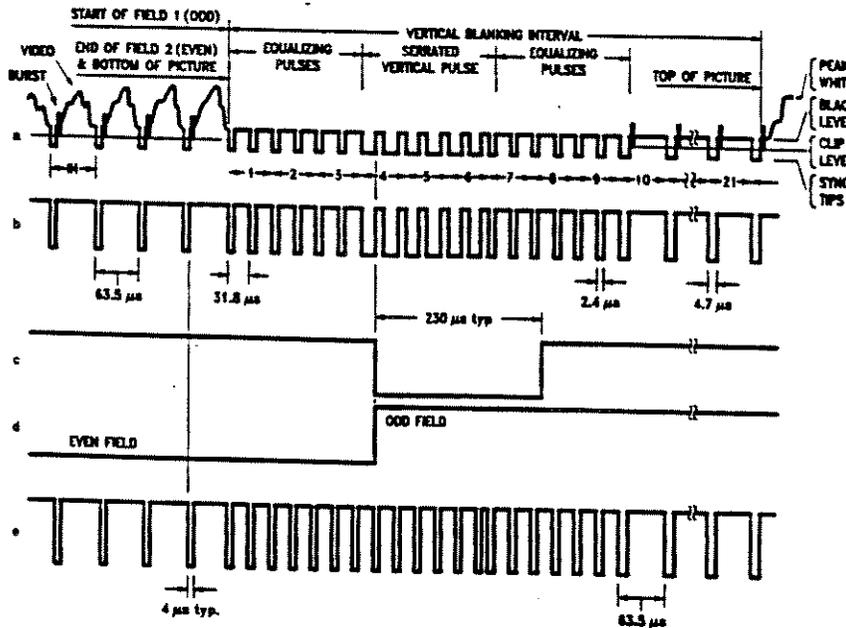


Figura 3.23 - Formas de Onda do LM1881

### Saída de Sincronismo Composto

A saída de sincronismo composto é uma reprodução da forma de onda do sinal de vídeo composto considerando-se apenas a porção abaixo do nível preto. Isso é feito deslocando-se os picos dos pulsos de sincronismo para o valor de  $1,5V_{DC}$  no pino 2, e usando-se um comparador com uma tensão de limiar um pouco acima deste valor para extrair apenas o pulso de sincronismo. A saída do comparador passa então por um buffer inversor e é fornecida no pino 1.

## Saída de Sincronismo Vertical

Uma saída de sincronismo vertical é obtida a partir da integração da forma de onda de sincronismo composto. A seção responsável pela geração do pulso de sincronismo vertical é mostrada na figura 3.22 (ver canto inferior esquerdo desta figura). Note que há dois comparadores nesta seção. Um comparador tem uma tensão de referência gerada internamente chamada  $V_1$  ligada a uma de suas entradas. O outro comparador tem uma outra tensão de referência gerada internamente chamada  $V_2$  (maior que  $V_1$ ) ligada a uma de suas entradas. Os dois comparadores possuem uma entrada em comum: as entradas não-inversoras são acionadas pelo integrador interno da seção. O integrador interno tem como entrada o mesmo sinal de sincronismo composto que é aplicado ao buffer de saída do pino 2, formado por série de pulsos positivos. O capacitor deste integrador é interno e sua corrente de carga e portanto a constante de tempo do integrador é regulada através do resistor externo  $R_{SET}$  ligado ao pino 6. A saída do integrador apresenta um valor de tensão baixo (ou melhor, menor que as tensões  $V_1$  e  $V_2$ ) durante as linhas horizontais normais dada a sua elevada constante de tempo. Durante os primeiros pulsos de equalização do intervalo de apagamento vertical, a tensão de saída do integrador ainda mantém-se em torno deste mesmo abaixo da referência  $V_1$ . Terminada a primeira seqüência de 6 pulsos de equalização segue-se a seqüência de 6 seqüência de pulsos serradores (ou pulsos de sincronismo vertical). O pulso serrador do sinal de sincronismo composto aplicado à entrada do integrador permanecem positivo por um tempo maior que os pulsos equalizadores e de sincronismo horizontal. Assim a partir do início do primeiro pulso serrador o integrador carrega seu capacitor a um valor de tensão bem mais alto. A saída do integrador durante este pulso assume um valor de tensão entre  $V_1$  e  $V_2$ , o que leva saída do comparador que tem  $V_1$  como referência a um valor alto. Este valor alto é então armazenado no flip-flop D, durante a borda de descida do primeiro pulso serrador. A saída Q do flip-flop D passa por uma porta OR e seta o flip-flop RS. A saída Q do flip-flop RS aciona um oscilador interno. O sinal deste oscilador, ligada a entrada reset do flip-flop RS, passa por um circuito divisor por oito, e realiza um reset neste flip-flop após oito ciclos de oscilação. A saída  $\bar{Q}$  do flip-flop RS é o sinal de sincronismo vertical real do LM1881 fornecida no pino 3.

O LM1881 também fornece um pulso de sincronismo vertical default para os casos em que o período de sincronismo vertical é mais longo que o usual no qual não são utilizados pulsos serradores. Com um tempo de sincronismo vertical maior, o integrador tem tempo para carregar o seu capacitor interno a um valor acima da tensão de referência  $V_2$ . Uma vez que neste caso não ocorrerá a borda de descida do pulso serrador para disparar o flip-flop D, o único sinal capaz de setar o flip-flop RS é passa a ser a saída do comparador default com referência  $V_2$ .

## Pulso Campo Ímpar/Par

Para detectar campos pares e ímpares o LM1881 novamente integra a forma de onda de sincronismo composto. Um capacitor é carregado durante o período existente entre pulsos de sincronismo, e é descarregado durante a ocorrência de tais pulsos. O período existente entre pulsos de sincronismo horizontal é suficiente para permitir que a tensão no capacitor atinja o valor limiar de um comparador cuja saída força um reset em flip-flop tipo T, que é também disparado pela forma de onda de sincronismo composto. Durante o intervalo de sincronismo vertical, o período entre os pulsos de sincronismo passa a ser menor e a saída do integrador não atinge mais o

valor limiar. Assim, a saída Q do flip-flop T é invertida a cada pulso de equalização. Uma vez que o período entre pulsos de sincronismo, para a última meia linha do campo ímpar, terá o mesmo efeito que um período entre pulsos de equalização, terá uma polaridade diferentes para campos sucessivos. Portanto comparando a polaridade da saída Q com o pulso de sincronismo vertical de saída do LM1881, pode-se gerar um índice para um campo par ou ímpar. O pino 7 permanece no nível baixo durante o campo par e em nível alto durante o campo ímpar.

### Pulso Porta para Salva de Cor

No sinal de vídeo composto, a salva de cor está localizada no *backporch* do período de apagamento horizontal. O backporch dura aproximadamente 4.8µs e é também o nível preto de referência para a próxima linha de varredura. O LM1881 gera um pulso que pode ser usado para recuperar a salva de cor ou ser usada em um circuito restaurador de nível DC do sinal de vídeo. Este pulso é fornecido no pino 5 e é obtido através da carga de um capacitor disparada na borda de término de um pulso de sincronismo horizontal. O pino 5 assume o nível baixo quando do término do pulso de sincronismo assim permanece por cerca de 4µs, retornando então ao nível alto.

### 3.4.2 O Gerador de Sincronismo LM1882

O LM1882 é um gerador de sincronismo de vídeo na forma de circuito integrado capaz de fornecer sinais de sincronismo vertical, horizontal e composto para televisores e monitores. Pulsos de equalização e serradores também podem ser incluídos no sinal de sincronismo composto. O diagrama de blocos simplificado e a pinagem do dispositivo são mostrados na figura 3.24. Note que se trata de um dispositivo programável com flexibilidade suficiente para suportar vários formatos de vídeo, a serem especificados a partir dos dados escritos pelo usuário em seu conjunto de 18 registradores de 12 bits. Embora completamente programável, é possível utilizar-se em uma configuração default que gera sinais de sincronismo segundo a especificação RS-170. Em nosso caso, limitaremos-nos a utilizá-lo em sua configuração padrão.

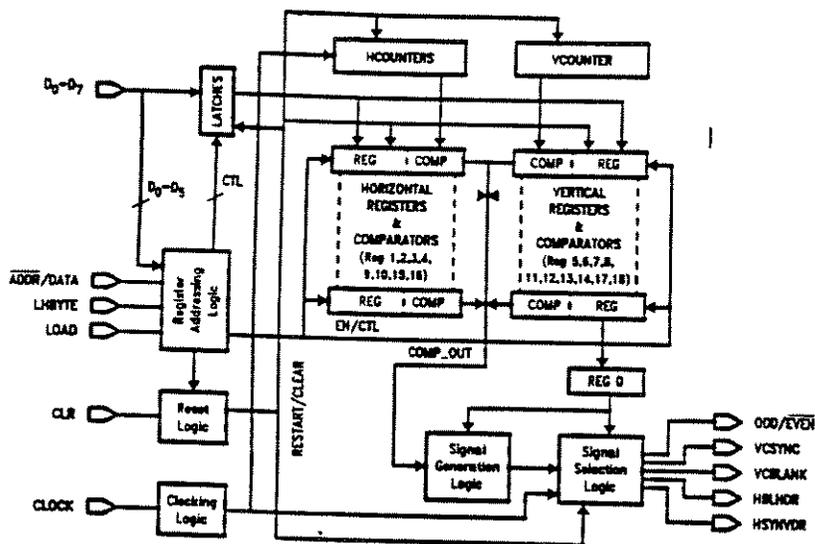


Figura 3.24 - diagrama de blocos do LM1882

### Descrição da Pinagem do LM1882

- **Entradas de Dados D0-D7:** Pinos de entrada de dados conectados ao registrador de endereço e aos registradores de dados.
- **ADDR-L/DATA-H:** Sinal que determina se o dado presente nas entradas de dados é um endereço (0) de registrador ou um dado (1) de configuração a ser escrito em um registrador de dados.
- **L/HBYTE:** Sinal que indica se o dado nas entradas D0-D7 constituem-se nos 8 bits menos significativos (0) ou nos 4 bits mais significativos do registrador de dados.
- **LOAD:** Sinal de controle que controla a armazenagem dos dados apresentados às entradas D0-D7.
- **CLOCK:** Entrada de relógio a partir do qual toda a temporização é derivada. Para configuração default deve ser utilizado um relógio de 14.31818.
- **CLR:** Inicializa o dispositivo quando assume o valor HIGH. A inicialização consiste em além de inicializar todos os contadores, comparadores e o registrador de endereços, fazer com que os registradores de dados assumam os valores que determinam a operação do dispositivo em sua configuração default.
- **ODD/EVEN:** Indica se o campo corrente é par ou ímpar.
- **VCSYNC:** Na configuração padrão fornece um sinal de sincronismo composto.
- **VCBLANK:** Na configuração padrão fornece um sinal de sincronismo vertical.
- **HBLHDR:** Na configuração padrão fornece um sinal de gating horizontal.
- **HSYNVDR:** Na configuração padrão fornece um sinal de sincronismo horizontal.

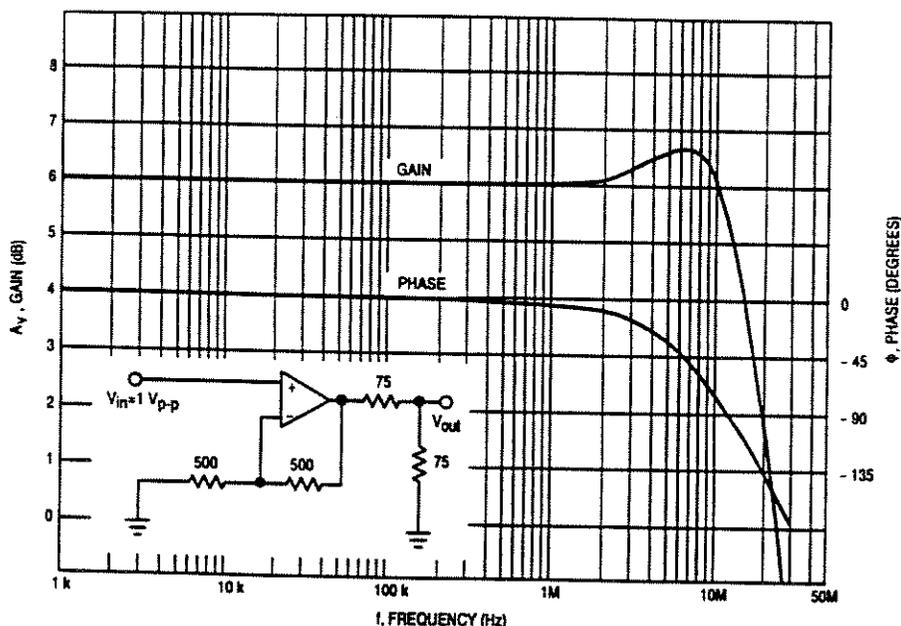


Figura 3.25 - Resposta em frequência do amplificador de vídeo MC14577

### 3.4.3 MC14577 - Amplificador de Vídeo

O MC14577 é composto de dois amplificadores operacionais com compensação em frequência interna otimizados para aplicações em vídeo. Um estágio de saída de alta corrente foi incorporado aos amplificadores de modo que eles são capazes de acionar cargas de 150 Ω de maneira direta. A resposta em malha fechada do MC14576 é mostrada na figura 3.25.

## 3.5 Projeto do Digitalizador de Vídeo

### 3.5.1 Diagrama de Blocos do Subsistema Digitalizador de Vídeo

A figura 3.26 mostra o diagrama de blocos do módulo digitalizador de vídeo. Ele é composto pela unidade de de entrada de vídeo, pela unidade de saída de vídeo, e pelos processadores de sincronismo.

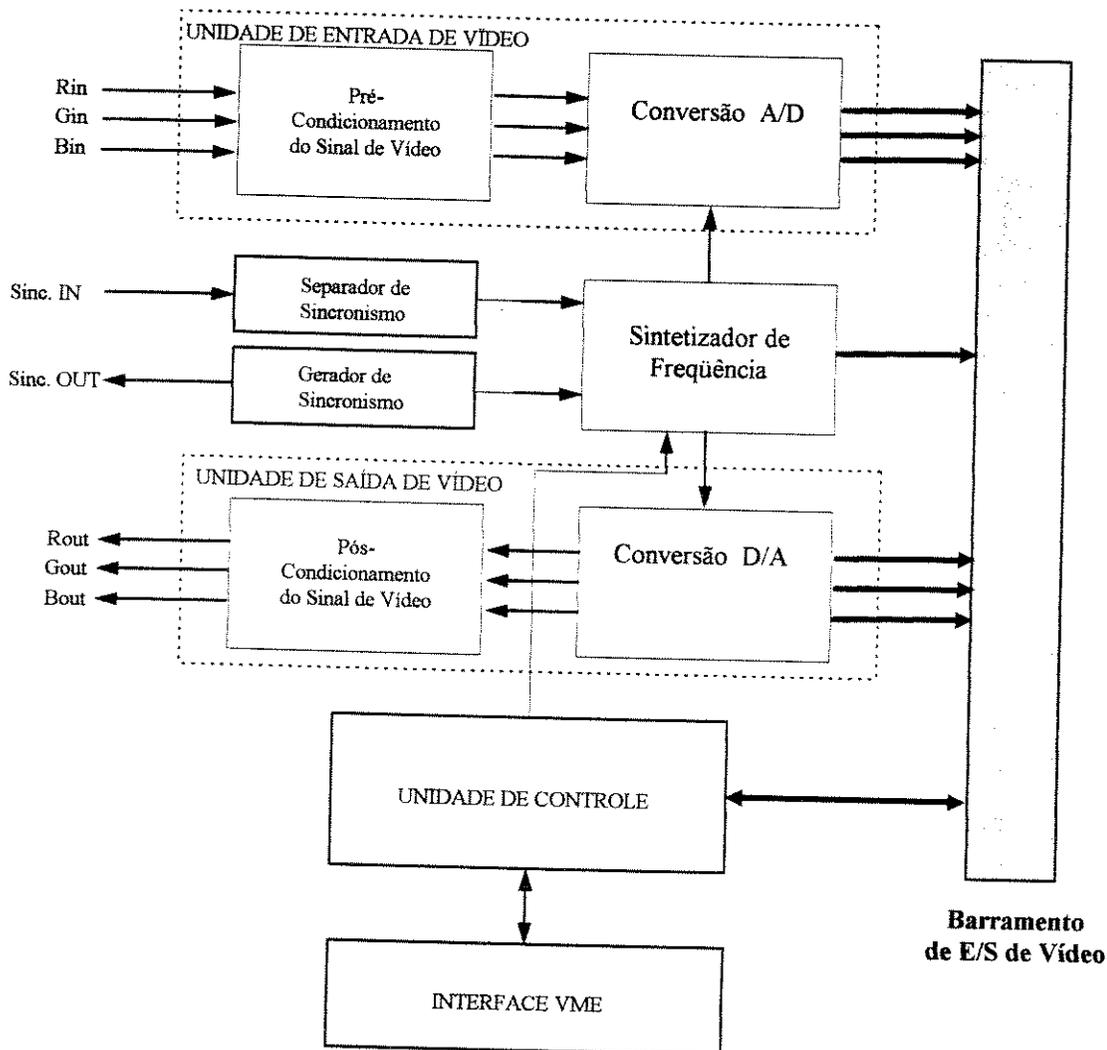


Figura 3.26 - Diagrama de Blocos de uma possível implementação do digitalizador

### **3.5.2 Unidade de Entrada de Vídeo**

A unidade de entrada de vídeo realiza a conversão A/D do sinal de vídeo analógico. É composta pelos conversores A/D, filtros passa-baixas e amplificadores de vídeo necessários para compensar as perdas nos filtros e para realizar o condicionamento da faixa de valores assumidos na entrada dos conversores A/D. A figura 3.27 detalha o esquema elétrico para uma possível implementação da unidade de entrada de vídeo.

### **3.5.3 Unidade de Saída de Vídeo**

A unidade de saída de vídeo é a responsável pela conversão D/A do sinal de vídeo digital. O esquema elétrico de uma possível implementação para esta unidade é mostrada na figura 3.28.

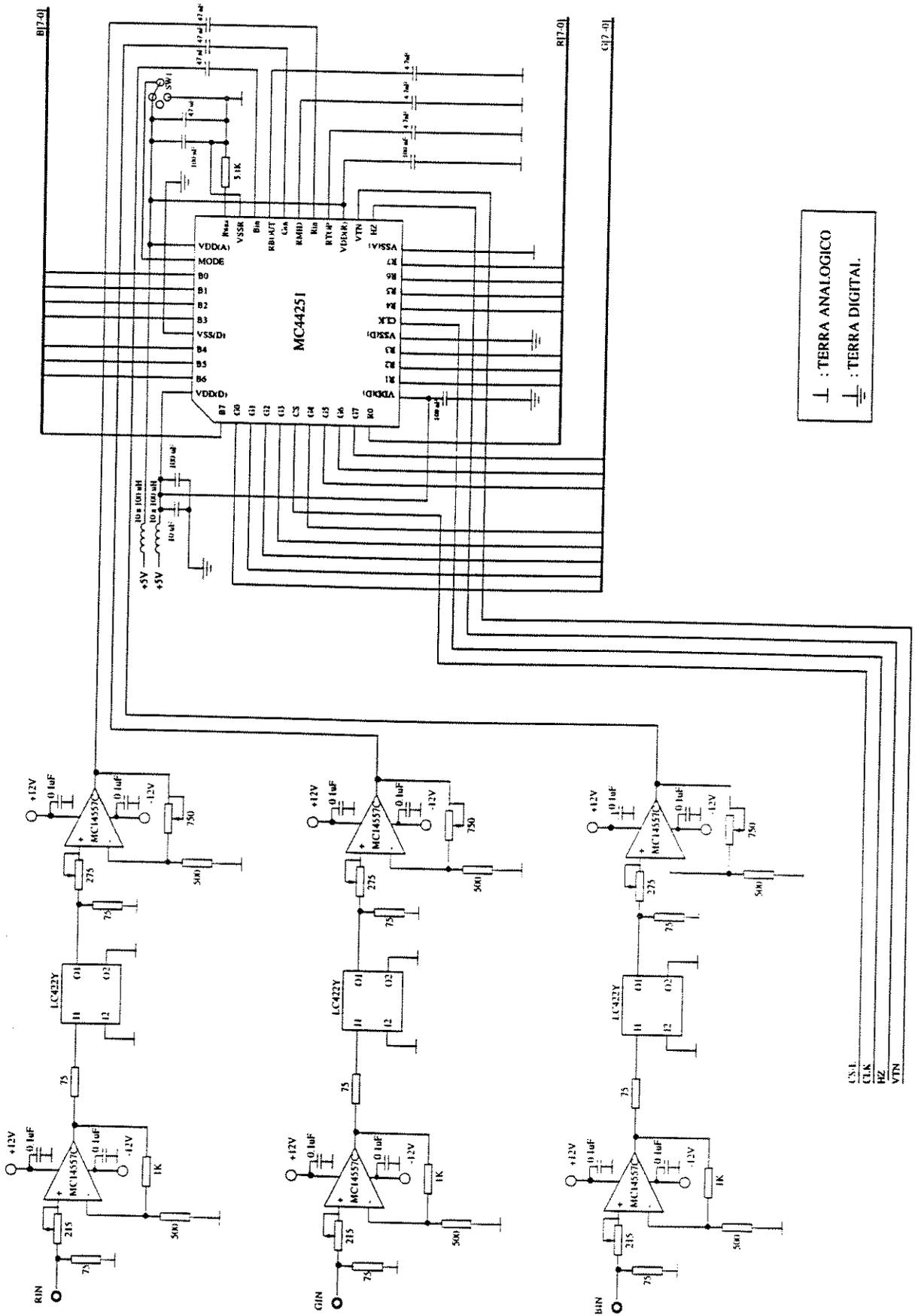


Figura 3.27 - Unidade de entrada de vídeo

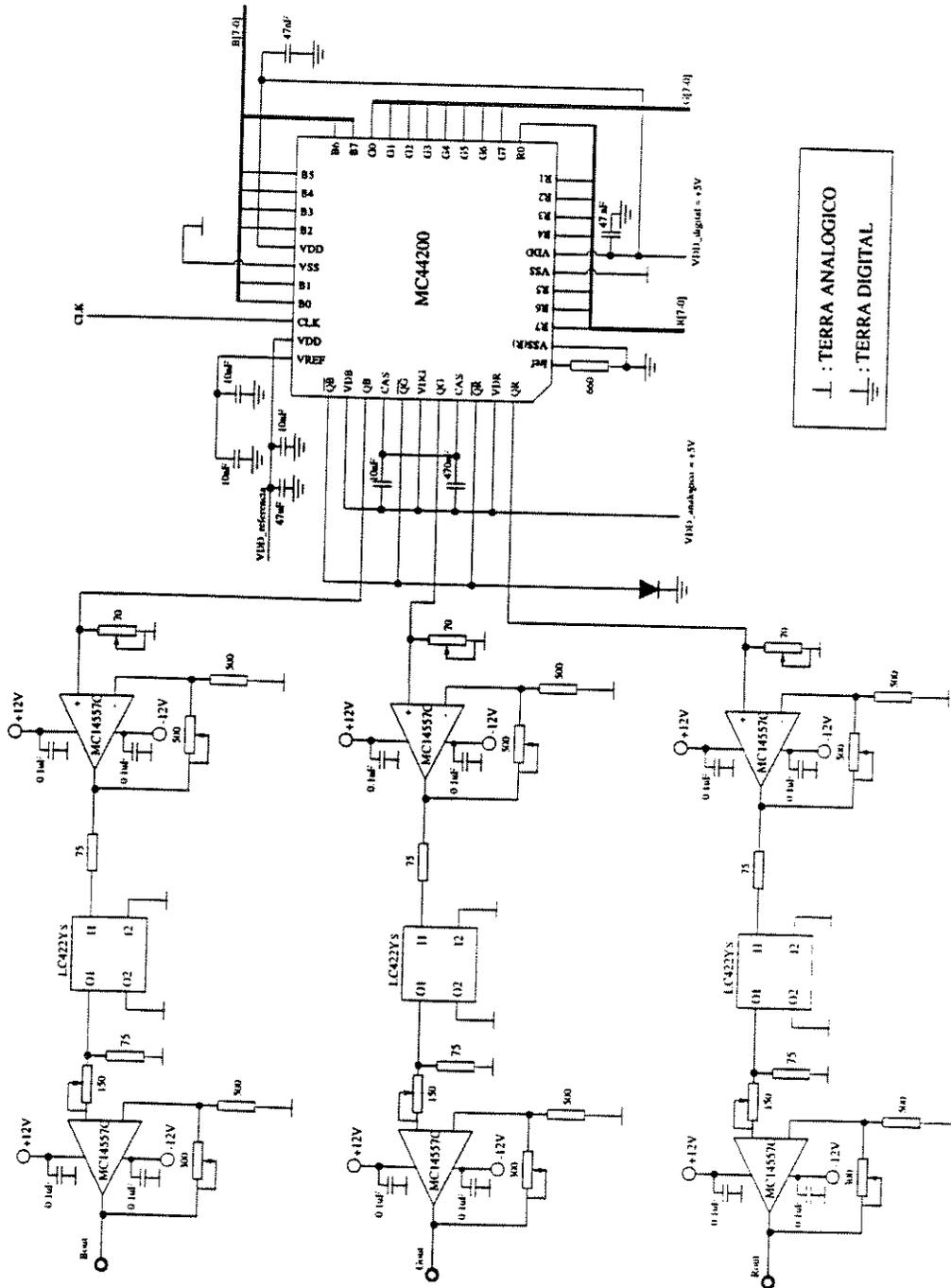


Figura 3.28 - Unidade de saída de vídeo

### 3.5.4 Processadores de Sincronismo/ Sintetizadores de frequência

A figura 3.29 ilustra uma possível implementação para a unidade processadora de sincronismo. O esquema elétrico do sintetizador de frequência é mostrado na figura 3.30

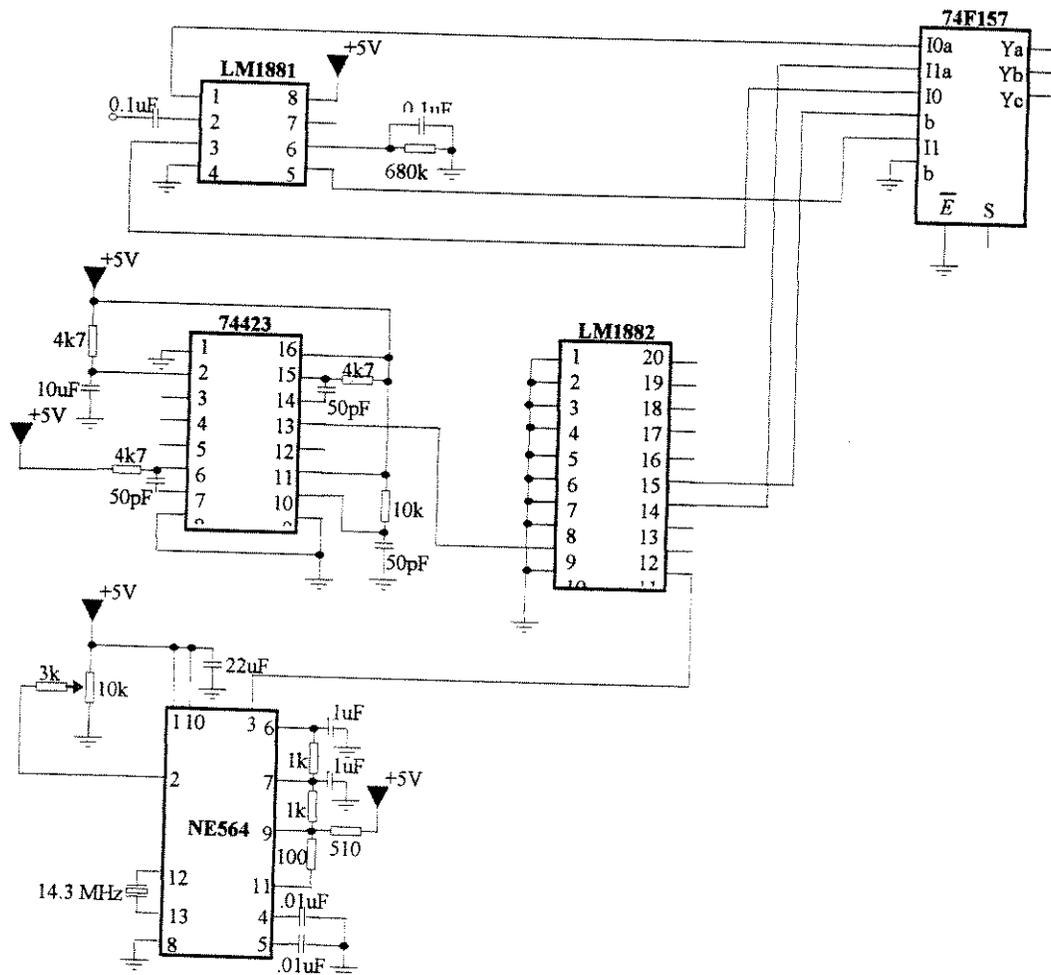


Figura 3.29 - Processadores de Sincronismo

**Figura 3.30** - Esquema elétrico do sintetizador de frequências

## 3.6 Bibliografia

- [1] Grebene A.B., "Bipolar and MOS Analog Integrated Circuit Design", Jonh Willey & Sons, 1984.
- [2] Sandbank C. P., "Digital Television", Jonh Willey & Sons, 1990.
- [3] Sarno P. E., "Interfaces de Entrada e Saída para um Codec de Vídeo", Tese Mestrado - UNICAMP, 1989
- [4] Egan W. F., "Frequency Syntesis by Phase Lock", Jonh Willey & Sons, 1981.
- [5] Signetics "Modeling the PLL", Signetics , 1988
- [6] Signetics "AN180 Frequency Syntesis with NE564", Signetics , 1988.
- [5] Motorola "Multimedia Device Data", Motorola Inc., 1995.
- [6] National "Linear Products Vol 3", National, 1990

# CAPÍTULO 4

## ARMAZENADOR DE QUADROS BASEADO EM MEMÓRIA SEMICONDUCTORA

No capítulo anterior discutimos como o sinal de vídeo analógico proveniente de uma câmera de TV pode ser convertido para o domínio digital, e vice-versa. O passo seguinte então é determinar-se uma maneira de armazenar o fluxo de dados de taxa elevada gerado pelo processo de captura de uma seqüência de quadros de vídeo. Conforme já discutimos, este é o mecanismo a ser utilizado para vencermos o gargalo existente no barramento de expansão dos PC's, sem fazer uso de compressão digital.

Os dispositivos de memória a serem utilizados nos vídeo digital são as memórias semicondutoras, já normalmente utilizadas na implementação dos sistemas de memória primária dos computadores pessoais, ou na implementação do chamado "refresh" de tela executado pelas placas controladoras de vídeo, ou seja:

- Memórias de escrita e leitura dinâmicas (DRAM)
- Memórias de escrita e leitura estáticas (SRAM)
- Memórias de vídeo (VideoRAM)
- Memórias Série (Field Memory)

A escolha do dispositivo a ser utilizado para a presente aplicação deve ser determinada basicamente por dois fatores:

- Qual a quantidade de memória requerida para armazenar a seqüência de quadros, ou seja da densidade e do custo do componente e;
- Quais são as características funcionais oferecidas pelo dispositivo de memória, que facilitam o seu interfaceamento com os conversores A/D e D/A do digitalizador de vídeo.

O presente capítulo é organizado da seguinte maneira: as primeiras quatro seções, discutem características dos dispositivos de memória acima mencionados, assim na seção 4.1 são tratadas as memórias dinâmicas, e uma atenção especial é dada as suas características de temporização. A seção 4.2 considera as memórias estáticas, e as seções 4.3 e 4.4 abordam dois tipos de memórias de uso específico para vídeo: a VideoRAM e a memória série Field Memory. As informações fornecidas nestas seções fornecem subsídios para o a tomada de decisões quando do projeto do módulo armazenador de quadros. As seções 4.5 e 4.6 descrevem um esboço inicial para o projeto de um módulo armazenador de quadros de vídeo baseado em memórias semicondutoras DRAM, capaz de interfacear-se tanto com o digitalizador descrito no capítulo 4, quanto com o barramento VME. O uso de memórias DRAM, conforme detalharemos mais adiante, deve-se principalmente ao menor custo e a maior densidade deste tipo de dispositivo.

## 4.1 Memórias dinâmicas - DRAM

As memórias dinâmicas armazenam informação na forma de carga elétrica em um capacitor formado pela capacitância intra-eletródica de um transistor MOS. Este capacitor possui perdas, e conseqüentemente, requer algum mecanismo de “refreshing” para restaurar periodicamente a carga armazenada antes que ela se degrade, e se perca a informação. Este arranjo embora inconveniente do ponto de vista funcional tem suas razões pelo fato de permitir dispositivos de maior densidade e menor custo.

Chips de memória de alta densidade exigem em razão do grande número de pinos de endereçamento requeridos, encapsulamentos fisicamente cada vez maiores. A solução encontrada pelos fabricantes de semicondutores para amenizar este problema foi multiplexar o barramento de endereços dos chips DRAM. Assim, um chip de 1M x 1 que necessitaria de um barramento de endereço de 20 bits pode ser encapsulado em um componente com somente 10 pinos de endereço.

A multiplexagem do barramento de endereços, embora possibilite a redução do número de pinos do encapsulamento, requer dois sinais adicionais para controlar o armazenamento do endereço da célula de memória desejada. Por exemplo, no caso do chip de 1M x 1, a armazenagem dos 10 bits menos significativos do endereço, conhecido como “endereço-linha”, é controlada pelo sinal  $\overline{\text{RAS}}$  (row address strobe). Os 10 bits mais significativos, que constituem o chamado endereço-coluna, são armazenados sob supervisão do sinal  $\overline{\text{CAS}}$  (column address strobe).

As figuras 4.1 e 4.2 mostram o arranjo interno e a pinagem de um chip de memória 1M x 1.

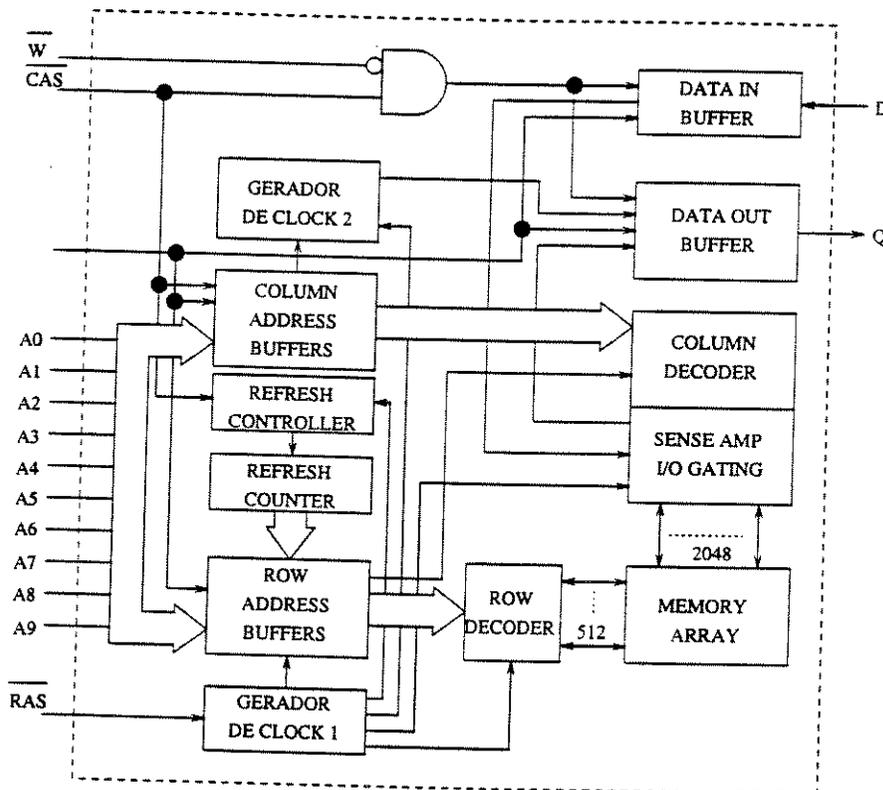


Figura 4.1 - Organização interna de um DRAM de 1Mx1

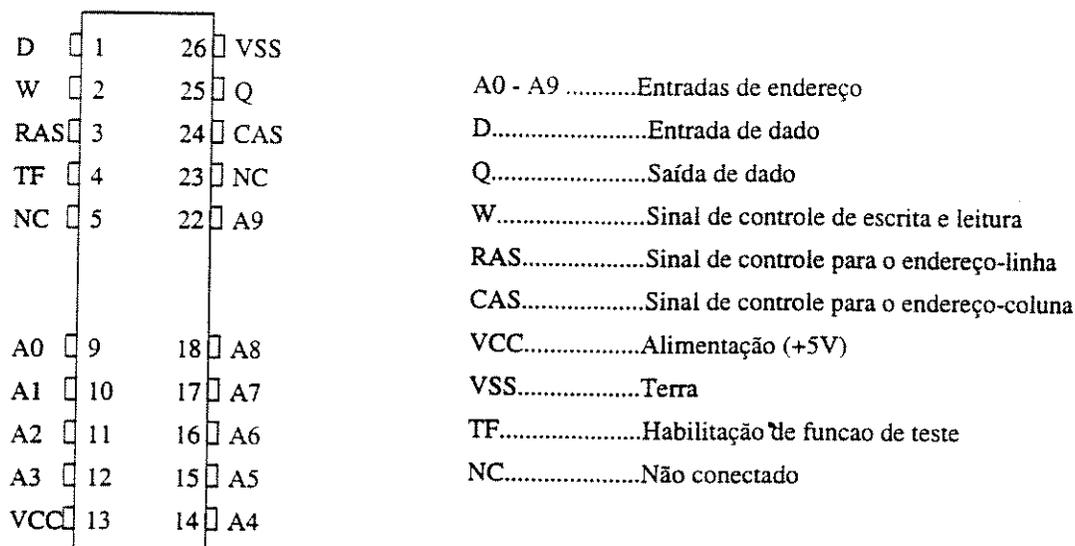


Figura 4.2 - Pinagem de um chip DRAM de 1Mx1

### 4.1.1 O ciclo de leitura convencional

A figura 4.3 mostra a versão simplificada do ciclo de leitura de um chip DRAM. O ciclo vai do instante “A” até instante “B” e tem duração mínima de  $t_{RC}$  segundos<sup>1</sup>. Numa breve explicação sobre as operações ilustradas na figura 4.3 diríamos que :

- A primeira operação a realizar-se é colocação do endereço-linha nos pinos de endereço do chip e o sinal RAS no estado “ativo-baixo” informando a lógica de controle interna do chip DRAM que o endereço-linha é válido.
- O endereço-coluna é então colocado no barramento de endereços e o sinal  $\overline{CAS}$  é colocado no estado “ativo-baixo”. Neste ponto o endereço completo está disponível a lógica interna do chip DRAM e;
- A célula de memória especificada é acessada e o dado requerido será colocado nos terminais de saída do dispositivo.

Nos instantes C e D os sinais de controle  $\overline{RAS}$  e  $\overline{CAS}$  retornam ao nível alto. Deve ser lembrado que a ordem na qual os sinais  $\overline{RAS}$  e  $\overline{CAS}$  são desabilitados não é relevante desde que os requisitos de temporização sejam atendidos.

<sup>1</sup> Cabe aqui uma observação: nas memórias dinâmicas, diferentemente do que ocorre nas memórias estáticas, o tempo de ciclo mínimo  $t_{RC}$  é muito maior que o tempo de acesso. Isto deve-se ao fato das memórias dinâmicas executarem uma operação interna de pré carga entre acessos sucessivos.

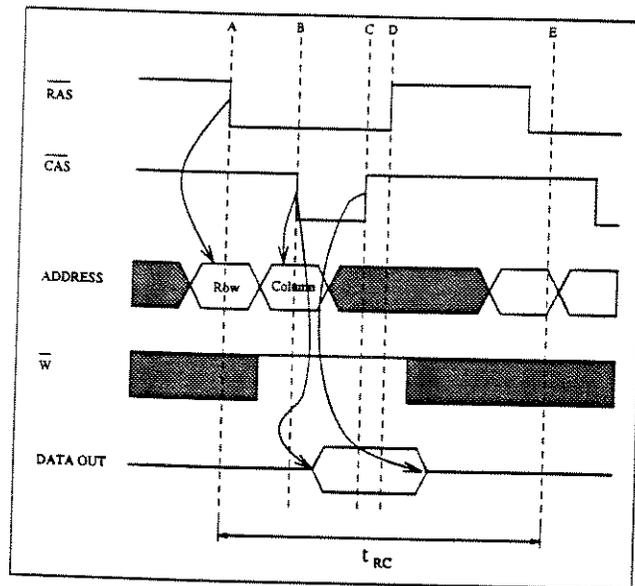


Figura 4.3 - Ciclo de leitura simplificado

Numa breve explicação sobre as operações ilustradas na figura 4.3 diríamos que :

- A primeira operação a realizar-se é colocação do endereço-linha nos pinos de endereço do chip e o sinal  $\overline{\text{RAS}}$  no estado “ativo-baixo” informando a lógica de controle interna do chip DRAM que o endereço-linha é válido.
- O endereço-coluna é então colocado no barramento de endereços e o sinal  $\overline{\text{CAS}}$  é colocado no estado “ativo-baixo”. Neste ponto o endereço completo está disponível a lógica interna do chip DRAM e;
- A célula de memória especificada é acessada e o dado requerido será colocado nos terminais de saída do dispositivo.

Nos instantes C e D os sinais de controle  $\overline{\text{RAS}}$  e  $\overline{\text{CAS}}$  retornam ao nível alto. Deve ser lembrado que a ordem na qual os sinais  $\overline{\text{RAS}}$  e  $\overline{\text{CAS}}$  são desabilitados não é relevante desde que os requisitos de temporização sejam atendidos.

### Temporização do barramento de endereços

A figura 4.4 mostra detalhes a respeito da temporização das entradas de endereços de uma memória dinâmica. Os requisitos de temporização são os mesmos de um “latch típico”, ou seja tempos de “setup”<sup>2</sup> e de “hold”<sup>3</sup>. No caso específico dos chips DRAM, estes requisitos são fornecidos através dos seguintes parâmetros:

- $t_{ASR}$  : O endereço-linha deve estar disponível nas entradas de endereço do chip pelo menos  $t_{ASR}$  segundos antes da borda de descida do sinal  $\overline{\text{RAS}}$ .
- $t_{RAH}$  (row address hold time) : Após a ativação do sinal  $\overline{\text{RAS}}$  o endereço-linha deve manter-se estável por pelo menos  $t_{RAH}$  segundos.
- $t_{ASC}$  (column address setup time): O endereço-coluna deve estar disponível nas entradas de endereço do chip DRAM pelo menos  $t_{ASC}$  segundos antes da borda de descida do sinal  $\overline{\text{CAS}}$ .

<sup>2</sup> O tempo de “setup” indica quanto tempo antes da habilitação do latch, a informação deve estar disponível em suas entradas.  
<sup>3</sup> O tempo de “hold” indica por quanto tempo as entradas devem ser mantidas estáveis após a habilitação do latch.

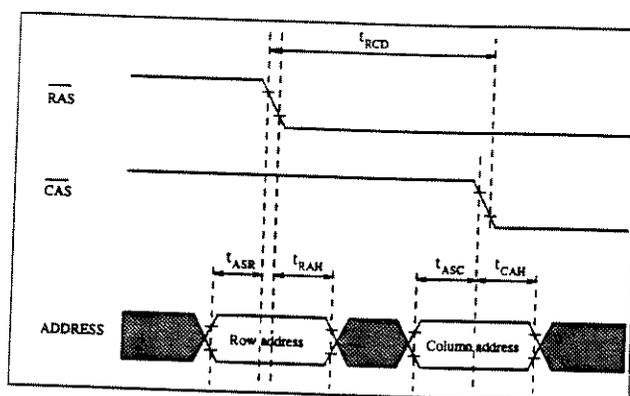


Figura 4.4 - Temporização das entradas de endereço

- $t_{CAH}$  (column address hold time): Após a ativação de  $\overline{CAS}$ , o endereço-coluna deve manter-se estável até que se passem os  $t_{CAH}$  segundos requeridos para que o endereço-coluna seja armazenado internamente.

Estes parâmetros podem também ser cotados com valores nulos ou até mesmo negativos. Se por exemplo o parâmetro  $t_{ASC}$  for cotado com o valor “-10ns”, isto significa que o sinal  $\overline{CAS}$  pode ser colocado no estado “ativo baixo” 10ns antes que o endereço esteja disponível nas entradas de endereço do chip DRAM.

