

1. Publicação nº INPE-3802-NTI/253	2. Versão	3. Data Fevereiro 1986	5. Distribuição ☒ Interna ☐ Externa ☐ Restrita
4. Origem DCA/DEA	Programa SUBORD		
6. Palavras chaves - selecionadas pelo(s) autor(es) MICROCOMPUTADORES MICROPROCESSADORES DE 16 BITS ACESSO DIRETO À MEMÓRIA CONTROLADOR PERIFÉRICOS			
7. C.D.U.: 681.326.32			
8. Título UM CONTROLADOR DE ACESSO DIRETO À MEMÓRIA PARA O MTSB		10. Páginas: 54 11. Última página: A.18 12. Revisada por	
9. Autoria Ronaldo Luiz Dias Cereda		13. Autorizada por Ricardo Azevedo Mendes Marco Antonio Raupp Diretor Geral	
Assinatura responsável			
14. Resumo/Notas Apresenta-se um projeto de Controlador de Acesso Direto à Memória (CADM), visando sua utilização em sistemas de microcomputadores que utilizem processadores de 16 bits da linha Texas Instruments. O CADM provê dois canais para periféricos que manipulem palavras de 8 bits, e incorpora facilidades para implementação de "daisy-chain".			
15. Observações			

ABSTRACT

A Direct Access Memory Controller (CADM) is presented, aiming at its utilization in microcomputer systems that use Texas Instruments' 16-bit processors. The CADM provides two channels for 8-bit peripherals and includes facilities for "daisy-chain" implementation.

•
•

SUMÁRIO

	<u>Pág.</u>
<u>SEÇÃO 1 - INTRODUÇÃO</u>	1
<u>SEÇÃO 2 - ARQUITETURA E DESCRIÇÃO FUNCIONAL</u>	4
2.1 - Módulo de Interface com o Sistema	4
2.2 - Módulo de Programação das Funções	7
2.3 - Árbitro	8
2.4 - Controlador	9
2.5 - Módulo de Composição/Decomposição de palavras e interfaces com os Periféricos	10
2.5.1 - Interface com os Periféricos	10
2.5.2 - Composição/Decomposição das Palavras	12
<u>SEÇÃO 3 - REGISTROS INTERNOS</u>	17
<u>SEÇÃO 4 - CONFIGURAÇÃO E INTERFACEAMENTO</u>	21
<u>SEÇÃO 5 - OPERAÇÃO</u>	23
<u>SEÇÃO 6 - CONCLUSÕES E RECOMENDAÇÕES</u>	27
REFERÊNCIAS BIBLIOGRÁFICAS	29
BIBLIOGRAFIA COMPLEMENTAR	31
APÊNDICE A - INFORMAÇÕES TÉCNICAS	



1 - INTRODUÇÃO

O CADM - Controlador de Acesso Direto à Memória foi projetado, no Departamento de Engenharia de Computação em Aplicações Espaciais - DCA/INPE, para atender às necessidades de acesso direto à memória em sistemas que utilizem o barramento BPISB (Maldonado et alii, 1984), memória principal organizada em palavras de 16 bits e periféricos com barramentos de dados de 8 bits. Para tanto, além de realizar as funções características dos controladores similares, o CADM dispõe também de recursos internos que permitem a composição e decomposição das palavras de memória (16 bits) a partir de, ou em "bytes" (8 bits) dos ou para os periféricos, durante uma operação de acesso direto à memória (ADM).

O CADM foi concebido visando sua utilização nos projetos SUBORD e COMINC (nos computadores MTSB/V2 e ASTRO L/V2, respectivamente), que estão sendo desenvolvidos no DCA/INPE, mas pode ser aplicado em quaisquer outros sistemas que apresentem um ambiente com as características anteriormente citadas.

O CADM dispõe de dois canais simétricos, com barramento bem definido, para a interação com os dispositivos periféricos. As interfaces propiciadas por este barramento são compatíveis com o Controlador de Disquete para Simples ou Dupla Densidade - CDS/D (Missawa, 1985) e também com os controladores para Disco Rígido, da Multidigit. Qualquer outro dispositivo que adote interface semelhante poderá, naturalmente, ser utilizado. Cada canal do CADM pode ser individualmente ajustado para cada tipo específico de utilização (disco flexível ou disco rígido). O ajuste é feito através de "jumpers". A prioridade entre os canais é fixada internamente.

A interface entre o CADM e o restante do sistema de computador no qual está inserido é realizada através do barramento padrão BPISB, e toda a interação com a Unidade Central de Processamento estabelece-se de duas maneiras básicas: através da CRU (Communication Register Unit), durante a programação do CADM ou dos dispositivos periféricos; e através dos sinais do BPISB - HOLD e HOLDA, durante a realização do

acesso direto à memória.

O CADM apresenta ainda facilidades para sua inserção numa cadeia de prioridades do tipo "daisy-chain", através da utilização dos sinais GRANTIN e GRANTOUT do BPISB.

Tabela 1.

TABELA 1

ESPECIFICAÇÕES DO CADM

PARÂMETROS	ESPECIFICAÇÕES
Palavra de Memória	16 bits
Palavra de Dados dos Periféricos	8 bits
Número de Canais	2
Prioridade dos Canais	Fixa ou Rotativa
Frequência de Relógio Utilizada	$f_u = 2,457 \text{ MHz}$
Tempo de ADM para uma palavra (16 bits)	$1,627 \mu s (p/ f_u)$
Taxa máxima de Transferência (em f_u)	614K palavras/s
Tecnologia dos Componentes	TTL
Tensão de Alimentação	$+ 5V \pm 5\% \text{ Volts}$
Endereços de CRU utilizados	(ver Tabela 2)

Na Seção 2 apresenta-se a arquitetura do CADM e uma descrição funcional detalhada de cada uma de suas partes. Na Seção 3 apresentam-se as informações necessárias à programação do CADM, incluindo-se formatos e endereços dos registros internos. Na Seção 4 abordam-se aspectos da configuração e interfaceamento do CADM nos sistemas aos quais será integrado. Na Seção 5 descreve-se a sequência de ações necessárias para que o CADM possa operar corretamente. Finalmente, na Seção 6, apresentam-se as conclusões do trabalho, bem como algumas recomendações de utilização e aplicações futuras.

O Apêndice A, no final do documento, contém as especificações do circuito INTEL-8257 utilizado na implementação do CADM.

2 - ARQUITETURA E DESCRIÇÃO FUNCIONAL

Procura-se, nesta seção, fornecer elementos para auxiliar a compreensão do funcionamento global do CADM. O CADM compõe-se basicamente de 5 grandes módulos funcionais, conforme mostra o diagrama de blocos da Figura 1. Os módulos mostrados são:

- a) módulo de interface com o sistema (MIS),
- b) módulo de programação das funções (MPF),
- c) árbitro (ABT),
- d) módulo controlador (MC),
- e) módulo de composição/decomposição das palavras e interface com os periféricos (MCDP/IP).

A seguir, passa-se a descrever cada um destes módulos com maior grau de detalhes.

2.1 - MÓDULO DE INTERFACE COM O SISTEMA

Este módulo-interface é constituído de "buffers/drivers" para recepção e transmissão de sinais entre o CADM e o BPISB. Estes componentes foram utilizados em obediência às normas estabelecidas pelo próprio padrão BPISB (Maldonado et alii, 1984).

Os sinais do BPISB que se relacionam com este módulo são detalhados na Seção 4.

É importante notar que o barramento de dados do sistema (16 bits), após transitar através deste módulo, é levado diretamente ao módulo de composição/decomposição de palavras, sem ter, portanto, qualquer interação com os outros módulos do CADM.

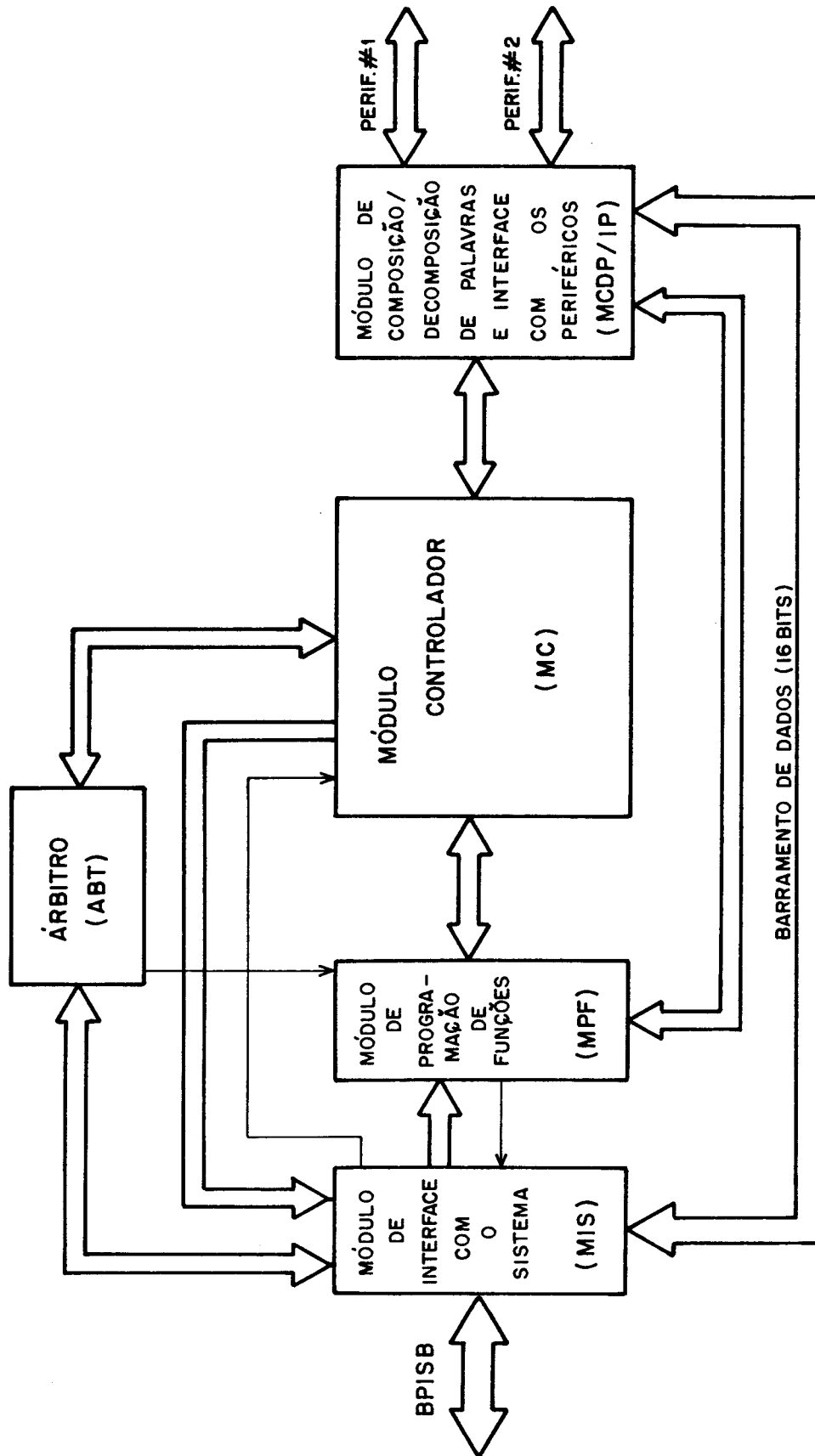


Fig. 1 - Diagrama de blocos do CADM

2.2 - MÓDULO DE PROGRAMAÇÃO DAS FUNÇÕES

Este é o módulo através do qual se programam todas as funções internas ao CADM e por onde se faz a programação dos periféricos ligados aos canais do CADM. Este módulo recebe, do MIS, todos os sinais relativos à CRU, além de alguns bits do barramento de endereços necessários às operações de entrada/saída serial realizadas através da CRU.

Um diagrama de blocos específico deste módulo é mostrado na Figura 2.

De maneira geral, este módulo é composto de um conjunto de registros - acessáveis através da CRU - que realizam basicamente dois tipos de funções:

- a) Paralelização/Serialização de dados: Os dados seriais provenientes da CRU são paralelizados e transformados em palavras de 8 bits destinadas à programação do "Módulo Controlador" do CADM ou à programação dos dispositivos periféricos. De forma inversa, as palavras de "status" provenientes do MC, ou dos dispositivos periféricos, são serializadas para a leitura através da CRU.
- b) Programação de funções internas: Neste caso, um conjunto de registros, acessáveis apenas para escrita, geram os sinais necessários à programação (ou leitura de "status") do MC do CADM e dos dispositivos periféricos. Além destes, há ainda os registros que contêm as informações destinadas ao MCDP e que devem igualmente ser programados antes do início de uma operação de acesso direto à memória.

Todas as informações necessárias à manipulação destes registros, tais como: finalidade, formatos, endereços e conteúdos, podem ser encontradas na Seção 3. Na Seção 5, são fornecidas informações complementares relativas à temporização e seqüenciamento de ações durante a utilização destes registros.

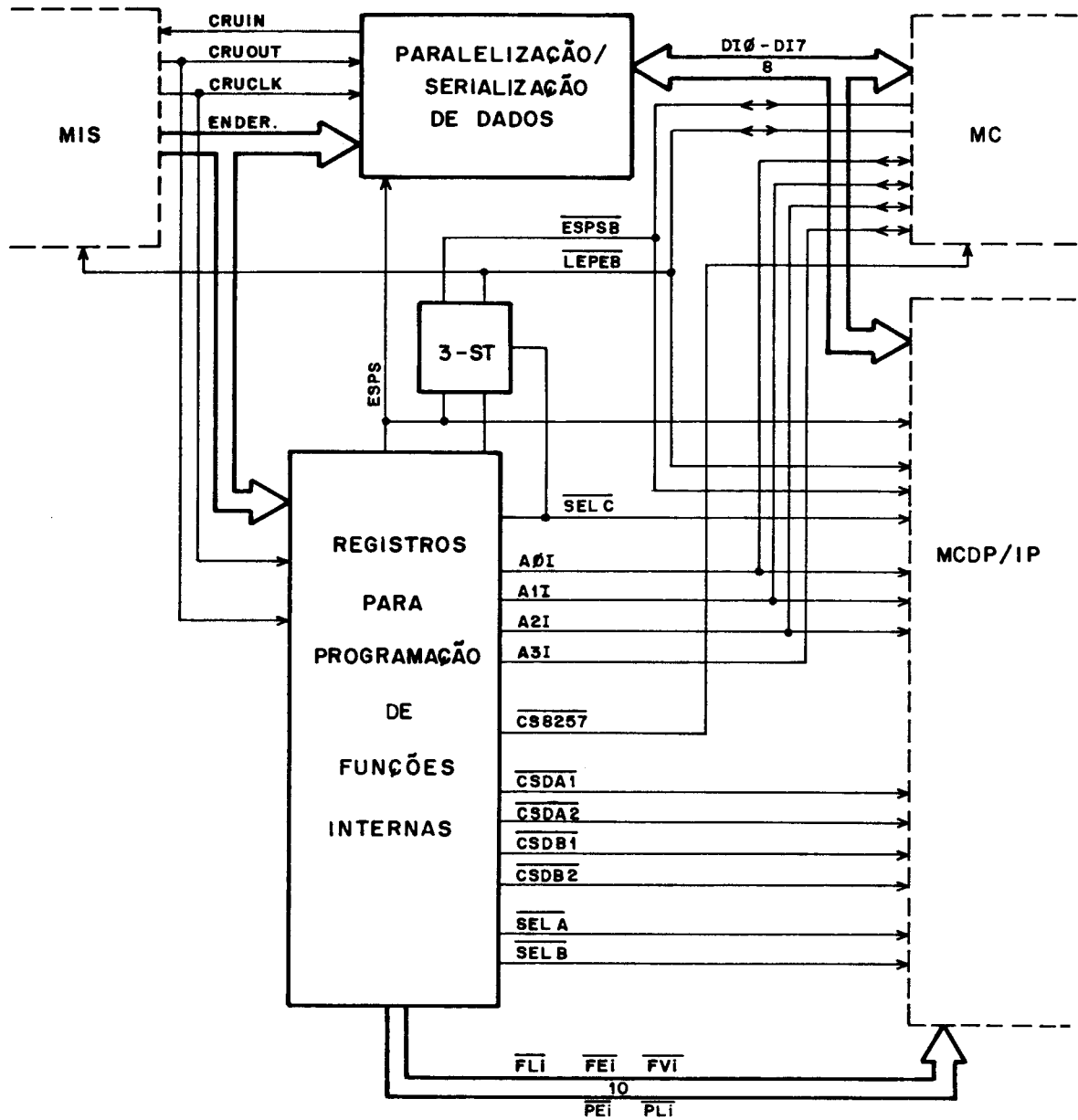


Fig. 2 - Módulo de programação das funções

2.3 - ÁRBITRO

Este módulo permite a inserção do CADM em um esquema de prioridades do tipo "daisy-chain", através da utilização dos sinais GRANTIN e GRANTOUT do BPISB. Além destes dois sinais, o árbitro necessita ainda de uma linha com características especiais, que deverá, a princípio, ser implementada utilizando uma das linhas do *barramento de propósitos gerais* (linhas de usuário) do BPISB. Este sinal deverá ser bidirecional, implementado em "open-collector" e receber o sinal HOLDA oriundo da UCP do sistema, porém com polaridade invertida (ativo alto). Este sinal será, no restante deste documento, referenciado como HOLDA*.

O árbitro recebe ainda, do MC, o sinal HRQI ("Hold Request" Interno) e o sinal de relógio ("CLOCK").

Em linhas gerais, o funcionamento do árbitro se dá como se segue:

- a) Supondo que a entrada GRANTIN esteja inicialmente com nível lógico "0": Ao receber a solicitação do controle do barramento (HRQI) proveniente do MC, o árbitro passa adiante a informação de solicitação através do sinal GRANTOUT (o qual, no final da cadeia de prioridades, deverá estar conectado à entrada HOLD do controlador mestre do sistema). Ao receber o reconhecimento do pedido feito através do sinal HOLDA* (assumindo o valor lógico "1"), o árbitro ativa o sinal de reconhecimento interno (HOLDAI), que é utilizado por todos os outros blocos que compõem o CADM, e imediatamente coloca em nível lógico "0" ("prende") o sinal HOLDA* do barramento.

Com a suspensão do pedido de HRQI, o sinal HOLDAI é desativado imediatamente e o sinal HOLDA* do barramento é "solto". Também o sinal GRANTOUT é desativado em consequência, a menos que haja alguma solicitação pendente de maior prioridade.

Durante a operação acima descrita, qualquer dispositivo da cadeia com menor prioridade fica impossibilitado de ganhar o

acesso ao barramento, inibido pela propagação do sinal GRANTOUT.

Qualquer dispositivo de maior prioridade que tente o acesso HOLDA*, quando então assumirá imediatamente o controle. Em outras palavras, não é permitido ao dispositivo mais prioritário interromper um acesso que esteja sendo realizado por um dispositivo de menor prioridade da cadeia.

- b) Supondo que a entrada GRANTIN esteja inicialmente com nível lógico "1": Neste caso o CADM em questão está impossibilitado de solicitar o controle do barramento, enquanto este valor permanecer indicando que um dispositivo mais prioritário já requereu o controle. Entretanto, o sinal de solicitação interno (HRQI) deverá ser mantido ativado até que a oportunidade de controle surja com a queda do nível lógico em GRANTIN.

Durante a permanência do nível alto em GRANTIN, o árbitro garante a propagação deste sinal pelos demais dispositivos da cadeia.

2.4 - CONTROLADOR

Este módulo é o responsável pela geração dos sinais de controle durante a realização de um ciclo de acesso direto à memória. É ele que recebe as solicitações de acesso vindas dos periféricos (através do MCDP/IP), testa se o canal está ou não habilitado, verifica as prioridades dos pedidos pendentes, envia a solicitação de controle do barramento (através do ABT) e, a partir do recebimento do reconhecimento do pedido feito, passa a gerar todos os sinais de controle necessários à realização do acesso direto à memória.

O MC é inteiramente projetado a partir do circuito INTEL-8257 (Programmable DMA Controller). Além deste circuito, este módulo compõe-se ainda de um registro ("latch") auxiliar, conforme recomendado pelo próprio fabricante, e de um oscilador a cristal que for

nece a frequência de relógio ($f=2,457$ MHz) necessária ao funcionamento do 8257.

No Apêndice A são fornecidas as especificações do circuito INTEL-8257 geradas pelo fabricante. A leitura destas especificações é imprescindível à utilização correta do CADM.

Como dito anteriormente, toda a programação (ou leitura da palavra de "status") do MC se faz através do MPF via CRU. Para a efetivação desta programação de forma adequada recomenda-se, além da leitura do Apêndice A, a observação das Seções 3 e 5 deste documento.