### Temporização do barramento de dados

Os requisitos de temporização do terminal de saída de dados são mostrados na figura 4.5.

- $t_{RAC}$ : O dado no terminal de saída de dados é considerado válido  $t_{RAC}$  segundos após a borda de descida do sinal  $\overline{RAS}$ .
- $t_{CAC}$ : O sinal  $\overline{CAS}$  tem basicamente duas funções : dar o comando para que o endereço-coluna seja armazenado internamente, e ativar os buffers de saída para que os dados estejam disponíveis no terminal de saída de dados. Por estas razões, o dado só será válido  $t_{CAC}$  segundos após a ativação do sinal  $\overline{CAS}$ .
- $t_{RCD}$ : Este parâmetro tem um valor mínimo e um valor máximo. O valor mínimo informa que deve existir um atraso de  $t_{RCD}$  entre as ativações dos sinais  $\overline{RAS}$  e  $\overline{CAS}$ . O valor máximo diz apenas que se o atraso entre ativações for maior que  $t_{RCD_{max}}$  o tempo de acesso será um valor maior que o especificado pelo fabricante.
- $t_{OFF}$ : Após a desativação do sinal  $\overline{CAS}$  a saída de dados flutuará  $t_{OFF}$  segundos depois.

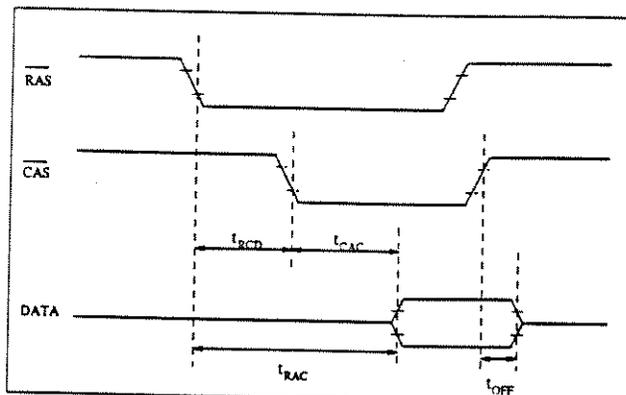


Figura 4.5 - Temporização das entradas de dados

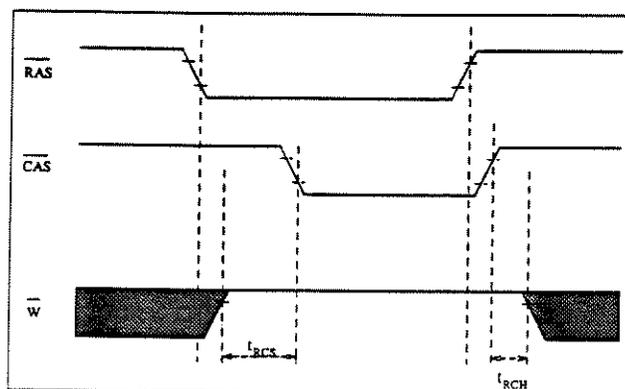


Figura 4.6 - Temporização da entrada W

### Temporização da entrada W

Conforme é mostrado na figura 4.6, a entrada  $\overline{W}$  deve estar no nível lógico "1"  $t_{RCS}$ <sup>4</sup> segundos antes da borda de descida do sinal  $\overline{CAS}$ . Após a desativação de  $\overline{CAS}$ , a entrada  $\overline{W}$  deve permanecer neste estado por pelo menos  $t_{RCH}$ <sup>5</sup> segundos.

<sup>4</sup> Read comand setup time

<sup>5</sup> Read comand hold time

## Temporização dos sinais RAS e CAS

A figura 4.7 mostra os requisitos de temporização dos sinais de controle  $\overline{\text{RAS}}$  e  $\overline{\text{CAS}}$ . Estes dois sinais controlam além das operações já mencionadas<sup>6</sup>, uma série de operações internas. Isto explica o porquê da existência de tantos parâmetros na figura 4.7. Felizmente do ponto de vista de quem apenas do usuário do chip, o diagrama pode ser encarado como um informe a respeito dos tempos em que se deve ativar e desativar os dois sinais, e de como eles interrelacionam-se.

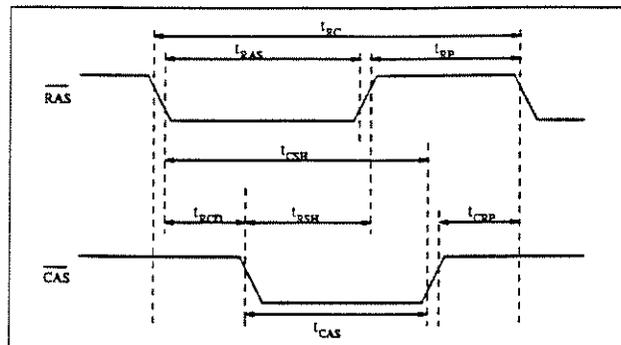


Figura 4.7 - Temporização das entradas  $\overline{\text{RAS}}$  e  $\overline{\text{CAS}}$

A seguir uma breve descrição dos tempos ilustrados na figura 4.7:

- $t_{RC}$ : Este parâmetro informa a duração mínima de um ciclo de leitura.
- $t_{RAS}$ : O sinal  $\overline{\text{RAS}}$  deve permanecer ativo por pelo menos  $t_{RAS_{min}}$  durante um ciclo de de leitura. Por outro lado esta ativação não pode durar mais que  $t_{RAS_{max}}$ , uma vez isto causaria problemas com as operações de refresh.
- $t_{RP}$ : O tempo de pré-carga designa o tempo que o sinal  $\overline{\text{RAS}}$  deve permanecer desativado antes que um novo ciclo de acesso possa ser iniciado.
- $t_{RSH}$ : Após a ativação do sinal  $\overline{\text{CAS}}$ , o sinal  $\overline{\text{RAS}}$  deve permanecer ativo por pelo menos  $t_{RSH}$  segundos.
- $t_{CAS}$ : Este parâmetro informa as larguras mínima e máxima do pulso de ativação do sinal  $\overline{\text{CAS}}$ .
- $t_{CRP}$ : O sinal  $\overline{\text{CAS}}$  deve ser desativado pelo menos  $t_{CRP}$  segundos antes da ativação do sinal  $\overline{\text{RAS}}$  do ciclo de acesso seguinte.
- $t_{CSH}$ : O sinal  $\overline{\text{CAS}}$  pode ser desativado até  $t_{CSH}$  segundos depois da borda de subida do sinal  $\overline{\text{RAS}}$ .

A figura 4.8 mostra um diagrama de temporização completo de um ciclo de leitura.

<sup>6</sup> Armazenamento das componentes de endereço e habilitação dos drives de saída de dados

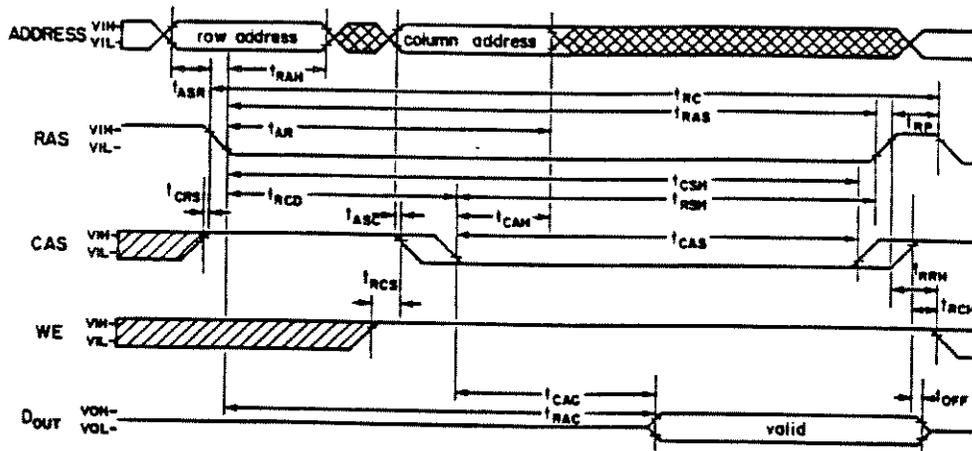


Figura 4.8 - Ciclo de leitura detalhado

### 4.1.2 O ciclo de escrita convencional

A figura 4.9 mostra o diagrama de temporização completo do ciclo de escrita de um chip DRAM. Ele é um pouco mais complexo que o do ciclo de leitura, uma vez que os requisitos para a entrada de dados e para o sinal  $\overline{W}$  são mais rigorosos. Para as entradas de endereços e para os sinais  $\overline{RAS}$  e  $\overline{CAS}$  os parâmetros são exatamente os mesmos. O diagrama da figura 4.10 mostra apenas os tempos da operação de escrita que se diferenciam da operação de leitura.

No que se refere a temporização da entrada  $\overline{W}$  diríamos:

- $t_{WCS}$ : O sinal  $\overline{W}$  deve estar em nível baixo pelo menos  $t_{WCS}$  segundos antes da borda de descida de  $\overline{CAS}$ .
- $t_{WP}$ : Define a largura mínima do pulso<sup>7</sup> do sinal  $\overline{W}$ .
- $t_{WCH}$ :  $\overline{W}$  não deve retornar ao nível alto antes que  $t_{WCH}$  segundos tenham se passado após a borda de descida de  $\overline{CAS}$ .
- $t_{RWL}$ :  $\overline{W}$  deve ser colocado em nível baixo pelo menos  $t_{RWL}$  segundos antes da borda de subida do sinal  $\overline{RAS}$ .
- $t_{CWL}$ : análogo ao parâmetro anterior, neste caso a referência é a borda de descida do sinal  $\overline{CAS}$ .

Quanto a temporização da entrada de dados vale notar que os dados devem ser válidos  $t_{DS}$ <sup>8</sup> segundos antes da borda de descida do sinal  $\overline{CAS}$  e devem manter-se estáveis por  $t_{DH}$ <sup>9</sup> segundos após a ativação de  $\overline{CAS}$ .

<sup>7</sup> Leia-se permanência no nível lógico baixo.

<sup>8</sup> Tempo de setup para entrada de dados.

<sup>9</sup> Tempo de hold da entrada de dados.

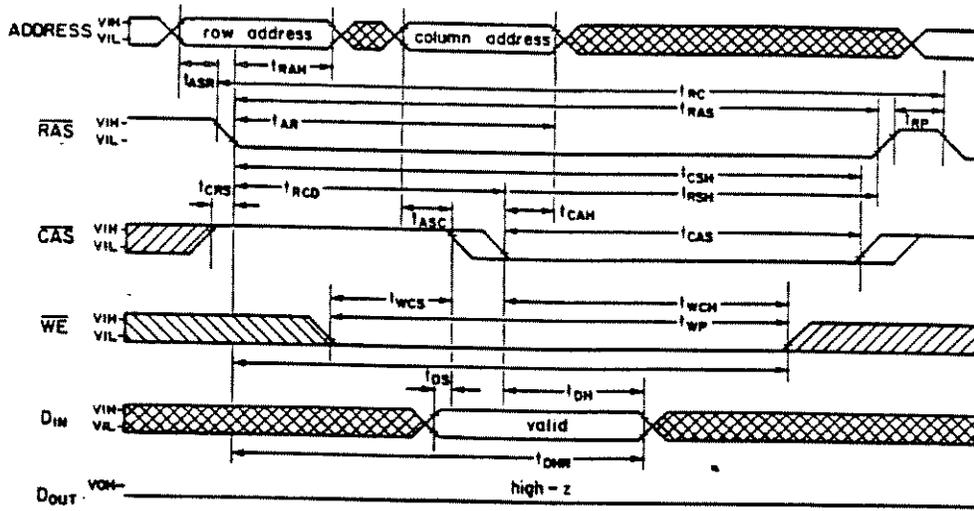


Figura 4.9 - Ciclo de escrita detalhado

### 4.1.3 O modo de operação página

O modo página é uma tentativa dos fabricantes de memórias para aumentar a velocidade de operação dos chips DRAM. O termo “página” vem do fato de várias células de memória com o mesmo endereço-linha serem acessadas de maneira sucessiva em um único ciclo de escrita ou leitura.

Conforme mostra a figura 4.10, os endereços-linha e coluna são aplicados como em um ciclo de acesso normal. Ocorre entretanto, que quando o sinal  $\overline{\text{CAS}}$  retorna é desabilitado, o sinal  $\overline{\text{RAS}}$  ao invés de retornar ao estado inativo, permanece ativado e assim sendo os dados da linha endereçada mantem-se disponíveis nos amplificadores sensores. Assim ao ativarmos novamente o sinal  $\overline{\text{CAS}}$  e fornecermos um segundo endereço-coluna, um outro dado poderá ser lido (ou escrito) sem que seja necessário fornecer o endereço-linha novamente. Outros pulsos de  $\overline{\text{CAS}}$  podem ser aplicados e de modo outras células de memória podem ser acessadas neste mesmo ciclo.

### 4.1.4 O modo de operação nibble

Além do modo página, outra forma de aumentar a velocidade oferecida em alguns chips DRAM é o modo nibble. Neste modo de operação ao fornecermos um endereço completo<sup>10</sup>, o dispositivo acessará automaticamente um conjunto de 4 células de memória em um único ciclo.

A figura 4.11 mostra os ciclos de leitura e escrita no modo nibble. Note que o ciclo de leitura no modo nibble é idêntico a um ciclo de leitura normal até a primeira borda de subida do sinal  $\overline{\text{CAS}}$ . Nesta primeira parte, a célula correspondente ao endereço completo fornecido é acessada. Nos três pulsos de  $\overline{\text{CAS}}$  subseqüentes ocorrerá o acesso das outras três células do conjunto.

<sup>10</sup> Endereço-linha + endereço-coluna.

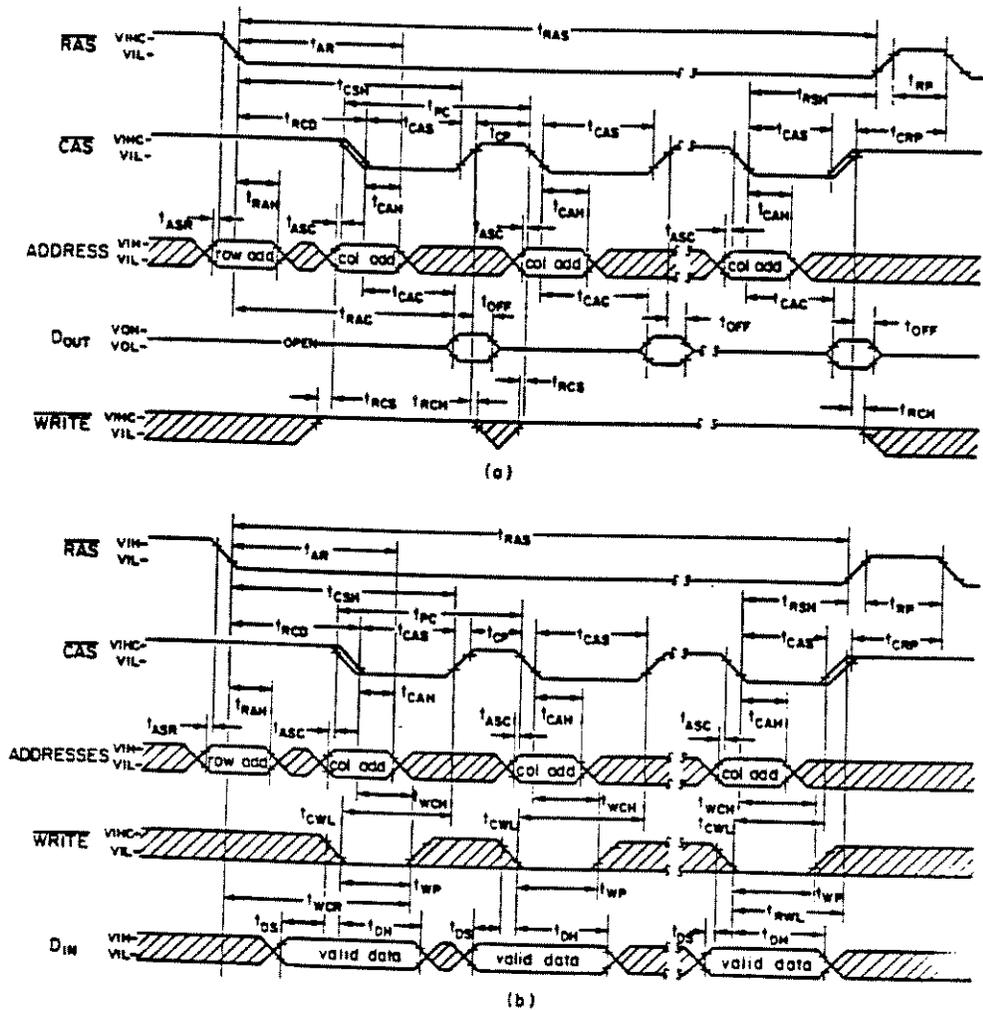


Figura 4.10 - Modo de operação paginado: (a) leitura ; (b) escrita.

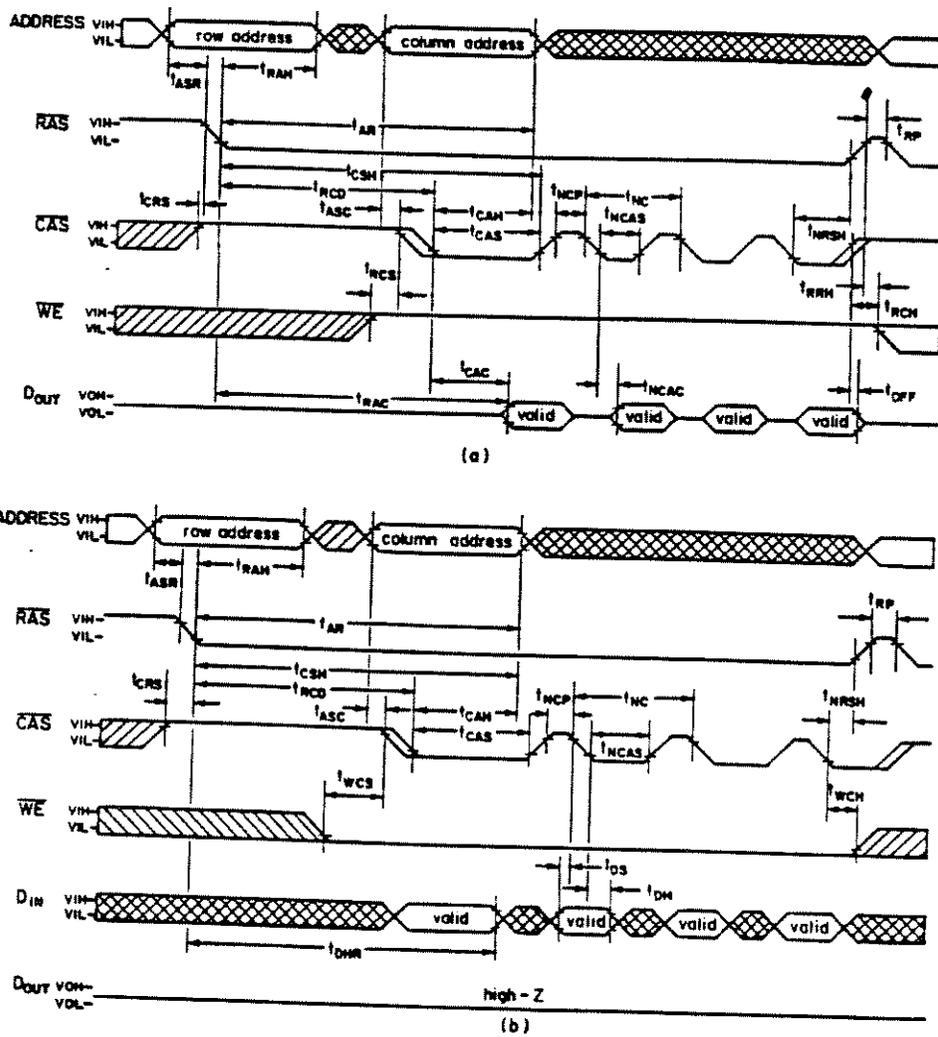


Figura 4.11 - Modo de operação nibble: (a) leitura; (b) escrita.

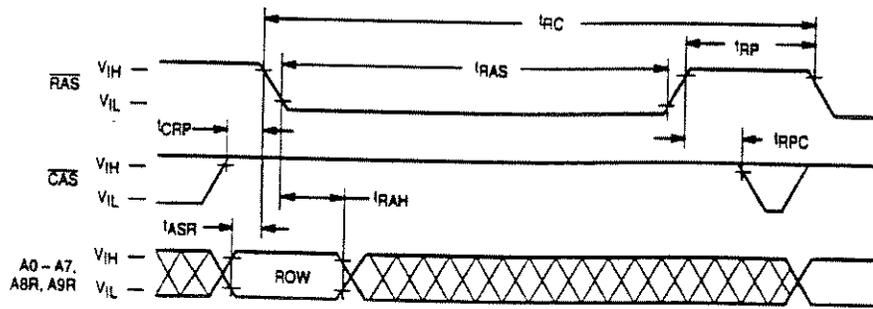


Figura 4.12 - Ciclo de *RAS-only-refresh*

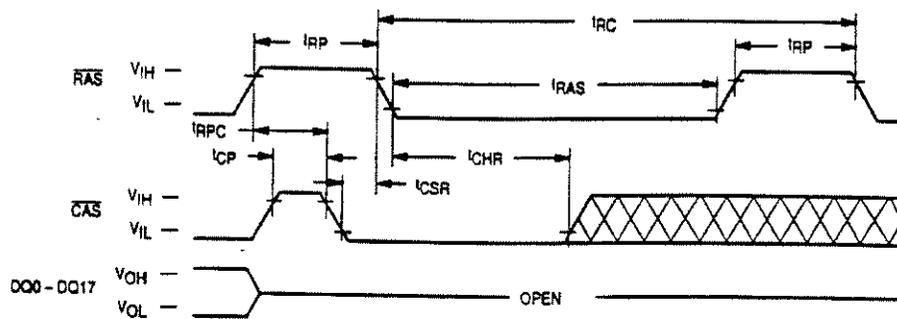


Figura 4.13 - Ciclo de *CAS-before-RAS*

### 4.1.5 Ciclos de reforço

Devido ao seu mecanismo de armazenamento de informação, as DRAM's requerem operações periódicas de reforço de dados. Nesta seção discutiremos dois meios de se realizar estas operações:

- *RAS-only refresh* e;
- *CAS-before-RAS refresh*.

A opção *ras only refresh* faz uso do fato de uma célula sofrer “refresh” sempre que for acessada, melhor ainda, todas as células da linha a qual pertence a célula em questão sofrerão “refresh”. Assim o “refresh” é feito realizando-se acessos às linhas conforme mostra a figura 4.12.

Na opção *cas-before-RAS refresh*, o sinal  $\overline{\text{CAS}}$  é ativado antes do sinal  $\overline{\text{RAS}}$ , e isto dispara um contador interno do chip DRAM que fornece os endereços das linhas que sofrerão “refresh”. Os pinos de dados e de endereços permanecerão em estado de alta impedância durante estes ciclos. Ciclos sucessivos de *cas-before-ras* auto-incrementam o contador interno de modo que a se realizar o “refresh” de todas as linhas do dispositivo. A figura 4.13 mostra as formas de onda típicas deste tipo de ciclo.

### 4.1.6 Características de chip's e módulos DRAM comerciais

Até este ponto discutimos de uma maneira geral a operação de dispositivos de memória dinâmica. No presente item, completaremos esta discussão fornecendo alguns dados a respeito de dispositivos disponíveis no mercado.

Em particular, trataremos do chip de memória DRAM MCM517400 e do módulo de memória SIMM MCM32800. As informações aqui apresentadas além de ilustrar o estudo básico a respeito de memórias dinâmicas realizado no item anterior, fornecem subsídios para a utilização destes componentes no projeto lógico do módulo armazenador de quadros de vídeo.

### Características do chip DRAM MCM517400 4Mx4

O MCM517400 é um chip de memória dinâmica de 16Mbits implementado com tecnologia CMOS de fabricação da Motorola. Este dispositivo é organizado de modo a armazenar até 4194304 palavras de 4 bits, possui 12 pinos de endereço (A0-A11) e suporta, em suas entradas e saídas, níveis de tensão TTL. O MCM517400 oferece, além dos ciclos de escrita e leitura convencionais, também o modo de operação paginado. No que se refere ao refresh, ele suporta tanto os ciclos *ras-only-refresh* quanto os ciclos *cas-before-ras-refresh*. Para se garantir a integridade dos dados armazenados requer-se a realização de 2048 ciclos de refresh a cada 32ms.

A tabela 4.1 lista os principais parâmetros de temporização do MCM517400 em suas versões de 60 e 70 nanosegundos<sup>11</sup>.

Tabela 4.1 - Parâmetros de temporização do MCM517400

Parâmetro de temporização	Símbolo	Versão 70 ns		Versão 60 ns	
		min	max	min	max
Row address setup time	t <sub>ASR</sub>	0ns	-	0ns	-
Row address hold time	t <sub>RAH</sub>	10ns	-	10ns	-
Column setup time	t <sub>ASC</sub>	0ns	-	0ns	-
Column address hold time	t <sub>CAH</sub>	10ns	-	15ns	-
Access time from RAS	t <sub>RAC</sub>	60ns	-	70ns	-
Access time from CAS	t <sub>CAC</sub>	15ns	-	20ns	-
RAS to CAS delay time	t <sub>RCD</sub>	20ns	45ns	20ns	50ns
Turn off delay	t <sub>OFF</sub>	0ns	15ns	0ns	15ns
Read command setup time	t <sub>RCS</sub>	0ns	-	0ns	-
Read command hold time	t <sub>RCH</sub>	0ns	-	0ns	-
Random read or write cycle time	t <sub>RC</sub>	110ns	-	130ns	-
RAS pulse width	t <sub>RAS</sub>	60ns	10us	70ns	10us
RAS precharge time	t <sub>RP</sub>	40ns	-	50ns	-
RAS hold time	t <sub>RSH</sub>	15ns	-	20ns	-
CAS pulse width	t <sub>CAS</sub>	15ns	10us	20ns	10us
CAS to RAS precharge time	t <sub>CRP</sub>	5ns	-	5ns	-
CAS hold time	t <sub>CSH</sub>	60ns	-	70ns	-
Write command setup time	t <sub>WCS</sub>	0ns	-	0ns	-
Write command pulse width	t <sub>WP</sub>	10ns	-	15ns	-
Write command hold time referenced to CAS	t <sub>WCH</sub>	10ns	-	15ns	-
Write command to RAS lead time	t <sub>RWL</sub>	15ns	-	20ns	-
Write command to CAS lead time	t <sub>CWL</sub>	15ns	-	20ns	-
Data in setup time	t <sub>DS</sub>	0ns	-	0ns	-
Data in hold time	t <sub>DH</sub>	10ns	-	15ns	-

<sup>11</sup> Este tempo corresponde ao tempo de acesso

## Características do módulo de memória SIMM MCM32800 8M x 32

Conforme mostra a figura 4.14 os módulos de memória SIMM (Single Inline Memory Module) são pequenas placas de circuito impresso que possuem chips de memória DRAM em paralelo, de modo a aumentar a largura do barramento de dados.

O MCM32800 é um módulo de memória dinâmica com capacidade para armazenar 8.388.608 palavras de 32 bits. Ele é um SIMM de 72 vias consistindo de 16 chips DRAM MCM517400 em paralelo, organizados como mostrado na figura 4.14. Os parâmetros de temporização deste módulo são os mesmos do chip MCM517400 listados na tabela 4.1.

## 4.2 Memórias estáticas - SRAM

As memórias estáticas costumam ser a primeira possibilidade considerada quando do projeto de sistemas de memórias, dado que são muito mais fáceis de se utilizar que as memórias dinâmicas. Ao contrário do que ocorre com as DRAM's, este tipo de memória não requer operações de refresh periódicas, e não necessitam de circuitos externos adicionais para executar a multiplexagem de endereço muito peculiares às memórias DRAM.

O diagrama básico de uma célula de memória estática implementada com tecnologia NMOS é mostrada na figura 2.1. A principal característica desta célula é que seis transistores são requeridos para armazenar um único bit de informação. Conforme veremos na próxima seção, as células da memória dinâmica armazenam seus dados na forma de uma carga elétrica na capacitância interna de um único transistor e portanto requerem muito menos transistores que as células das memórias estáticas. Em razão deste maior número de componentes por célula das memórias estáticas, um dispositivo de memória dinâmica com um dado tamanho de chip sempre armazenará uma quantidade de dados maior que um chip de memória estática de tamanho correspondente.

### 4.2.1 Características de uma memória estática típica

A pinagem e a organização interna de memória estática típica é mostrada na figura 2.2. Comentários sobre o chip escolhido e descrição da pinagem.

#### O ciclo de leitura

Uma versão simplificada do ciclo de leitura de um chip de memória estática típico é mostrada na figura 2.3. Para ler os dados a partir do dispositivo, o endereço apropriado deve ser colocado nas entradas de endereço, e o sinal de seleção de chip  $\overline{CS}$  bem como a linha de habilitação de saída  $\overline{OE}$  devem ser colocados no nível lógico "0". Assim o dado tornar-se-á válido quando o menor entre os tempos  $t_{CWL}$ ,  $t_{AA}$  e  $t_{CO}$  tiver sido satisfeito. Esta situação significa que o intervalo entre a colocação do endereço no barramento e a ativação do sinal de seleção  $\overline{CS}$  deve ser ativados o mais próximo possível caso queiramos que o tempo de acesso mínimo do chip em questão seja atingido.

### Características do módulo de memória SIMM MCM32800 8M x 32

Conforme mostra a figura 4.14 os módulos de memória SIMM (Single Inline Memory Module) são pequenas placas de circuito impresso que possuem chips de memória DRAM em paralelo, de modo a aumentar a largura do barramento de dados.

O MCM32800 é um módulo de memória dinâmica com capacidade para armazenar 8.388.608 palavras de 32 bits. Ele é um SIMM de 72 vias consistindo de 16 chips DRAM MCM517400 em paralelo, organizados como mostrado na figura 4.15. Os parâmetros de temporização deste módulo são os mesmos do chip MCM517400 listados na tabela 4.1.

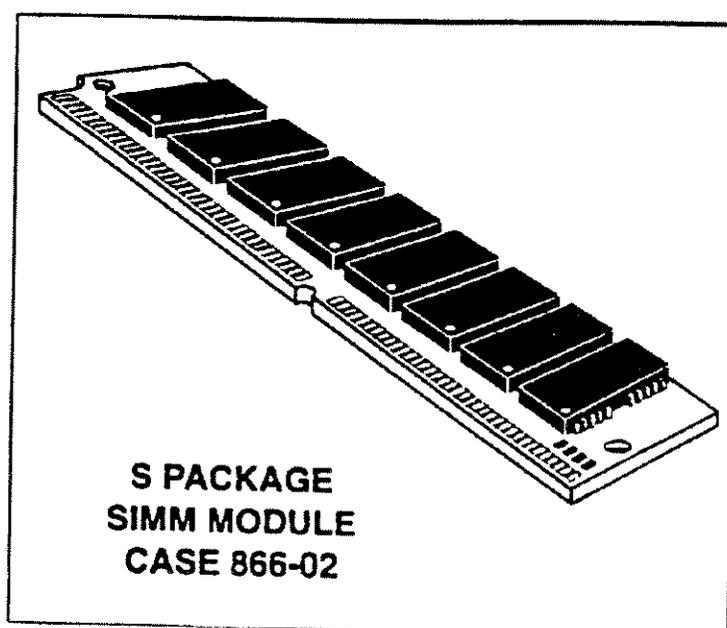


Figura 4.14 - Módulo SIMM de memória dinâmica

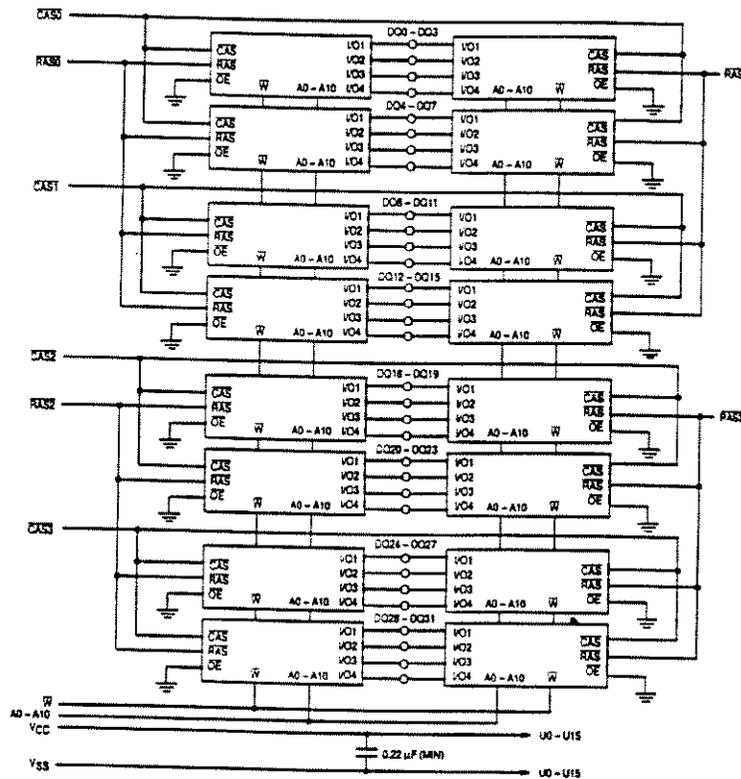


Figura 4.15 - Organização interna do MCM32800.

## 4.2 Memórias estáticas - SRAM

As memórias estáticas costumam ser a primeira possibilidade considerada quando do projeto de sistemas de memórias, dado que são muito mais fáceis de se utilizar que as memórias dinâmicas. Ao contrário do que ocorre com as DRAM's, este tipo de memória não requer operações de refresh periódicas, e não necessitam de circuitos externos adicionais para executar a multiplexagem de endereço muito peculiares às memórias DRAM.

O diagrama básico de uma célula de memória estática implementada com tecnologia NMOS é mostrada na figura 4.16. A principal característica desta célula é que seis transistores são requeridos para armazenar um único bit de informação. Conforme veremos na próxima seção, as células da memória dinâmica armazenam seus dados na forma de uma carga elétrica na capacitância interna de um único transistor e portanto requerem muito menos transistores que as células das memórias estáticas. Em razão deste maior número de componentes por célula das memórias estáticas, um dispositivo de memória dinâmica com um dado tamanho de chip sempre armazenará uma quantidade de dados maior que um chip de memória estática de tamanho correspondente.

### 4.2.1 Características de uma memória estática típica

A organização interna de um chip de memória estática típico é mostrada na figura 4.17

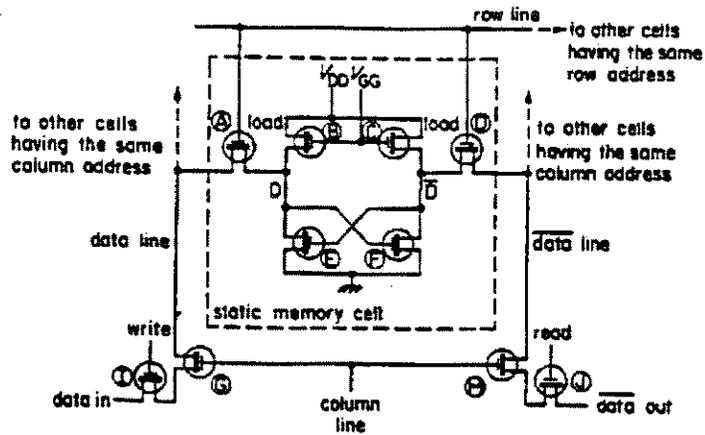


Figura 4.16 - Célula de memória estática implementada com tecnologia NMOS

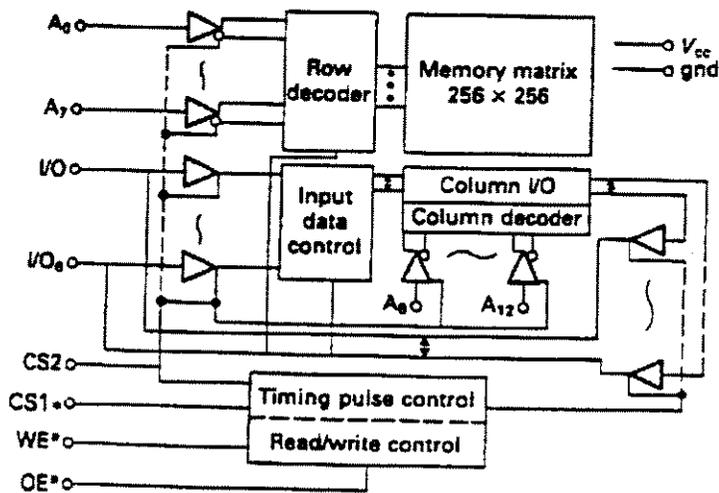


Figura 4.17 - Organização interna de uma SRAM típica.

### O ciclo de leitura

Uma versão simplificada do ciclo de leitura de um chip de memória estática típico é mostrada na figura 5.18. Para ler os dados a partir do dispositivo, o endereço apropriado deve ser colocado nas entradas de endereço, e o sinal de seleção de chip  $\overline{CS}$  bem como a linha de habilitação de saída  $\overline{OE}$  devem ser colocados no nível lógico "0". Assim o dado tornar-se-á válido quando o menor entre os tempos  $t_{CFL}$ ,  $t_{AA}$  e  $t_{CO}$  tiver sido satisfeito. Esta situação significa que o intervalo entre a colocação do endereço no barramento e a ativação do sinal de seleção  $\overline{CS}$  deve ser ativados o mais próximo possível caso queiramos que o tempo de acesso mínimo do chip em questão seja atingido.

A linha de habilitação das linhas de dados,  $\overline{OE}$ , faz com que as linhas de dados do dispositivo assumam um estado de baixa impedância  $t_{OLZ}$  segundos após a ativação. Esta linha presta-se principalmente a propósitos de contenção de barramento.

Um outro período crítico no ciclo de leitura ilustrado na figura 5.18 ocorre no fim do ciclo em questão. As linhas de dados “flutuarão”  $t_{OHZ}$  após a desativação de um entre os sinais  $\overline{CS}$  ou  $\overline{OE}$ . No caso das entradas de endereço variarem antes que as linhas de dados comecem a flutuar o chip mantém o conteúdo do barramento de dados por pelo menos  $t_{OH}$  segundos.

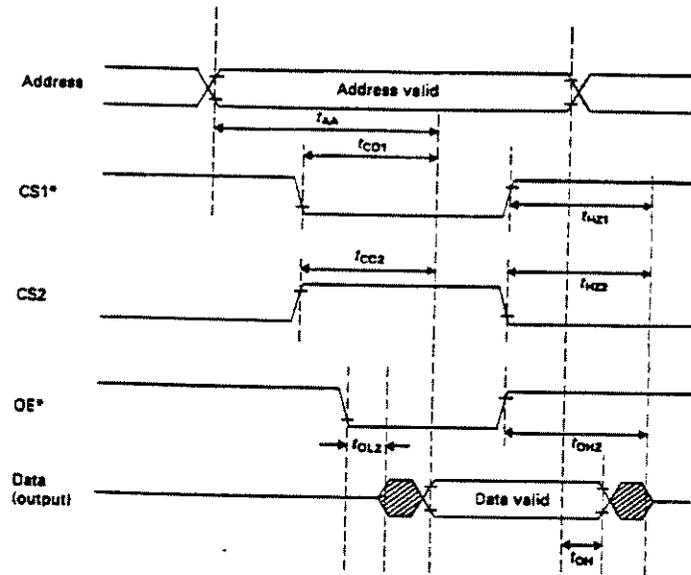


Figura 4.18 - Ciclo de leitura simplificado de uma SRAM

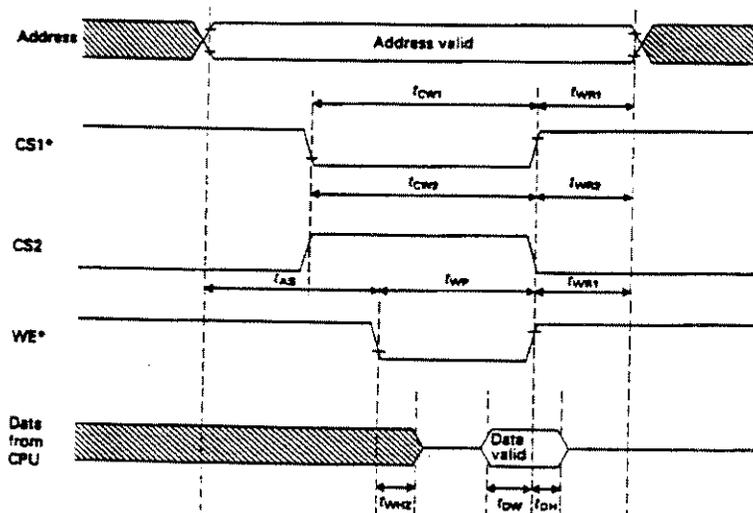


Figura 4.19 - Ciclo de escrita simplificado de uma SRAM

## O ciclo de escrita

A figura 4.19 mostra uma versão simplificada do diagrama de temporização de um ciclo de escrita de um chip SRAM típico. Um ciclo de escrita válido é iniciado quando a linha de seleção  $\overline{CS}$  ou a entrada  $\overline{WE}$  são colocadas no nível lógico “0”. Ou seja, um ciclo de escrita começa quando um dos dois sinais é ativado e termina quando um deles é desabilitado.

Como requisito de “guarda” das entradas  $\overline{CS}$  ou  $\overline{WE}$  tem-se que as entradas de endereço do chip devem ser válidas pelo menos  $t_{AS}$  segundos antes que um destes dois sinais seja ativado. Além disso, durante o ciclo de escrita, a entrada  $\overline{CS}$  deve manter-se ativada por pelo menos  $t_{CW}$  segundos e a entrada  $\overline{WE}$  por deve permanecer no nível “0” por  $t_{WP}$  segundos. A “palavra” a ser armazenada deve ser colocada nas linhas de dados da memória pelo menos  $t_{DW}$  segundos antes que do fim do ciclo de escrita (ou seja, borda de subida dos sinais  $\overline{WE}$  ou  $\overline{CS}$ ) e deve ser mantida por pelo menos  $t_{DH}$  segundos. Ao final de um ciclo de escrita, o conteúdo do barramento de endereços deve ser mantido por  $t_{WR}$  segundos.

## 4.3 VideoRAM

As VideoRAM's são dispositivos de memória de alta velocidade com acesso duplo que costumam ser utilizadas em algumas placas adaptadores de vídeo para PCs. A figura 4.20 mostra o diagrama de blocos simplificado de uma memória VideoRAM. Conforme ilustrado este dispositivo é composto basicamente por uma matriz de memória dinâmica com acesso de escrita e leitura convencional, organizada com “mxn” palavras<sup>12</sup> de “k” bits, e por um registrador de dados serial com “n” localizações de “k” bits, conhecido como SAM (Serial Access Memory). Uma interface entre as “n” colunas da matriz de memória dinâmica e as “n” localizações do registrador SAM permite que uma linha inteira desta memória DRAM interna seja escrita ou lida a partir do registrador SAM.

Assim, o componente suporta basicamente três tipos de operações:

- Acesso aleatório convencional (escrita e leitura) a partir da memória dinâmica;
- Acesso serial (entrada e saída) a partir do registrador SAM;
- E transferências de dados entre qualquer linha da matriz DRAM e o registrador SAM, efetuadas em ambos sentidos, ou seja, da memória para o registrador e do registrador para a memória.

A VideoRAM permite o acesso simultâneo e assíncrono, às portas DRAM e SAM, exceto durante as transferências entre esses dois elementos. Isto porque nas operações de transferência, as “n” colunas da memória dinâmica permanecem conectadas as “n” localizações do registrador serial.