2.5 - MÓDULO DE COMPOSIÇÃO/DECOMPOSIÇÃO DE PALAVRAS E INTERFACE COM OS PERIFÉRICOS

Este módulo realiza duas funções básicas e intimamente relacionadas entre si:

- a) interfaceamento entre o CADM e os sinais provenientes ou destinados aos periféricos;
- b) composição das palavras de memória (16 bits) a partir de dois "bytes" enviados pelo periférico - durante um ciclo de escrita por ADM - ou decomposição das palavras provenientes da memória em dois "bytes" a serem enviados aos periféricos - durante um ciclo de leitura por ADM.

Para facilitar o entendimento das funções e acrescentar maior clareza ao relato, são abordados separadamente os submódulos relacionados com cada uma destas funções básicas.

2.5.1 - INTERFACE COM OS PERIFÉRICOS

Este submódulo é constituído por um conjunto de "buffers/drivers" destinados à recepção e à transmissão de sinais entre o CADM e cada um dos periféricos que se ligam a seus canais. Foi estabelecido um barramento bem definido para esta interação, compatível com

os dispositivos periféricos mais comumente utilizados no DCA/INPE, cujos sinais são detalhados na Seção 4.

Este submódulo também é constituído de um circuito expensor do pulso do sinal de "reset" a ser enviado aos periféricos (sinais MRT e MR2) durante a inicialização do sistema. Estes sinais são gerados a partir de um multivibrador monoestável disparado pelo sinal RESET/POWERON do BPISB. Desta maneira a largura dos pulsos de MRT e MR2 pode ser ajustada através de um potenciômetro, de forma a atender as necessidades dos dispositivos periféricos utilizados.

O submódulo é constituído ainda de uma lógica destinada a geração automática dos endereços e dos sinais das linhas de seleção para os periféricos, durante a realização das operações de leitura/escrita dos seus registros de dados internos. Esta lógica atua da seguinte maneira: o barramento presente em cada canal do CADM possui três linhas de endereços (AD0, AD1 e AD2) que especificam quais os registros internos aos dispositivos periféricos que devem ser acessados a cada momento. Durante uma operação qualquer de transferência de dados de ou para os periféricos, o registro interno a ser acessado é o "registro de dados". Caso a transferência dos dados esteja sendo feita por ADM, cabe ao CADM a geração automática dos endereços deste registro, bem como a do sinal de seleção para os periféricos. Esta é a função realizada pela lógica em questão. Como este endereço difere para cada tipo de aplicação (AD0=AD1=AD2=0 para disco rígido; AD0=AD1=1 para disquete, AD2 não é utilizado), utiliza-se na geração dos endereços as informações fixadas pelos "jumpers" associados a cada um dos canais, os quais indicam o tipo de dispositivo periférico que está sendo utilizado. Deve-se lembrar que, para o caso de utilização de disquete, a geração automática de endereços é realizada internamente ao próprio CDS/D (Missawa, 1985), não sendo portanto necessária a geração pelo CADM.

2.5.2 - COMPOSIÇÃO/DECOMPOSIÇÃO DAS PALAVRAS

Como mencionado anteriormente, o CADM foi projetado para realizar acesso direto à memória em sistemas que utilizem memórias com palavras de 16 bits e periféricos com palavras de 8 bits. Desta maneira é necessária a composição ou a decomposição das palavras de memória a cada ciclo de ADM.

É muito importante notar que, por imposição de projeto, o CADM só solicita o controle do barramento do sistema (ou seja só efetiva o acesso direto à memória) uma vez a cada palavra de 16 bits (ou a cada dois "bytes" dos periféricos). Com isto consegue-se evitar paralizações desnecessárias na UCP do sistema.

Assim, num ciclo de escrita na memória, o primeiro "byte" vindo do periférico é armazenado internamente ao CADM e nenhuma solicitação ("DMA Request") é feita ao controlador; ao chegar o segundo "byte" oriundo do periférico, a palavra é composta - a partir deste "byte" e do anterior armazenado - e, com o envio da solicitação de ADM ao controlador, o acesso é consumado normalmente.

De forma semelhante, durante um ciclo de leitura da memória, a palavra completa (16 bits) é acessada numa operação normal de ADM, porém somente um "byte" desta palavra é transmitido ao periférico; o outro "byte" fica armazenado internamente ao CADM e só é transmitido quando for recebida uma segunda solicitação vinda do periférico. Nesta segunda transferência nenhuma solicitação é feita ao controlador.

Fica evidente então que o submódulo de composição / decomposição de palavras é constituído de duas partes básicas:

- a) os registros auxiliares para a composição ou decomposição das palavras, que armazenam os "bytes" quando necessário, conforme o tipo de acesso que está sendo realizado;
- b) o controle do canal, que controla a efetivação dos pedidos de

acesso ao "controlador", como explicado acima; simula os sinais para os periféricos quando estes pedidos não são efetivados, e seleciona os sinais a serem enviados ao periférico em cada caso.

Toda a lógica deste submódulo de composição/decomposição de palavras aparece duplicada no CADM, sendo cada réplica pertencente a cada um dos canais. A Figura 3 ilustra este submódulo.

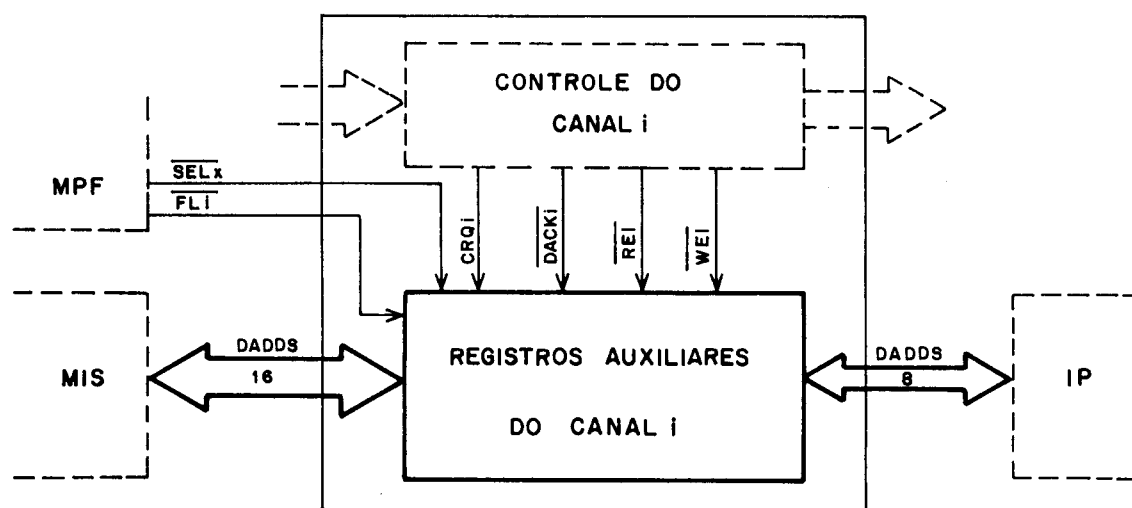


Fig. 3 - Submódulo de composição/decomposição de palavras

O bloco "controle do canal i" é detalhado na Figura 4. A seguir, explicam-se particularidades de seu funcionamento.

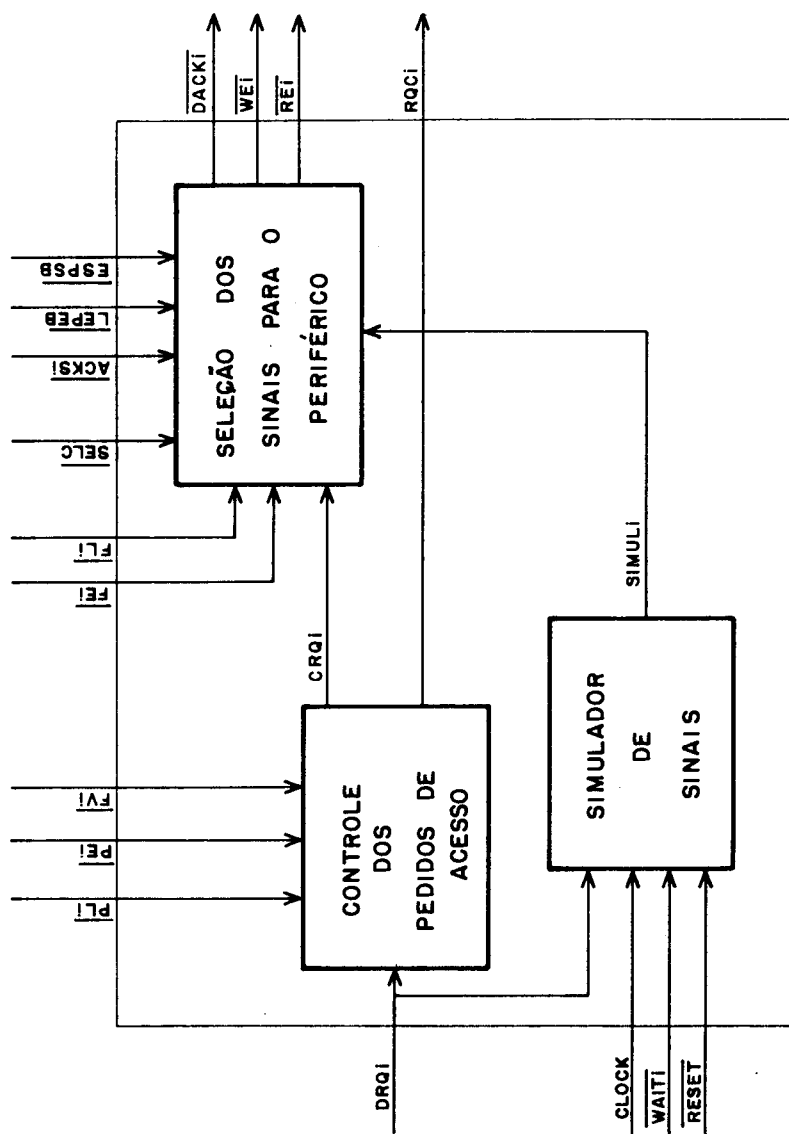


Fig. 4 - Controle do canal i

Este bloco recebe do MPF as informações sobre o tipo de acesso direto à memória que está sendo realizado - leitura ou escrita - através dos "flags" $\overline{FLI}$ ou $\overline{FEI}$, respectivamente. Os pulsos $\overline{PLI}$ e $\overline{PEI}$, provenientes do mesmo módulo, ocorrem automaticamente durante a programação destes "flags" e determinam a condição inicial do "control pedidos de acesso". Há uma terceira situação correspondente ao "Verify DMA Cycle" (ver Apêndice A), que é sinalizada pelo "flag" $\overline{FVI}$ e provoca a efetivação do ADM a cada solicitação chegada do periférico (DRQi), o que ocorre naturalmente a cada "byte" transferido.

Ao receber uma solicitação de acesso proveniente do periférico (DRQi), o "controle dos pedidos de acesso" passa ou não esta solicitação ao MC através do sinal RQCi. A decisão de passar/não passar esta solicitação é realizada alternadamente a partir da condição inicial estabelecida em cada caso (leitura ou escrita na memória), conforme discutido no início desta seção.

Quando a solicitação de acesso é passada ao controlador, os sinais de controle dos periféricos ($\overline{DACKi}$, $\overline{WEi}$ e $\overline{REi}$) originam-se deste MC (através de $\overline{ACKSi}$, $\overline{ESPSB}$ e $\overline{LEPEB}$ respectivamente) e quando não, devem ser simulados. Esta simulação é realizada pelo "simulador de sinais" que produz uma única saída (SIMULi), a qual é utilizada em cada caso seguindo a orientação estabelecida pelos "flags" $\overline{FLI}$ e $\overline{FEI}$.

A "seleção dos sinais para os periféricos" é comandada pelo sinal CRQi, originário do "controle dos pedidos de acesso", que estabelece a origem dos sinais a cada instante: oriundos do MC ou oriundos do "simulador".

Há ainda a situação em que os periféricos estão sendo programados (ou sendo feita a leitura de "status") pela UCP do sistema, através da CRU. Neste caso, sob o controle do sinal $\overline{SELC}$, do MPF, os sinais $\overline{ESPSB}$ e $\overline{LEPEB}$ passam a ser gerados também a partir deste MPF e selecionados (por $\overline{SELC}$) para envio aos periféricos através de $\overline{WEi}$ e $\overline{REi}$, respectivamente. Nesta situação deverá ser utilizado também o sinal $\overline{CSDx1}$ (gerado a partir do registro CSDx1 do MPF), conforme discutido na Seção 5 deste documento.

Finalizando, é importante notar que, como o acesso direto à memória só é efetivado a cada 2 "bytes" (ou uma palavra de memoria), o incremento dos "DMA Address Registers" (ver Apêndice A) internos ao MC deverá ser realizado também somando o valor 2 ao conteúdo anterior. Este efeito é conseguido deslocando de uma posição as linhas de endereços provenientes do circuito 8257 em relação ao barramento de endereços do sistema. Este deslocamento tem conseqüências diretas sobre a programação dos "DMA Address Registers", conforme discutido na Seção 5 deste documento.

3 - REGISTROS INTERNOS

Nesta seção procura-se dar todas as informações necessárias à manipulação dos registros internos ao CADM. Estes registros internos estão distribuídos por dois módulos distintos, o MC e o MPF, e devem ser programados sempre antes do início de qualquer operação de ADM.

Os registros internos ao MC são especificados no Apêndice A. A seguir passa-se às informações relativas aos registros internos ao MPF. A Tabela 2 sintetiza estas informações.

A coluna "endereços" da Tabela 2 especifica os endereços-base de CRU dos registros correspondentes, expressos em hexadecimal. Supõe-se que, nas operações por CRU, o CADM receba, da UCP do sistema, um sinal de seleção (DECADM) resultante da decodificação dos bits mais significativos dos endereços de CRU (A2-A7). Na coluna "endereços" especifica-se somente a parte do endereço-base relativa aos bits menos significativos (A8-A14), ou seja, o "byte" menos significativo do endereço-base de "software" de CRU.

Obs.: Os sinais SELA, SELB e SELC são funções das saídas destes registros, conforme explicitado abaixo:

SELA=CSDAT.CSDA2

SELB=CSDBT.CSDB2

SELC=SELA.SELB.CS8257

Deve-se ressaltar aqui, uma vez mais, que toda a função de programação (ou leitura de "status") do MC ou dos dispositivos periféricos é realizada através dos registros do MPF. A orientação para esta programação é dada na Seção 5, levando em conta as informações da Tabela 2.

Também é importante notar que todos os registros do MPF, com exceção de RD, RC1 e RC2, são apagados automaticamente durante a realização de um ciclo de ADM. Deve-se lembrar que, nestas situações, os endereços para os periféricos, bem como os seus sinais de seleção, também são automaticamente gerados (ver Seção 2.5.1).

TABELA 2

REGISTROS INTERNOS AO MPF

NOME DO REGISTRO	Nº DE BITS	ENDEREÇOS	FUNÇÃO
Registro de Dados (RD)	8	00	Registro que contém os dados a serem enviados ao MC ou aos periféricos durante sua programação, ou os dados recebidos do MC ou dos periféricos, durante uma leitura de "status".
Registro de Endereços (RE)	4	10	Suas saídas (A0I, A1I, A2I e A3I) são utilizadas para o endereçamento dos registros internos do MC ou dos dispositivos periféricos.
Registro de Controle do Canal 1 (RC1)	2	40	Suas saídas são decodificadas gerando sinais para o MCDP/IP. (ver Tabela 3).
Registro de Controle do Canal 2 (RC2)	2	50	(Idem ao anterior)
LEPE	1	18	Produce o sinal de controle de leitura utilizado na leitura de "status" do MC ou dos periféricos.
ESPS	1	1A	Produce o sinal de controle de escrita utilizado na programação do MC ou dos periféricos.
CS8257	1	1C	Produce o sinal de seleção ("Chip Select") para o 8257 (MC).
CSDA1	1	1E	Produce o sinal de seleção para o periférico do canal 1.

(continua)

Tabela 2 - Conclusão

NOME DO REGISTRO	Nº DE BITS	ENDEREÇOS	FUNÇÃO
CSDA2	1	20	Produz o segundo sinal de seleção para o periférico do canal 1.
CSDB1	1	22	Produz o sinal de seleção para o periférico do canal 2.
CSDB2	1	24	Produz o segundo sinal de seleção para o periférico do canal 2.

Os conteúdos dos registros RC1 e RC2 são decodificados, gerando os "flags" $\overline{FLI}$, $\overline{FEI}$ e $\overline{FVI}$ utilizados pelo MCDP/IP, conforme Seção 2.5.2.. Os pulsos $\overline{PEI}$ e $\overline{PLI}$ são gerados automaticamente durante a programação destes registros. Seus conteúdos especificam o tipo de acesso por ADM a ser realizado pelo canal correspondente e devem obedecer à Tabela 3.

TABELA 3

CONTEÚDO DOS REGISTROS RCi

CONTEÚDO		TIPO DE ADM
0	0	(ilegal)
0	1	Escrita na memória
1	0	Leitura da memória
1	1	"Verify"

Os registros associados a sinais de controle ou seleção (LEPE, ESPS, CS8257, CSDA1, CSDA2, CSDB1 e CSDB2) devem ser carregados com o valor "1" quando se quer ativar o sinal correspondente. Assim, por exemplo, quando o registro CS8257 tem o valor "1", o sinal de seleção ~~CS8257~~ está ativo (nível baixo).

4 - CONFIGURAÇÃO E INTERFACEAMENTO

O CADM quando integrado a um sistema de computador estará sempre localizado entre o barramento BPISB ("backplane" do sistema) e os controladores dos periféricos que o sistema utilizar, como ilustra a Figura 5.

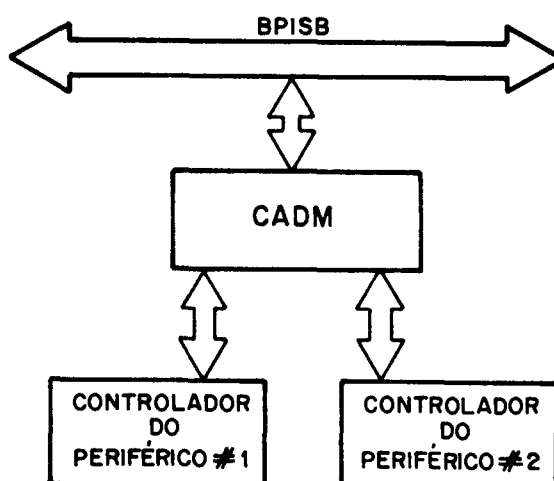


Fig. 5 - Aplicação típica do CADM.

Um número qualquer de CADMs pode ser utilizado com a implementação de uma cadeia de prioridades do tipo "daisy-chain".

O ambiente de utilização mostrado na Figura 5 define dois tipos de interfaces: interface com o barramento BPISB e interface com os barramentos dos periféricos.

No primeiro tipo de interação, a interface com o BPISB, estão compreendidos os seguintes sinais: barramento de dados (D0-D15); barramento de endereços (A0-A14); sinais relativos à CRU (CRUIN, CRUOUT, CRUCLK e TCRUCLK) que é por onde se estabelecem as operações de E/S controladas pela UCP do sistema; sinais de controle de memória (MEMEN, DBIN e WE) utilizados pelo CADM para a realização do ADM; o sinal

RESET/POWERON, utilizado para inicialização; os sinais $\overline{\text{GRANTIN}}$ e $\overline{\text{GRANTOUT}}$ utilizados pelo ABT na implementação da cadeia "daisy-chain"; e os sinais HOLDA* e $\overline{\text{DECADM}}$ que devem ser implementados nas linhas de usuário, segundo as especificações definidas nas Seções 2.3 e 3, respectivamente. Existe ainda a possibilidade de utilização de duas linhas de interrupções (INTi) que seriam conectadas às linhas INTRQi existentes nos barramentos dos periféricos.

O CADM utiliza também as linhas VCC e TERRA do barramento de alimentação do BPISB.

No que diz respeito aos barramentos dos periféricos, procurou-se estabelecer um padrão de interface bem definido, compatível com os dispositivos periféricos mais comumente utilizados no DCA/INPE. Estes barramentos são física e funcionalmente idênticos para os dois canais, e abrigam os seguintes sinais: barramento de dados (DALx0-DALx7); linhas de endereços (AD0 , AD1 e AD2); linhas de seleção ($\overline{\text{CSi}}$ e $\overline{\text{CSDx2}}$), sendo a primeira ativada com a ativação de $\overline{\text{CSDx1}}$ ou automaticamente durante o ADM, e a segunda gerada a partir do registro correspondente, do MPF; sinais $\overline{\text{WEi}}$ e $\overline{\text{REi}}$, para o controle da escrita ou leitura no periférico, respectivamente; sinal $\overline{\text{MRi}}$, de "reset"; sinal DRQi de solicitação de transferência de dados; sinal $\overline{\text{DACKi}}$ de reconhecimento à solicitação; sinal $\overline{\text{WAITi}}$ de pedido de espera ao CADM; e o sinal INTRQi de pedido de interrupção.