<sup>12</sup> Onde “m” é o número de linhas e “n” o de colunas da matriz de memória dinâmica

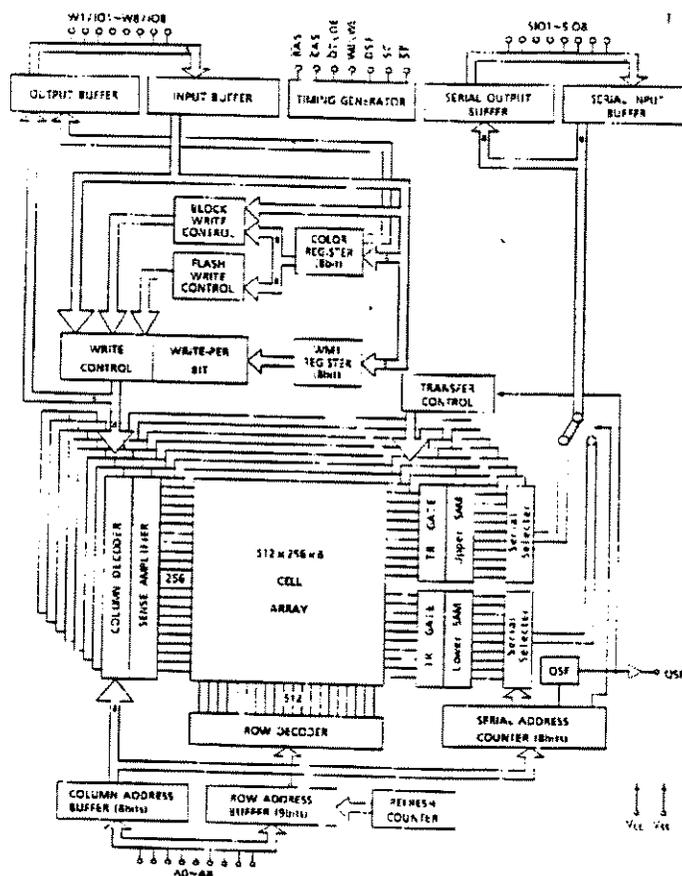


Figura 4.20 - Organização interna da VideoRAM TC528128AJ

No que se refere ao acesso via registrador serial, algumas VideoRAM's oferecem a opção de transferência de dados no modo registrador particionado. Esta operação permite a implementação de carga e recarga do registrador SAM em tempo real. Assim, fluxos de E/S de dados serial verdadeiramente contínuos podem ser suportados pelo componente, sem requisitos de temporização muito críticos. Sob modo de operação, o registrador SAM é dividido em uma "metade-alta" e em uma "metade-baixa", e enquanto uma metade está sendo carregada através da porta SAM, a outra metade pode ser lida via porta DRAM.

### 4.3.1 Operação da porta SAM

Os acessos a porta DRAM de uma VideoRAM são feitos de maneira similar àqueles feitos em um chip DRAM normal e que já foram descritos na seção 4.1. Assim nesta exposição sobre VideoRAM limitaremos-nos a tratar a operação da porta SAM.



Figura 4.21 - Operação da porta SAM no modo single register

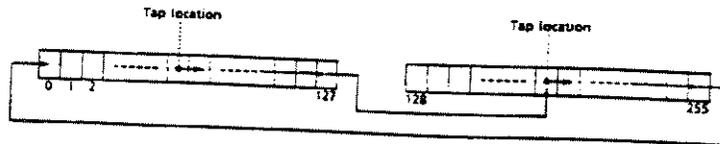


Figura 4.22 - Operação da porta SAM no modo particionado

### Operação da SAM no modo single register

Ao operar no modo single register, a porta SAM pode executar operações de entrada e saída serial independentemente das operações realizadas na porta DRAM. A operação de transferência anterior determina a direção do fluxo de dados na porta SAM. Assim se a operação de transferência anterior foi uma transferência de escrita ou pseudo-escrita (no sentido SAM→DRAM), a porta SAM operará no modo “entrada de dados”. Se por outro lado, a operação de transferência mais recente foi uma transferência de leitura (no sentido DRAM→SAM), a porta SAM operará no modo “saída de dados”.

Dados podem ser obtidos serialmente a partir da porta SAM, depois que uma operação de transferência de leitura for executada. Conforme mostra a figura 4.21, os dados são deslocados para fora do registrador SAM a partir de qualquer uma das suas  $n$  localizações (na figura 4.21 “ $n$ ” é igual a 256). Esta localização inicial é designada como localização *tap* e corresponde ao endereço-coluna especificado na borda de descida do sinal  $\overline{CAS}$  durante o ciclo de transferência de leitura.

O SAM é configurado como um registrador circular. O dado é “deslocado para fora” seqüencialmente a partir da localização *tap* até a localização mais significativa (255) e então volta para a localização menos significativa (0), e assim sucessivamente.

### Operação no modo registrador particionado

No modo registrador particionado, os dados podem ser deslocados para dentro ou para fora de uma metade do registrador SAM, enquanto uma operação de transferência particionada de leitura ou escrita está sendo executada na outra metade do registrador. Uma operação de transferência normal deve preceder as operações de transferência particionada, uma vez as transferências particionadas não alteram modo de operação da porta SAM definido pela transferência normal anterior. A figura 4.22 ilustra a operação no modo registrador particionado de uma SAM de 256 localizações, note que neste modo o registrador também é configurado em anel.

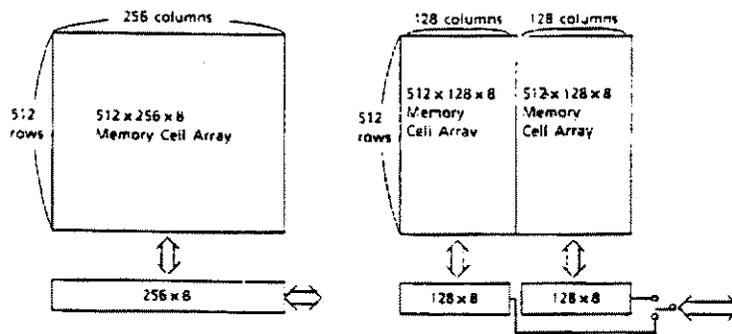


Figura 4.23 - (a) transferência normal, (b) transferência particionada

### 4.3.2 Operação de transferência entre a SAM e a DRAM interna

Conforme mostra a figura 4.23 as VideoRAM's podem realizar dois tipos de transferência bidirecionais internas entre SAM e DRAM. Por exemplo a VideoRAM TC528128AJ, durante uma operação de transferência normal (ou seja, não particionada), 256 palavras de 8 bits podem ser transportadas da DRAM para a SAM (transferência de leitura) ou da SAM para a DRAM (transferência de escrita). Durante as transferências particionadas, 128 palavras de 8 bits são transportadas da metade superior (ou inferior) da DRAM para a metade superior (ou inferior) da SAM ( transferência de leitura particionada) ou realizar esta mesma operação no sentido inverso (transferência de escrita particionada). O modo de transferência normal ou particionado é controlado pela entrada de sinal DSF.

#### Ciclo de transferência de leitura normal

Uma transferência de leitura consiste no transporte de uma linha inteira de dados da matriz DRAM para o registrador SAM. Esta operação é invocada mantendo-se os sinais  $\overline{CAS}$  no nível alto,  $\overline{DT/OE}$  no nível baixo,  $\overline{WB/WE}$  no nível alto e DSF no nível baixo durante a borda de descida do sinal  $\overline{RAS}$ .

O endereço-linha fornecido na borda de descida de  $\overline{RAS}$  determina qual será a linha da DRAM a ser transferida para a SAM. O endereço-coluna selecionado na borda de descida de  $\overline{CAS}$  determina o endereço inicial do ponteiro serial da SAM (localização *tap*).

A transferência é completada na borda de subida de  $\overline{DT/OE}$ . Ao completar-se esta transferência, a porta SAM é automaticamente setada para a operação no modo “saída de dados”. A figura 4.24 mostra um diagrama de blocos ilustrando a operação de uma transferência de leitura.

Durante um ciclo de transferência de leitura, que foi precedido por um ciclo de transferência de escrita (ver figura 4.25), o relógio SC deve ser mantido em nível constante (alto ou baixo). A próxima borda de subida do relógio SC deve ocorrer somente  $t_{TSD}$  segundos após a borda de subida do sinal  $\overline{DT/OE}$ .

Durante um ciclo de transferência de leitura, que foi precedido por um outro ciclo de transferência de leitura (ver figura 4.26), os dados da linha transferida previamente aparecem nos pinos SIO da porta SAM até que o sinal  $\overline{DT/OE}$  vá para o nível alto. Esta característica permite que o primeiro byte da nova linha de dados

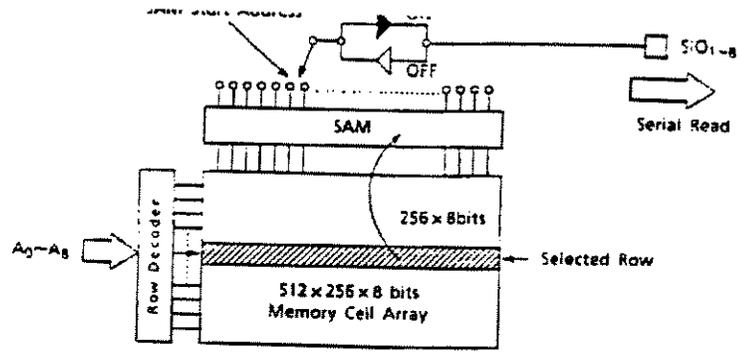


Figura 4.24 - Diagrama de blocos de uma transferência de leitura

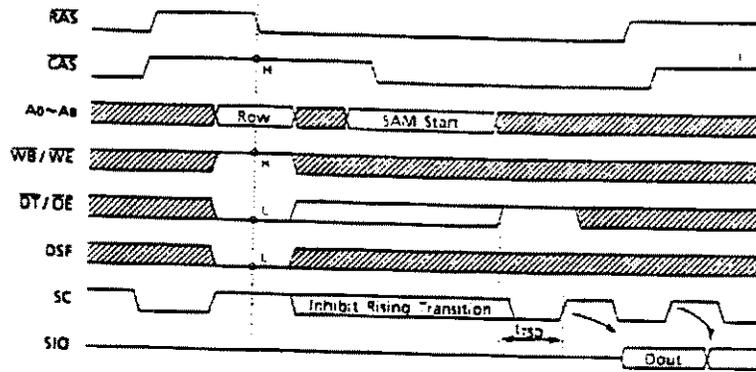


Figura 4.25 - Temporização de uma transferência de leitura normal

apareça na saída serial tão logo o último byte da linha previamente transferida tenha sido fornecido na saída serial sem perdas de temporização. Para fazer com que esse o fluxo de dados o mais contínuo possível, a borda de subida de  $\overline{DT}/\overline{OE}$  deve estar sincronizada com os sinais  $\overline{RAS}$ ,  $\overline{CAS}$  e as bordas de subida de SC subsequentes ( $t_{RTH}$ ,  $t_{CTH}$  e  $t_{TSL}/t_{TSD}$  devem ser satisfeitos), como mostra a figura 4.26. O modo de transferência particionado elimina estas restrições temporais.

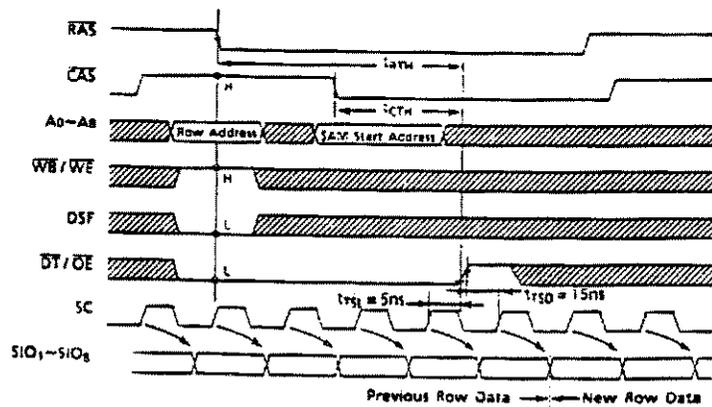


Figura 4.26 - Temporização de uma transferência de leitura normal em tempo real

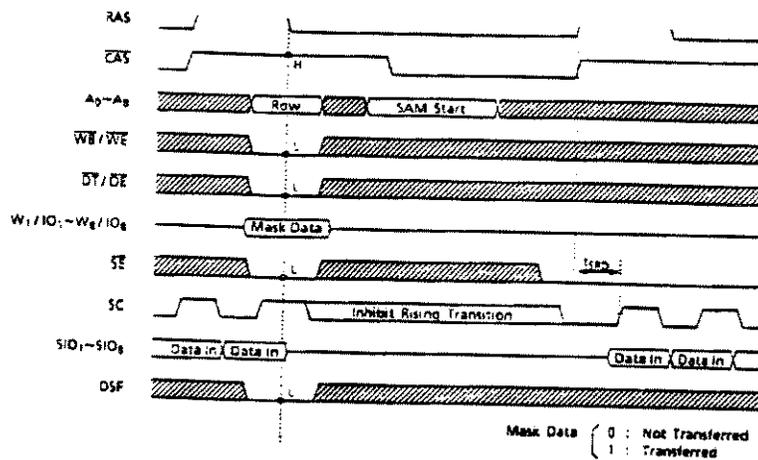


Figura 4.27 - Temporização de uma transferência de escrita normal

### Ciclo de transferência de escrita normal

Um ciclo de transferência de escrita consiste de um carregamento do conteúdo da SAM em uma linha de dados selecionada da matriz DRAM. Uma transferência de escrita é invocada mantendo-se os sinais  $\overline{CAS}$  no nível alto,  $\overline{DT/OE}$  no nível baixo,  $\overline{WB/WE}$  no nível baixo,  $\overline{SE}$  no nível baixo e DSF no nível baixo durante a borda de descida do sinal  $\overline{RAS}$ . A figura 4.27 mostra o diagrama de temporização para as operações de transferências de escritas normais.

O endereço-linha selecionado na borda de descida de  $\overline{RAS}$  determina o endereço da linha de dados da matriz DRAM para o qual será transferido o conteúdo do registrador SAM. O endereço-coluna selecionado na borda de descida de  $\overline{CAS}$  determina a localização tap da SAM. Após o término da transferência de escrita, a porta SAM é setada para o modo “entrada de dados” e bytes podem ser carregados serialmente através dos pinos SIO1-8 sincronizados pelo relógio SC.

Quando transferências de escrita consecutivas forem executadas, novos dados não podem ser colocados no registrador serial até que o ciclo de  $\overline{RAS}$  da transferência de escrita precedente tenha sido completado. Assim, o relógio SC deve permanecer constante no nível alto ou baixo até que  $t_{SRD}$  segundos tenham se passado após a borda de subida de  $\overline{RAS}$ .

### Ciclo de transferência de pseudo-escrita

Um ciclo de transferência de pseudo-escrita deve ser executado antes de carregar os dados no registrador serial depois que uma transferência de leitura tenha sido executada. A única finalidade do ciclo de pseudo-escrita é mudar o modo de operação da porta SAM para o modo “entrada de dados”, uma vez que nenhum transporte de dados ocorre em função de sua execução. Uma transferência de pseudo escrita é invocada sempre que mantivermos os sinais  $\overline{CAS}$  no nível alto,  $\overline{DT/OE}$  no nível baixo,  $\overline{WB/WE}$  no nível baixo,  $\overline{SE}$  no nível alto e DSF no nível baixo durante a borda de descida do sinal  $\overline{RAS}$ .

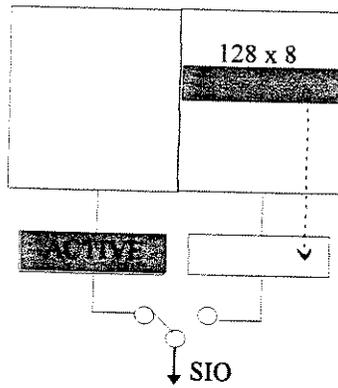


Figura 4.28 - Transferência de leitura particionada

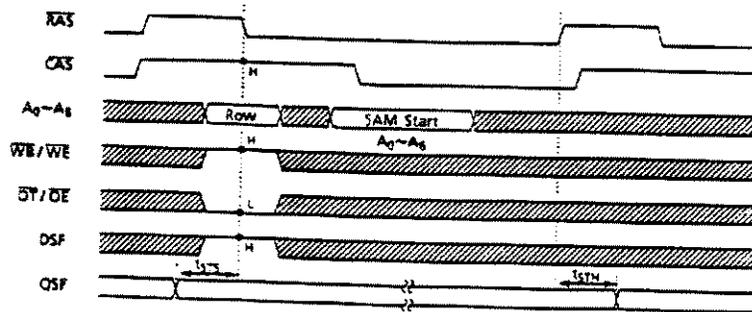


Figura 4.29 - Temporização de uma transferência de leitura particionada

### Ciclo de transferência de leitura particionado

Conforme ilustrado na figura 4.28 uma transferência de leitura particionada consiste do transporte de metade dos dados de uma linha da matriz DRAM para o registrador SAM particionado não ativo correspondente. Por ativo entenda-se a partição da registrador serial que está conectada aos pinos SIO da porta SAM. Assim, dados podem ser “deslocados para fora” da outra partição do registrador simultaneamente. A saída QSF indica qual é a partição que está ativa, assumindo “0” quando a partição ativa é metade inferior (com localizações menos significativas) do registrador e “1” em caso contrário.

O diagrama de temporização da transferência de leitura particionado é mostrado na figura 4.29. Durante este tipo de transferência, os sinais de controle da porta DRAM não precisam estar sincronizados com o relógio serial SC, e dessa forma elimina-se as restrições de temporização no caso em que se requer fluxos de saída de dados seriais verdadeiramente contínuos. A figura 4.30 mostra um exemplo de transferências particionadas de leitura consecutivas. Note que uma transferência de leitura normal deve preceder esta série, para que a porta SAM seja setada para a operação saída de dados.

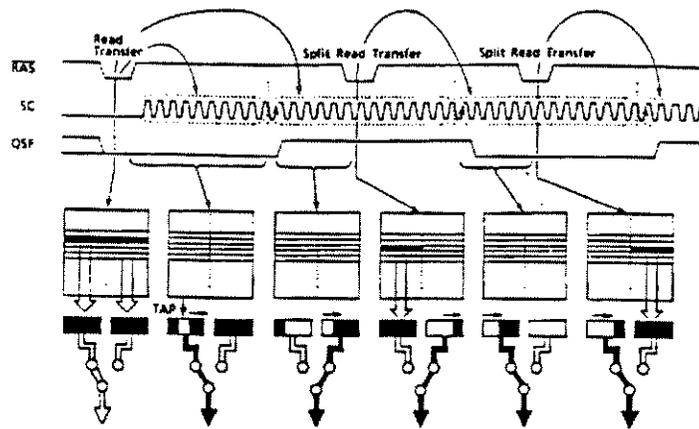


Figura 4.30 - Transferências de leitura particionadas consecutivas

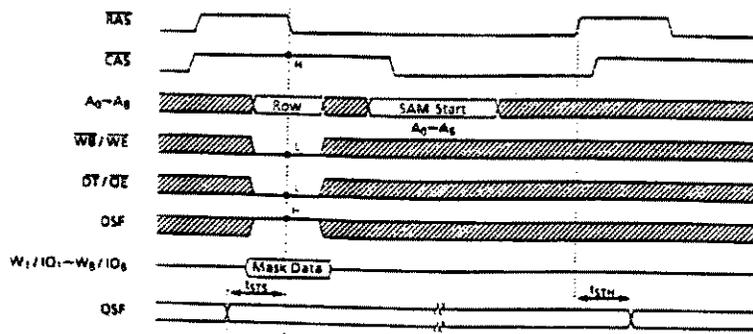


Figura 4.31 - Temporização de uma transferência de escrita particionada

### Ciclo de transferência de escrita particionado

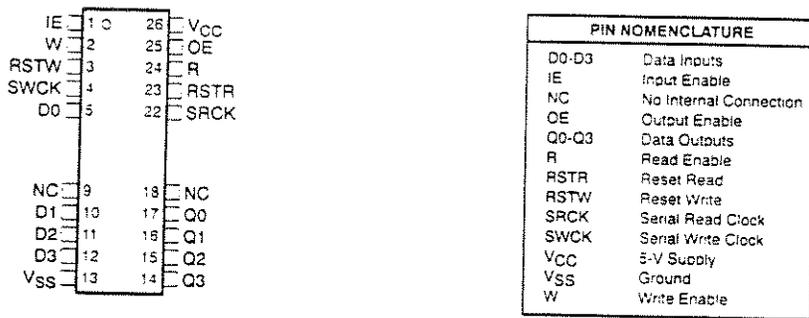
Uma transferência de escrita particionada carrega as palavras armazenadas na partição inativa do registrador serial a uma linha particionada da matriz DRAM.

Dados seriais podem ser deslocados para dentro da metade ativa do registrador ao mesmo tempo em que ocorre a transferência na porção inativa. Assim sendo, a operação das portas DRAM podem ser feitas de forma totalmente assíncrona e simultânea, o que permite a entrada serial de fluxos de dados verdadeiramente contínuos.

A figura 4.31 mostra o diagrama de um ciclo de transferência de escrita particionada.

## 4.4 Field Memory

Nesta seção trataremos de um outro tipo de memória de uso específico para vídeo, as chamadas Field Memories. A discussão, entretanto, não será geral como as feitas para os outros dispositivos e limitaremos-nos a mostrar algumas características do chip TMS4C1070 da Texas Instruments.



**Figura 4.32** - Pinagem da Field Memory TMS4C1070.

O TMS4C1070 é uma Field Memory com capacidade para armazenar até 262264 palavras de 4 bits. Embora o mecanismo de armazenagem de informação seja similar ao utilizado pelas memórias dinâmicas, o próprio dispositivo realiza automaticamente o refresh das células de memória, de modo que a Filed Memory parece ser completamente estática do ponto de vista do usuário.

A figura 4.32 ilustra a pinagem do TMS4C1070. Acessos de escrita e leitura ocorrem serialmente, e normalmente começam no endereço 0, depois que ponteiros os ponteiros de escrita e leitura são resetados através de operações RSTW e RSTR.

Os acessos de escrita e leitura podem, caso se queira, ocorrerem assincronamente. Quando o acesso de leitura ocorre com um atraso em relação ao acesso de escrita, o dispositivo comporta-se como uma memória FIFO (First-In First-Out). A quantidade de atraso determina o “comprimento” do sua memória FIFO.

O atraso mínimo entre a escrita e a leitura no dispositivo é de 600 ciclos do relógio de escrita serial SWCK. O atraso máximo é um campo completo (262144 ciclos de escrita) mais outros 119 ciclos de SWCK.

#### 4.4.1 Operação de escrita

Conforme ilustra a figura 4.33, a operação de escrita é controlada pelo relógio SWCK e pelos sinais RSTW, W e IE. A operação de escrita é estabelecida pulsando-se o relógio SWCK e mantendo-se os sinais W e IE no nível lógico alto, depois que de se realizar um reset no ponteiro de endereços de escrita através de um pulso no sinal RSTW. Cada operação de escrita deve conter pelo menos 120 ciclos de escrita, isto é, dois pulsos sucessivos do sinal RSTW devem estar separados por pelo menos 120 ciclos de escrita ativos (ciclos de SWCK) enquanto W estiver no nível alto.

A tabela 4.2 ilustra a função verdade para a operação de escrita. Podemos observar a partir desta tabela que a entrada W é usada para habilitar/desabilitar o incremento do ponteiro de endereços de escrita, e também para habilitar e desabilitar as operações de escrita na Field Memory. Um nível lógico alto em W serve para habilitar ambas funções, e um nível baixo leva a situação oposta.

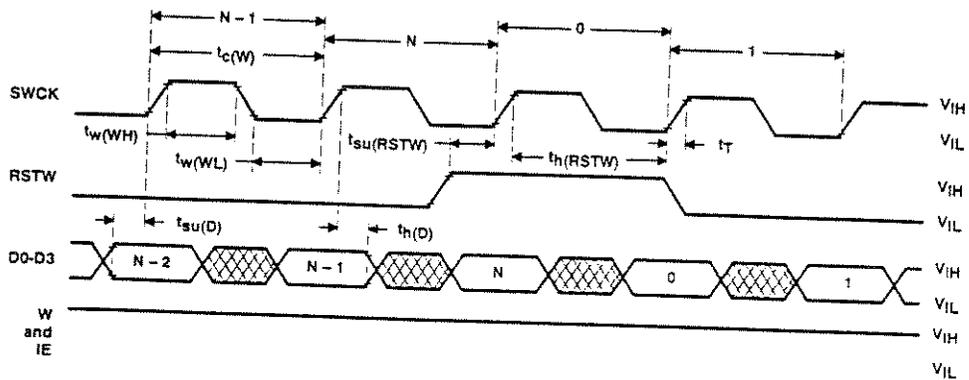


Figura 4.33 - Temporização de uma operação de escrita da Field Memory TMS4C1070

Tabela 4.2 - Função verdade da operação de escrita

<i>W</i>	<i>IE</i>	<i>Ponteiro de endereços de escrita</i>	<i>D0 - D3</i>
H	H	Incrementa o ponteiro de endereços	Armazena dados
H	L	Incrementa o ponteiro de endereços	Não armazena
L	X	Pára o ponteiro de endereços	Não armazena

O sinal *IE* é usado para habilitar/desabilitar a escrita na Field Memory. Um nível alto em *IE* habilita as operações de escrita, e um nível baixo desabilita esta operação.

O relógio *SWCK* controla o armazenamento das entradas de dados *D0-D3* do chip quando as entradas *W* e *IE* estão no nível alto. Além disso, *SWCK* também incrementa o ponteiro de endereços de escrita, *W* está no nível alto, qualquer que seja o estado da entrada *IE*.

### 4.4.2 Operação de leitura

A operação de leitura é controlada pelos sinais *SRCK*, *RSTR*, *R* e *OE*. O ciclo de leitura é realizado pulsando-se o relógio *SRCK* enquanto as entradas *R* e *OE* são mantidas no nível alto após uma operação de inicialização do ponteiro de endereços de leitura (pulso em *RSTR*). Da mesma forma que na escrita, cada operação de leitura deve conter pelo menos 120 ciclos de leitura. A tabela 4.3 mostra a função verdade para a operação de leitura. Note que as entradas *R* e *OE* desempenham, agora para operações de leitura, papéis análogos aos sinais *W* e *IE*.

Tabela 4.3 - Função verdade da operação de leitura

<i>W</i>	<i>IE</i>	<i>Ponteiro de endereços de leitura</i>	<i>D0 - D3</i>
H	H	Incrementa o ponteiro de endereços	Saída de dados
H	L	Incrementa o ponteiro de endereços	Alta impedância
L	H	Pára o ponteiro de endereços	Saída de dados
L	L	Pára o ponteiro de endereços	Alta impedância

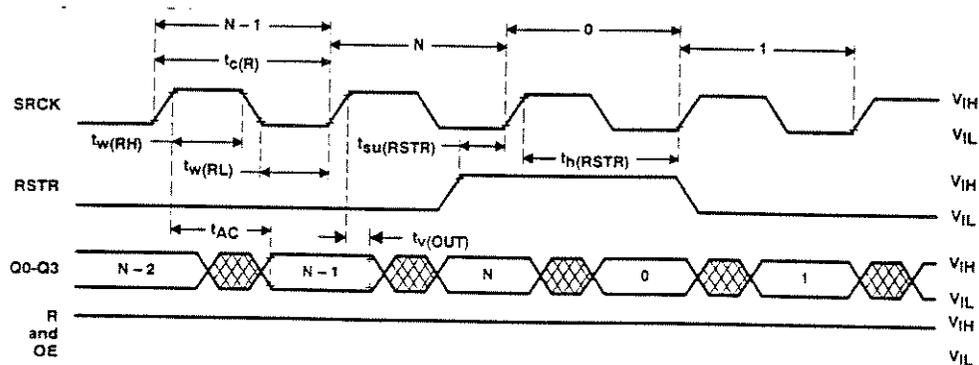


Figura 4.33 - Temporização de uma operação de escrita da Field Memory TMS4C1070

## 4.5 Questões a serem respondidas quando se projeta um armazenador de quadros de vídeo

Até aqui temos discutido os dispositivos de memória semicondutora que poderiam ser utilizados para armazenar vídeo na forma digital. A partir desta seção realizaremos uma mudança de enfoque, passando a discorrer a respeito de como utilizar as informações apresentadas no projeto de um módulo armazenador de quadros capaz de interfacear-se de maneira quase direta com os conversores A/D e D/A do módulo digitalizador de vídeo considerado no capítulo 3, e que possua uma interface com o barramento VME que compõe a espinha dorsal do sistema de aquisição e visualização.

Mas antes de passarmos ao projeto propriamente dito, chamaremos a atenção para algumas questões que devem ser respondidas ao iniciarmos o projeto do armazenador de quadros em questão. As principais questões a serem respondidas, quando se projeta um armazenador de quadros de vídeo baseado em memórias semicondutoras são as seguintes:

- Qual o tamanho total do armazenador, ou melhor qual a quantidade de memória requerida ?
- Que tipo de chips ou módulos de memória poderíamos utilizar, ou quais deles estão disponíveis?
- Que arranjos de multiplexagem/demultiplexagem do barramento de dados serão requeridos para acomodar a taxa de dados proveniente do módulo digitalizador de vídeo ?
- Quantos chips ou módulos de memória devem ser acomodados em uma única placa de circuito ? Que tamanho de placa utilizar ?

### 4.5.1 Quantidade de memória requerida

A quantidade de memória Q, requerida pelo módulo armazenador de quadros é determinada pelos seguintes parâmetros:

- Número de bits por amostra,  $n_{bits}$ .
- Duração da linha ativa de vídeo,  $t_{la}$ .
- Frequência de amostragem utilizada,  $f_s$ .
- Número de linhas ativas por quadro,  $N_{linhas}$ .
- Número de linhas por quadro,  $N_{colunas}$ .
- Números de quadros a serem capturados,  $N_{Quadros}$ .
- Esquema da digitalização, E, ou seja, se o sinal de vídeo será digitalizado na forma de componentes RGB (4:4:4), ou na forma de componentes  $Y C_B C_R$  (4:2:2). O parâmetro E assume os valores:
  - 1 para digitalização de vídeo composto,
  - 2 para digitalização de componentes  $Y C_B C_R$  e
  - 3 para digitalização de componentes RGB.

Assim, a quantidade Q em bytes será dada por:

$$Q = (E \cdot N_{Quadros} \cdot N_{linhas} \cdot N_{colunas} \cdot n_{bits})/8$$

onde  $N_{colunas}$  corresponde ao número de pels por linha e é dado pelo produto  $t_{la} \times f_s$ .

A tabela 4.4 mostra os valores de memória requeridos para se armazenar 1 segundo de vídeo ( 30 quadros ) digitalizado em 2 esquemas diferentes: RGB (4:4:4) e  $Y C_B C_R$  (4:2:2).

**Tabela 4.4 - Quantidade de memória requerida para armazenar 1 segundo de vídeo**

Esquema	Freq. amostragem	Número de pels por linha	Número de linhas	Q em bytes
RGB (4:4:4)	13.5Mhz	720	480	31.104.000
$Y C_B C_R$ (4:2:2)	13.5Mhz	720	480	20.736.000

### 4.5.2 Escolha do dispositivo de memória : a opção pelas DRAM's

Optou-se por utilizar memórias DRAM, por razões de custo e densidade. O armazenador de quadros deve armazenar pelo menos 30 quadros RGB amostrados com uma frequência de 13.5MHz (esquema 4:4:4 da recomendação 601). Conforme mostra a tabela 4.4, isto requer uma quantidade de memória que inviabiliza a utilização de memórias estáticas, VRAM's ou Field Memories.

Esta escolha entretanto, não está livre de problemas, uma vez que as DRAM's em razão da multiplexagem dos pinos de endereço e da necessidade de "refresh" periódico, são menos cê{o}modas de se utilizar. Além disso, há ainda o problema da velocidade de operação destes dispositivos: o armazenador deve ser capaz de receber um fluxo de bytes proveniente de um conversor A/D que digitaliza o sinal de TV a uma frequência de amostragem de 13.5MHz. Isto significa um byte a cada 70ns, para cada componente primária. Como seria possível armazenar este fluxo de dados em memórias DRAM cujo tempo de ciclo é da ordem de 110ns? Nota-se portanto, que o uso de DRAM para este caso não constitui um caminho sem espinhos.

Vale lembrar que não teríamos tais problemas se pudessemos optar pelo uso de memórias como a VRAM e a Field Memory, uma vez que, como já mencionamos estes dispositivos possuem uma porta de acesso serial que possibilita tanto a entrada quanto a saída do fluxo de dados gerados por conversores A/D e D/A de maneira direta.

Portanto o preço que pagamos ao optarmos pelas memórias DRAM é um aumento na complexidade dos circuitos auxiliares a serem fornecidos, dado que teremos de prover, via lógica externa, recursos que já estariam implementados caso utilizássemos memórias especiais para vídeo, como a VRAM ou a Field Memory.

### 4.5.2 Arranjos de Multiplexagem/Demultiplexagem

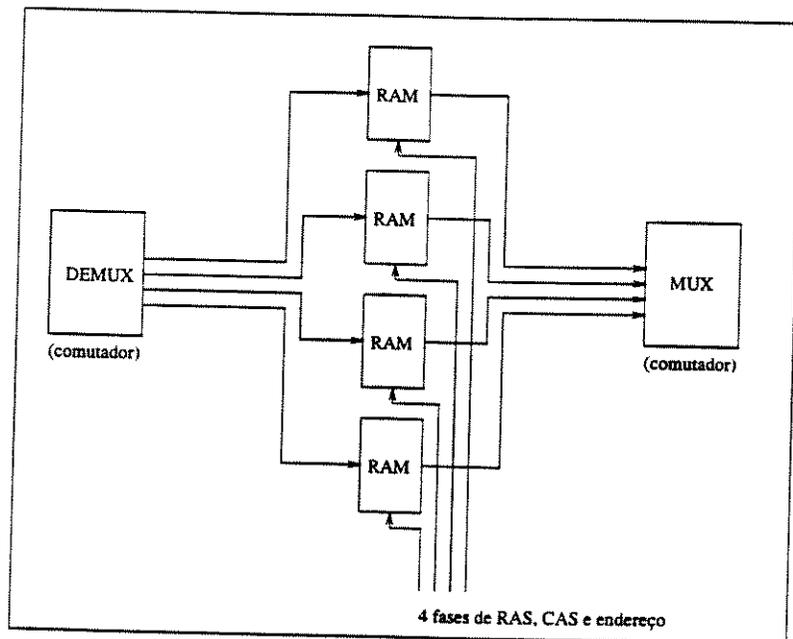
Dispositivos de memória dinâmicas de velocidades relativamente baixas podem acomodar às altas taxas de dados requeridas pelos formatos de vídeo digital através de uma demultiplexagem dos dados por um fator F dependente da razão entre estas duas quantidades.

Há basicamente duas maneiras para realizar esta demultiplexagem. A primeira é distribuir os dados aos dispositivos seqüencialmente, utilizando algum mecanismo que funcione como um comutador, conforme mostrado na figura 4.35a, com os dados sendo retirados da memória de maneira similar (comutador atuando como um multiplexador). A vantagem deste método é que um atraso mínimo pode ser atingido. Entretanto, esta solução requer múltiplas fases de relógios e endereços.

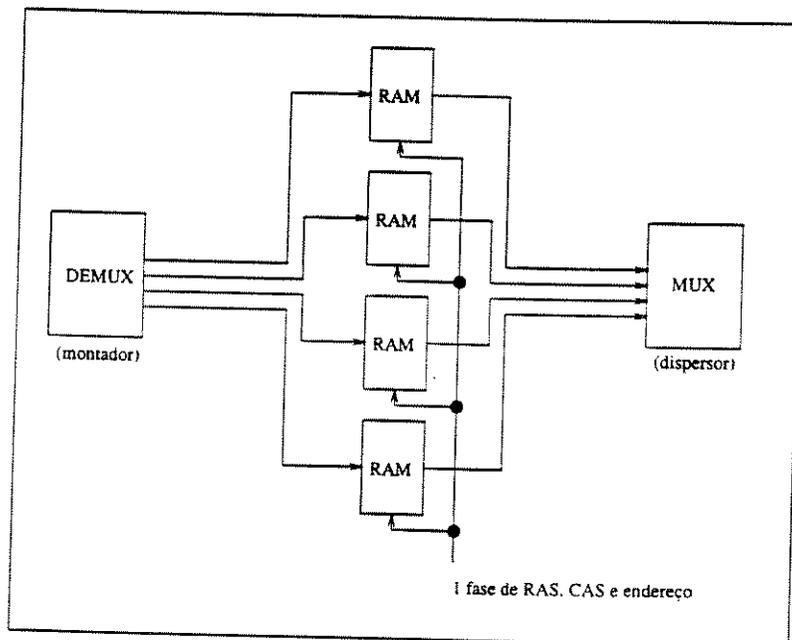
A outra maneira, é montar os dados do fluxo em blocos e armazená-los simultaneamente como mostra a figura 4.35b. Na leitura os blocos seriam desmontados recompondo-se o fluxo original. A vantagem deste método é apenas uma fase de relógio e endereço é requerida. A desvantagem é que introduz um atraso de F pulsos de relógio ao fluxo. Note que este atraso não representa qualquer problema em aplicações como sistemas armazenadores de seqüências de quadros como é o nosso caso.

### 4.5.3 Características mecânicas do módulo armazenador

As características mecânicas do módulo armazenador de quadros são as definidas para a placa 6U usada em sistemas VME.



(a)



(b)

Figura 4.35 - Métodos de multiplexagem do barramento de dados das DRAM.

## 4.6 Organização do armazenador de quadros

A organização do módulo armazenador de quadros proposto é mostrada na figura 4.36. Conforme ilustra o diagrama de blocos, o armazenador possui basicamente cinco seções:

1. Banco de Memórias DRAM,
2. Interface com o Barramento de E/S de Video,
3. Interface com o Barramento VME,
4. Gerador de Ciclos de Reforço,
5. Árbitro do Barramento Interno do Armazenador.

A seção “banco de memórias” tem a função básica de armazenar os dados fornecidos ao módulo através das interfaces com o barramentos VME e o barramento de video. Ela é constituída por um arranjo de módulos DRAM do tipo SIMM.

A seção “interface com o barramento de video” realiza a tarefa de compatilizar o fluxo de dados proveniente dos conversores A/D e D/A do módulo digitalizador ao barramento de dados interno do armazenador.

A seção “interface com o barramento VME” torna os dados armazenados no banco de memórias disponíveis a um módulo Mestre VME. Ela deve conter blocos bastante convencionais tais como decodificadores de endereço, seletores de dados sobre o barramento local e circuitos de controle que os sinis do barramento VME com os sinais de rAS e CAS do barramento local.

O armazenador deve possuir um controlador de refresh local que realiza de tempo em tempo um refreshing do tipo burst a fim de garantir a integridade dos dados armazenados.

O controle do barramento local deve ser arbitrado por uma seção de arbitragem interna de modo a se garantir o tratamento de ambiguidades, tais como o acesso simultâneo às duas interfaces, e ou o acesso de uma das interfaces quando for necessário um burst de refreshing.

Nos itens a seguir forneceremos algumas considerações acerca de possíveis implementações das seções citadas.

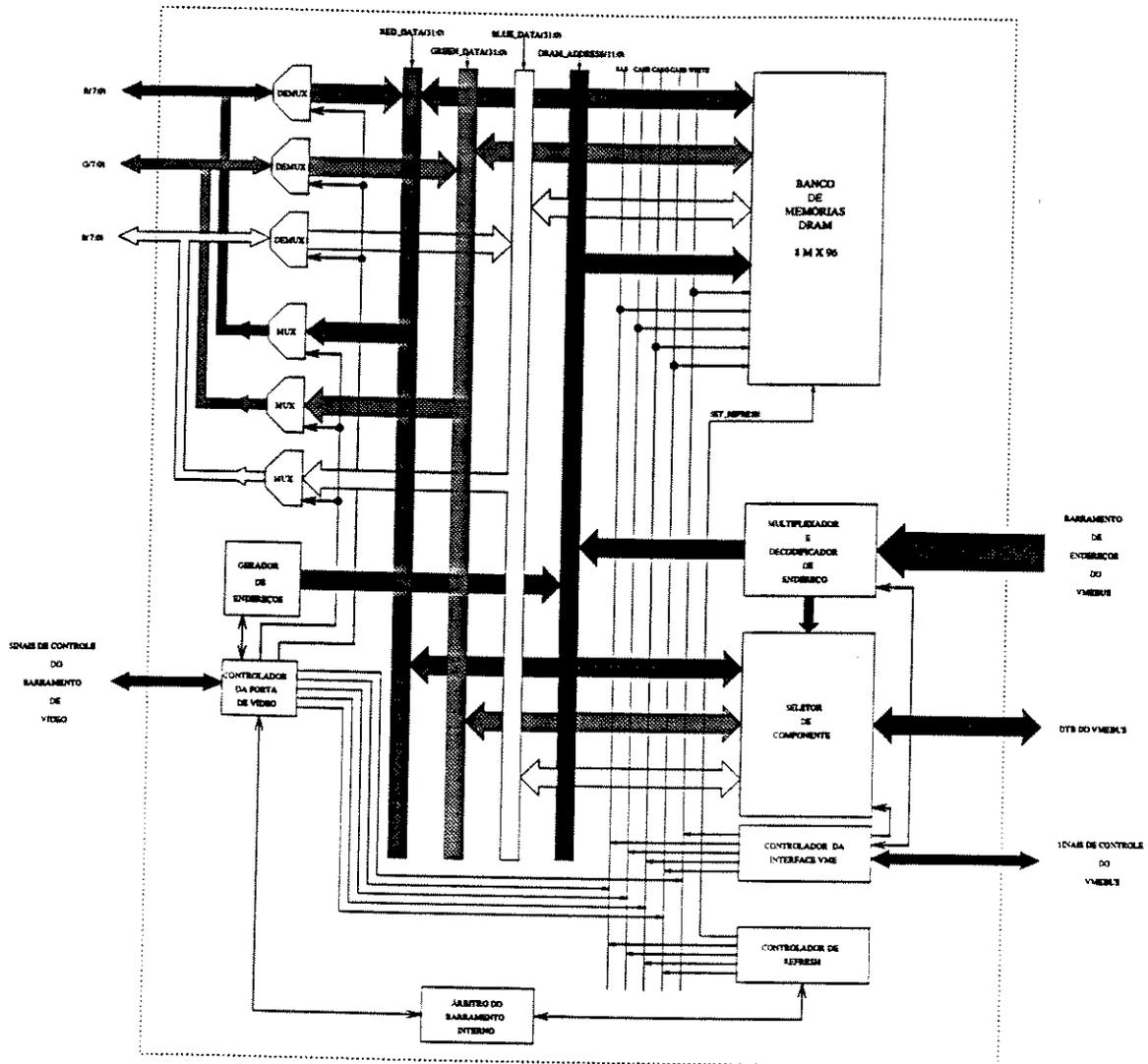


Figura 4.36 - Diagrama de Blocos do Armazenador de Quadros.

### 4.6.1 Implementação do Banco de Memórias DRAM's

A figura 4.37 mostra um digrama esquemático da seção “banco de memórias”. A implementação ilustrada utiliza três módulos SIMM MCM32800 de 8M x 32, o que permitiria o armazenamento de até 97 quadros no formato 4:4:4. O arranjo aloca um módulo SIMM para cada componente de vídeo e a seleção de quais módulos serão acessados é feita através das três linhas CAS(2:0)-L pertencentes ao barramento local.