A conexão física do CADM com os dispositivos periféricos é feita por cabos flexíveis ("flat-cable"), utilizando conectores de 50 pinos.

5 - OPERAÇÃO

Para que se estabeleça o início da operação do CADM, algumas ações manuais devem ser efetuadas.

Inicialmente deverão ser ajustados os "jumpers" relativos a cada canal, para especificar o tipo de aplicação desejado: disco rígido ou flexível. Seu posicionamento deve ser orientado como mostra a Figura 6.

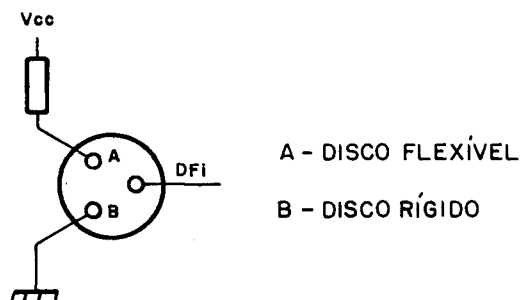


Fig. 6 - Posicionamento dos "jumpers".

Também deverá ser ajustada a largura dos pulsos de "reset" (sinais MRT e MRZ) para os periféricos, através do potenciômetro existente no MCDP/IP, levando em conta as exigências de cada aplicação.

Uma outra providência que deverá ser tomada antes do início da operação é o ajuste (quando possível) do pedido de espera (WAIT) de cada periférico. Além de atender às exigências particulares do periférico, este pedido de espera deverá ser utilizado, sempre que necessário, para evitar a ocorrência indesejável de pedidos consecutivos de acesso, devido à permanência de DRQi em S4 (ver "DMA Operation", Apêndice A).

Tendo sido efetuadas estas ações manuais, deverão ser programados o CADM e os dispositivos periféricos, de acordo com a aplicação desejada. Para a programação do MC do CADM é imprescindível o conhecimento das especificações contidas no Apêndice A.

Como citado anteriormente, a programação tanto do CADM quanto dos dispositivos periféricos é realizada por CRU, utilizando, portanto, as instruções correspondentes: LDCR, SBO e SBZ. Para a leitura de palavras de "status" utilizam-se STCR ou TB.

Todas as operações de programação ou leitura de palavras de status são feitas através do MPF, conforme mencionado anteriormente. Na Seção 3 os registros internos a este módulo estão relacionados, juntamente com todas as informações necessárias à sua utilização.

Para a programação do MC ou dos dispositivos periféricos a seguinte sequência comum de operações deverá ser efetuada a cada vez, e na ordem estabelecida:

- 1) carga da palavra de comando em RD,
- 2) carga do endereço relativo ao registro desejado em RE,
- 3) ativação do sinal de seleção correspondente,
- 4) ativação do sinal de controle de escrita,
- 5) desativação do sinal de controle,
- 6) desativação do sinal de seleção.

Analogamente, numa leitura de palavras de status a sequência será:

- 1) carga do endereço do registro desejado em RE,
- 2) ativação do sinal de seleção correspondente,
- 3) ativação do sinal de controle de leitura,
- 4) leitura da palavra em RD,
- 5) desativação do sinal de controle,

6) desativação do sinal de seleção.

Os ciclos de instrução dos processadores utilizados ga
rante a temporização exigida para estas operações.

Finalizando, chama-se a atenção para uma particularidade
de relativa à carga dos "DMA Address Registers" do 8257 no MC. Para
conseguir o efeito de incremento de 2 a cada ADM (ver Seção 2.5.2), os
endereços iniciais de memória devem ser carregados deslocados um bit à
direita, ou seja, o bit A14 do endereço deve corresponder à posição men
os significativa dos registros, e a posição mais significativa não deve
ser utilizada.

6 - CONCLUSÕES E RECOMENDAÇÕES

O CADM foi projetado, como dito na Seção 1, para operar em sistemas com memórias organizadas em palavras de 16 bits e periféricos com palavras de 8 bits. Seu circuito, no entanto, poderia ser reestruturado para operação com periféricos que manipulem palavras de 16 bits. Com exceção das modificações resultantes das necessidades de ordem física - expansão do campo de dados dos barramentos dos periféricos - por exemplo, as alterações no circuito limitariam-se à supressão do submódulo de composição/decomposição de palavras (Seção 2.5.2).

Um outro aspecto que deve ser notado é o do "overhead" causado na programação do CADM, devido à utilização da CRU como único meio de entrada/saída controlada pela UCP do sistema. Preferiu-se este enfoque ao uso de memória mapeada para não comprometer o espaçamento de memória do sistema, uma vez que para um sistema de testes como o MTSB a disponibilidade de memória é um parâmetro muito importante. Deve-se observar, contudo, que como a programação do CADM e dos periféricos só ocorre uma vez no início de cada bloco de palavras transferido por ADM, este "overhead" tende a ser insignificante.

REFERÊNCIAS BIBLIOGRÁFICAS

MALDONADO, J. C.; CRUZ, M. A. C.; MISSAWA, M.; CEREDA, R. L. D.;
"BPISB - Padronização de um Barramento para Microcomputadores"
São José dos Campos, INPE-3258-NTI/217 - 1984.

MISSAWA, M.; *"Um Controlador de Disquete para Simplex ou Dupla
Densidade - CDS/D"* São José dos Campos, INPE (Publicação res-
trita ao DCA/INPE) - 1985.

BIBLIOGRAFIA COMPLEMENTAR

TEXAS INSTRUMENTS; *"9900 family systems design and data book"*.
Dalas, Texas, c1978.

APÊNDICE A

INFORMAÇÕES TÉCNICAS



8257/8257-5 PROGRAMMABLE DMA CONTROLLER

- MCS-85® Compatible 8257-5
- 4-Channel DMA Controller
- Priority DMA Request Logic
- Channel Inhibit Logic
- Terminal Count and Modulo 128 Outputs
- Single TTL Clock
- Single + 5V Supply
- Auto Load Mode
- *± 81 Kbytes/sec.*

The Intel® 8257 is a 4-channel direct memory access (DMA) controller. It is specifically designed to simplify the transfer of data at high speeds for the Intel® microcomputer systems. Its primary function is to generate, upon a peripheral request, a sequential memory address which will allow the peripheral to read or write data directly to or from memory. Acquisition of the system bus is accomplished via the CPU's hold function. The 8257 has priority logic that resolves the peripherals requests and issues a composite hold request to the CPU. It maintains the DMA cycle count for each channel and outputs a control signal to notify the peripheral that the programmed number of DMA cycles is complete. Other output control signals simplify sector data transfers. The 8257 represents a significant savings in component count for DMA-based microcomputer systems and greatly simplifies the transfer of data at high speed between peripherals and memories.

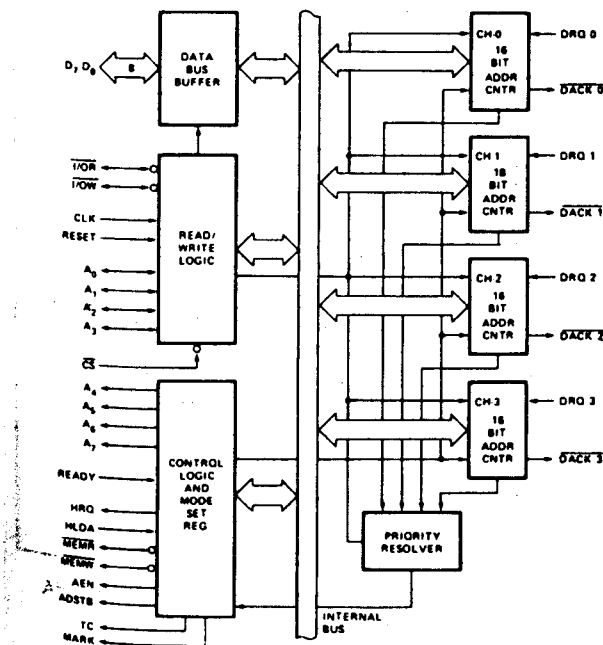


Figure 1. Block Diagram

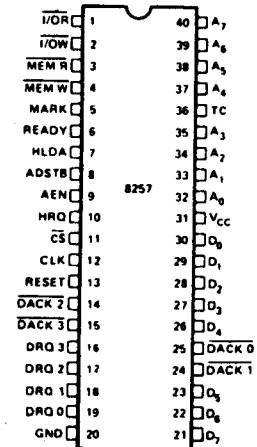


Figure 2. Pin Configuration



8257/8257-5

FUNCTIONAL DESCRIPTION

General

The 8257 is a programmable, Direct Memory Access (DMA) device which, when coupled with a single Intel® 8212 I/O port device, provides a complete four-channel DMA controller for use in Intel® microcomputer systems. After being initialized by software, the 8257 can transfer a block of data, containing up to 16,384 bytes, between memory and a peripheral device directly, without further intervention required of the CPU. Upon receiving a DMA transfer request from an enabled peripheral, the 8257:

1. Acquires control of the system bus.
2. Acknowledges that requesting peripheral which is connected to the highest priority channel.
3. Outputs the least significant eight bits of the memory address onto system address lines A_0-A_7 , outputs the most significant eight bits of the memory address to the 8212 I/O port via the data bus (the 8212 places these address bits on lines A_8-A_{15}), and
4. Generates the appropriate memory and I/O read/write control signals that cause the peripheral to receive or deposit a data byte directly from or to the addressed location in memory.

The 8257 will retain control of the system bus and repeat the transfer sequence, as long as a peripheral maintains its DMA request. Thus, the 8257 can transfer a block of data to/from a high speed peripheral (e.g., a sector of data on a floppy disk) in a single "burst". When the specified number of data bytes have been transferred, the 8257 activates its Terminal Count (TC) output, informing the CPU that the operation is complete.