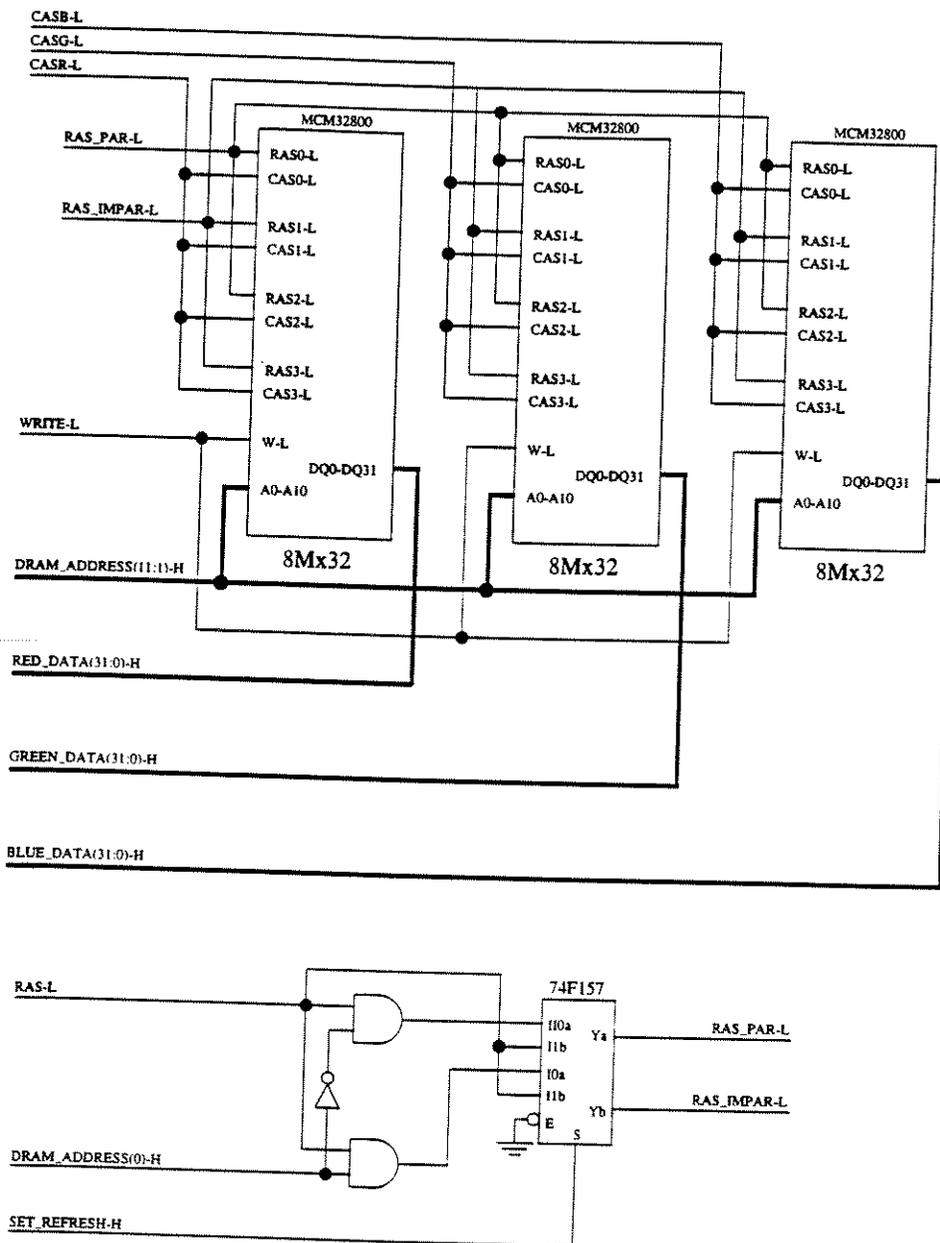


Figura 4.37 - Diagrama do banco de memórias DRAM's com capacidade de 50 Mbytes.

## 4.6.2 Interface com o Barramento de E/S de Vídeo

### Demultiplexador

A implementação de um bloco demultiplexador é mostrada na figura 4.38. Conforme pode ser visto na figura 4.36 um conjunto de três deste blocos deve realizar uma demultiplexagem de um fator  $F$  igual a 4 do fluxo de dados fornecido pelo digitalizador. Assim, um fluxo que originalmente possui 24 bits taxa de dados em torno de (1/70ns) é convertido em um fluxo com palavras de 32 bits e taxa de bits quatro vezes menor. Ele pode ser implementado com registradores paralelos 74F374 e segue a estratégia de montagem de palavras discutido no item 4.5.2.

### Multiplexador

Um bloco multiplexador é mostrado na figura 4.39. Ele realiza a função inversa do demultiplexador e pode ser implementado registradores 74F374.

### Gerador de Endereços

O gerador de endereços é um contador que fornece uma sequência de endereços linha e coluna requeridos pelo Banco de Memórias DRAM para armazenar as palavras de dados formadas pelas amostras de vídeo digital. Conforme mostra a figura 4.40, ele é implementado utilizando-se um arranjo de contadores síncronos 74F579.

### Circuitos de controle

As figuras 4.4.1 e 4.42 para os controladores do mux/demux e do gerador de endereços e dos sinais de RAS e CAS.

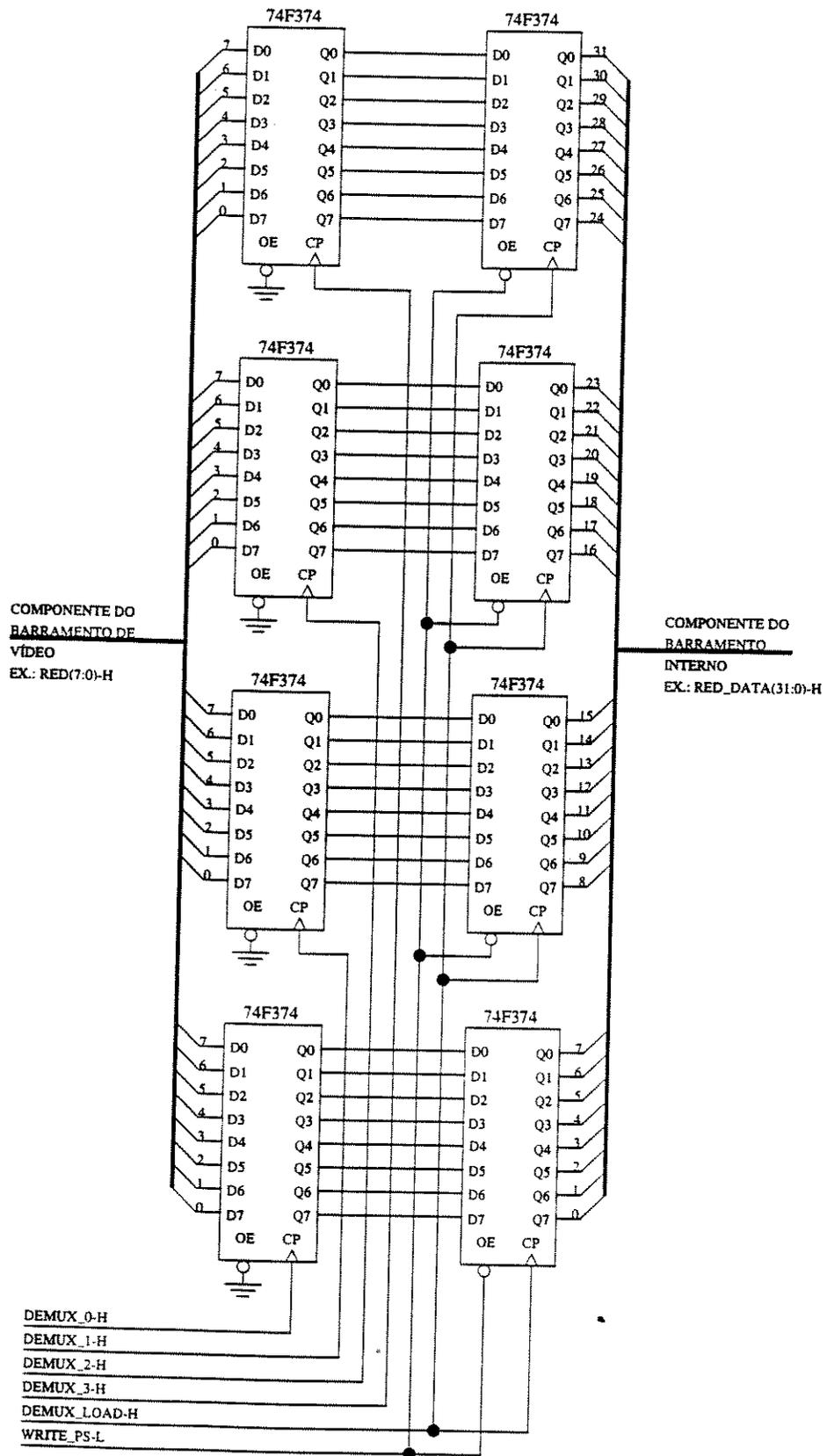


Figura 4.38 - Demultiplexador para uma componente

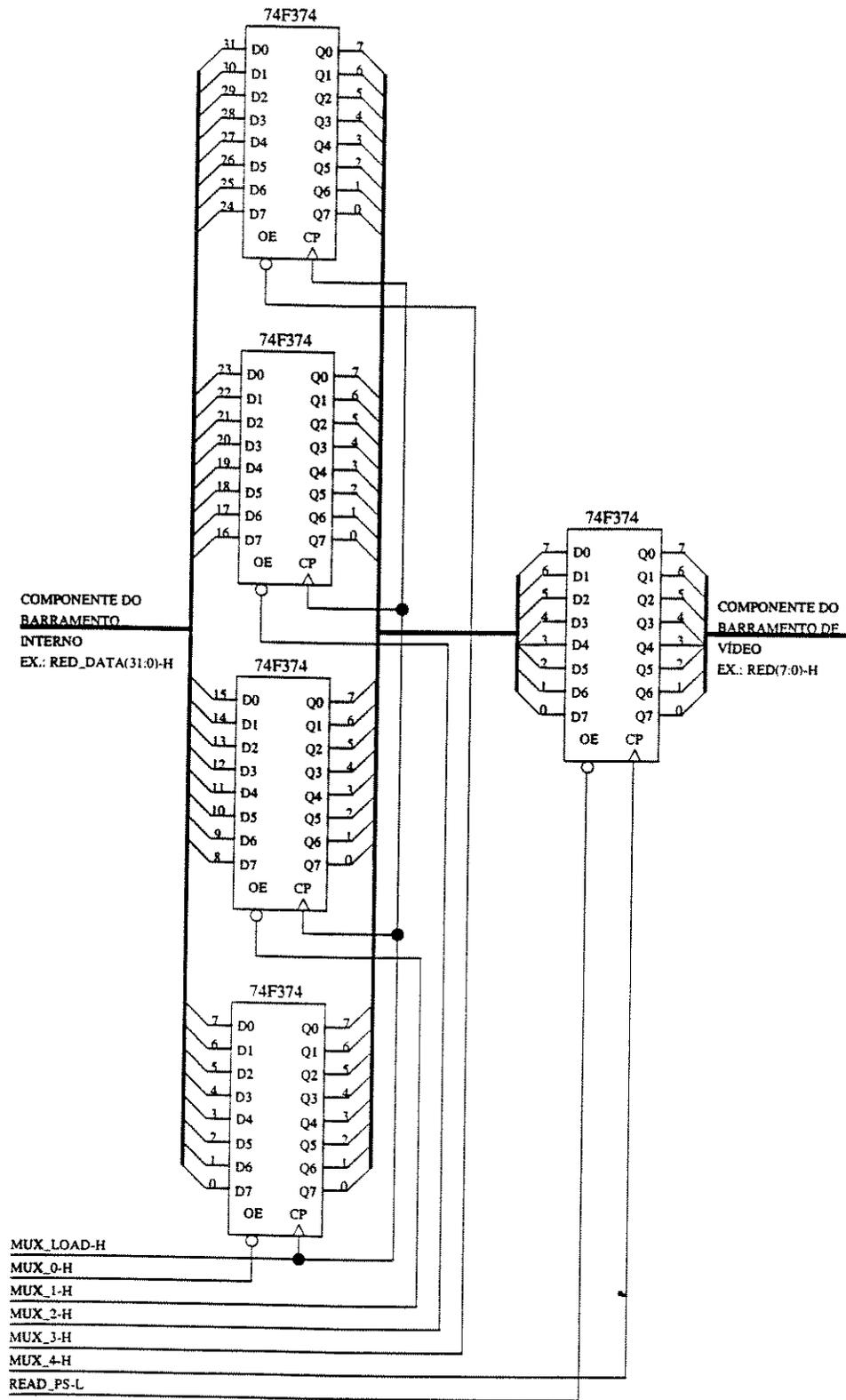


Figura 4.39 - Multiplexador para uma componente

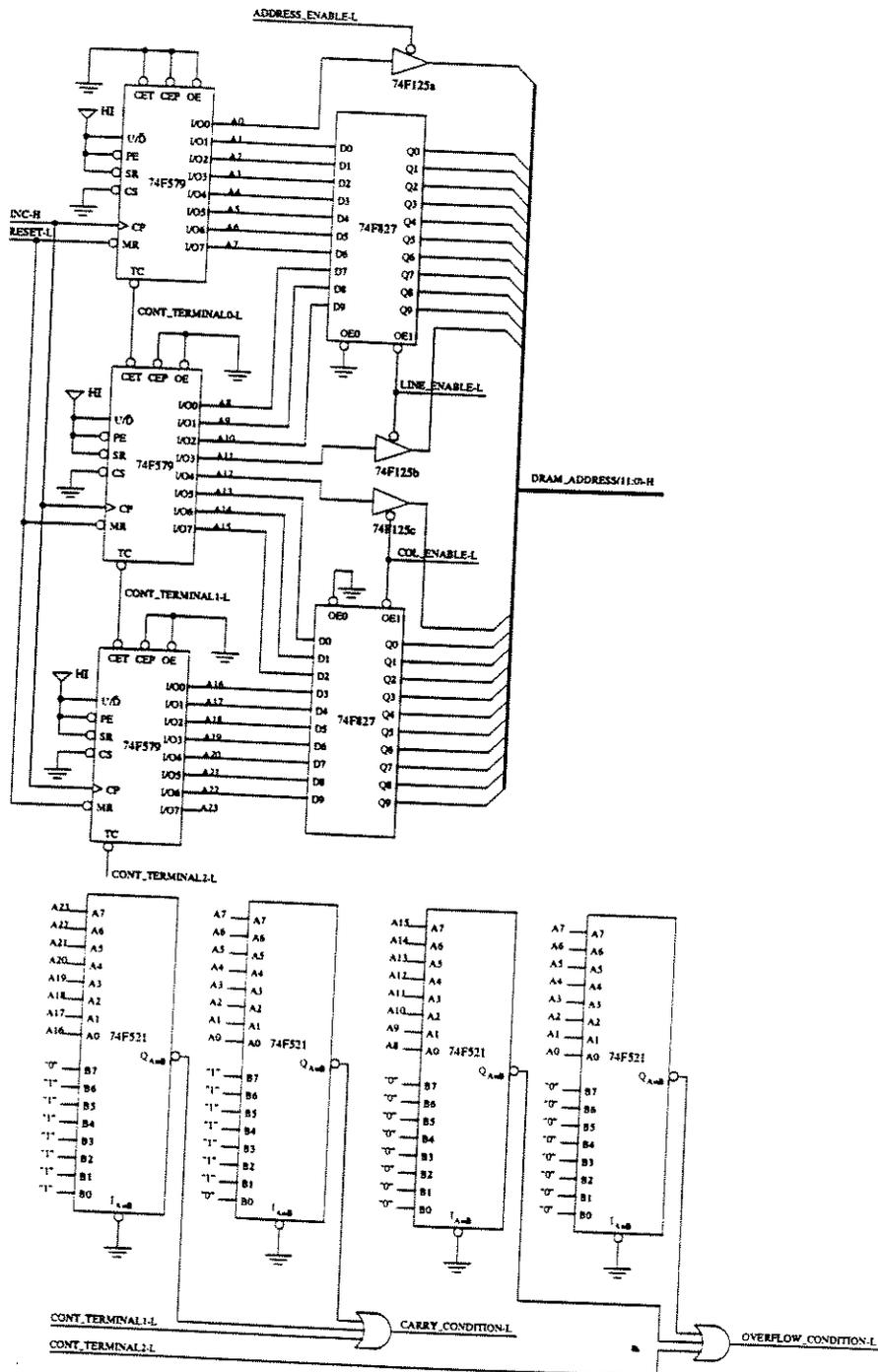


Figura 4.40 - Gerador de endereços

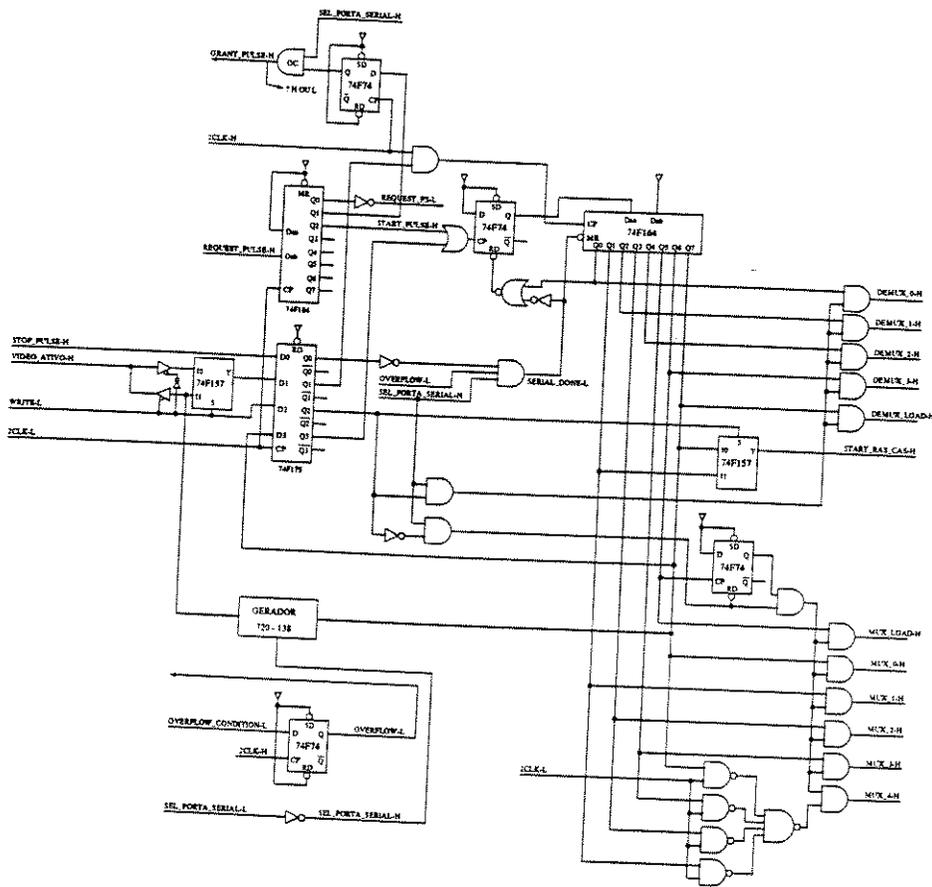


Figura 4.41 - Circuito de controle do demu/mux

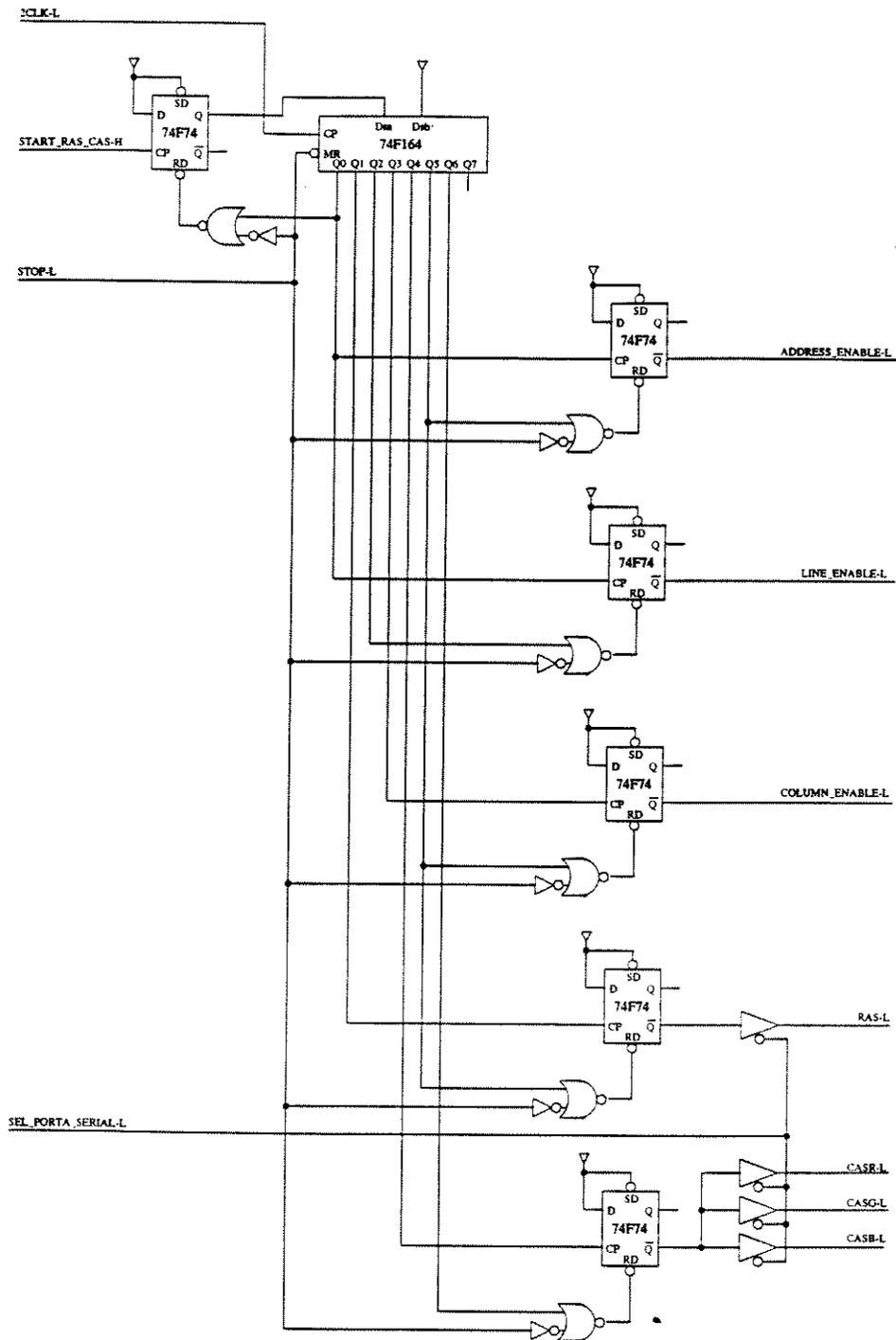


Figura 4.42 - Circuito de controle do gerador de endereços e geração de RAS e CAS.

### 4.6.3 Implementação da interface com o barramento VME

A interface com o Barramento VME é implementada através dos blocos: seletor de componentes, multiplexador /decodificador de endereços bem como o controlador da interface. Estes blocos desempenham as tarefas convencionais necessárias ao interafceamento com o barramento VME.

#### Seletor de componentes

A figura 5.43 mostra um possível diagrama esquemático para o seletor de componentes de dados.

#### Multiplexador e decodificador de endereços

A figura 5.44 mostra um possível diagrama esquemático para este bloco.

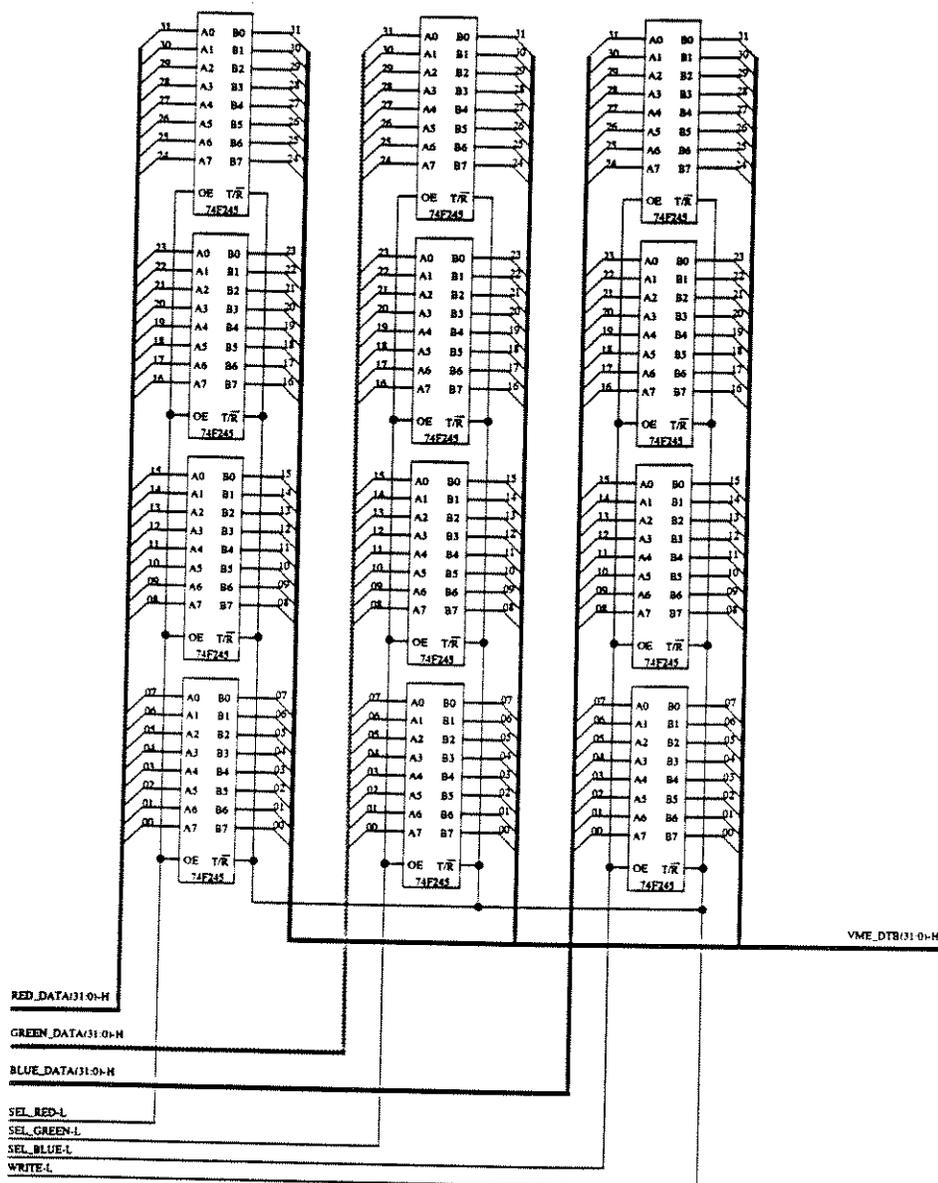


Figura 4.43 - Seletor de componentes de dados

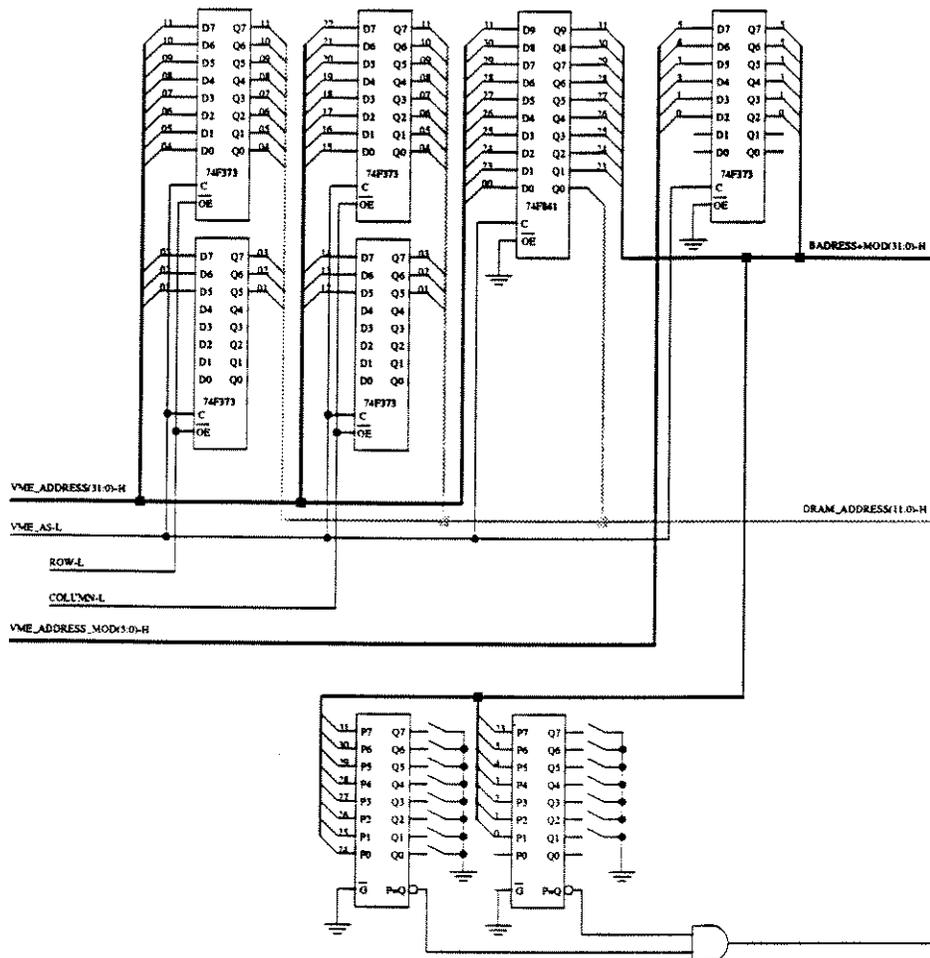


Figura 4.44 - Multiplexador e decodificador de endereços VME.

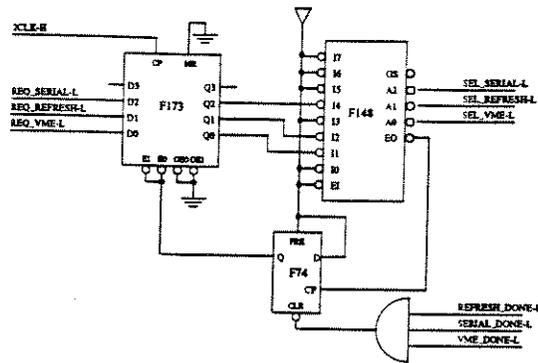


Figura 4.38 - Circuito de arbitragem do barramento interno do armazenador

#### 4.6.4 Implementação da seção de arbitragem do barramento local

Vide figura 5.45.

### Bibliografia

- [1] Motorola, "Dynamic RAMS and Memory Modules Data", Motorola Inc., 1994.
- [2] Texas Instruments, "MOS Memory Data Book", Texas Instruments Inc., 1988.
- [2] Hitachi, "Application of the Multiport Video RAM", Hitach Ltd., 1986.

# CAPÍTULO 5

## INTERFACE DE COMUNICAÇÃO COM O COMPUTADOR HOSPEDEIRO

### 5.1 Por que usar a interface SCSI

A interface de comunicação com o computador hospedeiro poderia ser implementada através de uma interface padrão como o SCSI Bus, a IEEE 488 (GPIB), ou até mesmo por meio da porta paralela Centronics disponível para conexão de impressoras em sistemas PC's. Dentre essas alternativas a mais interessante é a interface SCSI Bus, que além de ser a mais rápida, está disponível tanto em PC's (através de placa adaptadoras) quanto em estações SUN. Uma outra vantagem da SCSI é que outros periféricos como discos rígidos poderiam vir a ser integrados ao sistema dedicado através do barramento SCSI.

### 5.2 A Interface SCSI

Em 1981, a NCR Corporation e a Shugart Associates juntaram esforços a fim de desenvolver uma interface inteligente para disk drivers. Esta interface foi chamada inicialmente de SASI (Shugart Associates Systems Interface), e em dezembro do mesmo ano, estas duas empresas propuseram ao comitê de padrões X3T9 que aceitasse a SASI como documento de trabalho para o estabelecimento de um padrão ANSI. O subcomitê X3T9.2 iniciou então os trabalhos com a nova interface e renomeou-a com a sigla SCSI (Small Computer System Interface). O trabalho extensivo realizado pelo subcomitê resultou em várias melhorias e revisões para a SCSI, e em abril de 1984, a Revisão 14 foi submetida ao comitê de padrões X3T9 para dar início ao processo de aceitação. Este processo incluiu novas revisões até que em junho de 1986, a Revisão 17b foi aceita pela ANSI como um padrão oficial.

Ocorreu entretanto, que o padrão SCSI especificado pela ANSI, possuía muitos comandos e características cuja implementação não era obrigatória, e além disso, muitas outras características que foram identificadas como necessárias pelos fabricantes não haviam sequer sido definidas. A SCSI tentava contornar este problema, fornecendo ao desenvolvedor de sistemas a capacidade de adicionar os comandos e características que desejasse através de comandos do tipo "vendor-unique-command". Esta flexibilidade embora possa parecer positiva, pode levar a incompatibilidades entre produtos baseadas na SCSI, que tenham sido desenvolvidos por fabricantes diferentes.

Um grupo da indústria atento a este problema, definiu um conjunto de 18 comandos comuns, chamado de CCS (Common Command Set), que foi aceito pelo subcomitê X3T9.2 como parte integrante

da base para a proposta de uma extensão da SCSI. O subcomitê chamou esta extensão de SCSI-2, e trabalhou nela por alguns anos, até que em janeiro de 1994, a ANSI aprovou a Revisão 10L da SCSI-2 como padrão, passando a designá-la como X3.131-1994.

A SCSI-2 tem sido largamente utilizada por vários fabricantes de periféricos para computadores pessoais. Ela estendeu a especificação inicial do CCS, incluindo o suporte de dispositivos tais como CD-ROMs, scanners, dispositivos de comunicação, drives de memória óptica, etc. Oferece ainda como opção um barramento mais largo e mais rápido que o da SCSI original<sup>1</sup>, ou seja além do barramento normal de 8 bits, tem-se as opções de 16 e 32 bits. A SCSI-2 suporta taxas de dados de até 10 MBps sobre a implementação de 8 bits, o que significa que se pode chegar a uma taxa de 40 MBps utilizando-se o barramento de 32 bits. As implementações SCSI-2 que chegam aos 10MBps costumam ser chamadas de Fast SCSI, e as que fazem uso do barramento de 16 ou 32 bits são referenciadas como Wide SCSI. é possível ainda encontrar-se implementações “fast e wide” e até “slow e narrow”<sup>2</sup>.

A interface SCSI continua evoluindo, e a medida que surgem novas tecnologias de periféricos, adapta-se para acomodá-las, ainda que sempre exista um compromisso em se manter a compatibilidade com os padrões existentes. O subcomitê X3T9.2 já trabalha na proposta de uma nova interface que sucederá a SCSI-2. Conhecida como SCSI-3, planeja-se que ela suporte fibras ópticas, cabos mais longos, e um número de dispositivos SCSI por barramento maior que os oito atualmente permitidos. Ela será mais rápida podendo atingir taxas de dados de até 20 Mbps<sup>3</sup>.

O objetivo desta seção é realizar uma introdução à interface SCSI-2. As informações aqui expostas visam auxiliar e enriquecer a exposição das sugestões feitas para implementações da interface SCSI do sistema de aquisição/visualização, que serão apresentadas nas seções subseqüentes deste capítulo.

### 5.2.1 Características mecânicas

Os dispositivos SCSI são conectados em *daisy chained* por meio de um cabo comum de 50 condutores (referenciado como cabo A) e opcionalmente por meio de mais um cabo de 68 condutores (referenciado como cabo B). O cabo A transporta os sinais comuns a qualquer dispositivo SCSI, enquanto o cabo B é usado somente por dispositivos que suportam a opção Wide SCSI, embora dispositivos com largura de barramento diferentes possam ser misturados. No que se refer aos drivers/receivers são especificadas duas alternativas:

1. Drivers/receivers com terminação única, que permitem cabos de até 6 metros de comprimento, ou
2. Drivers/receivers com terminação diferencial, que permitem cabos de até 25 metros.

---

<sup>1</sup> Também referenciada como SCSI-1

<sup>2</sup> Implementações mais simples que atingem até 5 MBps e utilizam barramento de 8 bits

<sup>3</sup> Em implementações que utilizam o barramento de 8 bits.

### Requisitos para os cabos

A característica de impedância dos cabos deve estar entre 90 e 140 ohms. Um cabo plano de 50 condutores ou cabo trançado de 25 sinais pode ser utilizado para o cabo A. Para cabo B pode ser utilizado um cabo plano de 68 condutores ou um cabo trançado de 34 sinais.

### Requisitos para os conectores

Dois tipos de conectores são definidos: o blindado e o não-blindado. Os conectores blindados são usados para aplicações externas<sup>4</sup> nas quais requisitos como compatibilidade eletromagnética (EMC) e proteção contra descarga eletrostática (ESD) precisam ser atendidos. Os conectores não-blindados por sua vez, destinam-se a aplicações internas<sup>5</sup>.

As figuras 5.1 e 5.2 mostram os conectores blindados que ficam ligados aos cabos A e B e os que são fixados no chassi dos dispositivos SCSI.

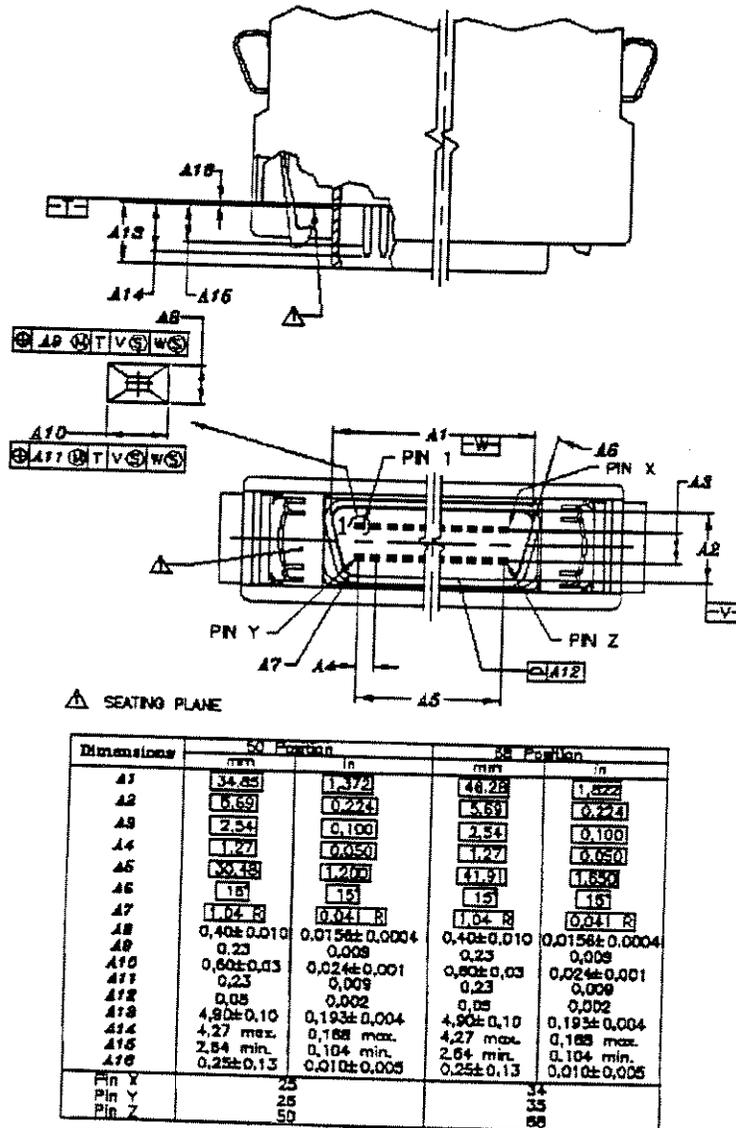
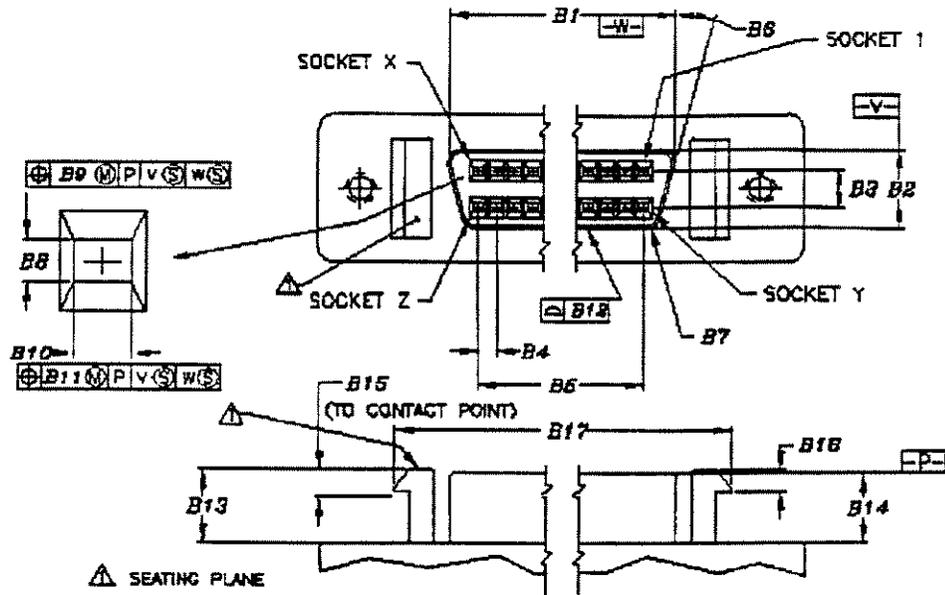


Figura 5.1 - Conector blindado do cabo A ou cabo B

<sup>4</sup> Como interligar o computador hospedeiro a periféricos que estão fora do gabinete do hospedeiro.  
<sup>5</sup> Como interligar o computador hospedeiro a dispositivos que estão no mesmo gabinete.



Dimensions	50 Position		68 Position	
	mm	in	mm	in
B1	34.70	1.366	45.13	1.816
B2	5.54	0.218	5.54	0.218
B3	2.54	0.100	2.54	0.100
B4	1.27	0.050	1.27	0.050
B5	30.48	1.200	41.91	1.650
B6	15°	15°	15°	15°
B7	1.00 R	0.039 R	1.00 R	0.039 R
B8	0.61±0.05	0.024±0.002	0.61±0.05	0.024±0.002
B9	0.15	0.006	0.15	0.006
B10	0.86±0.10	0.034±0.004	0.86±0.10	0.034±0.004
B11	0.15	0.006	0.15	0.006
B12	0.05	0.002	0.05	0.002
B13	5.10±0.05	0.201±0.002	5.10±0.05	0.201±0.002
B14	5.00±0.13	0.197±0.005	5.00±0.13	0.197±0.005
B15	1.65 max.	0.073 max.	1.65 max.	0.073 max.
B16	1.50±0.03	0.059±0.001	1.50±0.03	0.059±0.001
B17	42.29±0.10	1.665±0.004	53.72±0.10	2.115±0.004
Socket X	25		34	
Socket Y	26		35	
Socket Z	50		88	

NOTE - Dimensions B8 and B10 are the opening in the dielectric. The socket contacts (not shown) fit within the opening.

Figura 5.2 - Conector blindado fixado ao dispositivo

## Pinagem dos conectores dos cabos A e B

As associações dos sinais SCSI aos pinos dos conectores dos cabos A e B são ilustradas nas tabelas 5.1 até 5.4<sup>6</sup>.

*Tabela 5.1 - Associação dos sinais - Cabo A  
Implementação com terminação única*

<i>SINAL</i>	<i>Pino</i>	<i>Pino</i>	<i>SINAL</i>
GROUND	1	26	-DB(0)
GROUND	2	27	-DB(1)
GROUND	3	28	-DB(2)
GROUND	4	29	-DB(3)
GROUND	5	30	-DB(4)
GROUND	6	31	-DB(5)
GROUND	7	32	-DB(6)
GROUND	8	33	-DB(7)
GROUND	9	34	-DB(P)
GROUND	10	35	GROUND
GROUND	11	36	GROUND
RESERVED	12	37	RESERVED
OPEN	13	38	TERMPWR
RESERVED	14	39	RESERVED
GROUND	15	40	GROUND
GROUND	16	41	-ATN <sup>7</sup>
GROUND	17	42	GROUND
GROUND	18	43	-BSY
GROUND	19	44	-ACK
GROUND	20	45	-RST
GROUND	21	46	-MSG
GROUND	22	47	-SEL
GROUND	23	48	-C/D
GROUND	24	49	-REQ
GROUND	25	50	-I/O

<sup>6</sup> A pinagem ilustrada nestas tabelas não vale para os conectores sem blindagem de alta densidade utilizados no cabo A.

<sup>7</sup> (-) a frente do sinal indica que o sinal é ativo-baixo.

**Tabela 5.2 - Pinagem do conector - Cabo B**  
**Implementação com terminação única**

<i>SINAL</i>	<i>Pino</i>	<i>Pino</i>	<i>SINAL</i>
GROUND	1	35	GROUND
GROUND	2	36	-DB(8)
GROUND	3	37	-DB(9)
GROUND	4	38	-DB(10)
GROUND	5	39	-DB(11)
GROUND	6	40	-DB(12)
GROUND	7	41	-DB(13)
GROUND	8	42	-DB(14)
GROUND	9	43	-DB(15)
GROUND	10	44	-DB(P1)
GROUND	11	45	-ACKB
GROUND	12	46	GROUND
GROUND	13	47	-REQB
GROUND	14	48	-DB(16)
GROUND	15	49	-DB(17)
GROUND	16	50	-DB(18)
TERMPWRB	17	51	TERMPWRB
TERMPWRB	18	52	TERMPWRB
GROUND	19	53	-DB(19)
GROUND	20	54	-DB(20)
GROUND	21	55	-DB(21)
GROUND	22	56	-DB(22)
GROUND	23	57	-DB(23)
GROUND	24	58	-DB(P2)
GROUND	25	59	-DB(24)
GROUND	26	60	-DB(25)
GROUND	27	61	-DB(26)
GROUND	28	62	-DB(27)
GROUND	29	63	-DB(28)
GROUND	30	64	-DB(29)
GROUND	31	65	-DB(30)
GROUND	32	66	-DB(31)
GROUND	33	67	-DB(P3)
GROUND	34	68	GROUND

**Tabela 5.3 - Pinagem do conector - Cabo A**  
**Implementação com terminação diferencial**

<i>SINAL</i>	<i>Pino</i>	<i>Pino</i>	<i>SINAL</i>
GROUND	1	26	GROUND
+DB(0)	2	27	-DB(0)
+DB(1)	3	28	-DB(1)
+DB(2)	4	29	-DB(2)
+DB(3)	5	30	-DB(3)
+DB(4)	6	31	-DB(4)
+DB(5)	7	32	-DB(5)
+DB(6)	8	33	-DB(6)
+DB(7)	9	34	-DB(7)
+DB(P)	10	35	-DB(P)
DIFFSENS	11	36	GROUND
RESERVED	12	37	RESERVED
TERMPWR	13	38	TERMPWR
RESERVED	14	39	RESERVED
+ATN	15	40	-ATN
GROUND	16	41	GROUND
+BSY	17	42	-BSY
+ACK	18	43	-ACK
+RST	19	44	-RST
+MSG	20	45	-MSG
+SEL	21	46	-SEL
+C/D	22	47	-C/D
+REQ	23	48	-REQ
+I/O	24	49	-I/O
GROUND	25	50	GROUND

**Tabela 5.4 - Associação dos sinais ao Cabo B**  
**Implementação com terminação diferencial**

<i>SINAL</i>	<i>Pino</i>	<i>Pino</i>	<i>SINAL</i>
GROUND	1	35	GROUND
+DB(8)	2	36	-DB(8)
+DB(9)	3	37	-DB(9)
+DB(10)	4	38	-DB(10)
+DB(11)	5	39	-DB(11)
+DB(12)	6	40	-DB(12)
+DB(13)	7	41	-DB(13)
+DB(14)	8	42	-DB(14)
+DB(15)	9	43	-DB(15)
+DB(P1)	10	44	-DB(P1)
+ACKB	11	45	-ACKB
GROUND	12	46	DIFFSENS
+REQB	13	47	-REQB
+DB(16)	14	48	-DB(16)
+DB(17)	15	49	-DB(17)
+DB(18)	16	50	-DB(18)
TERMPWRB	17	51	TERMPWRB
TERMPWRB	18	52	TERMPWRB
+DB(19)	19	53	-DB(19)
+DB(20)	20	54	-DB(20)
+DB(21)	21	55	-DB(21)
+DB(22)	22	56	-DB(22)
+DB(23)	23	57	-DB(23)
+DB(P2)	24	58	-DB(P2)
+DB(24)	25	59	-DB(24)
+DB(25)	26	60	-DB(25)
+DB(26)	27	61	-DB(26)
+DB(27)	28	62	-DB(27)
+DB(28)	29	63	-DB(28)
+DB(29)	30	64	-DB(29)
+DB(30)	31	65	-DB(30)
+DB(31)	32	66	-DB(31)
+DB(P3)	33	67	-DB(P3)
GROUND	34	68	GROUND

### 5.2.2 Especificações elétricas

O barramento SCSI suporta duas especificações elétricas: uma com terminação única e a outra com terminação diferencial. A versão com terminação única usa níveis de sinal TTL, enquanto a versão com terminação diferencial usa sinais EIA RS-485 para permitir o uso de cabos mais longos.

#### Alternativa com terminação única (single ended)

Todos os sinais<sup>8</sup> devem ser terminados nas duas extremidades dos cabos, e esta terminação pode ser feita através de dois métodos:

- a) A terminação de cada sinal pode constituir-se de 220 ohms  $\pm 5\%$  para a linha TERMPWR (Terminator Power) e de 330 ohms  $\pm 5\%$  para a referência de terra, conforme mostra a figura 5.3, ou
- b) A terminação de cada sinal pode atender os requisitos a seguir (ver figura 5.4):
  1. Cada terminador fornece uma impedância característica entre 100 e 132 ohms.
  2. Os terminadores são alimentados pela linha TERMPWR e podem receber alimentação adicional a partir de outras fontes embora isto não seja necessário para que operem corretamente.

<sup>8</sup>Exceção feita àqueles que foram definidos como RESERVED, GROUND, ou TERMPWR).

3. A corrente disponível para qualquer driver de linha não deve exceder os 48 mA, quando ele ativa a linha derrubando sua tensão para 0,5V dc. Sendo que desta corrente apenas 44.8 mA devem estar disponíveis a partir dos dois terminadores da linha de sinal.
4. A tensão nas linhas de sinais quando desativadas deve ser de pelo menos 2,5 V dc.

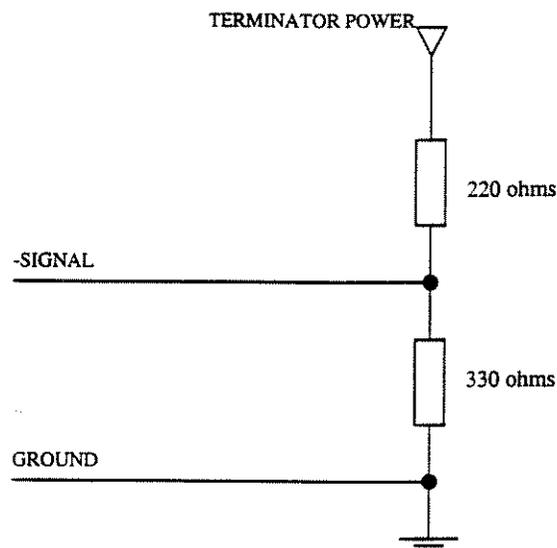


Figura 5.3 - Alternativa a) para implementação do terminador single ended

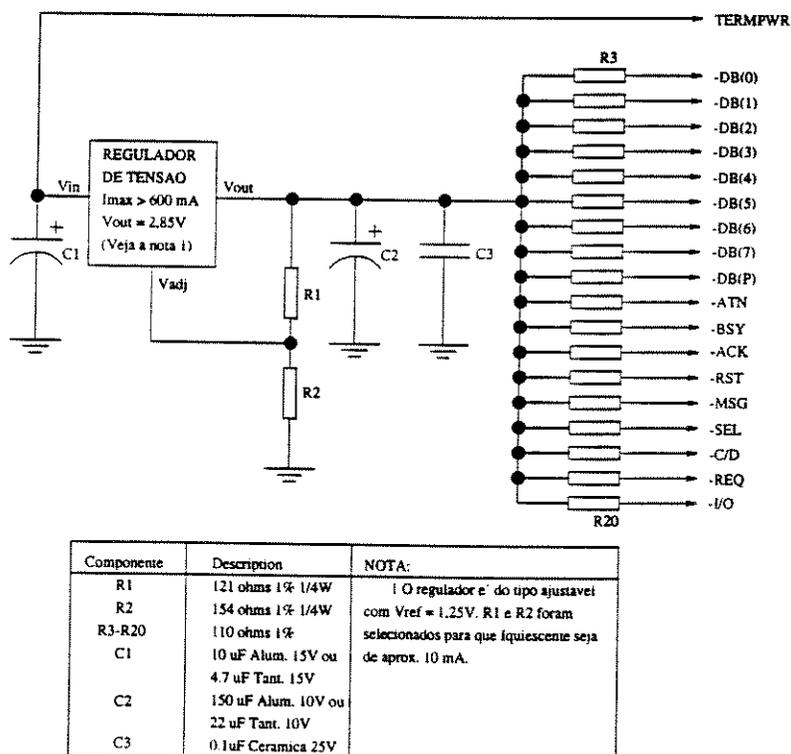
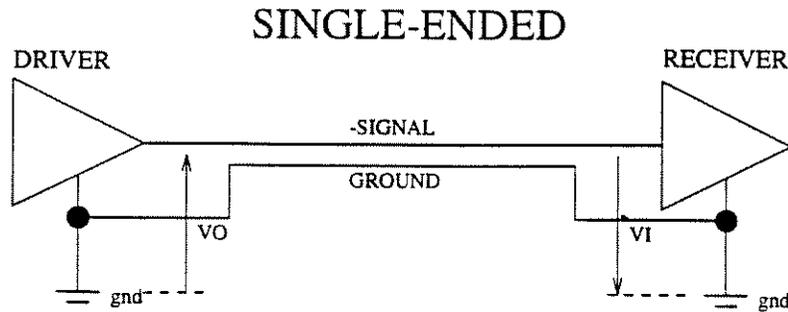


Figura 5.4 - Alternativa b) para implementação do terminador single ended



**Figura 5.5** - Tensões de entrada e saída para o terminador single ended

As condições acima devem ser encontradas na configuração de targets e initiators, assim que pelo menos um dispositivo for alimentado através da linha TERMPWR.

**Características de saída** Todos os sinais usam drivers open-collector ou three-state. Cada sinal acionado por um dispositivo SCSI terá as seguintes características de saída:

$V_{OL}$ (tensão de nível baixo de saída)	De 0,0V a 0,8V (sinal verdadeiro)
$V_{OH}$ (tensão de nível alto de saída)	De 2,5V a 5,25V (sinal falso)

$V_{OL}$  e  $V_{OH}$  devem ser medidas entre o terminal de saída (ligada à linha -SIGNAL) e a referência de terra do dispositivo SCSI (ver figura 5.5).

**Características de entrada** Cada sinal recebido por um dispositivo SCSI terá as seguintes características de entrada:

$V_{IL}$ (tensão de nível baixo de entrada)	De 0,0V a 0,8V (sinal verdadeiro)
$V_{IH}$ (tensão de nível alto de entrada)	De 2,0V a 5,25V (sinal falso)
$I_{IL}$ (corrente de entrada de nível baixo)	De -0,4mA a 0,0mA com $V_I = 0,5V$
$I_{IH}$ (corrente de entrada de nível alto)	De 0,0mA a 0,1mA com $V_I = 2,7V$
Capacitância de entrada máxima	25pF

### Alternativa Diferencial

Todos os sinais diferenciais são constituídos de duas linhas denotadas +SIGNAL e -SIGNAL. Um sinal será tomado como verdadeiro quando sua linha +SIGNAL for mais positiva que sua linha -SIGNAL. Todos os sinais devem ser terminados como mostra a figura 5.6, e a impedância dos terminadores diferenciais deve ser de 122 ohms. Estas características foram adicionadas segundo a EIA RS-485-1983, e permitem o uso de cabos mais longos.

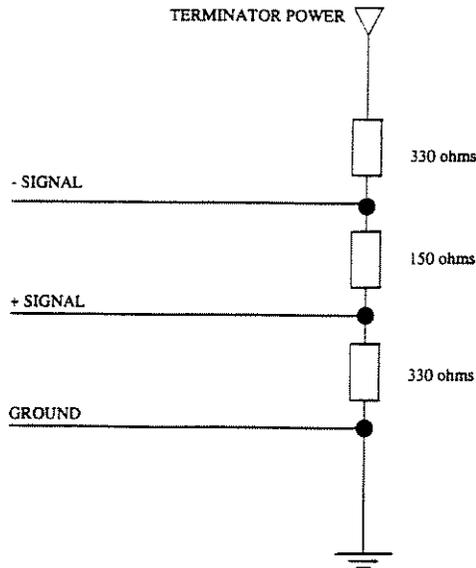


Figura 5.6 - Terminador diferencial

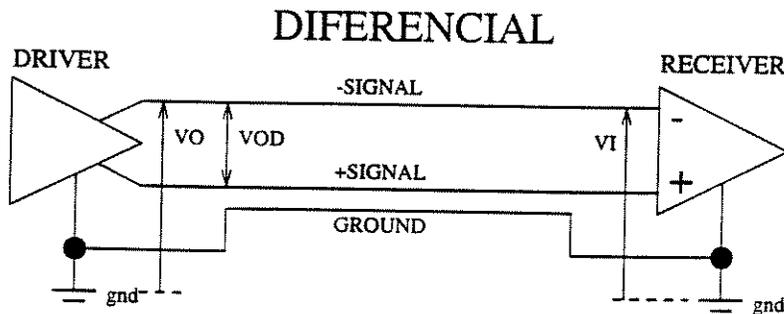


Figura 5.7 - Tensões de entrada e saída para o terminador diferencial

**Características de saída** Todo sinal diferencial ao ser acionado por um dispositivo SCSI deve ter as seguintes características:

$V_{OL}$ (tensão de nível baixo de saída)	Máximo de 1,7V com $I_{OL}$ (corrente de saída de nível baixo) = 55mA.
$V_{OH}$ (tensão de nível alto de saída)	Mínimo de 2,7V com $I_{OH}$ (corrente de saída de nível alto) = -55mA.
$V_{OD}$ (tensão de saída diferencial)	Mínimo de 1,0V com a tensões de modo comum na faixa de -7V a +12V

$V_{OL}$  e  $V_{OH}$  devem ser medidas entre o terminal de saída (ligado à linha -SIGNAL) e a referência de terra do dispositivo SCSI (ver figura 5.7).

**Características de entrada** Todo sinal diferencial recebido por um dispositivo SCSI terá as seguintes características:

$I_I$ (corrente de entrada)	Máximo de $\pm 2,0\text{mA}$
Capacitância de entrada máxima	25 pF

### Alimentação dos terminadores

Os initiators fornecerão alimentação aos contatos TERMPWR e TERMPWRB<sup>9</sup>. Esta alimentação deve ser fornecida através de um diodo para prevenir “back flow” de tensão para o dispositivo. Os targets não são obrigados a alimentar os terminadores, embora qualquer dispositivo SCSI possa fazê-lo.

### 5.2.3 Definições Lógicas da SCSI

Nos dois primeiros itens desta seção, tratamos das características físicas da interface SCSI. A partir do presente item, passaremos a discutir as características lógicas da interface. A SCSI é um barramento de entrada e saída local e o seu objetivo primário é fornecer ao computador hospedeiro uma independência com relação a um periférico dentro de uma classe de periféricos. Assim sendo, diferentes disk drivers, tape drivers, impressoras ou mesmo dispositivos de comunicação podem ser adicionados ao computador hospedeiro sem a necessidade de grandes modificações no hardware ou no software do sistema.

O barramento SCSI suporta um máximo de oito dispositivos, sendo que destes oito, apenas um par pode comunicar-se de cada vez. Cada dispositivo SCSI tem associado a si, um bit de identificação chamado de bit SCSI ID. O bit SCSI ID nada mais é que uma representação de um bit significativo do endereço de um dispositivo SCSI. Como pode ser visto na figura 5.8, cada bit ID referencia uma das linhas de sinal DB7-DB0<sup>10</sup>, com a linha DB7 representando o dispositivo de maior prioridade.

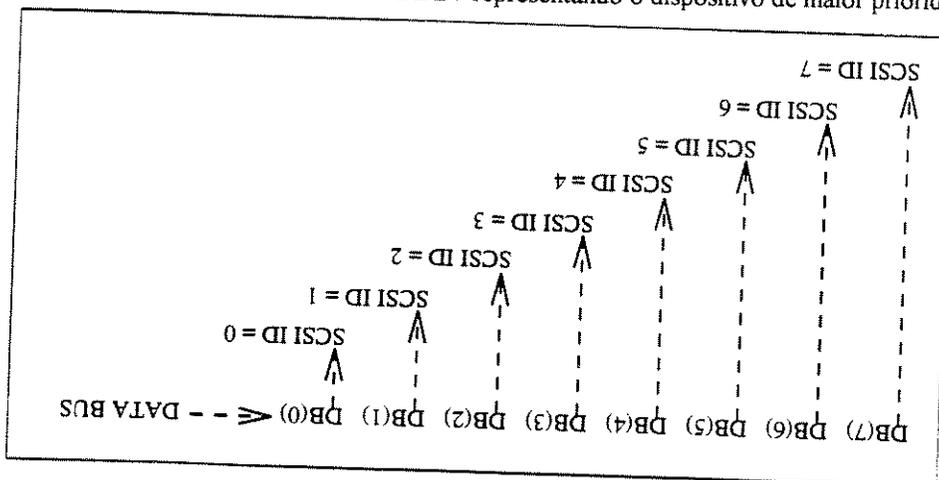


Figura 5.8 - Bits SCSI ID

<sup>9</sup> Se a opção wide SCSI for implementada.

<sup>10</sup> Os sinais DB7-DB0 além de transportar o bit de identificação de dispositivo SCSI ID, constituem-se no barramento de dados da SCSI, transportando informação de dados, comandos, mensagens, e status. Os sinais SCSI serão abordados mais detalhadamente no item 5.2.4

**Target e Initiator** Quando dois dispositivos SCSI se comunicam, um deles atua como initiator (o hospedeiro, por exemplo), e o outro atua como target (controlador de periférico). O initiator é o dispositivo que origina a operação E/S, e o target é o dispositivo que executa esta operação. A maioria dos dispositivos SCSI tem uma tarefa fixa como initiator ou target, mas alguns dispositivos são capazes de assumir os dois papéis.

**Configurações de Targets e Initiators** A SCSI foi projetada para ser um barramento E/S flexível, permitindo ao computador hospedeiro comunicar-se com periféricos, com outros hospedeiros, e ainda, que os periféricos do barramento possam comunicar-se entre si. Na figura 5.9 tem-se exemplos de quatro configurações distintas de targets e initiators para sistemas de E/S baseados na SCSI:

- Um Hospedeiro (initiator) e um único Controlador (target)
- Um Hospedeiro (initiator) e vários Controladores (targets)
- Vários Hospedeiros (initiators) e um único Controlador (target)
- Vários Hospedeiros (initiators) e vários Controladores (targets)

Pode-se observar, que nestes sistemas de E/S, os computadores hospedeiros costumam ser os initiators da configuração, enquanto os controladores de dispositivos em geral trabalham como targets do barramento SCSI.

## Níveis de endereçamento na SCSI

Há 3 níveis de endereçamento na arquitetura da SCSI conforme mostra a figura 5.10.

**Endereço do dispositivo SCSI:** No nível mais alto está o endereço que o dispositivo SCSI ocupa (ou reconhece) dentro do barramento, que é representado pelo já mencionado bit SCSI ID. Geralmente, o dispositivo fornece um meio (como por exemplo chaves ou jumpers) para selecionar um dos oito endereços disponíveis. Cada dispositivo no barramento SCSI deve estar associado a um único endereço.

**Logical Unit Number e Target Routine Number:** Em um nível intermediário temos então os *Logical Unit Number (LUN)* usadas para endereçar as *logical units* e os *Target Routine Number (TRU)* usadas para endereçar as *target routines*. Uma logical unit pode referir-se a uma parte do periférico, a um periférico, ou a um grupo contendo vários periféricos. Um máximo de oito logical units por target é permitido. O conceito de logical unit não é definido para um initiator. As target routines são processos que executam diretamente sobre o target e não estão associados a uma logical unit ou periférico em particular. As targets routines tem como principal propósito retornar informação a respeito do target. A SCSI permite que ao target até oito target routines.

**Logical Block Address** No nível mais baixo temos o Logical Block Address (LBA) que são usados para endereçar os chamados logical blocks. Um logical block refere-se a uma parte de um periférico (logical unit), e o seu tamanho bem como o número de logical blocks suportados por uma logical unit são definidos segundo as necessidades do projetista do sistema.

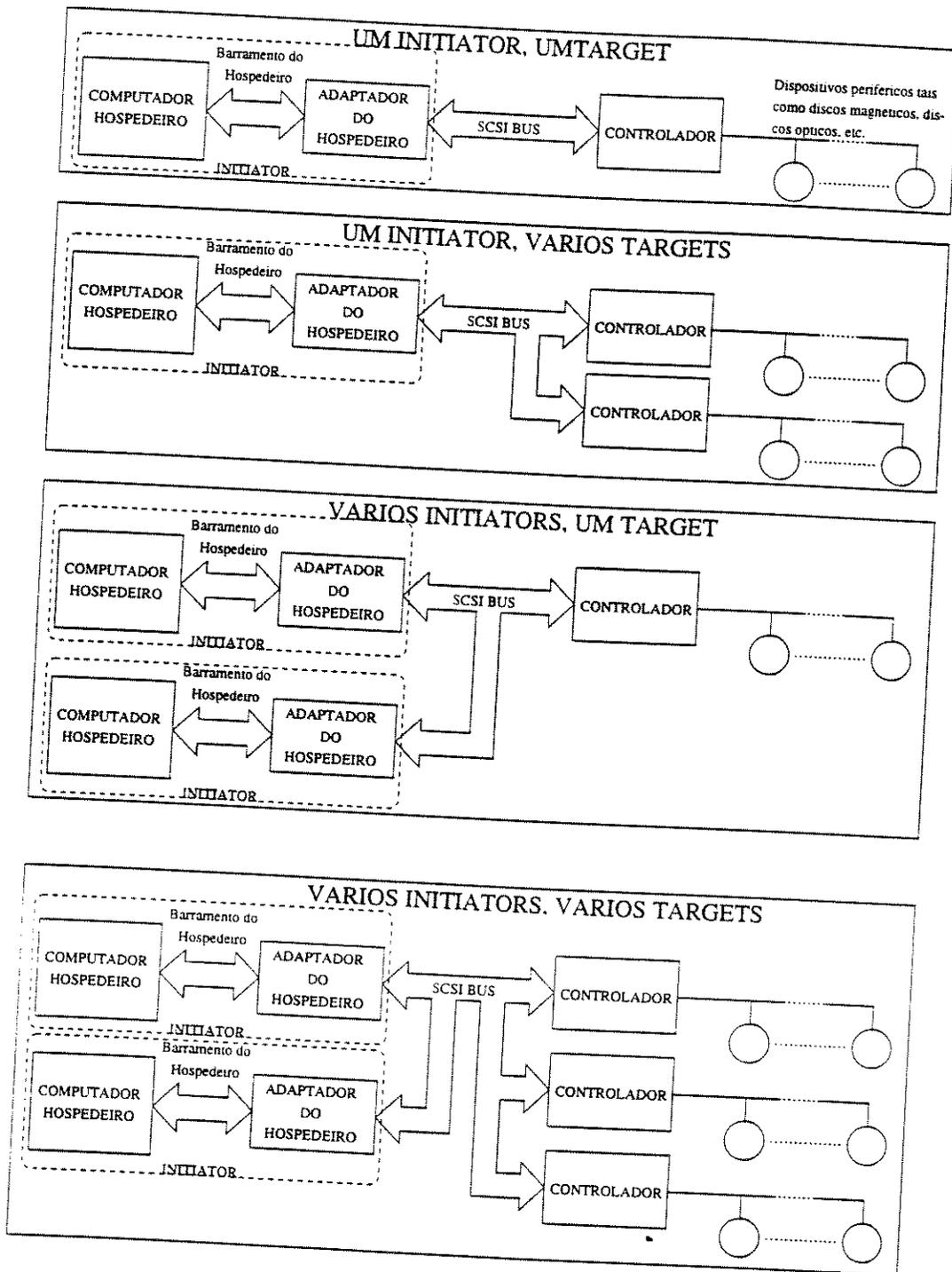


Figura 5.9 - Possíveis configurações de E/S de sistemas baseados na SCSI

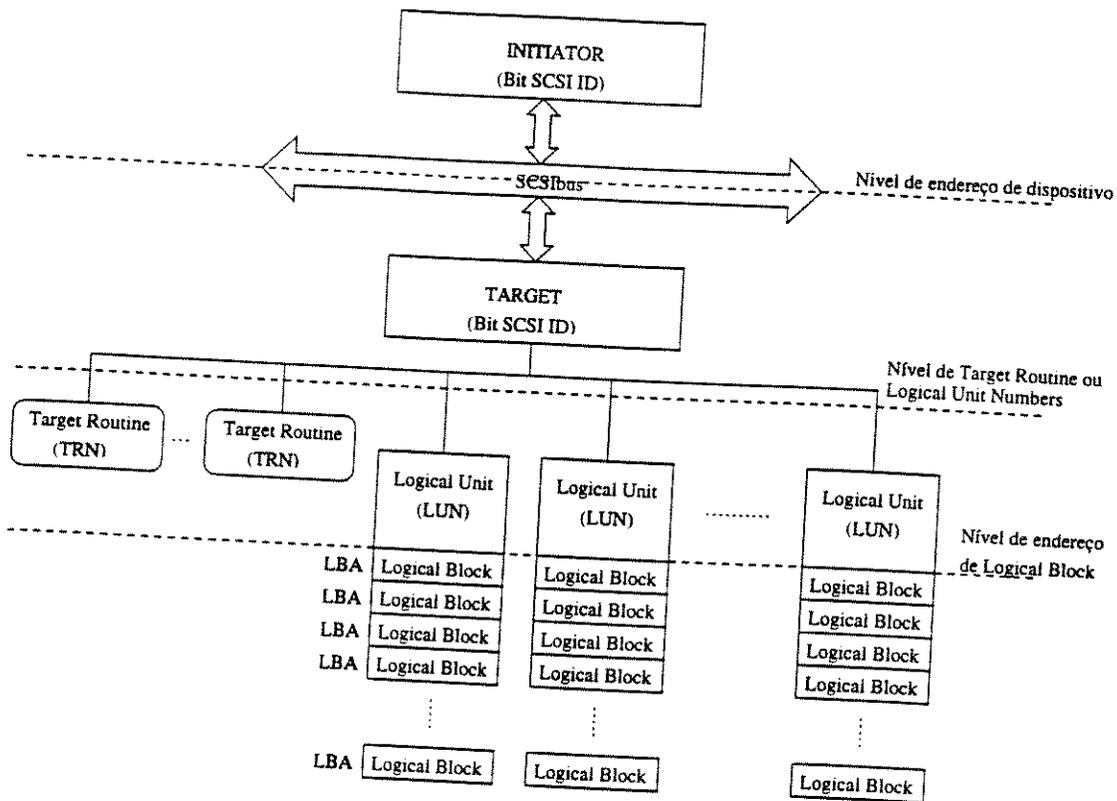


Figura 5.10 - Níveis de endereçamento da SCSI

## 5.2.4 Sinais da SCSI

Há um total de 18 sinais no cabo A e 29 no cabo B. Trinta e seis desses sinais são usados para transferência de dados, e os onze restantes são para sinalização de controle do barramento. Os sinais SCSI são descritos a seguir:

**DB(7-0,P) (Barramento de dados)** O initiator e o target usam este barramento de 8 bits bidirecional para transferência de dados. Essas linhas não transferem somente dados, comandos, informação de status e mensagens, mas também os bits SCSI ID durante as fases de seleção e reSeleção. A linha DB(P) representa o bit de paridade ímpar.

**DB(31-8,P1,P2,P3) (Barramento de dados)** Estas vinte e quatro linhas de dados constituem uma extensão para o barramento de dados DB(7-0,P). As linhas DB(P1,P2,P3) são bits de paridade para DB(15-8), DB(23-16), e DB(31-24) respectivamente. A paridade usada nos três bits referidos deve ser ímpar.

**Sinais de controle** Os onze sinais de controle podem ser divididos em três grupos funcionais:

- Grupo 1 - formado pelos sinais SEL e BSY que permitem ao initiator selecionar um target. Eles também são usados na fase de reSeleção.

- Grupo 2 - formado pelos sinais C/D, I/O, MSG, REQ, REQB ACK e ACKB, que controlam, como veremos mais adiante, as fases de transferência de informação.
- Grupo 3 - formado pelos sinais ATN e RST, que são as duas condições assíncronas do barramento SCSI.

**SEL (Select)** O initiator usa este sinal durante a fase de seleção do target que executará o comando. O target também pode usar o sinal SEL para reSelecionar o initiator do qual havia se desconectado anteriormente.

**BSY (Busy)** Um sinal “wired or” usado para indicar que o barramento SCSI está em uso (busy).

**C/D (Control/Data)** O target usa este sinal para indicar se a informação que está sendo transferida é de controle ou de dados. Quando C/D está ativado (verdadeiro) isto significa que a informação pode ser um comando, um byte de status, ou uma mensagem. Se C/D está desativado (falso), então a informação transferida é um dado.

**I/O (Input/Output)** O target usa este sinal para definir a direção da transferência de informação entre o initiator e o target. A direção da transferência é definida com relação ao initiator. Quando está ativado este sinal indica que a informação está sendo transferida para initiator a partir do target. Quando está desativado o sinal I/O indica que a transferência é do initiator para o target.

**MSG (Message)** O target usa este sinal para indicar ao initiator que a informação que está sendo transferida é uma Mensagem.

**REQ (Request)** O target no cabo A usa este sinal para requerer uma transferência de dados sob o handshake REQ/ACK entre o initiator e o target. Em resposta a ativação do sinal de Request pelo target, o initiator aceita o dado colocado no barramento durante uma “fase Data-In” ou coloca o dado no barramento durante uma “fase Data-Out”. O target continua a ativar o sinal REQ até que o initiator responda ativando o sinal ACK, a menos que se esteja realizando uma transferência de dados síncrona.

**REQB (Request)** Análogo ao sinal REQ para o cabo B.

**ACK (Acknowledge)** O initiator no cabo A usa este sinal para responder à ativação do sinal REQ pelo target selecionado. A ativação do sinal ACK pelo initiator indica que ele colocou um dado no barramento durante uma “fase Data-Out” ou que aceitou um dado a partir do barramento durante uma “fase Data-In”. O handshake REQ/ACK é usado para todas as transferências de informação entre initiator e target.

**ACKB (Acknowledge)** Análogo ao sinal ACK para o cabo B.

**ATN (Attention)** O initiator usa este sinal para informar ao target selecionado que uma mensagem está disponível. O target, de acordo com sua conveniência, requisitará esta mensagem usando a fase MESSAGE-OUT.

**RST (Reset)** Este sinal “wired or” pode ser usado por qualquer dispositivo do barramento SCSI, para indicar a condição Reset.

## Temporização dos sinais SCSI

As características de temporização dos sinais SCSI são mostradas na tabela 5.5.

*Tabela 5.5 - Temporização do SCSIbus*

Arbitration delay	2,4 us
Assertion period	90 ns
Bus clear delay	800 ns
Bus Free delay	800 ns
Bus set delay	1,8 us
Bus settle delay	400 ns
Cable skew delay	10 ns
Data release delay	400 ns
Deskew delay	45 ns
Disconnection delay	200 us
Hold time	45 ns
Negation period	90 ns
Power-on to selection time	10 s
Reset to selection time	(recomendado) 250 s
Reset hold time	(recomendado) 25 us
Selection abort time	200 us
Selection time-out delay	250 ms
Transfer period	(recomendado) Setado durante uma mensagem STDR
Fast assertion period	30 ns
Fast cable skew delay	5 ns
Fast deskew delay	20 ns
Fast hold time	10 ns
Fast negation period	30 ns

Os parâmetros de temporização dos sinais SCSI listados na tabela 5.5 são descritos a seguir:

**Arbitration delay:** Tempo mínimo que um dispositivo SCSI deve esperar a partir da ativação do sinal BSY, para poder examinar as linhas DB(7-0) e verificar se ele venceu o processo de arbitragem do barramento.

**Assertion period:** Largura mínima dos pulsos de ativação dos sinais REQ e ACK durante transferências no modo síncrono.

**Bus clear delay:** Tempo máximo que um dispositivo SCSI pode gastar para desativar todos os seus sinais do barramento, quando receber uma solicitação para tanto. Tal solicitação pode correr em três situações:

- Quando a fase *BUS FREE* é detectada;
- Quando o sinal SEL é recebido de outro dispositivo SCSI durante a fase Arbitration; e
- Quando uma ativação do sinal RST é percebida

**Bus Free delay:** Tempo mínimo que um dispositivo SCSI deve esperar após a detecção da fase BUS FREE, para poder ativar o sinal BSY e levar o barramento para a fase Arbitration.

**Bus set delay:** Tempo máximo para que um dispositivo SCSI ative a linha BSY e o seu bit SCSI ID levando o barramento para a fase ARBITRATION, contado a partir da detecção da fase BUS FREE.

**Bus settle delay:** Tempo mínimo a se esperar para que o barramento se estabilize após a mudança de certos sinais de controle.

**Cable skew delay:** A diferença máxima no tempo de propagação de dois sinais do barramento SCSI quaisquer medida entre dois dispositivos SCSI quaisquer.

**Data release delay:** Tempo máximo para que um initiator desative as linhas de dados após a transição do sinal I/O de “falso” para “verdadeiro”.

**Deskew delay:** Tempo mínimo para a propagação de certos sinais.

**Disconnection delay:** Tempo mínimo que um target, que anteriormente participava de uma arbitragem, deve esperar após a desativação do sinal BSY para atender uma mensagem DISCONNECT enviado pelo initiator.

**Hold time:** Tempo mínimo que existir entre a ativação do sinal REQ (ou ACK) e a mudança das linhas de dados, ou seja por quanto tempo o dado é válido após a ativação de REQ (ou ACK) durante transferências no modo síncrono.

**Negation Period:** O target e o initiator devem “negar” os sinais REQ e ACK respectivamente por pelo menos este tempo durante as transferências no modo síncrono.

**Power on to selection time:** Tempo máximo recomendado para que o sistema após ser ligado, possa responder com os dados e status adequados aos comandos TESTE UNIT READY, INQUIRY, e REQUEST SENSE.

**Reset to selection time:** Tempo máximo recomendado para que o sistema após uma condição de Hard Reset, possa responder com os dados e status adequados aos comandos TESTE UNIT READY, INQUIRY, e REQUEST SENSE.

**Reset to hold time:** Largura mínima do pulso de ativação do sinal RST.

**Selection abort time:** Tempo máximo, contado a partir da detecção de seleção mais recente, pelo qual um target ( ou initiator deve aguardar uma resposta (ativação do sinal BSY) do dispositivo a ser selecionado ou reselectionado.

**Selection time out delay:** O tempo mínimo que um dispositivo SCSI deve esperar pela resposta BSY durante uma fase de SELECTION ou RESELECTION antes de iniciar um procedimento de time out.

**Transfer period:** Tempo mínimo permitido entre as bordas de subida de pulsos de REQ ou ACK sucessivos, durante transferências no modo síncrono.

## 5.2.5 Fases do barramento SCSI

A comunicação sobre o barramento SCSI é controlada por uma seqüência de estados assumidos por este barramento, que na terminologia SCSI são chamados de fases. O barramento SCSI tem oito fases distintas:

1. BUS FREE
2. ARBITRATION
3. SELECTION
4. RESELECTION
5. COMMAND
6. DATA
7. STATUS
8. MESSAGE

As fases são controladas através dos sinais de controle definidos no item anterior, e o barramento SCSI não pode estar simultaneamente em mais de uma fase.

Os valores dos sinais SEL, BSY, e I/O e a ordem em que ocorrem as fases determinam quando se entra nas fases *BUS FREE*, *ARBITRATION*, *SELECTION* e *RESELECTION* (ver figura 5.11). A seqüência em que estas fases ocorrem está sujeita a algumas restrições de modo que:

Só é possível entrar-se nas fases *SELECTION* e *RESELECTION* a partir da fase *ARBITRATION*. Por sua vez, a entrada na fase *ARBITRATION* só pode ser feita a partir da fase *BUS FREE*. E finalmente, a entrada na fase *Bus Free* pode ser feita a partir de qualquer outra fase.

Portanto, para se determinar qual é a fase corrente, devemos saber a priori qual era a fase anterior, e quais são os valores correntes dos sinais de controle. O initiator e o target utilizam estes sinais para fazer o barramento SCSI mudar de uma fase para a outra.

A tabela 5.6 mostra os valores dos sinais SEL, BUSY e I/O durante as fases Bus Free, Arbitration, Selection e ReSelection. Conforme é ilustrado, quando os sinais SEL e BSY assumem ambos o valor "falso", isto indica que se está na fase Bus Free. A partir deste ponto, os sinais SEL, BSY, e I/O são usados para selecionar-se as outras fases. Um dispositivo SCSI disputa o controle do barramento SCSI, ativando o sinal BSY e o seu bit SCSI ID. Após um arbitration delay, o dispositivo examina o dado que se encontra no barramento. Se o seu bit SCSI ID é o de maior prioridade, ele vence a arbitragem, e ativa o sinal SEL. O barramento entrará na fase *SELECTION* ou *RESELECTION*, dependendo do valor assumido pelo sinal I/O. Se I/O for "verdadeiro" a próxima fase assumida pelo barramento será a fase *SELECTION*, em caso contrário segue-se para a fase *RESELECTION*.

**Tabela 5.6 : Fases e Sinais de Controle**

<i>FASE</i>	<i>SEL</i>	<i>BSY</i>	<i>I/O</i>
BUS FREE	0	0	0
ARBITRATION	0	1	0
SELECTION	1	X	0
RESELECTION	1	X	1

Obs.: 0 = falso; 1 = verdadeiro; X = irrelevante

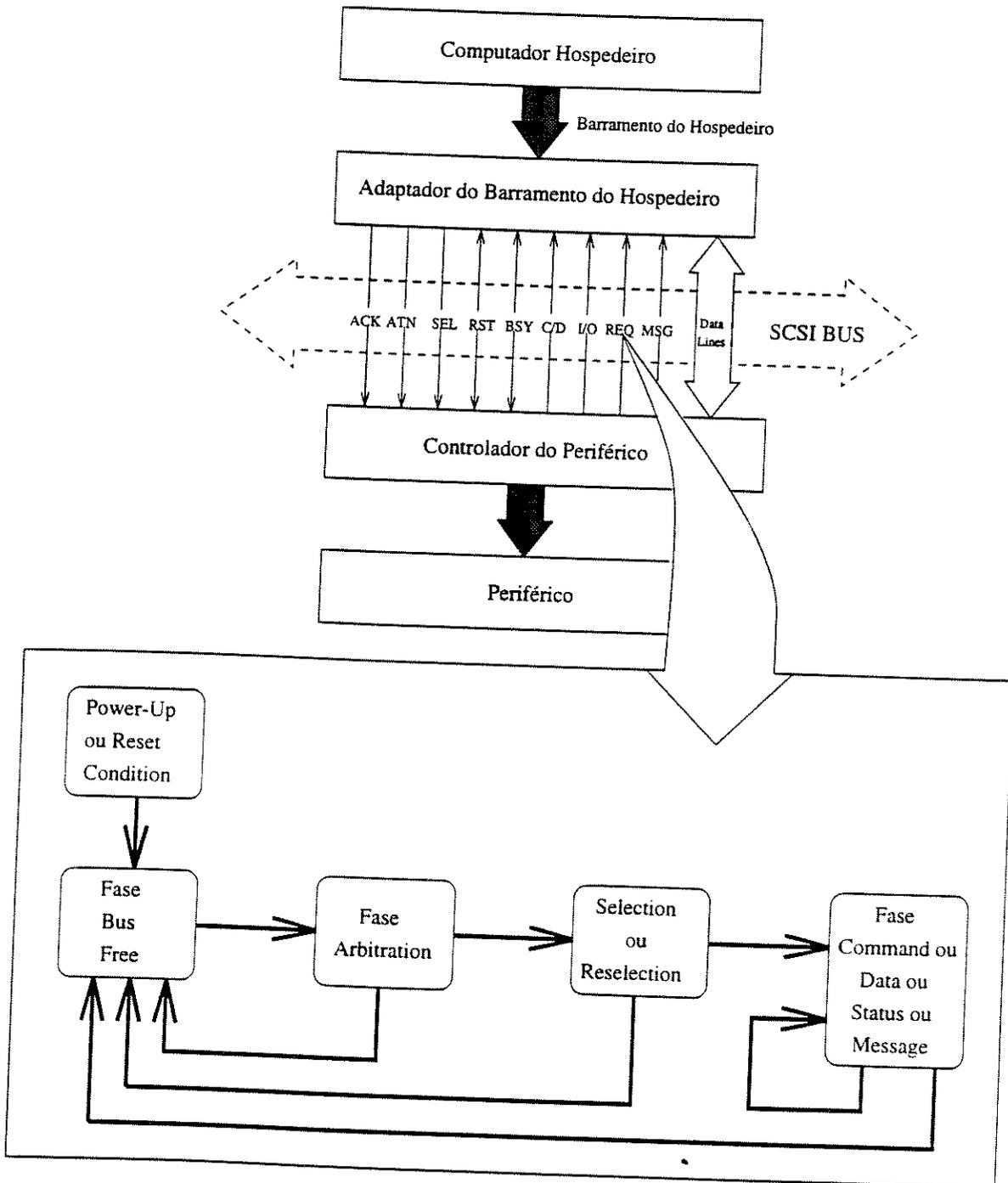


Figura 5.11 - Sequência de fases do barramento SCSI

A seguir detalharemos as tarefas realizadas durante cada uma das fases.

### **Fase Bus Free**

Esta fase indica que nenhum processo de E/S está ocorrendo sobre o barramento SCSI, e inicia-se quando os sinais SEL e BSY assumem ambos o valor “falso” por pelo menos 400ns (bus settle delay) . Os dispositivos desativam todos os sinais do barramento quando começa a fase *BUS FREE*.

### **Fase Arbitration**

A fase *Arbitration* permite a um dispositivo, competir pelo controle do barramento SCSI, a fim de iniciar ou retomar um processo de E/S. Quando o dispositivo SCSI deseja adquirir o controle do barramento ele realiza o seguinte procedimento:

1. O dispositivo SCSI espera a ocorrência da fase *BUS FREE*.
2. O dispositivo requisita o controle do barramento SCSI, ativando simultaneamente os sinais BSY e o sinal de dado DBx correspondente ao seu bit SCSI ID.
3. Passado um *Arbitration delay* , o dispositivo lê o conteúdo das linhas DB(7-0) a fim de verificar se ele é o dispositivo de maior prioridade a requerer o controle do barramento (lembrando que DB7 representa o dispositivo de maior prioridade). Se isto for verdade, ou seja se no byte lido a partir do barramento de dados não houver nenhum bit SCSI ID de maior prioridade ativado, o dispositivo vencerá a arbitragem e habilitará o sinal SEL. Caso contrário, o dispositivo desativa todos os sinais e retorna ao passo 1.
4. O dispositivo que venceu a arbitragem espera por um tempo igual a “um bus clear delay mais um bus settle delay” após a ativação do sinal SEL, para então alterar o valor de qualquer outro sinal.

### **Fase Selection**

Esta fase permite ao initiator selecionar um target a fim de iniciar a execução de uma função a ser realizada pelo target.

Como já mencionado anteriormente, o dispositivo SCSI que venceu a arbitragem mantém os sinais BSY e SEL ativados e espera por pelo menos “um bus clear delay mais um bus settle delay” até que se encerre a fase *ARBITRATION*. Este dispositivo SCSI torna-se então o initiator do processo de E/S se mantiver o sinal I/O com o valor “falso”.

Na fase *SELECTION*, o initiator coloca nas linhas DB(7-0) um byte no qual somente a linha correspondente ao seu bit ID e a do target a ser selecionado são verdadeiras. Ele ativa então o sinal ATN indicando que uma fase *MESSAGE OUT* se seguirá a fase *SELECTION*. Espera então por dois ou mais

deskew delays e desativa o sinal BUSY. O initiator aguarda pelo menos um bus settle delay antes de procurar pela resposta do target selecionado, ou seja ativação da linha BSY.

O target determina que foi selecionado, quando o sinal SEL e o seu bit ID forem ambos verdadeiros enquanto os sinais BSY e o I/O são falsos por pelo menos um bus settle delay. Detectada a seleção o target pode examinar o barramento de dados para verificar o bit SCSI ID do initiator que requereu a seleção. O target selecionado responde ao initiator, ativando o sinal BSY antes que expire o selection abort time, contado a partir da última detecção de seleção.

Após detectar a ativação do sinal BSY pelo target, o initiator desativa o sinal SEL e libera as linhas DB(7-0). O target ao perceber que o sinal SEL foi desativado, ativa a linha REQ para levar o barramento a uma fase de transferência de informação.

**Procedimentos de time-out** São especificados dois procedimentos opcionais de time-out para abortar o processo de seleção, caso o initiator não receba a resposta do target, após ter esperado por pelo menos um selection time-out delay:

- a) O initiator pode simplesmente ativar o sinal RST, ou
- b) O initiator pode manter os sinais SEL e ATN ativados, desativar as linhas DB(7-0), e aguardar pela resposta do target por mais um tempo igual a “um selection abort time mais dois skew time”. Se não obtiver novamente a resposta, desativa SEL e ATN e libera o barramento para a fase BUS FREE.

## Fase Reselection

Esta fase permite ao target reconectar-se ao initiator, a fim de continuar uma operação de E/S iniciada anteriormente por este mesmo initiator que foi suspensa pelo target<sup>11</sup>. Após vencer a arbitragem, o dispositivo tornar-se-á o target do processo de E/S corrente se ativar o sinal I/O.