The 8257 offers three different modes of operation: (1) DMA read, which causes data to be transferred from memory to a peripheral; (2) DMA write, which causes data to be transferred from a peripheral to memory; and (3) DMA verify, which does not actually involve the transfer of data. When an 8257 channel is in the DMA verify mode, it will respond the same as described for transfer operations, except that no memory or I/O read/write control signals will be generated, thus preventing the transfer of data. The 8257, however, will gain control of the system bus and will acknowledge the peripheral's DMA request for each DMA cycle. The peripheral can use these acknowledge signals to enable an internal access of each byte of a data block in order to execute some verification procedure, such as the accumulation of a CRC (Cyclic Redundancy Code) checkword. For example, a block of DMA verify cycles might follow a block of DMA read cycles (memory to peripheral) to allow the peripheral to verify its newly acquired data.

Block Diagram Description

1. DMA Channels

The 8257 provides four separate DMA channels (labeled CH-0 to CH-3). Each channel includes two sixteen-bit registers: (1) a DMA address register, and (2) a terminal count register. Both registers must be initialized before a channel is enabled. The DMA address register is loaded with the address of the first memory location to be accessed. The value loaded into the low-order 14-bits of the terminal count register specifies the number of DMA cycles minus one before the Terminal Count (TC) output is activated. For instance, a terminal count of 0 would cause the TC output to be active in the first DMA cycle for that channel. In general, if N = the number of desired DMA cycles, load the value $N-1$ into the low-order 14-bits of the terminal count register. The most significant two bits of the terminal count register specify the type of DMA operation for that channel.

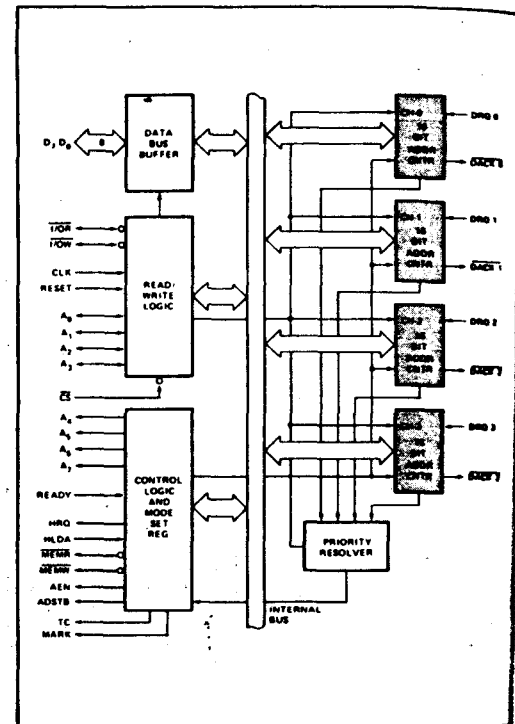


Figure 3. 8257 Block Diagram Showing DMA Channels



8257/8257-5

These two bits are not modified during a DMA cycle, but can be changed between DMA blocks.

Each channel accepts a DMA Request (DRQn) input and provides a DMA Acknowledge (DACKn) output.

(DRQ 0-DRQ 3)

DMA Request: These are individual asynchronous channel request inputs used by the peripherals to obtain a DMA cycle. If not in the rotating priority mode then DRQ 0 has the highest priority and DRQ 3 has the lowest. A request can be generated by raising the request line and holding it high until DMA acknowledge. For multiple DMA cycles (Burst Mode) the request line is held high until the DMA acknowledge of the last cycle arrives.

(DACK 0 - DACK 3)

DMA Acknowledge: An active low level on the acknowledge output informs the peripheral connected to that channel that it has been selected for a DMA cycle. The DACK output acts as a "chip select" for the peripheral device requesting service. This line goes active (low) and inactive (high) once for each byte transferred even if a burst of data is being transferred.

2. Data Bus Buffer

This three-state, bi-directional, eight bit buffer interfaces the 8257 to the system data bus.

(D₀-D₇)

Data Bus Lines: These are bi-directional three-state lines. When the 8257 is being programmed by the CPU, eight-bits of data for a DMA address register, a terminal count register or the Mode Set register are received on the data bus. When the CPU reads a DMA address register, a terminal count register or the Status register, the data is sent to the CPU over the data bus. During DMA cycles (when the 8257 is the bus master), the 8257 will output the most significant eight-bits of the memory address (from one of the DMA address registers) to the 8212 latch via the data bus. These address bits will be transferred at the beginning of the DMA cycle; the bus will then be released to handle the memory data transfer during the balance of the DMA cycle.

BIT 15	BIT 14	TYPE OF DMA OPERATION
0	0	Verify DMA Cycle
0	1	Write DMA Cycle
1	0	Read DMA Cycle
1	1	(Illegal)

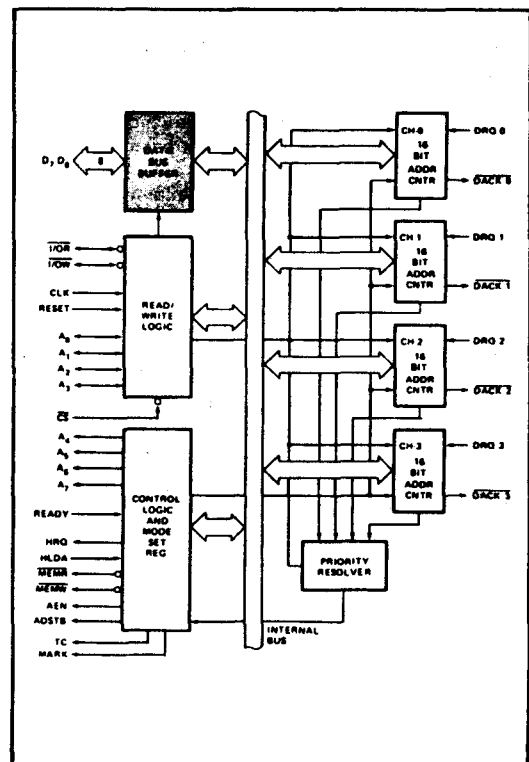


Figure 4. 8257 Block Diagram Showing Data Bus Buffer



8257/8257-5

3. Read/Write Logic

When the CPU is programming or reading one of the 8257's registers (i.e., when the 8257 is a "slave" device on the system bus), the Read/Write Logic accepts the I/O Read ($\overline{IOR}$) or I/O Write ($\overline{IOW}$) signal, decodes the least significant four address bits, (A_0-A_3), and either writes the contents of the data bus into the addressed register (if $\overline{IOW}$ is true) or places the contents of the addressed register onto the data bus (if $\overline{IOR}$ is true).

During DMA cycles (i.e., when the 8257 is the bus "master"), the Read/Write Logic generates the I/O read and memory write (DMA write cycle) or I/O Write and memory read (DMA read cycle) signals which control the data link with the peripheral that has been granted the DMA cycle.

Note that during DMA transfers Non-DMA I/O devices should be de-selected (disabled) using "AEN" signal to inhibit I/O device decoding of the memory address as an erroneous device address.

($\overline{IOR}$)

I/O Read: An active-low, bi-directional three-state line. In the "slave" mode, it is an input which allows the 8-bit status register or the upper/lower byte of a 16-bit DMA address register or terminal count register to be read. In the "master" mode, $\overline{IOR}$ is a control output which is used to access data from a peripheral during the DMA write cycle.

($\overline{IOW}$)

I/O Write: An active-low, bi-directional three-state line. In the "slave" mode, it is an input which allows the contents of the data bus to be loaded into the 8-bit mode set register or the upper/lower byte of a 16-bit DMA address register or terminal count register. In the "master" mode, $\overline{IOW}$ is a control output which allows data to be output to a peripheral during a DMA read cycle.

(CLK)

Clock Input: Generally from an Intel® 8224 Clock Generator device. (ϕ 2 TTL) or Intel® 8085A CLK output.

(RESET)

Reset: An asynchronous input (generally from an 8224 or 8085 device) which disables all DMA channels by clearing the mode register and 3-states all control lines.

(A_0-A_3)

Address Lines: These least significant four address lines are bi-directional. In the "slave" mode they are inputs which select one of the registers to be read or programmed. In the "master" mode, they are outputs which constitute the least significant four bits of the 16-bit memory address generated by the 8257.

($\overline{CS}$)

Chip Select: An active-low input which enables the I/O Read or I/O Write input when the 8257 is being read or programmed in the "slave" mode. In the "master" mode, $\overline{CS}$ is automatically disabled to prevent the chip from selecting itself while performing the DMA function.

4. Control Logic

This block controls the sequence of operations during all DMA cycles by generating the appropriate control signals and the 16-bit address that specifies the memory location to be accessed.

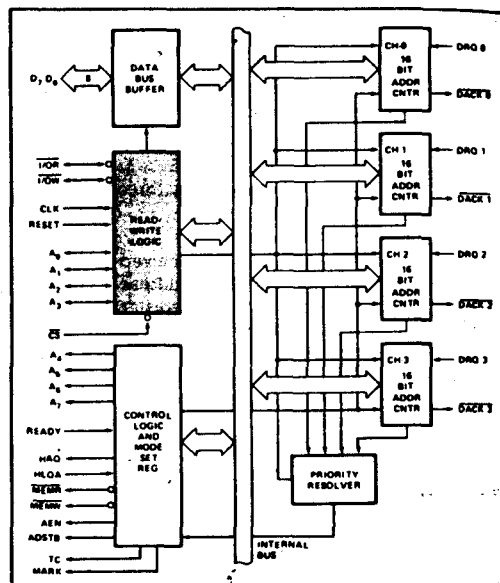


Figure 5. 8257 Block Diagram Showing Read/Write Logic Function



8257/8257-5

(A₄-A₇)

Address Lines: These four address lines are three-state outputs which constitute bits 4 through 7 of the 16-bit memory address generated by the 8257 during all DMA cycles.

(READY)

Ready: This asynchronous input is used to elongate the memory read and write cycles in the 8257 with wait states if the selected memory requires longer cycles. READY must conform to specified setup and hold times.

(HRQ)

Hold Request: This output requests control of the system bus. In systems with only one 8257, HRQ will normally be applied to the HOLD input on the CPU. HRQ must conform to specified setup and hold times.

(HLDA)

Hold Acknowledge: This input from the CPU indicates that the 8257 has acquired control of the system bus.

(MEMR)

Memory Read: This active-low three-state output is used to read data from the addressed memory location during DMA Read cycles.

(MEMW)

Memory Write: This active-low three-state output is used to write data into the addressed memory location during DMA Write cycles.

(ADSTB)

Address Strobe: This output strobes the most significant byte of the memory address into the 8212 device from the data bus.