Durante a fase RESELECTION, o target coloca nas linhas DB(7-0) um byte onde somente o seu bit SCSI ID e do initiator a ser reselectionado são verdadeiros. O target espera que se passem dois *deskew delays* e desativa o sinal BSY. Feito isto, o target aguarda por pelo menos um *bus settle delay* para então procurar pela resposta do initiator (ou seja uma ativação da linha BSY).

O initiator determina que está sendo reselectionado quando os sinais SEL, I/O e o seu bit SCSI ID são “verdadeiros”, enquanto o sinal BSY mantém-se “falso”, por pelo menos um *bus settle delay*. O initiator reselectionado pode então determinar o target que o está reselectionando, verificando o conteúdo dos sinais DB(7-0). O initiator responde ao target ativando o sinal BSY.

---

<sup>11</sup> Isto é o target desconnectou-se permitindo uma fase *BUS FREE*, antes que a operação E/S fosse totalmente completada

O target, ao detectar que o initiator ativou BSY, também ativa o sinal BSY e passados dois *deskew delays* desativa o sinal SEL. A partir deste instante, o target está livre para modificar o conteúdo das linhas DB(7-0) e do sinal I/O. O initiator ao perceber a desativação de SEL, desativa BSY. O target permanece com o sinal BSY ativado até completar o processo de E/S corrente.

**Procedimento de time-out** Existem duas opções de *time out* para levar o barramento SCSI à fase *BUS FREE*, caso o target não receba a resposta do initiator reselectionado antes que o *selection time out delay* se expire:

1. O target simplesmente ativa o sinal RST, ou
2. O target continua ativa o sinal SEL e I/O e desativa as linhas de dados. Ele espera por mais “um *selection abort time* mais dois *deskew delays*”, e se não receber resposta desativa os sinais SEL e I/O levando o barramento para a fase *BUS FREE*.

### Fases de transferência de informação

As fases COMMAND, DATA, STATUS, e MESSAGE são referenciadas como fases de transferência de informação e conforme mostra a tabela 5.7, são controladas pelos sinais MSG, C/D e I/O.

**Tabela 5.7 - Fases de transferência de informação e sinais de controle**

Fase	Direção	MSG	C/D	I/O
DATA OUT	Para o target	0	0	0
DATA IN	A partir do target	0	0	1
COMMAND	Para o target	0	1	0
STATUS	A partir do target	0	1	1
MESSAGE OUT	Para o target	1	1	0
MESSAGE IN	A partir do target	1	1	1

Obs.: 0 = falso; 1 = verdadeiro;  
Para estas fases, SEL é falso e BSY é verdadeiro

O target pelo fato de acionar os três sinais controla as mudanças de uma fase para outra, embora o initiator possa requisitar uma fase de MESSAGE OUT ativando o sinal ATN. Estas fases servem aos seguintes propósitos:

**Fase COMMAND** - Esta fase permite ao target requisitar uma informação de comando a partir do initiator.

**Fase DATA** - Refere-se as fases DATA IN e DATA OUT. A fase DATA IN permite ao target requisitar que dados sejam enviados para o initiator a partir do target. A fase DATA OUT permite ao target requisitar que dados sejam enviados do initiator para o target.

**Fase STATUS** - Esta fase permite ao target requisitar que informação de status seja enviada do target para o initiator.

**Fase MESSAGE** - Refere-se as fases MESSAGE IN e MESSAGE OUT. A fase MESSAGE IN permite ao target requisitar que uma mensagem seja enviada para o initiator a partir do target. A fase MESSAGE OUT permite ao target requisitar que uma mensagem seja enviada do initiator para o target. Uma mensagem é uma estrutura de dados que transporta informação para gerenciamento da conexão lógica do initiator com o target.

Durante as fases de transferência de informação os sinais de handshake REQ/ACK seguem um protocolo que rege a operação de transferência. Este protocolo é chamado, segundo a terminologia SCSI, de modo de transferência de informação. São definidos dois modos de transferência: o assíncrono e o síncrono.

**Modo de transferência assíncrono** O modo assíncrono é o modo de transferência assumido “por default” e todos os comandos, mensagens, e informação de status são transferidas neste modo. A figura 5.12 ilustra o comportamento dos sinais SCSI em uma transferência no modo assíncrono. Em todas as fases de transferência de informação, o target controla a direção da transferência por meio do sinal I/O. Quando o sinal for verdadeiro, a informação é transferida do target para o initiator. Quando o sinal I/O for falso, a informação é transferida do target para o initiator. Em uma transferência para o initiator, o target seta as linhas DB(7-0,P) com os valores desejados, espera pelo menos “um *deskew delay* mais um *cable skew delay*”, e ativa o sinal REQ. Os sinais de dados DB(7-0,P) permanecem válidos até que o initiator ative o sinal ACK. Quando o target perceber a ativação de ACK, ele pode mudar o conteúdo das linhas de dados e desativar o sinal REQ. Ao perceber que REQ foi desativado, o initiator desativa o sinal ACK, e o target pode então reiniciar o ciclo. Em uma transferência para o target, o target inicialmente ativa o sinal REQ, requisitando a informação. Em resposta, o initiator coloca o dado desejado nas linhas DB(7-0,P) do barramento, e após um tempo igual a “um *deskew delay* mais um *cable skew delay*” ativa o sinal ACK. O initiator manterá o dado no barramento enquanto o sinal REQ mantiver-se verdadeiro. Ao perceber que o initiator ativou o sinal ACK, o target lê o dado nas linhas DB(7-0), e desativa o sinal REQ. Ao perceber esta desativação o initiator pode mudar ou liberar as linhas de dados, e desativa o sinal ACK. Ao perceber que o sinal ACK foi desativado o target pode continuar a transferência ativando novamente o sinal REQ, requisitando nova informação.

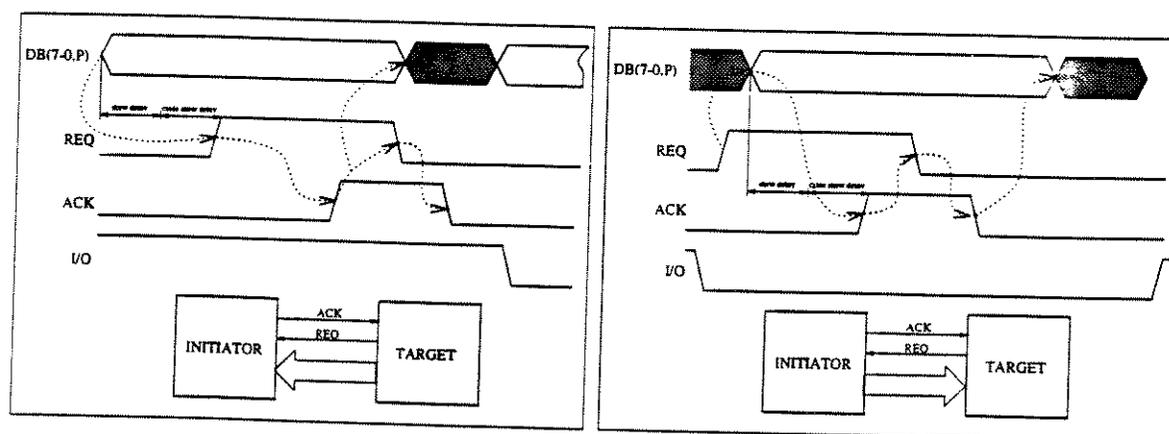


Figura 5.12 - Temporização de uma transferência no modo assíncrono

**Modo de transferência síncrono** A transferência no modo síncrono é opcional e pode ser utilizada somente na fase DATA. Um target que suporte o modo síncrono pode negociar com o initiator para que a transferência de dados seja feita sincronamente através de um protocolo de mensagens. Para começar a requisição pelo modo síncrono, o target envia ao initiator uma mensagem *Synchronous Data Transfer Request* contendo o seu *transfer period* e o seu *offset REQ/ACK*. O initiator pode responder com uma mensagem *Reject* (caso ele não suporte o modo síncrono), ou responder com uma nova mensagem *Synchronous Data Transfer Request* através da qual informará seu *transfer period* e seu *offset REQ/ACK*. O modo síncrono estabelecer-se-á com o maior dos *transfer periods* e com o menor dos *offset REQ/ACK*. Por *offset REQ/ACK* entenda-se o número máximo de pulsos de REQ, que podem ser enviados pelo target em avanço ao número de pulsos de ACK recebidos do initiator. Se durante uma transferência o número de REQ exceder o número de pulsos ACK de um *offset REQ/ACK*, o target não ativará o sinal REQ, até que a borda de subida do próximo pulso ACK seja recebida. Para que uma transferência síncrona seja completada com sucesso é necessário que o número de pulsos REQ enviados pelo target seja igual ao número de pulsos ACK recebidos. Em uma transferência do target para o initiator, o target primeiramente coloca nas linhas DB(7-0), o dado desejado. Passados “um *deskew delay* mais um *cable skew delay*” o target ativa o sinal REQ. Os sinais de dados DB(7-0,P) são mantidos estáveis por pelo menos “um *deskew delay* mais um *cable skew delay* mais um *hold time*”, contados a partir da borda de subida do sinal REQ. O target mantém REQ ativado por pelo menos um *assertion period*, e passado este tempo ele poderá desativar o sinal REQ e alterar as linhas de dados. O initiator por sua vez, deverá ler as linhas DB(7-0,P) respeitando o *hold time*, e deverá responder ao target com um pulso de ACK. Em uma transferência para o target, o initiator após perceber a borda de subida do pulso REQ, coloca o dado desejado nas linhas DB(7-0,P). Passados então “um *deskew delay* mais um *cable skew delay*”, o initiator ativa o sinal ACK. As linhas DB(7-0) permanecerão estáveis por pelo menos “um *deskew delay* mais um *cable skew delay* mais um *hold time*”, contados a partir da borda de subida do pulso de ativação de ACK. O sinal ACK deve ser mantido ativado por pelo menos um *assertion period*, e a partir disso as linhas de dados poderão ser alteradas. O target deverá ler as linhas DB(7-0,P) respeitando o *hold time* contado a partir da transição de ACK para verdadeiro.

**Transferência de dados no modo wide** O modo wide é opcional e pode ser usado somente na fase DATA. São usadas mensagens para se determinar o uso modo wide pelos dispositivos SCSI, e para determinar a largura do caminho de dados durante a fase DATA. No modo wide podem ser estabelecidas transferências de 16 ou 32 bits, e Dispositivos que suportam transferências de 32 bits também suportam as transferências de 16 bits. Além disso todos os dispositivos devem suportar transferências de 8 bits. Durante as transferências de 16 bits, o primeiro byte lógico de dados para cada fase DATA será transferido ao longo dos sinais DB(7-0,P) do cabo A e o segundo byte lógico será transferido nos sinais DB(15-8,P1) do cabo B. Nas transferências de 32 bits, o primeiro byte lógico de dados para cada fase DATA será transferido ao longo dos sinais DB(7-0,P) do cabo A e o segundo, terceiro e quarto bytes lógicos de dados serão transferido nos sinais DB(15-8,P1), DB(23-16, P2), e DB(31-24,P3) pertencentes ao cabo B, respectivamente. Para assegurar que a integridade dos dados, certas requisitos sobre os sinais REQ/ACK no cabo A e REQ/ACKB no cabo B devem ser atendidos:

- a) O mesmo modo de transferência deve ser usado para cabos A e B.
- b) Os procedimentos descritos para os modos assíncrono e síncrono para o cabo A (sinais REQ, ACK e DB(7-0,P)), também são válidos para o cabo B (sinais REQ, ACK e DB(31-8,P1,P2,P3)).
- c) O target deve assegurar que o número de handshakes REQ/ACK e REQB/ACKB sejam iguais antes de mudar o barramento para outra fase.

## 5.2.6 Condições do barramento SCSI

O barramento SCSI tem duas condições: Attention e Reset. Estas condições podem alterar a sequência de fases do barramento, e são meios que o initiator possui para forçar o target a alterar a execução de um processo de E/S.

### Condição Attention

Através desta condição o initiator informa que tem uma mensagem para o target. O initiator cria esta condição ao ativar o sinal Attention (ATN) durante qualquer fase com exceção das fases Arbitragem e *Bus Free*. O target responde com uma fase MESSAGE OUT como se segue:

- a) Se ATN tornar-se verdadeiro durante a fase COMMAND, o target entrará na fase MESSAGE OUT depois de transferir parte ou todos os bytes do bloco descritor de comando<sup>12</sup>.
- b) Se ATN tornar-se verdadeiro durante a fase DATA, o target entrará na fase MESSAGE OUT assim que lhe for mais conveniente ( freqüentemente na fronteira de um logical block). O initiator deve continuar com os handshakes REQ/ACK até detectar a mudança de fase.
- c) Se ATN torna-se verdadeiro durante uma fase STATUS, o target entrará na fase MESSAGE OUT depois que o initiator reconhecer o byte de status.
- d) Se ATN torna-se verdadeiro durante uma fase MESSAGE IN, o target vai para a fase MESSAGE OUT antes de enviar a sua mensagem.
- e) Se ATN torna-se verdadeiro durante a fase SELECTION antes do initiator desativar o sinal BSY, o target entra na fase MESSAGE OUT imediatamente após a fase SELECTION.
- f) Se ATN torna-se verdadeiro durante a fase RESELECTION, o target entrará na fase MESSAGE OUT após enviar a mensagem IDENTIFY.

### Condição Reset

A condição de Reset pode ser gerada por qualquer dispositivo SCSI, ativando-se o sinal RST, e leva o barramento para a fase *BUS FREE*.

---

<sup>12</sup>O bloco descritor de comando será descrito no item 6.2.9.

O efeito que a condição de Reset tem sobre os processos de E/S não terminados, reservas, e modos de operação dos dispositivos SCSI é determinado pela opção utilizada na implementação da condição Reset. Há duas opções para a implementação:

**Opção Hard Reset:** Um dispositivo que implementa a opção Hard Reset executa o seguinte procedimento quando detecta uma condição de Reset:

- Aborta todos os processos de E/S incompletos inclusive os que estão na fila.
- Cancela todas as reservas de dispositivos.
- Retornar todas os modos de operação dos dispositivos SCSI para as condições default.

**Opção Soft Reset:** Um dispositivo que implementa esta opção segue o procedimento:

- Tenta completar os processos de E/S incompletos que foram completamente identificados.
- Preserva todas as reservas de dispositivos
- Preserva todos os modos de operação.
- Preserva as informações necessárias para retomar a execução dos processos que estavam na fila antes da condição Reset.

### 5.2.7 Ponteiros SCSI

A arquitetura SCSI define dois conjuntos de ponteiros dentro do initiator . Conhecidos como ponteiros ativos e ponteiros salvos, cada um destes conjuntos é formado por três ponteiros:

- um ponteiro de comando,
- um ponteiro de dado e
- um ponteiro de status.

**Ponteiros Ativos** Os ponteiros ativos apontam para o próximo byte de comando, dado, e status a ser transferido entre a memória do initiator e o target. Há somente um conjunto de ponteiros ativos para cada initiator.

**Ponteiros Salvos** Há um conjunto de ponteiro salvos para cada processo de E/S. Um ponteiro de comando salvo aponta para o início do Bloco Descritor de Comando de seu processo de E/S. Um ponteiro de status salvo aponta para o início da área de status de seu processo de E/S.

No início de cada processo de E/S o ponteiro de dado salvo aponta para o início da área de dados. Ele permanece com este valor até que o target envie uma mensagem Save Data Pointer. Ao receber esta mensagem o initiator salva armazena o valor do ponteiro de dado ativo no ponteiro de dado salvo.

O target pode restaurar os ponteiros ativos com os valores armazenados nos ponteiros salvos enviando ao initiator uma mensagem Restore Pointers.

## 5.2.8 Sistema de mensagens

O sistema de mensagens permite a comunicação entre um iniciator e um target com o propósito de gerenciamento da ligação lógica existente entre eles. As mensagens são enviadas durante a fase Message e podem ter um, dois ou múltiplos bytes de comprimento. Durante uma fase Message, podem ser enviadas um ou mais mensagens, mas o envio de uma mensagem de múltiplos bytes não possa ser quebrado em duas ou mais fases Message. Conforme mostra a tabela 5.8, através do primeiro byte da mensagem pode-se determinar qual é o seu comprimento em bytes.

**Tabela 5.8 - Comprimento das mensagens**

<b>Primeiro byte</b>	<b>Formato da Mensagem</b>
00h	mensagem de um-byte (COMMAND COMPLETE)
01h	mensagens extendidas
02h - 1Fh	mensagens de um-byte
20h - 2Fh	mensagens de dois-bytes
30h - 7Fh	Reservado
80h - FFh	Mensagem de um-byte (IDENTIFY)

As mensagens suportadas pela SCSI são listadas na tabela 5.9. O primeiro byte da mensagem identifica também qual é a mensagem a ser executada. A tabela 5.9 indica ainda, o sentido em que se dá o envio da mensagem e quais devem ser obrigatoriamente suportadas.

**Tabela 5.9 - Mensagens da SCSI**

<b>Código do Primeiro Byte</b>	<b>Suporte</b>		<b>Mensagem</b>	<b>Direção</b>
	<b>Initiator</b>	<b>Target</b>		
06h	opt.	obr.	ABORT	Out
0Dh	opt.	opt.	ABORT TAG	Out
0Ch	opt.	obr.	BUS RESET DEVICE	Out
0Eh	opt.	opt.	CLEAR QUEUE	Out
00h	obr.	obr.	COMMAND COMPLETE	In
04h	opt.	opt.	DISCONNECT	In ou Out
80h - 7Fh	obr.	obr.	IDENTIFY	In ou Out
23h	opt.	opt.	IGNORE WIDE RESIDUE	In
0Fh	opt.	opt.	INITIATE RECOVERY	In ou Out
05h	obr.	obr.	INITIATOR DETECTED ERROR	Out
0Ah	opt.	opt.	LINKED COMMAND COMPLETE	In
0Bh	opt.	opt.	LINKED COMMAND COMPLETE (WITH FLAG)	In
09h	obr.	obr.	MESSAGE PARITY ERROR	Out
***	opt.	opt.	MODIFY DATA POINTER	In
08h	obr.	obr.	NO OPERATION	Out
21h	opt.	opt.	HEAD OF QUEUE TAG	Out
22h	opt.	opt.	ORDERED QUEUE TAG	Out
20h	opt.	opt.	SIMPLE QUEUE TAG	Out
10h	opt.	opt.	RELEASE RECOVERY	Out
03h	opt.	opt.	RESTORE POINTER	In
02h	opt.	opt.	SAVE DATA POINTER	In
***	opt.	opt.	SYNCHRONOUS DATA TRANSFER REQUEST	In ou Out
11h	opt.	opt.	TERMINATE I/O PROCESS	Out
***	opt.	opt.	WIDE DATA TRANSFER REQUEST	In ou Out

OBS.: In = Target to Initiator, Out = Initiator to Target.

opt. = opcional

obr. = obrigatório

\*\*\* = mensagens extendidas

Bit \ Byte	7	6	5	4	3	2	1	0
0	Extended message (01h)							
1	Extended message (n)							
2	Extended message (y)							
3	----- Extended message arguments -----							
n+1								

Figura 5.13 - Formato de uma mensagem estendida

O valor *01h* no primeiro byte de uma mensagem indica o começo de uma mensagem estendida de múltiplos bytes. O número mínimo de bytes a ser enviada a uma mensagem neste formato é três. O formato e os códigos das mensagens estendidas são mostrados na figura 5.13 e na tabela 5.10.

Tabela 5.10 - Códigos das mensagens estendidas

Código (y)	Mensagem
00h	MODIFY DATA POINTER
01h	SYNCHRONOUS DATA TRANSFER REQUEST
03H	WIDE DATA TRANSFER REQUEST

A primeira mensagem a ser enviada pelo initiator (ou pelo target) após a fase SELECTION (ou RESELECTION) será uma mensagem IDENTIFY, ABORT, ou BUS RESET DEVICE. Se o target (ou initiator) receber qualquer outra mensagem ele levará o barramento para a fase *BUS FREE*.

Quando a primeira mensagem enviada for IDENTIFY, ela pode ser seguida por outras mensagens, tais como um par de mensagens SYNCHRONOUS DATA TRANSFER REQUEST para negociar a mudança do modo de transferência para a opção síncrona. Como veremos mais adiante a mensagem IDENTIFY estabelece uma conexão entre o initiator e uma logical unit ou uma target routine de um dado target.

A seguir tem-se uma descrição de algumas das mensagens listadas na tabela 5.9. As mensagens omitidas referem-se a características da SCSI que não serão abordados em nossa discussão, por uma questão de escopo de nosso trabalho.

**Abort:** O initiator envia esta mensagem ao target para abortar a operação corrente. Se um número de unidade lógica tiver sido identificado, todos os dados e informação de status pendentes são abortados e o target leva o barramento para a fase *Bus Free*. Se uma unidade lógica não tiver sido identificada quando

do recebimento desta mensagem, o target simplesmente leva o barramento para a fase *Bus Free*. Nesta operação não é transferida nenhuma mensagem de término ou de status.

**Bus Device Reset:** O initiator envia esta mensagem para um target para cancelar todos os processos de E/S (comandos) associados a este dispositivo SCSI. Esta mensagem força o target para um estado inicial sem operações pendentes com qualquer initiator. O target então, leva o barramento SCSI para a fase *Bus Free*.

**Command Complete:** O target envia esta mensagem ao initiator para indicar que a execução de um comando terminou e que enviou uma informação de status válida ao initiator.

**Disconnect:** O target envia a mensagem DISCONNECT ao initiator para indicar que a ligação física existente entre eles será quebrada e que uma reconexão será requisitada mais tarde para que a operação corrente seja completada. A pós enviar esta mensagem, target desconecta-se do initiator, negando a linha BSY (Busy).

**Identify:** O initiator ou o target enviam estas mensagens para estabelecer uma ligação lógica entre um initiator e uma logical unit ou target routine. O formato da mensagem Identify é mostrado na tabela 5.11.

**Tabela 5.11 - Formato para a mensagem Identify**

Bit	7	6	5	4	3	2	1	0
	Identify	DiscPriv	LUN/TAR	Reservado	Reservado	LUNTRN		

O bit 7 ao ser setado para “1” identifica mensagem Identify. O bit 6 é setado para “1” pelo initiator quando ele quer indicar ao target que suporta desconexão e reselection. Um valor “0” para o bit 6 especifica que o target não suportará desconexão. Se o bit 5 for igual a “1”, isto indica que o processo de E/S está direcionado a uma logical unit do target. Caso o bit 5 seja “0”, o direcionamento se fará para uma target routine. Os bits 2, 1 e 0 contem o número da logical unit (LUN) quando o bit 5 é “1”, ou número da target routine quando o bit 5 é “0”.

**Ignore Wide Residue:** Esta mensagem deve ser enviada pelo target para indicar que o número de bytes válidos enviados no último handshake REQ/ACK - REQB/ACKB é menor que a largura de transferência negociada anteriormente.

**Initiator Detected Error:** O initiator envia esta mensagem para informar ao target que ocorreu um erro que requer que o target repita a operação. Embora a integridade dos ponteiros correntes não possa ser garantida uma mensagem Restore Pointers ou uma mensagem Disconnect seguida de reconexão, levará os ponteiros ao estado anterior.

**Linked Command Complete:** Esta mensagem é enviada pelo target ao initiator para indicar que a execução de um comando *linked* terminou e que um byte status foi enviado ao initiator. O initiator setará os ponteiros para o início do próximo comando *linked*.

**Linked Command Complete with Flag:** Esta mensagem é enviada pelo target ao initiator para indicar que a execução de um comando *linked with flag* terminou e que um byte status foi enviado ao initiator. O initiator setará os ponteiros para o início do próximo comando *linked*. Esta mensagem é tipicamente utilizada para causar uma interrupção no initiator entre dois comandos ligados.

**Parity Error:** O initiator envia esta mensagem para indicar que um ou mais bytes na última mensagem recebida contem um erro de paridade.

**Message Reject:** O initiator ou o target envia esta mensagem para indicar que a última recebida foi inapropriada ou não foi implementada.

**Modify Data Pointer:** Esta mensagem é enviada pelo target para requisitar ao initiator que o argumento nela enviado, seja somado (em complemento de 2) ao conteúdo do ponteiro de dados ativo.

**No Operation:** O initiator envia esta mensagem ao target quando o target está requisitando uma mensagem e o initiator não qualquer outra mensagem válida para enviar.

**Restore Pointers:** Ao receber esta mensagem do target, o initiator copia o conteúdo do conjunto de ponteiros salvos ( de dados, de comando, e de status) mais recente, para os ponteiros ativos correspondentes.

**Save Data Pointer:** Ao receber esta mensagem do target, o initiator copia o conteúdo do ponteiro de dados ativo para o ponteiro de dados salvo do processo de E/S correspondente. O target envia esta mensagem antes de transmitir uma mensagem Disconnect.

**Synchronous Data Transfer Request:** O formato desta mensagem é mostrado na tabela 5.12 a seguir:

Tabela 5.12 - Formato para a mensagem SDTR

Bit	7	6	5	4	3	2	1	0
Byte								
0	(01h) mensagem extendida							
1	(03h) comprimento da mensagem extendida							
2	(01h) identificação da mensagem SDTR							
3	Transfer period factor							
4	REQ/ACK offset							

Uma troca de mensagens (SDTR) entre um initiator e um target é estabelecida sempre que se deseja mudar o modo de transferência de dados para a opção síncrona. Esta troca de mensagens SDTR estabelece o transfer period (o maior deles) e REQ/ACK offset (o menor deles) que serão utilizados nas transferências de dados síncronas realizadas entre os dois dispositivos. O transfer period de um dispositivo

é tempo mínimo permitido entre as bordas de subida dos pulsos sucessivos de REQ (ou ACK) enviados durante uma transmissão. O transfer period é igual ao transfer period factor vezes 4.

O REQ/ACK offset de um dispositivo é o número máximo de pulsos de REQ que podem ser enviados pelo target antes que a borda de subida do pulso de ACK correspondente seja recebida.

**Wide Data Transfer Request:** O formato da mensagem WDTR é mostrada na tabela 5.13 a seguir:

**Tabela 5.13 - Formato para a mensagem WDTR**

Bit	7	6	5	4	3	2	1	0
Byte								
0	(01h) mensagem estendida							
1	(02h) comprimento da mensagem estendida							
2	(03h) identificação da mensagem WDTR							
3	(m) Transfer width exponent							

Uma troca de mensagens (WDTR) entre um initiator e um target estabelece um acordo entre dois dispositivos SCSI a respeito da largura do barramento de dados a ser usada nas transferências realizadas durante a fase DATA. A largura da transferência (transfer width) de um dispositivo é indicada através do byte transfer width exponent enviado na mensagem WDTR, sendo de 8 bits para  $m = 00h$ , de 16 bits para  $m = 01h$ , e de 32 bits para  $m = 02h$ . O dispositivo que envia a primeira mensagem WDTR seta sua largura de transferência para a máxima largura que é capaz de acomodar. Se o outro dispositivo suportar esta largura, ele retorna o mesmo valor na sua mensagem WDTR, caso contrário ele retorna um valor menor (referente a maior largura suportada pelo segundo dispositivo). A largura retornada pelo segundo dispositivo será a adotada durante as transferências de dados.

### 5.2.8 Comandos da SCSI

Comandos são os meios pelos quais um dispositivo initiator inicia a execução de uma operação realizada pelo dispositivo target.

Um initiator comunica um comando ao target enviando-lhe uma estrutura de dados conhecida como Bloco Descritor de Comando (BDC), durante a fase COMMAND. O target decodifica este BDC e executa a(s) tarefa(s) nele especificada(s). Completada(s) a(s) tarefa(s) o target deve retornar uma informação de status ao initiator, informando-lhe como foi a execução do comando, ou seja se o comando foi completado com ou sem sucesso. Um Bloco Descritor de Comando contém basicamente um código de identificação de comando e informações suplementares requeridas para a execução do comando. A figura 5.14 mostra os formatos dos BDC dos comandos dos grupos 0, 1 e 2.

Bit	7	6	5	4	3	2	1	0
0	Identificador do comando							
1	Logical Unit Number (MSB)							
2	Logical block address(se requerido)							
3	(LSB)							
4	Transfer length (se requerido)							
	Parameter list length (se requerido)							
	Allocation length (se requerido)							
5	Byte de Controle							

Bit	7	6	5	4	3	2	1	0
0	Identificador do comando							
1	Logical Unit Number					Reserved		
2	(MSB)							
3	Logical block address(se requerido)							
4	(LSB)							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Byte de Controle							

Bit	7	6	5	4	3	2	1	0
0	Identificador do comando							
1	Logical Unit Number					Reserved		
2	(MSB)							
3	Logical block address(se requerido)							
4	(LSB)							
5	Reserved							
6	Reserved							
7	Transfer length (se requerido)							
8	Parameter list length (se requerido)							
9	Allocation length (se requerido)							
10	Reserved							
11	Byte de Controle							

Figura 5.14 - Blocos Descritores de Comando para os grupos 0, 1, e 2

**Identificador de comando** O primeiro byte de um BCD é chamado de “identificador de comando”. Conforme mostra a tabela 5.14 a seguir, este byte possui um campo de 3 bits para indicar qual é o grupo do comando (ou seja grupo 0, grupo 1 ou grupo 2) , e portanto qual o formato de seu BCD, e um campo de 5 bits contendo o código do comando propriamente dito.

Tabela 5.14 - Byte identificador de comando

Bit	7	6	5	4	3	2	1	0
Byte	Código do grupo			Código do comando				

**Logical Unit Number:** O logical unit number, conforme já mencionado pode ser definido através da mensagem IDENTIFY . O target deve ignorar o logical unit number especificado no BDC se já recebeu uma mensagem IDENTIFY. Nestes casos é recomendado, que este campo do BDC seja preenchido com zero.

**Logical Block Address:** Os endereços dos logical blocks de uma logical unit (como por exemplo, um periférico) começam com logical block “zero” e são contiguos até o último logical block da logical unit em

questão. Como pode ser observado na figura 5.14, o BDC de um comando do grupo 0 (6 bytes) contem um endereço de logical block de 21 bits, enquanto os BDC's de comandos do grupo 1 (10 bytes) e do grupo 2 (12 bytes) possuem um campo de endereço de logical block de 32 bits.

**Transfer length:** O campo transfer length especifica a quantidade de dados a ser transferida e usualmente refere-se ao número de blocos.

**Parameter list length:** Este campo especifica o número de bytes a serem enviados durante a fase DATA OUT (do initiator para o target). Este campo é utilizado para especificar o comprimento em bytes de uma lista de parâmetros enviada para o target.

**Allocation length:** Este campo informa ao target o número de bytes que o initiator alocou para os dados que lhe serão retornados.

**Byte de controle:** O último byte de um BDC é o chamado byte de controle. A tabela 5.15 mostra a especificação deste campo do BDC.

Tabela 5.15 - Byte de controle

<b>Bit</b>	7	6	5	4	3	2	1	0
<b>Byte</b>								
	Vendor-specific		Reservado				Flag	Link

O bit Link é usado em processos de E/S formados por vários comandos, os chamados Linked Commands, cuja implementação é especificada como opcional. Um bit Link com valor “um”, indica que o initiator requer a continuação do processo de E/S e que o target deve ir para a fase COMMAND após completar o comando corrente.

O bit Flag, quando o bit Link tiver o valor “um”, especifica qual é a mensagem que o target deve retornar ao initiator quando completar com sucesso a execução de um dos comando ligados do processo de E/S. Se o bit Flag for “zero” o target retornará a mensagem Linked Command Complete. Se o bit Flag for “um”, o target retornará a mensagem Linked Command Complete (With Flag). Quando o bit Link tiver o valor “zero”, o bit Flag deve ser zero.

### 5.2.9 Exemplo de execução de um comando SCSI

A execução de um comando pelo target envolve os seguintes passos:

1. Aquisição e decodificação da informação de comando,
2. Transferência de dados e
3. Geração e retorno da informação de estado.

Um dos comandos SCSI mais comuns é o comando READ. A figura 5.15 ilustra o comportamento dos sinais do barramento SCSI durante a execução de um comando READ. As operações realizadas na execução deste comando ao longo das diversas fases são comentadas a seguir:

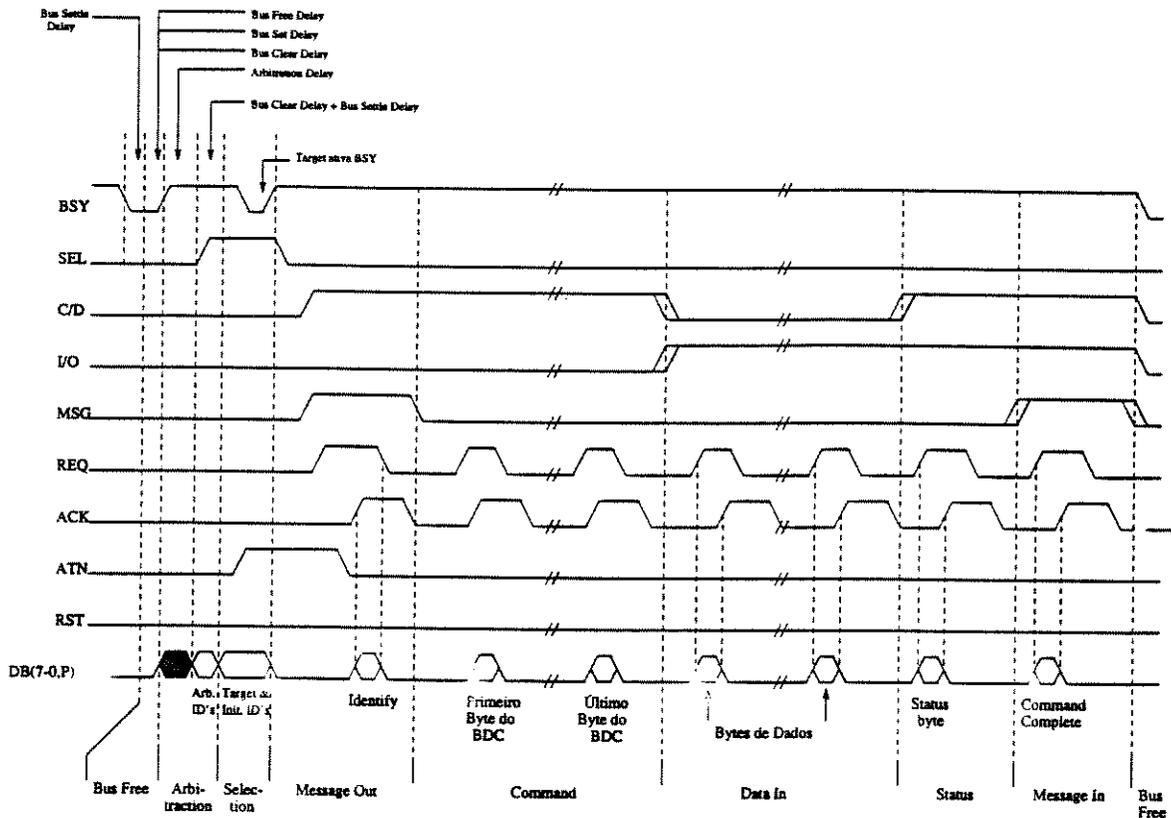


Figura 5.15 - Exemplo de execução de um comando SCSI

1. **Bus Free** O initiator espera pela fase *Bus Free*. Esta fase é reconhecida quando os sinais SEL e BSY têm o valor falso por pelo menos um *bus settle delay*.
2. **Arbitration** O initiator requisita então o controle do barramento SCSI, ativando os sinais BSY e a linha DB(x) correspondente ao seu bit ID. Após um *Arbitration delay*, o initiator examina as linhas de dados do barramento. Se nenhum bit ID de maior prioridade estiver ativado, o initiator vence a arbitragem e leva o barramento a fase Selection, ativando o sinal SEL.
3. **Selection** Na fase Selection, o initiator ativa as linhas de dados com os IDs correspondentes ao target a ser selecionado e ao próprio initiator. Após um pequeno atraso, o initiator nega o sinal BSY. Quando o target detecta esta configuração de sinais, ou seja SEL verdadeiro, BSY falso, I/O falso e o seu bit ID ativado, ele ativa o sinal BSY e armazena o bit ID do initiator em um registrador interno. O initiator ao detectar que BSY foi ativado, libera o barramento de dados e desativa o sinal SEL.
4. **Message Out** Ainda durante a fase Selection, o initiator pode ativar o sinal ATN para informar ao target, informando que deseja enviar-lhe uma mensagem. O target atendendo a esta condição pode

levar o barramento a fase Message Out e receber uma mensagem Identify do initiator. Esta mensagem informa o target a respeito de qual de suas lógicas ou target routine será acessada. é neste ponto que se estabelece a ligação lógica “initiator/logical unit ou target routine”, e um conjunto de ponteiros do initiator é associado ao processo de E/S. No caso do comando READ a ligação é estabelecida com uma logical unit.

5. **Command** Ao entrarmos na fase COMMAND, o target ativa o sinal REQ. Em resposta, o initiator procura o primeiro byte do BCD armazenado em sua memória no endereço especificado pelo seu ponteiro de comando ativo. Ao encontrá-lo, o initiator coloca o primeiro byte do BCD no barramento de dados (DB7-0,P), ativa o sinal ACK e incrementa o ponteiro de comando ativo. Os bytes seguintes do BCD são transferidos de maneira idêntica ao primeiro byte. Completada a transferência de todo o BCD, o target interpreta-o e identifica o comando descrito pelo BCD como sendo um READ. Procura então o primeiro bloco de dados na lógica unit que foi especificada, e ao localizar os dados, torna o sinal C/D falso e o sinal I/O verdadeiro, levando o barramento para a fase DATA IN.
6. **Data In** Esta fase é mantida enquanto os dados requisitados pelo initiator estiverem sendo transferidos. O target coloca o primeiro byte do primeiro bloco de dados especificado no BCD no barramento de dados e ativa o sinal REQ. Em resposta, o initiator aceita o byte de dados colocado no barramento, transfere-o para o endereço de memória especificado pelo ponteiro de dados ativo, e ativa o sinal ACK. Quando o target detecta que o sinal de ACK foi desativado, ele desativa o sinal REQ e coloca o segundo byte de dados no barramento. O initiator ao detectar a desativação de REQ, incrementa seu ponteiro de dados e desativa o sinal ACK. O target então ativa o sinal REQ e inicia o processo de transferência do segundo byte de dados para a memória do initiator. Este processo continua até que todos os bytes dos blocos de dados especificados no BCD tenham transferidos.
7. **Status** Após transferir todos dados requisitados pelo initiator, o target entra na fase Status. Nesta fase o target seguindo o mesmo handshake REQ/ACK da fase Data In envia um byte de status ao initiator indicando que não a transferência ocorreu sem problemas.
8. **Message Out** O target, após transferir o byte de status, entra na fase Message In. Ele coloca uma mensagem Command Complete no barramento de dados e ativa o sinal REQ. O initiator aceita esta mensagem, e ativa ACK. O target então desativa REQ, liberando o barramento de dados. O initiator em resposta desativa ACK. Ao perceber esta desativação o target desativa o sinal BSY, levando o barramento para a fase BUS FREE.

### 5.3 Considerações a cerca da implementação da interface SCSI

A interface SCSI pode ser implementada através de circuitos de interface paralela como o chip PIT 68230 (ver apêndice A), ou através de chips controladores SCSI como o Am53c94 (ver apêndice B). Uma opção

mais atraente, entretanto, é a implementação da SCSI através da aquisição módulos VME prontos como o MVME162.

### 5.3.1 Módulo MVME162 - Controlador Master VME com Interface SCSI

O MVME162 é baseado no microprocessador MC68040 ou MC68LC040. As diversas versões do MVME162 possuem de 1 a 4 MB de DRAM com proteção de paridade, 8 KB de SRAM, interface para Ethernet (transceiver), 2 portas seriais com interface EIA-232-D ou EIA-530, 6 temporizadores (tick), 1 soquete para PROM, 1MB de memória Flash, interface para SCSI Bus com DMA, controlador de VMEbus e 512 KB de SRAM. O I/O do MVME162 é conectado ao conector P2 do VMEbus. A notação é a convencional para as palavras de dados : Byte (8 bits , de 0 a 7 sendo o 0 o menos significativo); Word (16 bits , de 0 a 15) e Longword ( 32 bits, de 0 a 31).

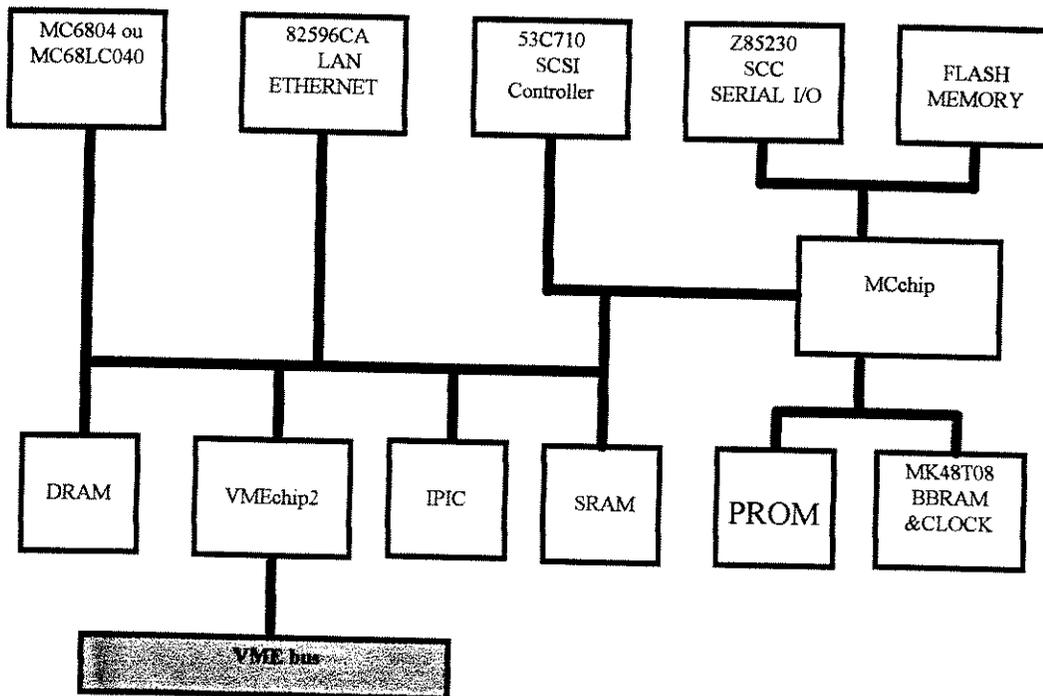


Figura 5.16 - Diagrama de Blocos do MVME162

- Estrutura dos dados de barramento : possui 32 bits (síncrono).
- Processador : MC68040 (este possui co-processador de ponto flutuante) ou MC68LC040.
- Opção sem Interface VMEbus : na presença do VMEchip2 pode-se inibir a interface.
- Opções de memória :

DRAM ( 1 MB não-entrelaçada a 4 MB entrelaçada)

SRAM (512 KB não-entrelaçada ); proteção de bateria : 200 dias.

PROM e FLASH (1 MB de memória Flash e 512KB x 8 JEDEC)

- Interface VMEbus e VMEchip2 : o VMEchip2 provê a interface entre o barramento local e o VMEbus.
- Interfaces I/O : inclui portas seriais, Ethernet e SCSI.
- Interface de Comunicação Serial : possui 2 interfaces seriais ( conector DB-25).
- Interface IP ( Industry Pack) : possui 4 Ips.
- Interface Ethernet : tem capacidade de DMA.
- Interface SCSI : Os subsistemas podem incluir drivers para disco rígido ou flexível.(Terminação SCSI : adaptador P2)

## 5.4 Bibliografia

- [1] *B. Beims & P. LaViolette, "Standart SCSI bus facilities peripheral control", EDN, April 4, 1985.*
- [2] *B. Beims & P. LaViolette, "SCSI protocol and controller ease bus arbitration", EDN, March 21, 1985.*
- [3] *D. McNutt, "SCSI and Beyond", Byte: 112-116, Aug 1994.*
- [4] *M. Wray, "The use of SCSI bus in high data transfer applications", Electronic Engeneering: 51-57, Oct 1989.*
- [5] *J. Lohmeyer, "Use SCSI devices for multiprocessor smart-I/O systems", EDN: 183-188, Jan 1985.*
- [6] *Advanced Micro Devices , Inc. , "3-Volt System Logic for Personal Computer Data Book", 1993*

# CAPÍTULO 6

## CONCLUSÕES

Neste trabalho estudamos diversos aspectos relevantes na obtenção do hardware de um sistema de aquisição e visualização de vídeo digital que atenda as necessidades, de certa forma atípicas, de um grupo de pesquisa com atividades ligadas à codificação de imagem. Foram considerados desde problemas encontrados nas soluções de placa disponíveis para PC's, até pontos importantes para possíveis implementações de tais sistemas.

No que se refere aos problemas encontrados nas soluções de placa, observou-se que estes deviam-se principalmente ao fato do barramento de expansão ISA/EISA do PC não suportar às altas taxas de dados geradas pela digitalização do sinal de vídeo, e em razão de requerer-se a capturas dos quadros sem o uso de compressão. Dentro deste cenário, a obtenção de um sistema de aquisição através de um projeto dedicado a satisfação destas necessidades específicas pareceu-nos uma alternativa bastante atraente, a que nos motivou a iniciar um estudo visando-se colher informações acerca dos problemas e das formas de implementação de um sistema como este.

Particionou-se então o problema em três questões principais: a) conversão A/D e D/A de sinais de vídeo; b) armazenamento de amostras de vídeo digital ; e c) obtenção de interfaces comunicação com o computador hospedeiro. Ao longo do estudo destas questões, procurou-se levantar informações acerca dos componentes chaves necessários, dos padrões de digitalização de vídeo, e das interfaces de comunicação. Procuramos ainda elaborar ainda que de maneira inicial uma proposta para de implementação de um sistema de aquisição e visualização dedicado. Quanto aos resultados obtidos a partir de nosso estudo, diríamos que eles se concentram mais fortemente no que se refere aos componentes chaves. De modo que o presente trabalho deve ser encarado como um sólido ponto de partida à implementação de sistemas de aquisição de vídeo.

Embora a situação no que se refere a velocidade do barramento dos PC's tenha se alterado sensivelmente com a chegada de novos barramentos de expansão como o VESA local bus e o PCI bus, dado a fato das aplicações a que se destinam as placas de aquisição de vídeo comerciais, ou seja multimídia e edição de vídeo semi-profissional, não serem prejudicadas pelo uso de compressão, e ainda da relativa facilidade de sua implementação, é pouco provável que surjam no mercado soluções que não façam uso da tecnologia de compressão. Ou seja, um sistema dedicado pode ser atraente mesmo nesta nova situação. Além disso, o conhecimento adquirido na implementação de tal sistema poderia representar um passo inicial para a obtenção de sistemas mais sofisticados capazes de tratar sinais de HDTV.

# **APÊNDICE A:**

## **Exemplo de implementação SCSI usando o PI/T 68230**

# Standard SCSI bus facilitates peripheral control

---

*The SCSI bus lets CPUs and peripherals transfer information using standard commands, and the command protocol makes it easy to design a bus controller. In this, the second of a 2-part series, you'll see how to use a programmable, parallel I/O interface chip in a SCSI bus controller. By using software and a relatively inexpensive host-adaptor circuit, you can interface many types of low-cost peripherals from various manufacturers with your computer. And when you change peripherals, you won't have to make hardware and software modifications.*

---

Bob Beims and Pat LaViolette, *Motorola*

As a designer using the SCSI bus, you have two options for bus control: Use specialized, complex bus controllers or build your own circuits. As described in part 1 of this series (see EDN, March 21, 1985, page 227), the SCSI command protocol sends command-description messages to target processors, which process the commands. Because any target processor can release the bus while processing a command, other bus activity can continue until that I/O process is complete. Simple  $\mu$ P interface chips like the Intel 8255A and Motorola MC6821 can't handle the arbitration required for this bus activity, for

although they provide parallel I/O lines for data transfer, they lack some of the control operations required by the SCSI command protocol.

The MC68230 programmable interface/timer chip, on the other hand, provides another approach to building your own circuits. It contains both data lines and control signals and handles bidirectional data transfers with a double-buffered, I/O-port architecture. This makes it suitable for SCSI bus-control operations. The one bidirectional I/O port on the chip uses four control lines and automatically supervises the Req\*-Ack\* (request-acknowledge) protocol for the SCSI bus.

This chip also controls timing operations for the 68000  $\mu$ P family. It uses the CPU's asynchronous, nonmultiplexed address and data buses. Fig 1 shows the chip and indicates the functionally grouped signal pins. The interface/timer chip and the 68000  $\mu$ P communicate via the standard interfacing techniques that apply to most  $\mu$ P chips. The CPU provides a chip-select signal ( $\overline{CS}$ ) and indicates the transfer direction with the read/write signal ( $R/\overline{W}$ ). Information travels to and from the chips over data-bus lines  $D_7$  through  $D_0$ . The interface circuit connects the CPU's  $A_5$  through  $A_1$  address lines to the chip's register-select lines,  $RS_5$  through  $RS_1$ , in order to address directly its 32 internal registers. The chip's  $\overline{DTACK}$  output line indicates the end of a data-transfer operation.

In conjunction with the SCSI bus circuits, the host

The MC68230 chip contains data lines and control signals and provides a double-buffered I/O-port architecture for bidirectional data transfers.

computer uses additional logic chips and assembly-language control software for a 68000-based computer. In addition to managing the data bus, the host computer's SCSI bus circuits (the host adapter) generate four signals and sense the levels of five other bus-control signals. When the host adapter supports bus arbitration and reselection, the Sel\* (select) and Bsy\* (busy) lines must be bidirectional. If you use parity for data-transfer operations, each bus interface must generate as well as detect the parity bits.

Although the programmable interface/timer chip provides three 8-bit I/O ports, six of the port C lines also act as control signals. The TOUT and PIRQ lines signal the CPU as interrupts occur in the 24-bit timer or at one of the I/O ports. The TIACK and PIACK lines provide interrupt-acknowledge signals to the chip. The DMAREQ line signals a direct-memory-access (DMA) request to a separate DMA control chip that then transfers information between the interface/timer chip and the computer's main memory. The interface cir-

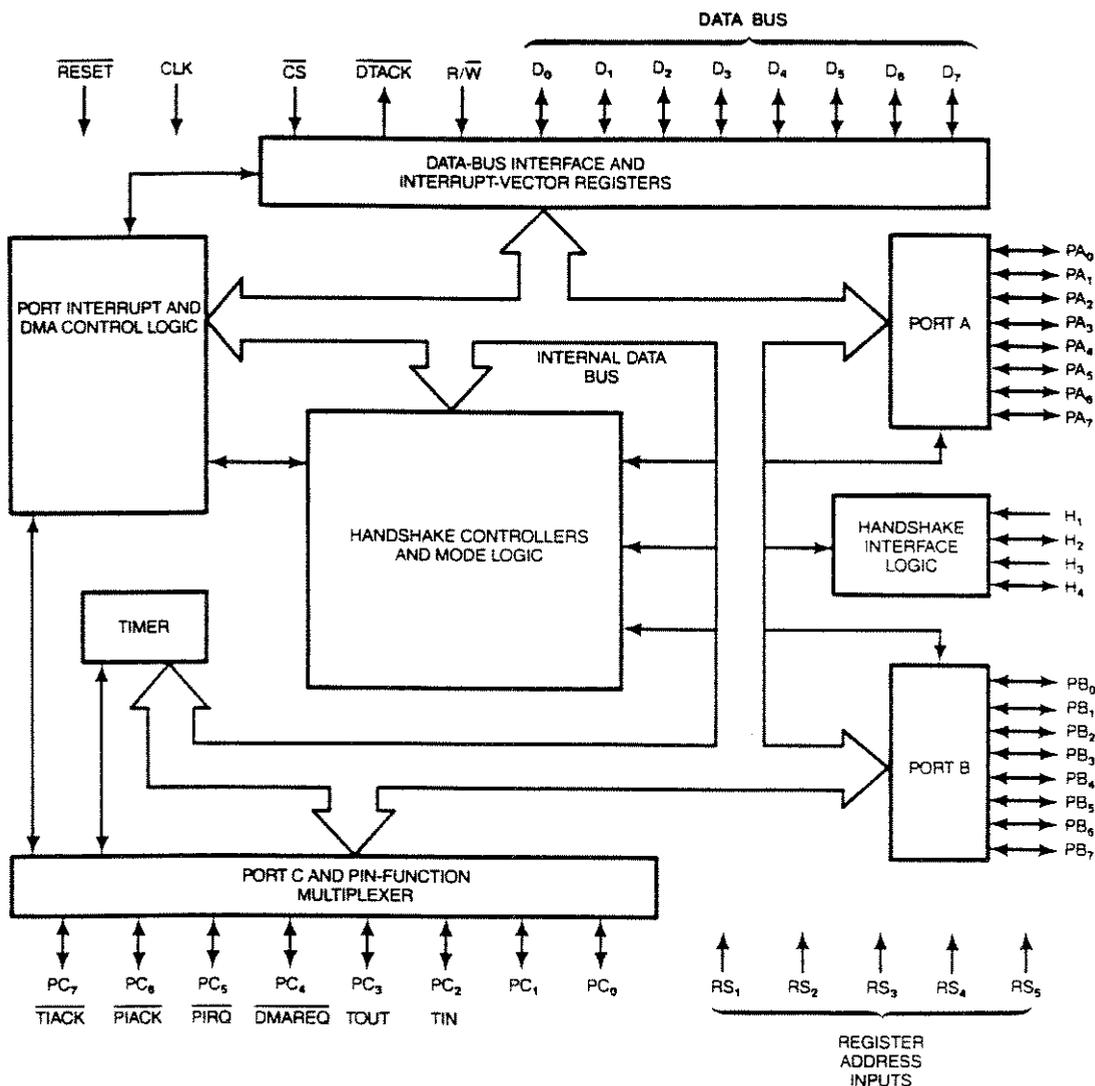


Fig 1—In the MC68230 programmable interface/timer chip, the port A lines control the interface. The handshaking signals control the port I/O operations.

cuits that you use to connect the DMA chip and host computer depend on the computer system you use (see Refs 1, 2, and 3).

### Ports afford 2-way transfers

You can operate the interface/timer chip's parallel A and B I/O ports in several ways. For the application under discussion here, use mode 2, in which the computer employs the port A data-direction register (PADDR, register 2—consult the listing starting on page 250) to designate each line at port A for either input or output operations. (For a description of the interface/timer chip's other modes, see Ref 3.) The interface/timer chip controls the port A lines with the bits in the port A data register (PADR, register 8).

Port B also provides bidirectional signal lines and includes double buffers for both input and output transfers. This particular architecture provides one byte for immediate use and stores the following byte. The process allows the interface/timer chip to send one byte to port B while the computer loads it with the next byte. For input operations, the double buffers let the chip transfer one byte to the computer while a peripheral transfers another byte to the chip. Port B's input and output buffers are independent.

Four handshake signals, H<sub>1</sub> through H<sub>4</sub>, manage the bidirectional data transfer between port B and the SCSI bus. The H<sub>1</sub> and H<sub>2</sub> lines provide output-control signals, and the H<sub>3</sub> and H<sub>4</sub> lines provide input-control signals. When the interface/timer chip sets port B for bidirectional I/O, the H<sub>1</sub> line controls the data flow. Port B's data-direction register (PBDDR) isn't used.

For output operations (Fig 2), the chip activates the signal to its active logic level its H<sub>2</sub> output to signal a peripheral that output data is available. The peripheral negates the H<sub>1</sub> input, gets the data from bus, and activates H<sub>1</sub> to tell the interface/timer that it has received the information. The peripheral checks the H<sub>2</sub> output again to see if another byte is ready. If the output buffers are empty, the H<sub>2</sub> output line isn't activated again until more information is available.

As long as the H<sub>1</sub> line remains activated, the interface/timer chip disables its output buffers and performs input transfers that use a similar process (Fig 3). The chip activates its H<sub>4</sub> output when possible to receive data. At that point, the peripheral places a data byte on the port B bus lines and activates the interface/timer chip's H<sub>3</sub> input, which then loads the data into the chip's port. After transferring a byte to the chip, the peripheral checks the H<sub>4</sub> output

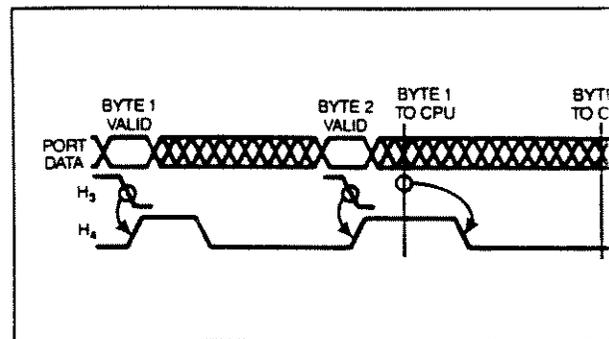


Fig 3—Port B's input timing diagram shows H<sub>3</sub> and H<sub>4</sub> engaging the handshaking protocol. The H<sub>3</sub> signal puts a byte into port B.

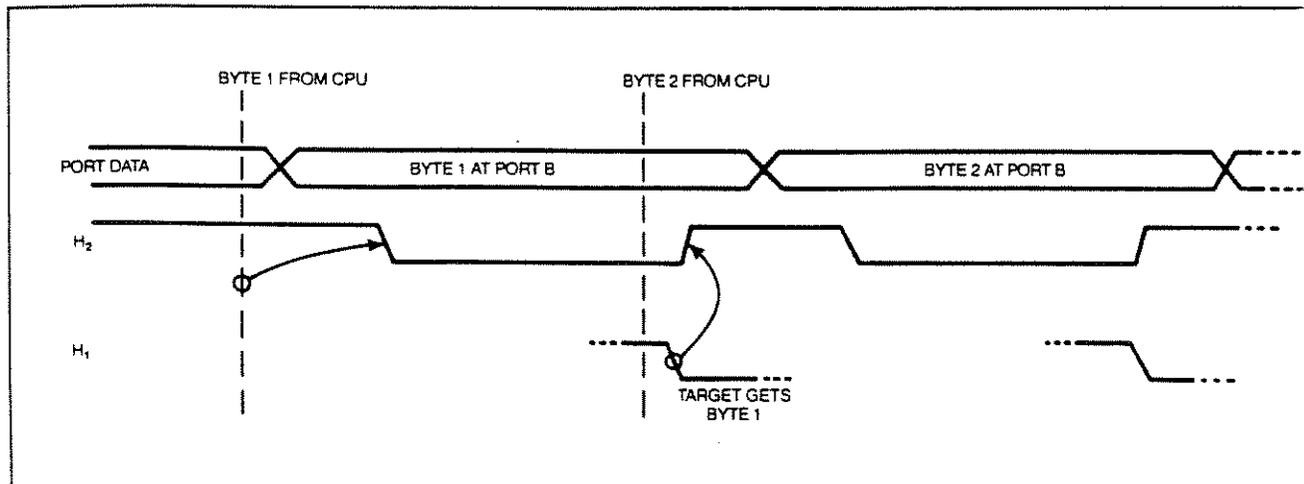


Fig 2—The port B output-timing relationships start when the target receives the first byte from the CPU, and the interface/timer chip activates the H<sub>2</sub> output. The peripheral activates H<sub>1</sub> to get the first data byte.

---

*For input operations, the double buffers let the interface/timer chip transfer one byte to the computer while a peripheral transfers another byte to the chip.*

---

and gets ready to transmit another byte when  $H_1$  is activated. The  $H_1$  output won't be activated again until the chip's double buffers can accept another byte.

The  $\mu P$  (or, if you choose, a DMA controller) transfers data to the SCSI host adapter by writing it in the interface/timer chip's port B data register (PBDR, register 9). Similarly, the  $\mu P$  receives input data from the PBDR. When the input buffers have data for the computer, or when space is available in the output buffers, the interface/timer chip places status bits in the port-status register (PSR, register 13), which the CPU monitors by using polling software. The interface/timer chip also interrupts the CPU when the program sets the proper control bits. If an application requires high-speed data transfer, a DMA controller chip uses the interface/timer chip's DMAREQ signal to control data flow. Under ideal conditions, this keeps the output buffers full and the input buffers empty. Interrupts modify the situation, however.

The interface/timer chip contains a 24-bit timer for interrupt signaling, operating-system timekeeping, pulse measuring, and time-out detection. The last two operations are useful in an SCSI-bus controller. Monitoring the period of the Bsy\* signal generated by a disk controller lets you determine the correct interleaving-time factor for high-speed data transfers. If a target doesn't respond to its selection signals within a set period, a time-out detection routine uses a timer to interrupt the CPU. This watchdog timer prevents the computer from waiting in an endless loop for a response from a nonexistent or inoperative peripheral device.

Fig 4 shows the host-adaptor circuit design that uses the programmable interface/timer chip. When the computer initializes the host adapter's circuits, port C's  $PC_0$

and  $PC_1$  lines generate enable and interrupt-clear (INTCLR) outputs. When the CPU or a power-on condition resets the interface/timer chip, all port pins are set up as input lines. The port A and port B lines and the  $PC_0$  and  $PC_1$  lines go to the logic-one state because they are set by internal pull-up resistors on each pin. The logic-one condition of the  $PC_0$  line causes the host adapter circuit to disable the Rst\*, Atn\*, Sel\*, Bsy\*, and Ack\* outputs to the SCSI bus. This prevents these signals from disrupting other devices on the bus while the CPU sets up the interface/timer chip when you turn on the computer.

The  $PC_1$  output controls the interrupt part of the circuit and disables interrupts that come from the parity chip, SCSI reset, and message detector, until the interface is ready for them. The software described in the listing controls the interface/timer chip's  $PC_1$  output (INTCLR), which allows the circuit to pass interrupts to the CPU only when the interface participates in a SCSI operation. The INTCLR signal also blocks parity-error, message-transmission, and SCSI-reset interrupts that come from bus operations independent of this interface. When the host adapter isn't in use, the CPU disables the interrupt-detecting circuits.

The buffers between the port A lines and the data bus invert the signals and provide the open-collector drivers that the SCSI bus standard requires. The Sel\* ( $PA_3$ ) and Bsy\* ( $PA_4$ ) outputs both pull the appropriate identification-select (ID Select) jumper line to the logic-zero state. This action identifies the adapter for other initiators and targets and supports the arbitration and selection phases. Furthermore, the Bsy\* signal switches the data-bus buffers to the input mode so that the interface/timer chip detects other initiators' ID signals on the bus during the arbitration phase. The same circuits let the chip read its own jumper-selected ID bit when the computer initializes the host adapter.

#### The Req\*/Ack\* control block

The Fig 4 circuit provides the timing relationships and level conversions that let the adapter and target transfer information according to the handshaking protocol. The I/O signal routes the Req\* and Ack\* signals to either the  $H_1$  and  $H_2$  lines for output transfers or to the  $H_3$  and  $H_4$  lines for input transfers. The four I/O control signals ( $H_1$  through  $H_4$ ) follow the Req\*-Ack\* protocol, with the following exception.

When the host adapter starts to transfer a block of data, both of the interface/timer chip's input buffers are empty and both output buffers are full. The target

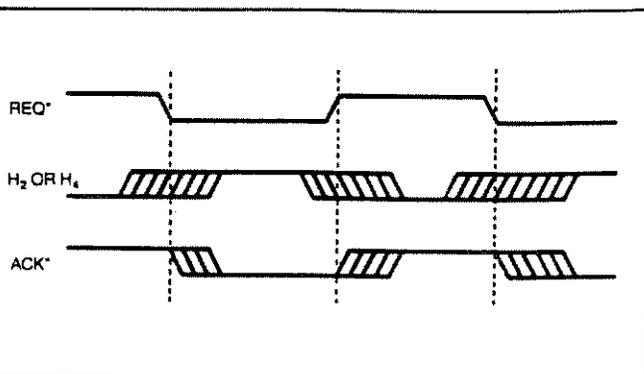


Fig 5—The interface circuit delays the Ack\* signal beyond the negative edge of the Req\* signal. This complies with the special bus-timing requirements for some peripherals and lets all SCSI-based devices use the host adapter.

requests a transfer, and the host adapter complies by activating the Ack\* signal. The interface/timer chip's controller then prepares for the next transfer by negating H<sub>2</sub> or H<sub>4</sub> without waiting for the target to negate the Req\* line. This violates the SCSI transfer protocol and causes timing problems for some target devices. Two cross-coupled NAND gates, however, generate the proper timing relationship for the Ack\* output. Fig 5 illustrates how the Ack\* signal is delayed until after negation of Req\*, regardless of the level on the H<sub>2</sub> or H<sub>4</sub> line.

The two D flip-flops shown in the Req\*/Ack\* control block in Fig 4 delay the H<sub>1</sub> input signal to meet the requirements of all SCSI target devices. In response to an activated Req\* signal, the initiator places data on the bus and activates Ack\*. The data on the bus must remain stable until Req\* is negated, after which the data can usually change; however, some controllers require stable data for a short time after the Req\* signal is negated. The flip-flops delay only the negated edges of Req\* signals sent to the H<sub>1</sub> input. An 8-MHz clock signal provides a data-hold time between 125 and 250 nsec, but this time delay doesn't affect data-transfer rates. Fig 6 shows the control-signal timing for both input and output operations.

#### The data-buffer block

The bus buffers use the enable signal from the interface/timer chip's PC<sub>0</sub> output for overall control, and a combination of the SCSI bus's I/O and Bsy\* signals selects the data-transfer direction. The two 74S38 open-collector NAND gates drive the data bus, and a 74LS240 receives and inverts data for the interface/timer chip. To prevent the 74LS240's input buffers and the interface/timer chip from driving the port-B pins at the same time, the I/O signal also controls the chip's H<sub>1</sub> input. Gating the I/O and Req\* signals provides an interlock that forces port B into the input mode whenever a bus device asserts the I/O line.

To support the full SCSI bus specification, a host adapter must disconnect itself from the bus when its target device requires additional time to complete a command. Later, the host adapter responds to the target's reselection sequence and completes the operation. The 5-input NOR gate detects the reselection phase, in which the target device asserts the I/O, Sel\*, and the host adapter's ID signals and negates the Bsy\* signal. In supporting optional SCSI-bus features, the MC68153 bus-interrupter module handles interrupts and acknowledges them for the CPU. For example,

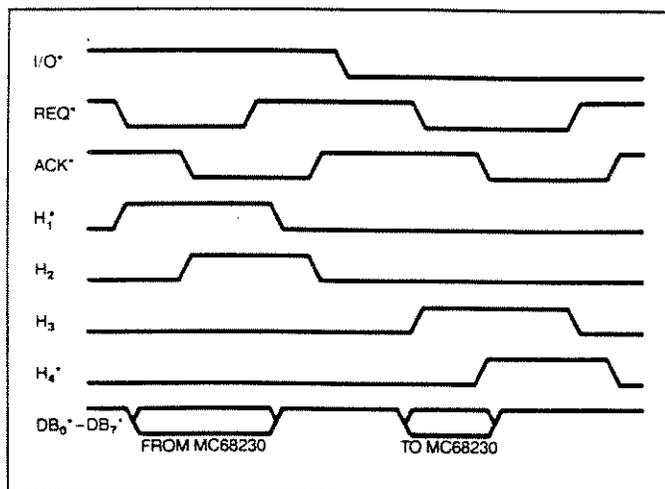


Fig 6—This timing diagram for the host adapter indicates the interface/timer chip and the SCSI bus control-signal timing relationships.

when the host adapter detects the start of a reselection sequence, the bus-interrupter module sends the CPU a priority interrupt to start the reselection software routine.

#### Parity errors yield interrupt

When your bus system requires you to use parity, the host adapter generates and checks for odd parity during all bus phases except Bus-Free and Arbitration. A 74S280 parity-generator/checker chip performs the calculations and generates an interrupt when it detects a parity error. The 74S280 drives the open-collector parity line during the host adapter's data transmissions and checks the parity of data during input operations. Because the interface/timer chip has double-buffered data registers, you can't determine which byte caused the parity error; the computer simply ends its present operation and tries again. If the host detects a parity error during a reselection phase, it ignores the target and waits for another reselection sequence that provides a valid parity bit.

#### Error detection

In addition to detecting the host-reselection phase and parity errors, the bus-interrupter module also monitors the Rst\* signal and detects messages sent to the host. The interrupts free the CPU from the job of periodically polling the adapter's status bits. The reset interrupt is critical here because the computer must respond to the Rst\* signal more quickly than is possible with polling software.

---

*When the input buffers have data for the computer, the interface/timer chip sets status bits in the port-status register that the CPU monitors with polling software.*

---

The message interrupt simplifies error-detecting software. For example, if an error occurs in the middle of a block transfer, the target stops the transfer and sends the host an error message. The interface/timer chip accepts the target's status information as part of the data block (it makes no distinction between control and data bytes), and the target negates the Bsy\* line and so terminates the transfer. The data-input software routine won't receive all the expected data from the interface/timer chip, and this keeps the computer in an endless loop unless the adapter detects the status message. The software could detect the end of a transfer by testing the state of the C\*/D or Bsy\* signals, but polling steps take time and slow down the overall transfer rate.

#### **Bus software**

The software (see listing) used with the interface/timer chip supports neither the full SCSI command set nor all of the message protocols, but it provides the commonly used commands and allows read/write operations with such random-access devices as disks. The program is uncomplicated because the interface/timer chip and the other chips take care of the handshaking, signaling, and data-buffering operations.

Because this routine uses software polling of the interface/timer chip, it isn't written for real-time data-transfer operations. However, you can program the chip to generate an interrupt or a DMA-request signal when the proper status bits are set. This relieves the computer of some major housekeeping operations. Because of software requirements for context switching, interrupt-based software requires more processing time, and transfer rates don't reach their maximum. Nevertheless, an interrupt-based system may be sufficient for some operating systems. You obtain the highest transfer rates by using a DMA controller that requires little of the software's time. With such a DMA controller, you can achieve transfer rates as high as 1M bytes/sec.

We tested the Fig 4 interface circuit on an MC68000-based computer with an 8-MHz clock, using a DTC1403D disk controller and SA1004 disk in order to evaluate both the hardware and software. When we used the optimum interleaving time factor, the circuit achieved a transfer rate of about 80k bytes/sec. This is much slower than the 1M-byte/sec transfer rate of the disk system alone or the 1.5M-byte/sec rate that the SCSI bus allows.

The software routines transfer data at about 200k

bytes/sec, but this is reduced to 110k bytes/sec when you put the read/write routines in a diagnostic loop to test the system. When you use an Adaptec 5500 controller and a Miniscribe 2012 hard disk with a DMA controller and a different software loop, the data rate increases to 250k bytes/sec, still below the maximum. Some of this difference arises from the arbitration among devices for the computer's main buses and from software time delays. You achieve the maximum transfer rate only when your DMA controller has its own local memory bus for the interface/timer chip.

The SCSI bus connects small, low-cost computer systems to intelligent peripheral controllers and provides moderate performance. By using the relatively inexpensive host-adapter circuits and software described here, you can add many types of economical peripherals from various manufacturers to your computer—and when you change peripherals, you don't need to make hardware or software changes.

---

#### **References**

1. *M68000 16/32-Bit Microprocessor Programmer's Reference Manual*, 4th ed, Prentice-Hall Inc, Englewood Cliffs, NJ 07632, 1984.
2. "MC68000 16-Bit Microprocessor Data Sheet," Motorola Semiconductor Products Inc, Austin, TX, 1984.
3. "MC68230 Parallel Interface/Timer (PI/T) Data Sheet," Motorola Semiconductor Products Inc, Austin, TX, 1983.

---

#### **Authors' biographies**

*Bob Beims is an application design engineer covering M68000 products for Motorola in Austin, TX. He has a BS in physics from Kansas Wesleyan and a BSEE from Kansas State University. Bob has a son and daughter with whom he spends his spare time.*



*Pat LaViolette is employed by Motorola as a system design manager for  $\mu$ Cs. He graduated from the University of Nebraska and enjoys woodworking and waterskiing as leisure activities.*



# LISTING: SCSI HOST-ADAPTER DRIVER ROUTINE

```

* GENERAL I/O ROUTINE FOR A SCSI HOST ADAPTER USING THE MC68230.
* THIS ROUTINE IS POSITION INDEPENDENT AND REENRANT.
*
* INPUT PARAMETERS:
* 00.W - BYTES/BLOCK COUNT
* 01.W - (15:8) TARGET ID, [7:0] INITIATOR IO MASK
* 02.W - COMMAND DESCRIPTOR BLOCK (CDB) ADDRESS
* 03.W - I/O BUFFER ADDRESS
* 04.W - SCSI DATA PORT ADDRESS (PDR)
*
* OUTPUT PARAMETERS:
* 00.B - COMPLETION STATUS BYTE
* 01.B - COMPLETION MESSAGE BYTE
*
* OTHER REGISTERS USED:
* (SAVED AND RESTORED)
* 02.W - READ, WRITE OR STATUS ROUTINE JUMP OFFSET
* 03.W - CDB BYTE COUNT
* 04.W - DATA TRANSFER BYTE COUNT
*
* THE PI/T SHOULD BE CONFIGURED PRIOR TO CALLING THIS ROUTINE AS FOLLOWS:
*
* PCCR - MODE 2, M1 & M4 ACTIVE LOW, M2 & M3 ACTIVE HIGH, TRANSFERS DISABLED
* PSRR - INTERRUPTS AND DMAREQ'S DISABLED.
* PADR - BITS 7-4 OUTPUT, BITS 3-0 INPUTS.
* PCDR - BIT 0 OUTPUT.
* PACR - M2 INTERLOCKED MODE, M1S = 1 WHEN EITHER OUTPUT BUFFER IS EMPTY.
* PBCR - M4 INTERLOCKED MODE, M3S = 1 WHEN EITHER INPUT BUFFER IS FULL.
*
* ALL REMAINING PORT C BITS MAY BE CONFIGURED AS DESIRED.
* THE TIMER IS NOT USED BY THIS ROUTINE.
*
* CONTROL BIT AND REGISTER OFFSET DEFINITIONS FOR THE MC68230 INTERFACE:

```

```

FFFFFFFF MODE_REG EQU -18 MODE CONTROL REGISTER (PCCR).
FFFFFFFF CNTL_REG EQU -02 CONTROL SIGNAL REGISTER (PADR).
00000000 DATA_REG EQU +00 DATA PORT REGISTER (PDR).
00000004 IDNT_REG EQU +04 ARBITRATION REGISTER (PBAR).
00000006 ENBL_REG EQU +06 ENABLE REGISTER (PCDR)
00000008 STAT_REG EQU +08 BUFFER STATUS REGISTER (PSR).

00000007 RST EQU 7 RESET
00000006 ATN EQU 6 ATTENTION
00000005 SELO EQU 5 SELECT (OUTPUT)
00000004 BSYO EQU 4 BUSY (OUTPUT)
00000003 SELI EQU 3 SELECT (INPUT)
00000002 BSYI EQU 2 BUSY (INPUT)
00000001 C_D EQU 1 CONTROL/DATA
00000000 MSG EQU 0 MESSAGE

00000000 M1S EQU 0 M1 STATUS BIT (READY FOR DATA).
00000002 M3S EQU 2 M3 STATUS BIT (DATA WAITING).

00000030 ENBL EQU 0 HOST ADAPTER ENABLE BIT.

REGLIST REG 02=04/A0=A2 6 REGISTERS TO BE SAVED/RESTORED.
SECTION 1
XDEF SCSI ENTRY POINT.
1 00000000 48E738E0 SCSI MOVEM.L REGLIST,-(SP) SAVE THE REGISTERS WE WILL BE USING.
*
* DETERMINE WHAT KIND OF OPERATION IS TO BE PERFORMED AND THE PROPER BYTE
* COUNT IF IT INVOLVES DATA TRANSFERS. ALSO SET 02 TO THE PROPER INDEX
* VALUE TO JUMP TO THE READ, WRITE OR STATUS ROUTINES. IF A RESET HAS BEEN
* REQUESTED, THE RESET IS PERFORMED REGARDLESS OF THE CURRENT SCSI STATUS.
*
1 00000004 B1FC00000000 CMP.L #0,A0 RESET BUS?
1 00000004 670000CC BEQ.L RESET

1 0000000E 1010 MOVE.B (A0),D0 GET THE FIRST BYTS OF THE CDB.
1 00000010 7605 MOVEQ #6-1,D3 SET THE SIX BYTE CDB COUNT - 1.
1 00000012 7404 MOVEQ #STATUS-XFER,D2 SET THE DEFAULT OFFSET FOR NO DATA XFER.
1 00000014 4244 CLR.W 04 CLEAR THE UPPER BYTE OF D4 AND
1 00000016 18230004 MOVE.B 4(A0),D4 GET THE NUMBER OF BLOCKS (NOT GROUP 1).

1 0000001A 0C000008 CMP.B #08,D0 READ?
1 0000001E 6606 BNE CDB1
1 00000020 343C003E MOVE.W #READ-XFER,D2 LOAD THE READ DATA ROUTINE'S OFFSET.
1 00000024 6004 BRA CDB2

1 00000026 0C000004 CDB1 CMP.B #04,D0 WRITE?
1 0000002A 6608 BNE ARBRATE
1 0000002C 343C0050 MOVE.W #WRITE-XFER,D2 LOAD THE WRITE DATA ROUTINE'S OFFSET.

1 00000030 C8C3 CDB2 MULL 00,D4 NUMBER OF BLOCKS * BYTES/BLOCK.
1 00000032 5344 SUBQ.W #1,D4 ADJUST FOR DBCC LOOP COUNTER.

*
* WAIT FOR THE BUS TO ENTER THE BUS FREE PHASE AND ARBRATE FOR IT.
*
* THE ASSERTION OF BSY MUST OCCUR AS QUICKLY AS POSSIBLE (WITHIN 1.1 USEC)
* AFTER THE DETECTION OF BUS FREE, SO PREPARE TO ASSERT IT BEFORE WE CHECK
* THE BUS.
*
1 00000034 102A0004 ARBRATE MOVE.B ENBL_REG(A2),D0 READ PORT C AND GET READY
1 00000038 08800000 BCLR #0,D0 TO ASSERT THE ENABLE BIT.
1 0000003C 08E40004FFFE BSET #BSYO,CNTL_REG(A2) GET READY TO ASSERT THE BSY OUTPUT.
1 00000042 08E40005FFFE BSET #S.MODE_REG(A2) ENABLE INPUT TRANSFERS (TO NEGATE ACK+).

1 00000048 082A0002FFFE WTFREE BTST #BSYI,CNTL_REG(A2) BUS FREE?
1 0000004E 66FB BNE WTFREE

1 00000050 15400005 MOVE.B D0,ENBL_REG(A2) ENABLE THE HOST ADAPTER AND ASSERT BSY+.
1 00000054 E019 RDR.B #9,D0 WAIT FOR 2 USEC (22 CLOCK NOP).
1 00000056 C32A0004 AND.B D1,IDNT_REG(A2) READ AND MASK THE IO BITS LOWER THAN ME.

```

```

1 000005A 6708
1 000005C 08EAD0000006
1 0000062 60E4
      BSET SELECT
      BSET #ENBL,ENBL_REG(A2) ANY HIGHER THAN ME?
      BRA WTFREE THEN DISABLE THE INTERFACE
                                AND WAIT FOR THE NEXT BUS FREE PHASE.
*
* SELECT THE TARGET AND WAIT UNTIL IT RESPONDS.
*
1 0000064 08EAD005FFFE SELECT BSET #SELO,CNTL_REG(A2) ASSERT SELECT.
1 000006A E059 ROR.W #B,D1 GET THE TARGET ID
1 000006C 0302 BSET D1,DATA_REG(A2) AND PREPARE TO ASSERT IT TO THE BUS.
1 000006E 08AA0004FFFE BCLR #BSY0,CNTL_REG(A2) NEGATE BSY+.
1 0000074 08EAD000FFFE BSET #D,MODE_REG(A2) CHANGE M1 SENSE TO ENABLE OUTPUT BUFFERS
*
1 000007A 082AD002FFFE WAITBSY BTST #BSY1,CNTL_REG(A2) WAIT UNTIL THE CONTROLLER RESPONDS.
1 0000080 67F8 BEQ WAITBSY
*
1 0000082 08AA0005FFFE BCLR #SELO,CNTL_REG(A2) NEGATE SELECT.
1 0000088 08AA0000FFFE BCLR #D,MODE_REG(A2) CHANGE M1 SENSE LEVEL BACK TO LOW.
1 000008E 08EAD004FFFE BSET #4,MODE_REG(A2) ENABLE OUTPUT TRANSFERS.
*
* TRANSFER THE COMMAND DESCRIPTOR BLOCK.
*
1 0000094 082AD0000008 COMMAND BTST #M15,STAT_REG(A2) WAIT FOR THE M15 BIT SET (READY FOR DATA
1 000009A 67F8 BEQ COMMAND SIGNAL FROM THE PI/T).
*
1 000009C 1498 MOVE.B (A0)+,(A2) SEND THE NEXT CDB BYTE.
1 000009E 51C8FFFA DBF D3,COMMAND UNTIL THEY'RE ALL GONE.
*
* READ OR WRITE DATA OR PROCEED TO THE STATUS PHASE.
*
1 00000A2 4E8B20FE XFER JMP XFER(PC,D2,4) JUMP TO READ, WRITE OR STATUS ROUTINE.
*
* READ THE STATUS BYTE.
*
1 00000A6 082AD0002008 STATUS BTST #M15,STAT_REG(A2) WAIT FOR M15 BIT SET (DATA WAITING).
1 00000AC 67F8 BEQ STATUS
*
1 00000AE 1012 MOVE.B (A2),D0 READ THE STATUS BYTE.
*
* READ THE MESSAGE BYTE.
*
1 00000B0 082AD0002008 MESSAGE BTST #M15,STAT_REG(A2) CHECK THE M15 BIT AGAIN.
1 00000B6 67F8 BEQ MESSAGE
*
1 00000B8 1212 MOVE.B (A2),D1 READ THE COMPLETION MESSAGE BYTE.
*
* RETURN TO THE CALLING ROUTINE.
*
1 00000BA 08EAD0000006 FINISH BSET #ENBL,ENBL_REG(A2) DISABLE THE HOST ADAPTER,
1 00000C0 022AD0CFFFE AND.B #3CF,MODE_REG(A2) DISABLE DATA TRANSFERS,
*
1 00000C6 4CD8071C FINISH1 MOVE.L (SP)+,REG15 RESTORE THE PREVIOUS MACHINE STATUS,
1 00000CA 4E75 RTS AND RETURN.
*
* RESET THE BUS (REGARDLESS OF ITS PREVIOUS STATE).
*
1 00000CC 7819 RESET MOVEQ #25,D4 SET A COUNTER FOR >25 USEC @ 8 MHz.
1 00000CE 03F90007FFFE BSET #RST,CNTL_REG ASSERT RST+.
1 00000D4 51CCFFFE WAIT25 DBF D4,WAIT25 WAIT.
1 00000D8 08B80007FFFE BCLR #RST,CNTL_REG NEGATE RST+.
1 00000DE 60E4 BRA FINISH1
*
* READ THE NUMBER OF BYTES OF DATA INDICATED BY D4 FROM THE TARGET.
*
1 00000E0 7602 READ MOVEQ #M15,D3 SET THE BIT NUMBER AND STATUS REGISTER
1 00000E2 41EAD008 LEA STAT_REG(A2),A0 ADDRESS TO SPEED THE "BTST" INSTRUCTION.
*
1 00000E6 0710 RDBYTE BTST D3,(A0) WAIT UNTIL DATA IS WAITING.
1 00000E8 67FC BEQ RDBYTE
*
1 00000EA 1202 MOVE.B (A2),(A1)+ PUT THE NEXT BYTE INTO THE BUFFER
1 00000EC 51CCFFFB DBF D4,RDBYTE AND CONTINUE UNTIL WE'VE GOT THEM ALL.
*
1 00000F0 6084 BRA STATUS THEN GO READ THE STATUS INFORMATION.
*
* WRITE THE NUMBER OF BYTES OF DATA INDICATED BY D4 TO THE TARGET.
*
1 00000F2 7600 WRITE MOVEQ #M15,D3 SET THE BIT NUMBER AND STATUS REGISTER
1 00000F4 41EAD008 LEA STAT_REG(A2),A0 ADDRESS TO SPEED THE "BTST" INSTRUCTION.
*
1 00000F8 0710 WRTBYTE BTST D3,(A0) WAIT UNTIL THERE IS ROOM FOR MORE DATA.
1 00000FA 67FC BEQ WRTBYTE
*
1 00000FC 1499 MOVE.B (A1)+,(A2) SEND THE NEXT BYTE FROM THE BUFFER
1 00000FE 51CCFFFB DBF D4,WRTBYTE AND CONTINUE UNTIL THEY'RE ALL GONE.
*
1 0000102 60A2 BRA STATUS THEN GO READ THE STATUS INFORMATION.

```



## **APÊNDICE B:**

### **Chip Controlador SCSI - Am53C94/Am53c96**



T-52-33-27

Advanced  
Micro  
Devices

# Am53C94/Am53C96

## High Performance SCSI Controller

### DISTINCTIVE CHARACTERISTICS

- Plug compatible with NCR 53C94/53C96
- 5 MB per second SCSI transfer rate
- 20 MB per second DMA transfer rate
- 16-bit DMA Interface plus 2 bits of parity
- Flexible bus architecture, supports a three bus architecture
- Single ended SCSI bus supported by Am53C94
- Single ended and differential SCSI bus supported by Am53C96
- Selection of multiplexed or non-multiplexed address and data bus
- High current drivers (48 mA) for direct connection to the single ended SCSI bus
- Supports Disconnect and Reselect commands
- Supports burst mode DMA operation with a threshold of 8
- Supports 3 byte tagged queuing as per the SCSI-2 specification
- Supports group 2 and 5 command recognition as per the SCSI-2 specification
- Advanced CMOS process for low power consumption
- Am53C94 available in 84 pin PLCC package
- Am53C96 available in 100 pin PQFP package

### GENERAL DESCRIPTION

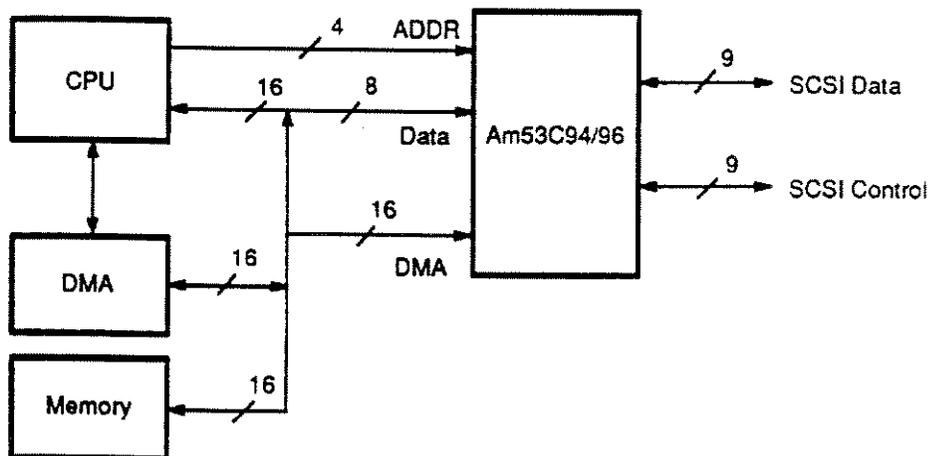
The High Performance SCSI Controller (HPSC) has a flexible three bus architecture. The HPSC has a 16 bit DMA interface, an 8 bit host data interface and an 8 bit SCSI data interface. The HPSC is designed to minimize host intervention by implementing common SCSI sequences in hardware. An on-chip state machine reduces protocol overheads by performing the required sequences in response to a single command from the host. Selection, Reselection, Information Transfer and Disconnection commands are directly supported.

The 16 byte internal FIFO further assists in minimizing host involvement. The FIFO provides a temporary stor-

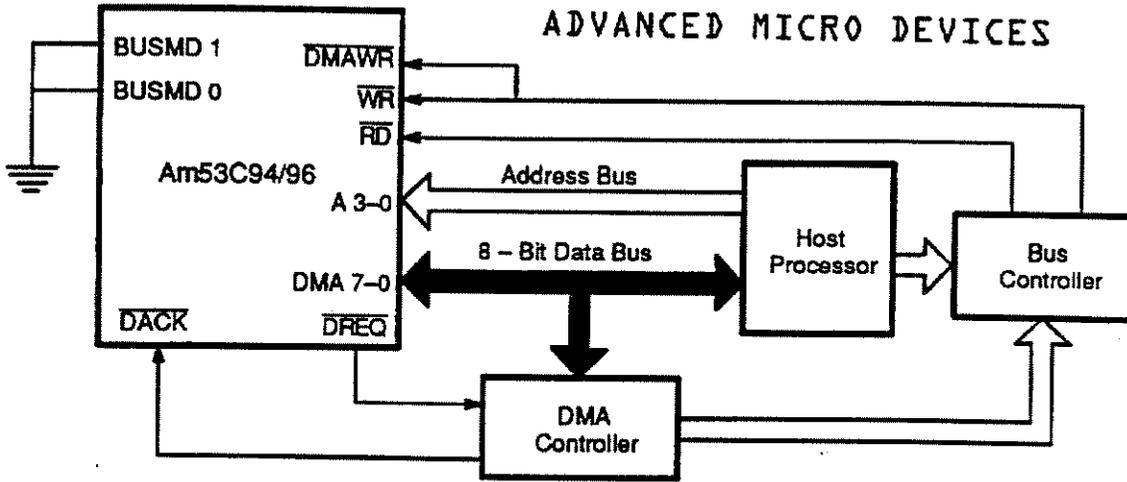
age for all command, data, status and message bytes as they are transferred between the 16 bit host data bus and the 8 bit SCSI data bus. During DMA operations the FIFO acts as a buffer to allow greater latency in the DMA channel. This permits the DMA channel to be suspended for higher priority operations such as DRAM refresh or reception of an ISDN packet.