(AEN)

Address Enable: This output is used to disable (float) the System Data Bus and the System Control Bus. It may also be used to disable (float) the System Address Bus by use of an enable on the Address Bus drivers in systems to inhibit non-DMA devices from responding during DMA cycles. It may be further used to isolate the 8257 data bus from the System Data Bus to facilitate the transfer of the 8 most significant DMA address bits over the 8257 data I/O pins without subjecting the System Data Bus to any timing constraints for the transfer. When the 8257 is used in an I/O device structure (as opposed to memory mapped), this AEN output should be used to disable the selection of an I/O device when the DMA address is on the address bus. The I/O device selection should be determined by the DMA acknowledge outputs for the 4 channels.

(TC)

Terminal Count: This output notifies the currently selected peripheral that the present DMA cycle should be the last cycle for this data block. If the TC STOP bit in the Mode Set register is set, the selected channel will be automatically disabled at the end of that DMA cycle. TC is activated when the 14-bit value in the selected channel's terminal count register equals zero. Recall that the low-order 14-bits of the terminal count register should be loaded with the values (n-1), where n = the desired number of the DMA cycles.

(MARK)

Modulo 128 Mark: This output notifies the selected peripheral that the current DMA cycle is the 128th cycle since the previous MARK output. MARK always occurs at 128 (and all multiples of 128) cycles from the end of the data block. Only if the total number of DMA cycles (n) is evenly divisible by 128 (and the terminal count register was loaded with n-1), will MARK occur at 128 (and each succeeding multiple of 128) cycles from the beginning of the data block.

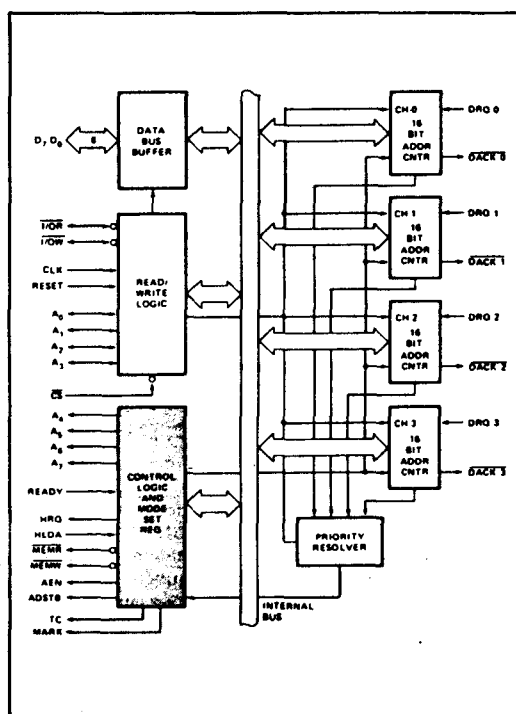
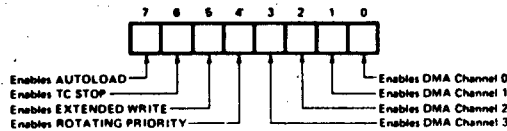


Figure 6. 8257 Block Diagram Showing Control Logic and Mode Set Register



5. Mode Set Register

When set, the various bits in the Mode Set register enable each of the four DMA channels, and allow four different options for the 8257:

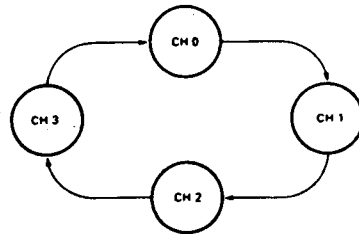


The Mode Set register is normally programmed by the CPU after the DMA address register(s) and terminal count register(s) are initialized. The Mode Set Register is cleared by the RESET input, thus disabling all options, inhibiting all channels, and preventing bus conflicts on power-up. A channel should not be left enabled unless its DMA address and terminal count registers contain valid values; otherwise, an inadvertent DMA request (DRQn) from a peripheral could initiate a DMA cycle that would destroy memory data.

The various options which can be enabled by bits in the Mode Set register are explained below:

Rotating Priority Bit 4

In the Rotating Priority Mode, the priority of the channels has a circular sequence. After each DMA cycle, the priority of each channel changes. The channel which has just been serviced will have the lowest priority.



If the ROTATING PRIORITY bit is not set (set to a zero), each DMA channel has a fixed priority. In the fixed priority mode, Channel 0 has the highest priority and Channel 3 has the lowest priority. If the ROTATING PRIORITY bit is set to a one, the priority of each channel changes after each DMA cycle (not each DMA request). Each channel moves up to the next highest priority assignment, while the channel which has just been serviced moves to the lowest priority assignment:

	CHANNEL → JUST SERVICED	CH-0	CH-1	CH-2	CH-3
Priority → Assignments	Highest ↑ Lowest	CH-1 CH-2 CH-3 CH-0	CH-2 CH-3 CH-0 CH-1	CH-3 CH-0 CH-1 CH-2	CH-0 CH-1 CH-2 CH-3

Note that rotating priority will prevent any one channel from monopolizing the DMA mode; consecutive DMA cycles will service different channels if more than one channel is enabled and requesting service. There is no overhead penalty associated with this mode of operation. All DMA operations began with Channel 0 initially assigned to the highest priority for the first DMA cycle.

Extended Write Bit 5

If the EXTENDED WRITE bit is set, the duration of both the MEMW and I/OW signals is extended by activating them earlier in the DMA cycle. Data transfers within micro-computer systems proceed asynchronously to allow use of various types of memory and I/O devices with different access times. If a device cannot be accessed within a specific amount of time it returns a "not ready" indication to the 8257 that causes the 8257 to insert one or more wait states in its internal sequencing. Some devices are fast enough to be accessed without the use of wait states, but if they generate their READY response with the leading edge of the I/OW or MEMW signal (which generally occurs late in the transfer sequence), they would normally cause the 8257 to enter a wait state because it does not receive READY in time. For systems with these types of devices, the Extended Write option provides alternative timing for the I/O and memory write signals which allows the devices to return an early READY and prevents the unnecessary occurrence of wait states in the 8257, thus increasing system throughput.

TC Stop Bit 6

If the TC STOP bit is set, a channel is disabled (i.e., its enable bit is reset) after the Terminal Count (TC) output goes true, thus automatically preventing further DMA operation on that channel. The enable bit for that channel must be re-programmed to continue or begin another DMA operation. If the TC STOP bit is not set, the occurrence of the TC output has no effect on the channel enable bits. In this case, it is generally the responsibility of the peripheral to cease DMA requests in order to terminate a DMA operation.

Auto Load Bit 7

The Auto Load mode permits Channel 2 to be used for repeat block or block chaining operations, without immediate software intervention between blocks. Channel 2 registers are initialized as usual for the first data block; Channel 3 registers, however, are used to store the block re-initialization parameters (DMA starting address, terminal count and DMA transfer mode). After the first block of DMA cycles is executed by Channel 2 (i.e., after the TC output goes true), the parameters stored in the Channel 3 registers are transferred to Channel 2 during an "update" cycle. Note that the TC STOP feature, described above, has no effect on Channel 2 when the Auto Load bit is set.



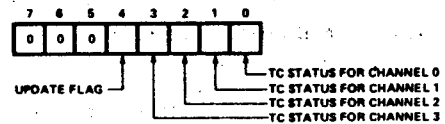
8257/8257-5

If the Auto Load bit is set, the initial parameters for Channel 2 are automatically duplicated in the Channel 3 registers when Channel 2 is programmed. This permits repeat block operations to be set up with the programming of a single channel. Repeat block operations can be used in applications such as CRT refreshing. Channels 2 and 3 can still be loaded with separate values if Channel 2 is loaded before loading Channel 3. Note that in the Auto Load mode, Channel 3 is still available to the user if the Channel 3 enable bit is set, but use of this channel will change the values to be auto loaded into Channel 2 at update time. All that is necessary to use the Auto Load feature for chaining operations is to reload Channel 3 registers at the conclusion of each update cycle with the new parameters for the next data block transfer.

Each time that the 8257 enters an update cycle, the update flag in the status register is set and parameters in Channel 3 are transferred to Channel 2, non-destructively for Channel 3. The actual re-initialization of Channel 2 occurs at the beginning of the next channel 2 DMA cycle after the TC cycle. This will be the first DMA cycle of the new data block for Channel 2. The update flag is cleared at the conclusion of this DMA cycle. For chaining operations, the update flag in the status register can be monitored by the CPU to determine when the re-initialization process has been completed so that the next block parameters can be safely loaded into Channel 3.

8. Status Register

The eight-bit status register indicates which channels have reached a terminal count condition and includes the update flag described previously.



The TC status bits are set when the Terminal Count (TC) output is activated for that channel. These bits remain set until the status register is read or the 8257 is reset. The UPDATE FLAG, however, is not affected by a status register read operation. The UPDATE FLAG can be cleared by resetting the 8257, by changing to the non-auto load mode (i.e., by resetting the AUTO LOAD bit in the Mode Set register) or it can be left to clear itself at the completion of the update cycle. The purpose of the UPDATE FLAG is to prevent the CPU from inadvertently skipping a data block by overwriting a starting address or terminal count in the Channel 3 registers before those parameters are properly auto-loaded into Channel 2.

The user is cautioned against reading the TC status register and using this information to reenables channels that have not completed operation. Unless the DMA channels are inhibited a channel could reach terminal count (TC) between the status read and the mode write. DMA can be inhibited by a hardware gate on the HRQ line or by disabling channels with a mode word before reading the TC status.