Parity on the DMA bus is optional. Parity can either be generated and checked or it can be simply passed through.

### SYSTEM BLOCK DIAGRAM

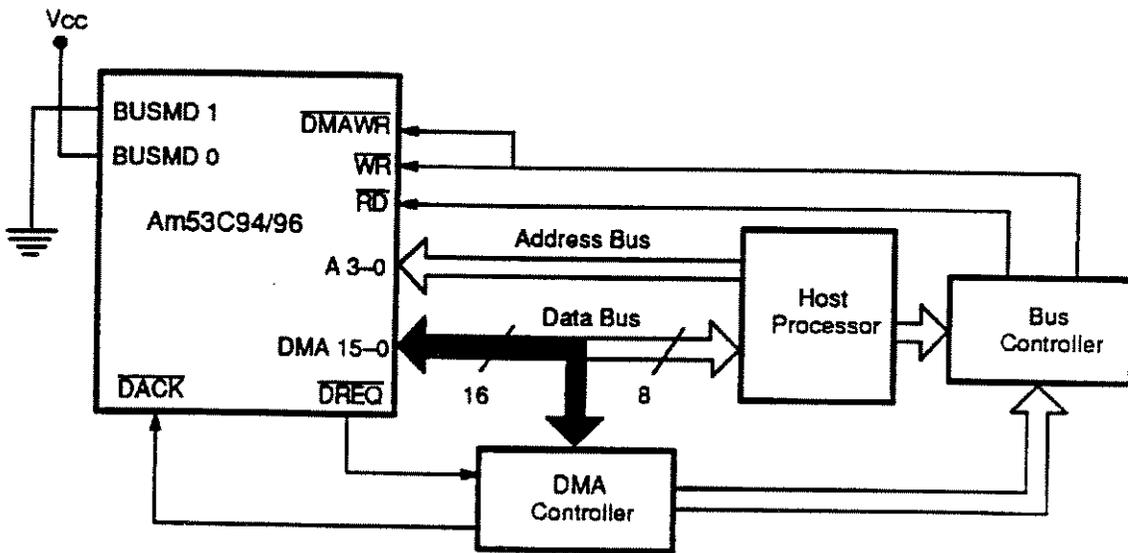


16506A-001A



Bus Mode 0

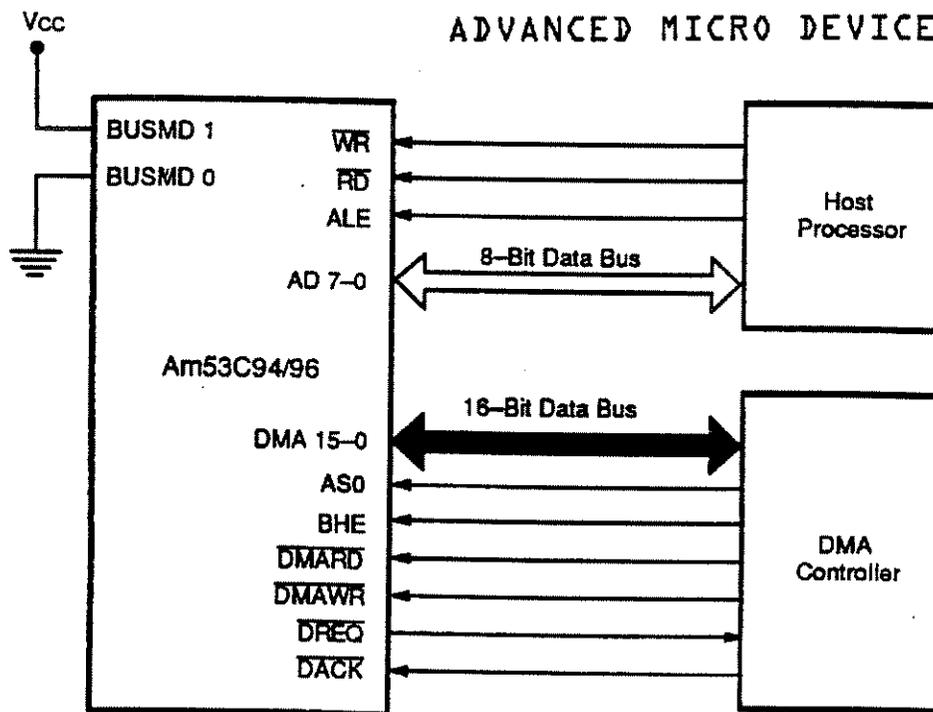
16506A-002A



Bus Mode 1

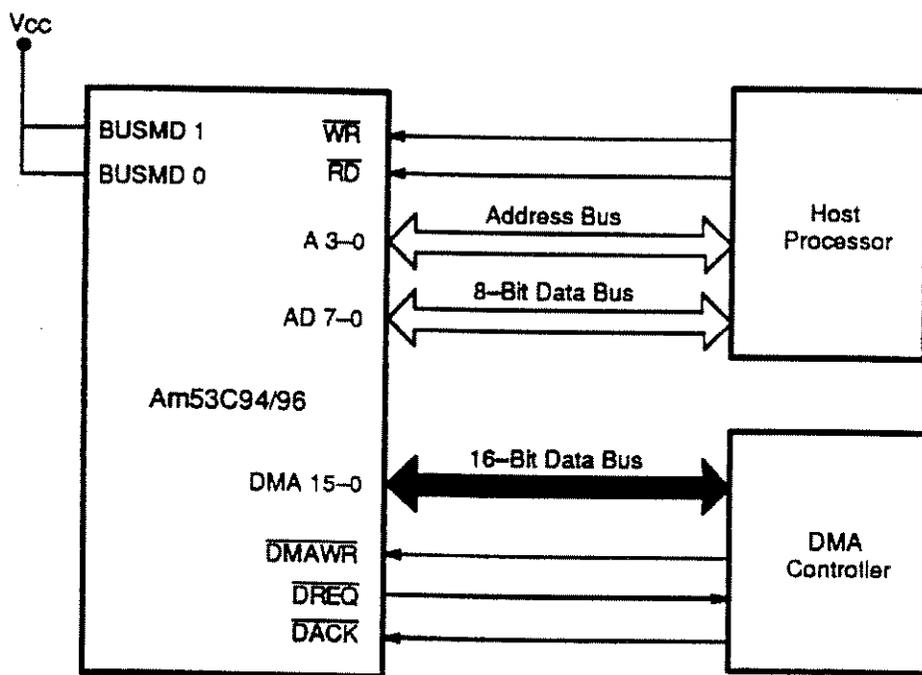
16506A-003A

ADVANCED MICRO DEVICES



16506A-004A

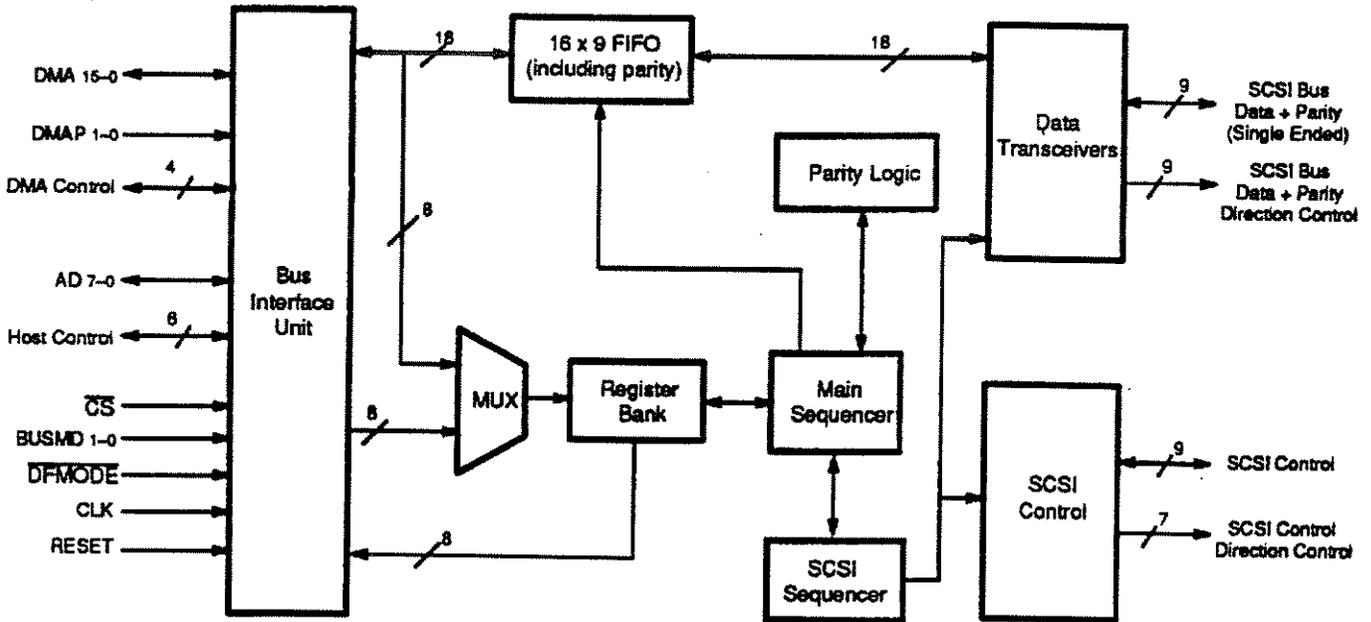
Bus Mode 2



16506A-005A

Bus Mode 3

ADVANCED MICRO DEVICES

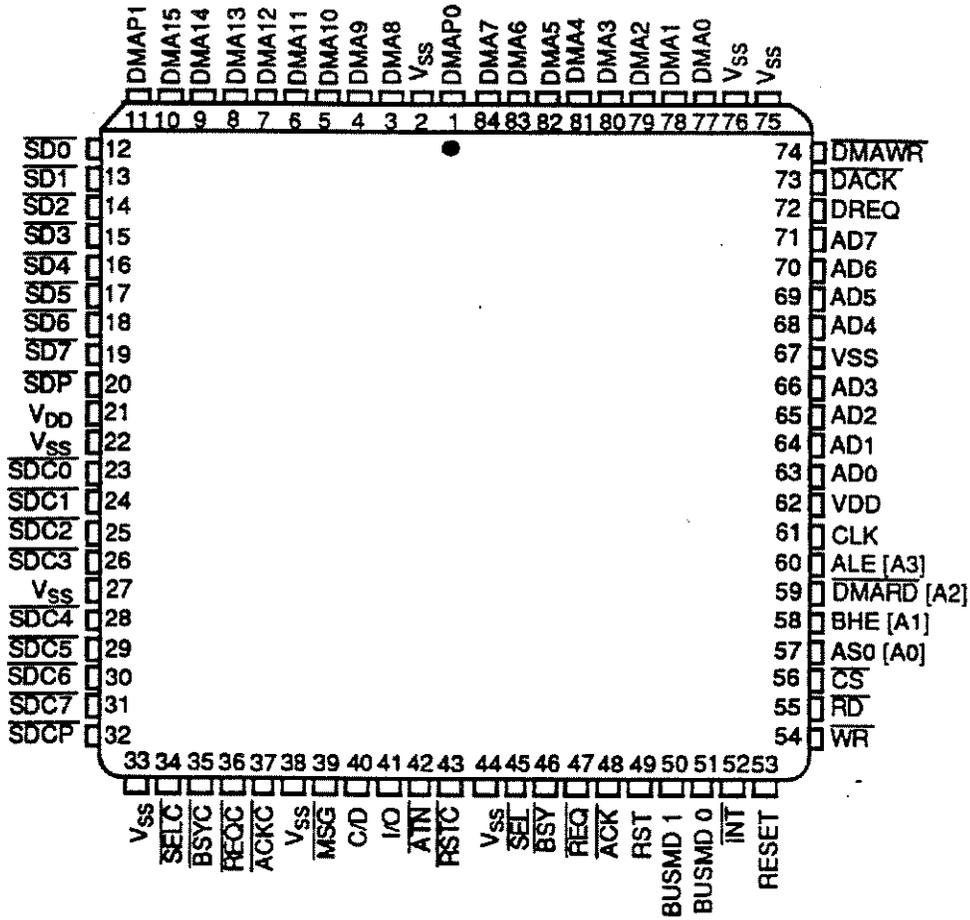


16506A-006A

CONNECTION DIAGRAM  
Am53C94 (Top View)

59E D ■ 0257525 0043027 25T ■ AMD1

PLCC ADVANCED MICRO DEVICES



16506A-007A

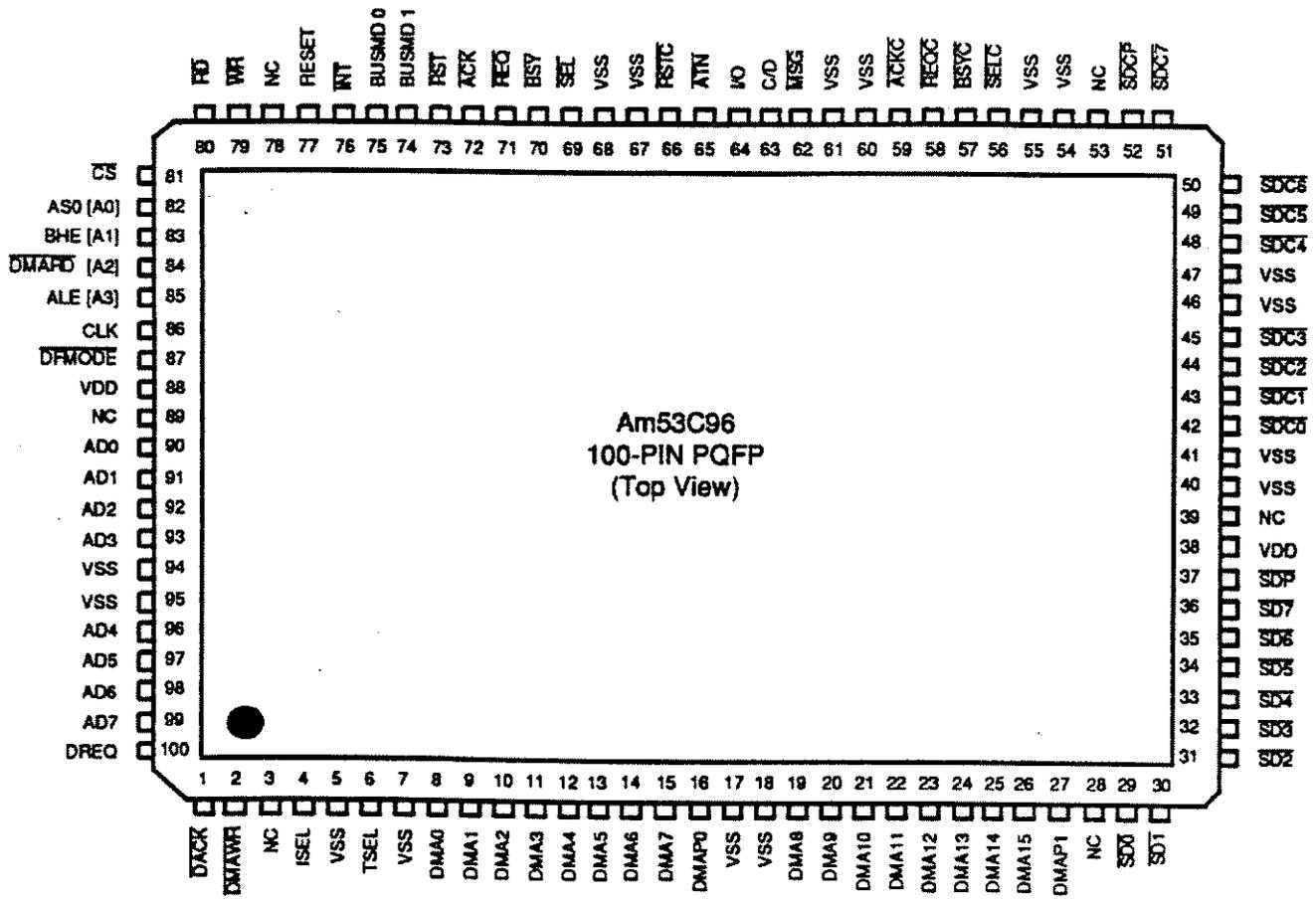


ADVANCE INFORMATION

CONNECTION DIAGRAM  
Am53C96 (Top View)

59E D ■ 0257525 0043028 196 ■ AMD1

PQFP ADVANCED MICRO DEVICES



16506A-008A

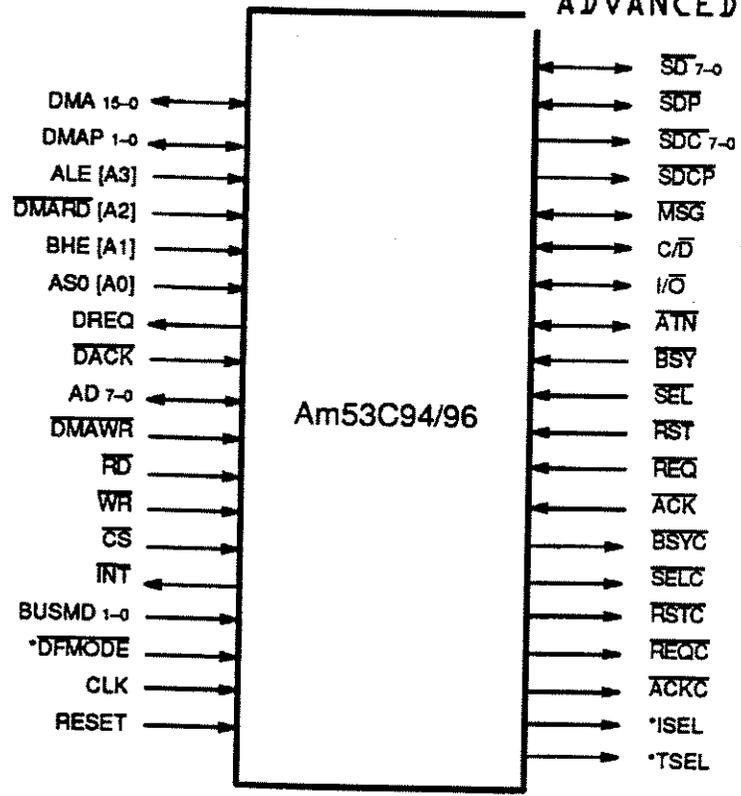
RELATED AMD PRODUCTS

Part Number	Description
Am53C80A	4 MBytes/sec Asynchronous CMOS SCSI Controller
Am33C93A	5 MBytes/sec Async/Synchronous CMOS SCSI Controller
Am85C80	Combination SCSI Controller (Am53C80A) and ESCC (Am85C30)
Am85C30	Enhanced Serial Communications Controller (ESCC)

LOGIC SYMBOL

59E D ■ 0257525 0043029 022 ■ AMD1

ADVANCED MICRO DEVICES



\* Pins available on the Am53C96 only

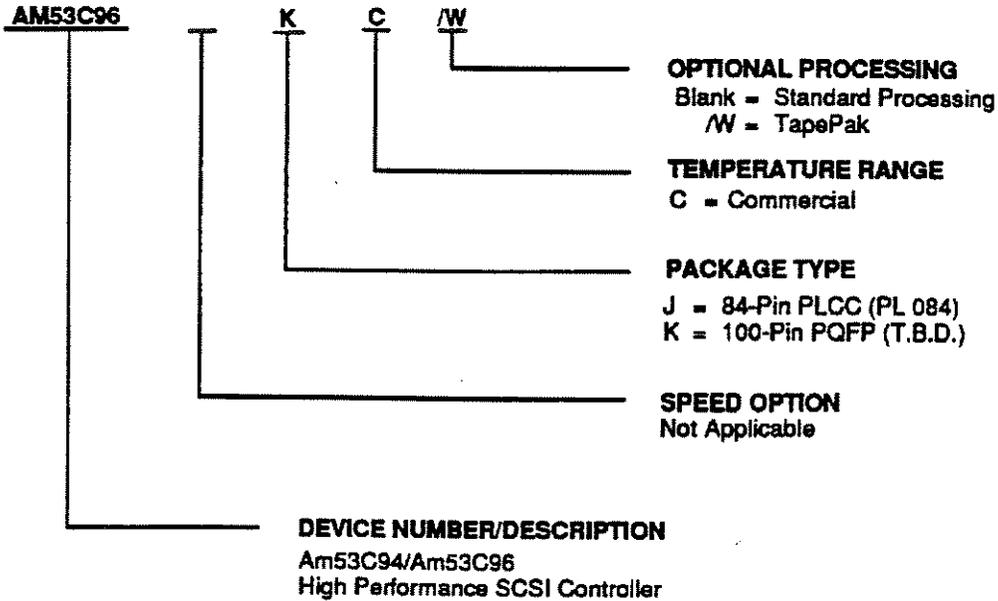
16506A-009A



**ORDERING INFORMATION**  
**Standard Products**

**ADVANCED MICRO DEVICES**

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



Valid Combinations	
AM53C94	JC
AM53C96	KC, KC/W

**Valid Combinations**

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.



**PIN DESCRIPTION**

**ADVANCED MICRO DEVICES**

**Host Interface Signals**

The following is the decoding for the BHE and AS0 inputs:

BHE	AS0	Bus Used
1	1	Upper Bus – DMA 15-8, DMAP 1
1	0	Full Bus – DMA 15-0, DMAP 1-0
0	1	Reserved
0	0	Lower Bus – DMA 7-0, DMAP 0

**DMA 15-0**

**Data / DMA Bus (Input/Output, Active High, Internal Pullup)**

The configuration of this bus depends on the Bus Mode 1-0 (BUSMD 1-0) inputs. When the device is configured for single bus operation, the host can access the internal register set on the lower eight lines and the DMA accesses can be made to the FIFO using the entire bus. When using the Byte Mode via the BHE and A0 inputs the data can be transferred on either the upper or lower half of the DMA 15-0 bus.

**DMAP 1-0**

**Data/DMA Parity Bus (Input/Output, Active High, Internal Pullup)**

These lines are odd parity for the DMA 15-0 bus. DMAP 1 is the parity for the upper half of the bus (DMA 15-8) and DMAP 0 is the parity for the lower half of the bus (DMA 7-0).

**ALE [A3]**

**Address Latch Enable [Address 3] (Input, Active High)**

This is a dual function input. When the device is configured for single bus operation this input acts as ALE. As ALE, this input latches the address on the AD 7-0 bus on its Low going edge. When the device is configured for dual bus operation this input acts as A3. As A3, this input is the third bit of the address bus.

**DMARD [A2]**

**DMA Read [Address 2] (Input, Active Low [Active High])**

This is a dual function input. When the device is configured for single bus operation this input acts as DMARD. As DMARD, this input is the read signal for the DMA 15-0 bus. When the device is configured for dual bus operation this input acts as A2. As A2, this input is the second bit of the address bus.

**BHE [A1]**

**Bus High Enable [Address 1] (Input, Active High)**

This is a dual function input. When the device is configured for single bus operation this input acts as BHE. As BHE, this input along with AS0 indicates on which lines the data transfer is to take place. When the device is configured for dual bus operation this input acts as A1. As A1, this input is the first bit of the address bus.

**AS0 [A0]**

**Address Status [Address 0] (Input, Active High)**

This is a dual function input. When the device is configured for single bus operation this input acts as AS0. As AS0, this input along with BHE indicates on which lines the data transfer is to take place. When the device is configured for dual bus operation this input acts as A0. As A0, this input is the zeroth bit of the address bus.

**DREQ**

**DMA Request (Output, Active High, Tri-State)**

This output signal to the DMA controller will be active during DMA read and write cycles. During a DMA read cycle it will be active as long as there is a word (or a byte in the byte mode) in the FIFO to be transferred to memory. During a DMA write cycle it will be active as long as there is an empty space for a word (or a byte in the byte mode) in the FIFO.

**DACK**

**DMA Acknowledge (Input, Active Low)**

This input signal from the DMA controller will be active during DMA read and write cycles. The DACK signal is used to access the DMA FIFO only and should never be active simultaneously with the CS signal, which accesses the registers only.

**AD 7-0**

**Host Address Data Bus (Input/Output, Active High, Internal Pullup)**

This bus is used only in the dual bus mode. This bus allows the host processor to access the device's internal registers while the DMA bus is transferring data. When using the multiplexed bus mode, these lines can be used for address and data. When using a non-multiplexed bus mode these lines can be used for the data only.

**DMAWR**

**DMA Write (Input, Active Low)**

This signal writes the data on the DMA 15-0 bus into the internal FIFO when DACK is also active. When in the single bus mode this signal must be tied to the WR signal.

**$\overline{RD}$**   
 Read (Input, Active Low)

This signal reads the internal device registers and places their contents on the data bus, when either  $\overline{CS}$  signal or  $\overline{DACK}$  signal is active.

**$\overline{WR}$**   
 Write (Input, Active Low)

This signal writes the internal device registers with the value present on the data bus, when the  $\overline{CS}$  signal is also active.

**$\overline{CS}$**   
 Chip Select (Input, Active Low)

This signal enables the read and write of the device registers.  $\overline{CS}$  enables access to any register (including the FIFO) while the  $\overline{DACK}$  enables access only to the FIFO.  $\overline{CS}$  and  $\overline{DACK}$  should never be active simultaneously in the single bus mode, they may however be active simultaneously in the dual bus mode provided the  $\overline{CS}$  signal is not enabling access to the FIFO.

**$\overline{INT}$**   
 Interrupt (Output, Active Low, Open Drain)

This signal is a non-maskable interrupt flag to the host processor. This signal is latched on the output on the high going edge of the clock. This flag may be cleared by reading the Interrupt Status Register (ISTAT) or by performing a device reset (hard or soft). This flag is not cleared by a SCSI reset.

**BUSMD 1-0**  
 Bus Mode (Input, Active High)

These inputs configure the device for single bus or dual bus operation and the DMA width.

BUSMD1	BUSMD0	Bus Configuration
1	1	Two buses: 8-bit Host Bus & 16-bit DMA Bus Register Address on A 3-0 & Data on AD Bus
1	0	Two buses: Multiplexed & byte control Register Address on AD 3-0 & Data on AD Bus
0	1	Single bus: 8-bit Host Bus & 16-bit DMA Bus Register Address on A 3-0 & Data on DMA Bus
0	0	Single bus: 8-bit Host Bus & 8-bit DMA Bus Register Address on A 3-0 & Data on DMA Bus

**$\overline{DFMODE}$**   
 Differential Mode (Input, Active Low)

This input is available only on the Am53C96. This input configures the SCSI bus to either single ended or differential. When this input is active, the device operates in the differential SCSI mode. The SCSI data is available on the SD 7-0 lines and the high active transceiver enables on the SDC 7-0 outputs. When this input is inactive, the device operates in the single-ended SCSI mode. The SCSI input data is available on SD 7-0 lines and the output data is available on SDC 7-0 lines. In the single-ended SCSI mode, the SD 7-0 and the SDC 7-0 buses can be tied together externally.

**CLK**  
 Clock (Input)

Clock input used to generate all the internal device timings. The maximum frequency of this input is 25 MHz. A minimum of 10MHz is required to maintain the SCSI bus timings.

**RESET**  
 Reset (Input, Active High)

This input when active resets the device. The RESET input must be active for at least two CLK periods after the voltage on the power inputs has reached Vcc minimum.

**SCSI Interface Signals**

**$\overline{SD}$  7-0**  
 SCSI Data (Input/Output, Active Low, Schmitt Trigger)

When the device is configured in the Single-Ended SCSI Mode ( $\overline{DFMODE}$  inactive) these pins are defined as inputs for the SCSI data bus. When the device is configured in the Differential SCSI Mode ( $\overline{DFMODE}$  active) these pins are defined as a bidirectional SCSI data bus.

**$\overline{SDP}$**   
 SCSI Data Parity (Input/Output, Active Low, Schmitt Trigger)

When the device is configured in the Single-Ended SCSI Mode ( $\overline{DFMODE}$  inactive) this pin is defined as the input for the SCSI data parity. When the device is configured in the Differential SCSI Mode ( $\overline{DFMODE}$  active) this pin is defined as bidirectional SCSI data parity.

**$\overline{SDC}$  7-0**  
 SCSI Data Control (Output, Active Low, Open Drain)

When the device is configured in the Single-Ended SCSI Mode ( $\overline{DFMODE}$  inactive) these pins are defined as outputs for the SCSI data bus. When the device is configured in the Differential SCSI Mode ( $\overline{DFMODE}$  active) these pins are defined as direction controls for the external differential transceivers. In this mode, a

signal High state corresponds to an output to the SCSI bus and a Low state corresponds to an input from the SCSI bus.

**SDCP**

**SCSI Data Control Parity (Output, Active Low, Open Drain)**

When the device is configured in the Single-Ended SCSI Mode ( $\overline{\text{DFMODE}}$  inactive) this pin is defined as an output for the SCSI data parity. When the device is configured in the Differential SCSI Mode ( $\overline{\text{DFMODE}}$  active) this pin is defined as the direction control for the external differential transceiver. In this mode, a signal high state corresponds to an output to the SCSI bus and a low state corresponds to an input from the SCSI bus.

**MSG**

**Message (Input/Output, Active Low, Schmitt Trigger)**

This is a bidirectional signal with a 48 mA output driver. It is an output in the target mode and a Schmitt trigger input in the initiator mode.

**C/ $\overline{\text{D}}$** 

**Command/Data (Input/Output, Schmitt Trigger)**

This is a bidirectional signal with a 48 mA output driver. It is an output in the target mode and a Schmitt trigger input in the initiator mode.

**I/ $\overline{\text{O}}$** 

**Input/Output (Input/Output, Schmitt Trigger)**

This is a bidirectional signal with a 48 mA output driver. It is an output in the target mode and a Schmitt trigger input in the initiator mode.

**ATN**

**Attention (Input/Output, Active Low, Schmitt Trigger)**

This signal is a 48 mA output in the initiator mode and a Schmitt trigger input in the target mode. This signal will be asserted when the device detects a parity error or it can be asserted via certain commands. In the target mode this pin is an input.

**BSY**

**Busy (Input, Active Low, Schmitt Trigger)**

This is a SCSI input signal with a Schmitt trigger.

**SEL**

**Select (Input, Active Low, Schmitt Trigger)**

This is a SCSI input signal with a Schmitt trigger.

**RST**

**Reset (Input, Active Low, Schmitt Trigger)**

This is a SCSI input signal with a Schmitt trigger.

**REQ**

**Request (Input, Active Low, Schmitt Trigger)**

This is a SCSI input signal with a Schmitt trigger.

**ACK**

**Acknowledge (Input, Active Low, Schmitt Trigger). This is a SCSI input signal with a Schmitt trigger.**

**BSYC**

**Busy Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. When the device is configured in the Single-Ended SCSI Mode ( $\overline{\text{DFMODE}}$  inactive) this pin is defined as a  $\overline{\text{BSY}}$  output for the SCSI bus. When the device is configured in the Differential SCSI Mode ( $\overline{\text{DFMODE}}$  active) this pin is defined as the direction control for the external differential transceiver. In this mode, a signal High state corresponds to an output to the SCSI bus and a Low state corresponds to an input from the SCSI bus.

**SEL $\overline{\text{C}}$** 

**Select Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. When the device is configured in the Single-Ended SCSI Mode ( $\overline{\text{DFMODE}}$  inactive) this pin is defined as a  $\overline{\text{SEL}}$  output for the SCSI bus. When the device is configured in the Differential SCSI Mode ( $\overline{\text{DFMODE}}$  active) this pin is defined as the direction control for the external differential transceiver. In this mode, a signal High state corresponds to an output to the SCSI bus and a Low state corresponds to an input from the SCSI bus.

**RSTC**

**Reset Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. The Reset SCSI command will cause the device to drive  $\overline{\text{RSTC}}$  active for 25–40 microseconds, which will depend on the CLK frequency and the conversion factor. When the device is configured in the Single-Ended SCSI Mode ( $\overline{\text{DFMODE}}$  inactive) this pin is defined as a  $\overline{\text{RST}}$  output for the SCSI bus. When the device is configured in the Differential SCSI Mode ( $\overline{\text{DFMODE}}$  active) this pin is defined as the direction control for the external differential transceiver. In this mode, a signal High state corresponds to an output to the SCSI bus and a Low state corresponds to an input from the SCSI bus.

**REQC**

**Request Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. This signal is asserted only in the target mode.

**ACKC**

**Acknowledge Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. This signal is asserted only in the initiator mode.



**ISEL**  
 Initiator Select (Output, Active High)

This signal is available on the Am53C96 only. It is active whenever the device is in the initiator mode. In the differential mode this signal is used to enable the initiator signals  $\overline{ACKC}$  and  $\overline{ATNC}$  and the device also drives these signals.

**TSEL**  
 Target Select (Output, Active High)

This signal is available on the Am53C96 only. It is active whenever the device is in the target mode. In the differential mode this signal is used to enable the target signals  $\overline{REQC}$ ,  $\overline{MSG}$ ,  $\overline{C/D}$  and  $\overline{I/O}$  and the device also drives these signals.

**FUNCTIONAL DESCRIPTION**

**ADVANCED MICRO DEVICES**

**Register Map**

Address (Hex.)	Operation	Register
00	Read	Current Transfer Count Register LSB
00	Write	Start Transfer Count Register LSB
01	Read	Current Transfer Count Register MSB
01	Write	Start Transfer Count Register MSB
02	Read/Write	FIFO Register
03	Read/Write	Command Register
04	Read	Status Register
04	Write	SCSI Destination ID Register
05	Read	Interrupt Status Register
05	Write	SCSI Timeout Register
06	Read	Internal State Register
06	Write	Synchronous Transfer Period Register
07	Read	Current FIFO Internal State Register
07	Write	Synchronous Offset Register
08	Read/Write	Control Register 1
09	Write	Clock Factor Register
0A	Write	Forced Test Mode Register
0B	Read/Write	Control Register 2
0C	Read/Write	Control Register 3 Rev. ID Register
0F	Write	Data Alignment Register

Not all registers in this device are both readable and writable. Some read only registers share the same address with write only registers. The registers can be accessed by asserting the  $\overline{CS}$  signal and then asserting either  $\overline{RD}$  or  $\overline{WR}$  signal depending on the operation to be performed. Only the FIFO Register can be accessed by asserting either  $\overline{CS}$  or  $\overline{DACK}$  in conjunction with  $\overline{RD}$  and  $\overline{WR}$  signals or  $\overline{DMARD}$  and  $\overline{DMAWR}$  signals. The register address inputs are ignored when  $\overline{DACK}$  is used but must be valid when  $\overline{CS}$  is used.

ADVANCED MICRO DEVICES

**COMMANDS**

The device commands can be broadly divided into two categories, DMA commands and non-DMA commands. DMA commands are those which cause data movement between the host memory and the SCSI bus while non-DMA commands are those that cause data movement between the device FIFO and the SCSI bus. The MSB of the command byte differentiates DMA commands from non-DMA commands.

**Summary of Commands**

Command	Command Code (Hex.)	
	Non-DMA Mode	DMA Mode
<b>Initiator Commands</b>		
Information Transfer	10	90
Initiator Command Complete Steps	11	91
Message Accepted	12	-
Transfer Pad Bytes	18	98
Set $\overline{ATN}$	1A	-
Reset $\overline{ATN}$	1B	-
<b>Target Commands</b>		
Send Message	20	A0
Send Status	21	A1
Send Data	22	A2
Disconnect Steps	23	A3
Terminate Steps	24	A4
Target Command Complete Steps	25	A5
Disconnect	27	A7
Receive Message	28	A8
Receive Command Steps	29	A9
Receive Data	2A	AA
Receive Command Steps	2B	AB
Target Abort DMA	04	84

Command	Command Code (Hex.)	
	Non-DMA Mode	DMA Mode
<b>Idle State Commands</b>		
Reselect Steps	40	C0
Select without $\overline{ATN}$ Steps	41	C1
Select with $\overline{ATN}$ Steps	42	C2
Select with $\overline{ATN}$ and Stop Steps	43	C3
Enable Selection/Reselection	44	C4
Disable Selection/Reselection		45
Select with $\overline{ATN3}$	46	C6
<b>General Commands</b>		
No Operation	00	80
Clear FIFO	01	81
Reset Device	02	82
Reset SCSI bus	03	83



### ABSOLUTE MAXIMUM RATINGS

Storage Temperature	-55 to +125°C
Ambient Operating Temperature	0 to +70°C
Maximum Vcc	-0.5 to +7.0 V
DC Voltage Applied to Any Pin	-0.5 to (Vcc + 0.3) V
Input Static Discharge Protection (Human body model: 100 pF at 1.5 K Ohms)	3000 V pin to pin

*Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.*

### OPERATING RANGES

#### Commercial (C) Devices

Ambient Temperature (T <sub>A</sub> )	0 to +70°C
Supply Voltage (Vcc)	+4.75 to +5.75 V

*Operating ranges define those limits between which the functionality of the device is guaranteed.*

### ADVANCED MICRO DEVICES

**DC CHARACTERISTICS over operating range unless otherwise specified**

Parameter Symbol	Parameter Description	Pin Names	Test Conditions	Min.	Max.	Unit
I <sub>CCS</sub>	Static Supply Current				8.0	mA
I <sub>CCD</sub>	Dynamic Supply Current				50	mA
I <sub>LU</sub>	Latch Up Current			-100	+100	mA

**SCSI Pins**

V <sub>IH</sub>	Input High Voltage	All SCSI Inputs		2.0		V
V <sub>IL</sub>	Input Low Voltage	All SCSI Inputs			0.8	V
V <sub>IHST</sub>	Input Hysteresis	All SCSI Inputs	4.75 V < V <sub>CC</sub> < 5.25 V	300		mV
V <sub>OH</sub>	Output High Voltage	SD 7-0, SDP	I <sub>OH</sub> = -2 mA	2.4	V <sub>CC</sub>	V
V <sub>SOL1</sub>	SCSI Output Low Voltage	SD 7-0, SDP	I <sub>OL</sub> = 4 mA	V <sub>SS</sub>	0.4	V
V <sub>SOL2</sub>	SCSI Output Low Voltage	SDC 7-0, SDCP, MSG, C/D, I/O, ATN, RSTC, SELC, BSYC, ACKC and REQC	I <sub>OL</sub> = 48 mA	V <sub>SS</sub>	0.5	V
I <sub>IL</sub>	Input Low Leakage	All SCSI Inputs	0.5 V < V <sub>IN</sub> < 2.7 V	-10	+10	μA
I <sub>IH</sub>	Input High Leakage	All SCSI Inputs	0.5 V < V <sub>IN</sub> < 2.7 V	-10	+10	μA
I <sub>OZ</sub>	High Impedance Leakage	All SCSI Inputs	0 V < V <sub>OUT</sub> < V <sub>CC</sub>	-10	+10	μA

**Bidirectional Pins**

V <sub>IH</sub>	Input High Voltage	DMA 15-0, DMAP 1-0 and AD 7-0		2.0	V <sub>CC</sub> + 0.5	V
V <sub>IL</sub>	Input Low Voltage	DMA 15-0, DMAP 1-0 and AD 7-0		V <sub>SS</sub> - 0.5	0.8	V
V <sub>OH</sub>	Output High Voltage	DMA 15-0, DMAP 1-0 and AD 7-0	I <sub>OH</sub> = -2 mA	2.4	V <sub>CC</sub>	V
V <sub>OL</sub>	Output Low Voltage	DMA 15-0, DMAP 1-0 and AD 7-0	I <sub>OL</sub> = 4 mA	V <sub>SS</sub>	0.4	V
I <sub>IL</sub>	Input Low Leakage	DMA 15-0, DMAP 1-0 and AD 7-0	V <sub>IN</sub> = V <sub>IL</sub>	-400	+10	μA
I <sub>IH</sub>	Input High Leakage	DMA 15-0, DMAP 1-0 and AD 7-0	V <sub>IN</sub> = V <sub>IH</sub>	-10	+10	μA
I <sub>OZ</sub>	High Impedance Leakage	DMA 15-0, DMAP 1-0 and AD 7-0		-100	400	μA

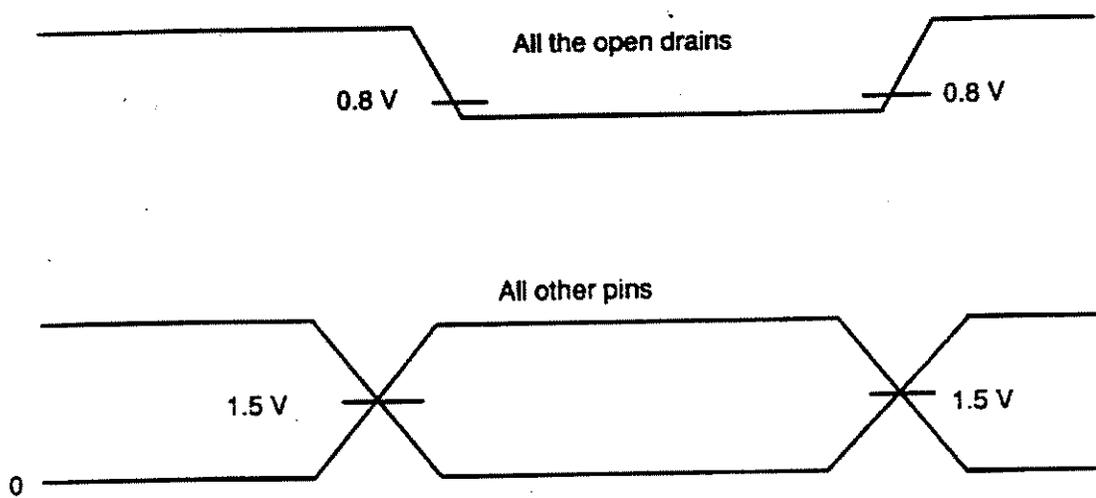
**Output Pins**

V <sub>OH</sub>	Output High Voltage	DRQ, ISEL, TSEL and INT	I <sub>OH</sub> = -2 mA	2.4	V <sub>CC</sub>	V
V <sub>OL</sub>	Output Low Voltage	DRQ, ISEL, TSEL and INT	I <sub>OL</sub> = 4 mA		0.4	V
I <sub>OZ</sub>	High Impedance Leakage	DRQ, ISEL, TSEL and INT	0 V < V <sub>OUT</sub> < V <sub>CC</sub>	-10	+10	μA

DC CHARACTERISTICS over operating range unless otherwise specified (Continued)

Parameter Symbol	Parameter Description	Pin Names	Test Conditions	Min.	Max.	Unit
<b>Input Pins</b>						
$V_{IH}$	Input High Voltage	A 3-0, $\overline{CS}$ , $\overline{RD}$ , $\overline{WR}$ , $\overline{DMAWR}$ , CLK, BUSMD 1-0, DACK, RESET, and DFMODE		2.0	$V_{CC} + 0.5$	V
$V_{IL}$	Input Low Voltage	A 3-0, $\overline{CS}$ , $\overline{RD}$ , $\overline{WR}$ , $\overline{DMAWR}$ , CLK, BUSMD 1-0, DACK, RESET, and DFMODE		$V_{SS} + 0.5$	0.8	V
$I_{IL}$	Input Low Leakage	A 3-0, $\overline{CS}$ , $\overline{RD}$ , $\overline{WR}$ , $\overline{DMAWR}$ , CLK, BUSMD 1-0, DACK, RESET, and DFMODE	$V_{IN} = V_{IL}$	-10	+10	$\mu A$
$I_{IH}$	Input High Leakage	A 3-0, $\overline{CS}$ , $\overline{RD}$ , $\overline{WR}$ , $\overline{DMAWR}$ , CLK, BUSMD 1-0, DACK, RESET, and DFMODE	$V_{IN} = V_{IH}$	-10	+10	$\mu A$

SWITCHING TEST WAVEFORMS



16506A-010A