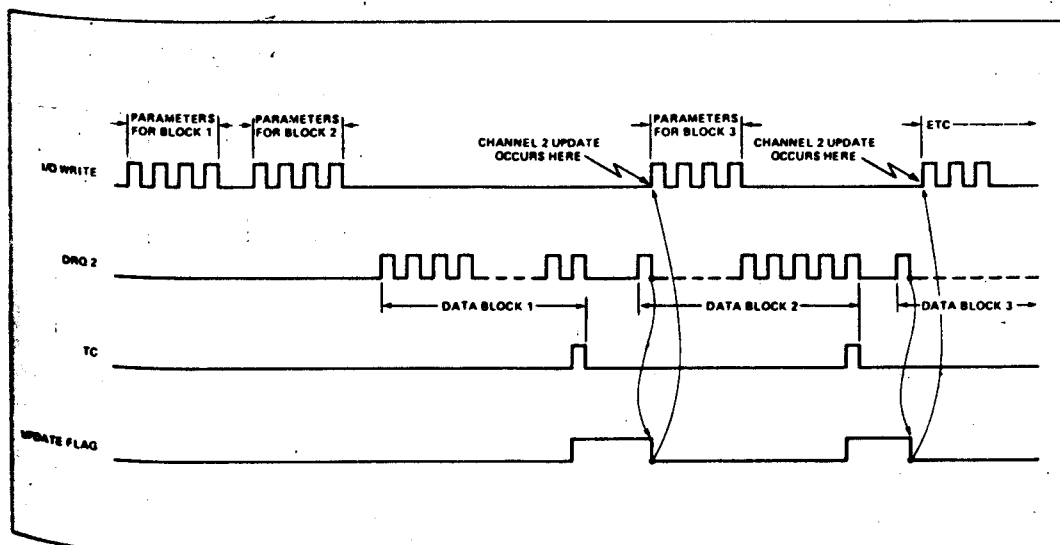


Figure 7. Autoload Timing



8257/8257-5

OPERATIONAL SUMMARY

Programming and Reading the 8257 Registers

There are four pairs of "channel registers": each pair consisting of a 16-bit DMA address register and a 16-bit terminal count register (one pair for each channel). The 8257 also includes two "general registers": one 8-bit Mode Set register and one 8-bit Status register. The registers are loaded or read when the CPU executes a write or read instruction that addresses the 8257 device and the appropriate register within the 8257. The 8228 generates the appropriate read or write control signal (generally I/OR or I/OW while the CPU places a 16-bit address on the system address bus, and either outputs the data to be written onto the system data bus or accepts the data being read from the data bus. All or some of the most significant 12 address bits A_4 - A_{15} (depending on the systems memory, I/O configuration) are usually decoded to produce the chip select ($\overline{CS}$) input to the 8257. An I/O Write input (or Memory Write in memory mapped I/O configurations, described below) specifies that the addressed register is to be programmed, while an I/O Read input (or Memory Read) specifies that the addressed register is to be read. Address bit 3 specifies whether a "channel register" ($A_3 = 0$) or the Mode Set (program only)/Status (read only) register ($A_3 = 1$) is to be accessed.

The least significant three address bits, A_0 - A_2 , indicate the specific register to be accessed. When accessing the Mode Set or Status register, A_0 - A_2 are all zero. When accessing a channel register bit A_0 differentiates between the DMA address register ($A_0 = 0$) and the terminal count register ($A_0 = 1$), while bits A_1 and A_2 specify one of the

CONTROL INPUT	$\overline{CS}$	I/OW	I/OR	A_3
Program Half of a Channel Register	0	0	1	0
Read Half of a Channel Register	0	1	0	0
Program Mode Set Register	0	0	1	1
Read Status Register	0	1	0	1

four channels. Because the "channel registers" are 16-bits, two program instruction cycles are required to load or read an entire register. The 8257 contains a first/last (F/L) flip flop which toggles at the completion of each channel program or read operation. The F/L flip flop determines whether the upper or lower byte of the register is to be accessed. The F/L flip flop is reset by the RESET input and whenever the Mode Set register is loaded. To maintain proper synchronization when accessing the "channel registers" all channel command instruction operations should occur in pairs, with the lower byte of a register always being accessed first. Do not allow $\overline{CS}$ to clock while either I/OR or I/OW is active, as this will cause an erroneous F/L flip flop state. In systems utilizing an interrupt structure, interrupts should be disabled prior to any paired programming operations to prevent an interrupt from splitting them. The result of such a split would leave the F/L F/F in the wrong state. This problem is particularly obvious when other DMA channels are programmed by an interrupt structure.

8257 Register Selection

REGISTER	BYTE	ADDRESS INPUTS				F/L	'BI-DIRECTIONAL DATA BUS							
		A ₃	A ₂	A ₁	A ₀		D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
CH-0 DMA Address	LSB	0	0	0	0	0	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
	MSB	0	0	0	0	1	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈
CH-0 Terminal Count	LSB	0	0	0	1	0	C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀
	MSB	0	0	0	1	1	C ₁₅	C ₁₄	C ₁₃	C ₁₂	C ₁₁	C ₁₀	C ₉	C ₈
CH-1 DMA Address	LSB	0	0	1	0	0	Same as Channel 0							
	MSB	0	0	1	0	1								
CH-1 Terminal Count	LSB	0	0	1	1	0								
	MSB	0	0	1	1	1								
CH-2 DMA Address	LSB	0	1	0	0	0	Same as Channel 0							
	MSB	0	1	0	0	1								
CH-2 Terminal Count	LSB	0	1	0	1	0								
	MSB	0	1	0	1	1								
CH-3 DMA Address	LSB	0	1	1	0	0	Same as Channel 0							
	MSB	0	1	1	0	1								
CH-3 Terminal Count	LSB	0	1	1	1	0								
	MSB	0	1	1	1	1								
MODE SET (Program only)	—	1	0	0	0	0	AL	TCS	EW	RP	EN3	EN2	EN1	EN0
STATUS (Read only)	—	1	0	0	0	0	0	0	0	UP	TC3	TC2	TC1	TC0

* A_0 - A_{15} : DMA Starting Address, C_0 - C_{15} : Terminal Count value (N-1), Rd and Wr: DMA Verify (00), Write (01) or Read (10) cycle selection.
AL: Auto Load, TCS: TC STOP, EW: EXTENDED WRITE, RP: ROTATING PRIORITY, EN3-EN0: CHANNEL ENABLE MASK, UP: UPDATE FLAG, TC3-TC0: TERMINAL COUNT STATUS BITS.



8257/8257-5

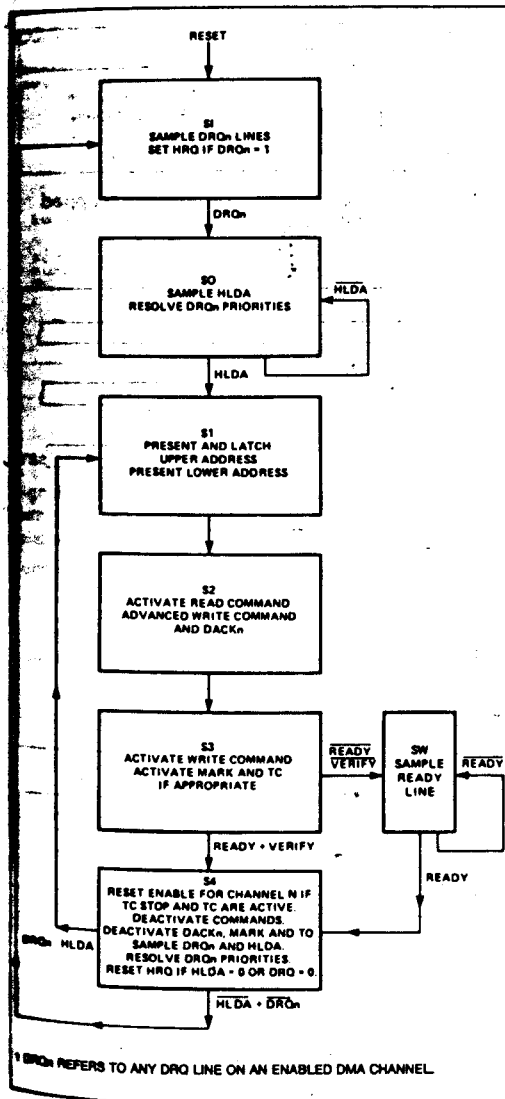


Figure 8. DMA Operation State Diagram

DMA OPERATION

Single Byte Transfers

A single byte transfer is initiated by the I/O device raising the DRQ line of one channel of the 8257. If the channel is enabled, the 8257 will output a HRQ to the CPU. The 8257 now waits until a HLDA is received insuring that the system bus is free for its use. Once HLDA is received the DACK line for the requesting channel is activated (LOW). The DACK line acts as a chip select for the requesting I/O device. The 8257 then generates the

read and write commands and byte transfer occurs between the selected I/O device and memory. After the transfer is complete, the DACK line is set HIGH and the HRQ line is set LOW to indicate to the CPU that the bus is now free for use. DRQ must remain HIGH until DACK is issued to be recognized and must go LOW before S4 of the transfer sequence to prevent another transfer from occurring. (See timing diagram.)

Consecutive Transfers

If more than one channel requests service simultaneously, the transfer will occur in the same way a burst does. No overhead is incurred by switching from one channel to another. In each S4 the DRQ lines are sampled and the highest priority request is recognized during the next transfer. A burst mode transfer in a lower priority channel will be overridden by a higher priority request. Once the high priority transfer has completed control will return to the lower priority channel if its DRQ is still active. No extra cycles are needed to execute this sequence and the HRQ line remains active until all DRQ lines go LOW.

Control Override

The continuous DMA transfer mode described above can be interrupted by an external device by lowering the HLDA line. After each DMA transfer the 8257 samples the HLDA line to insure that it is still active. If it is not active, the 8257 completes the current transfer, releases the HRQ line (LOW) and returns to the idle state. If DRQ lines are still active the 8257 will raise the HRQ line in the third cycle and proceed normally. (See timing diagram.)

Not Ready

The 8257 has a Ready input similar to the 8080A and the 8085A. The Ready line is sampled in State 3. If Ready is LOW the 8257 enters a wait state. Ready is sampled during every wait state. When Ready returns HIGH the 8257 proceeds to State 4 to complete the transfer. Ready is used to interface memory or I/O devices that cannot meet the bus set up times required by the 8257.

Speed

The 8257 uses four clock cycles to transfer a byte of data. No cycles are lost in the master to master transfer maximizing bus efficiency. A 2MHz clock input will allow the 8257 to transfer at a rate of 500K bytes/second.

Memory Mapped I/O Configurations

The 8257 can be connected to the system bus as a memory device instead of as an I/O device for memory mapped I/O configurations by connecting the system memory control lines to the 8257's I/O control lines and the system I/O control lines to the 8257's memory control lines.

This configuration permits use of the 8080's considerably larger repertoire of memory instructions when reading or loading the 8257's registers. Note that with this connection, the programming of the Read (bit 15) and Write (bit 14) bits in the terminal count register will have a different meaning:



8257/8257-5

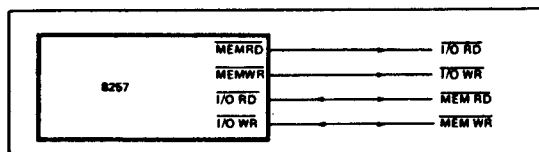


Figure 9. System Interface for Memory Mapped I/O

BIT 15 READ	BIT 14 WRITE	
0	0	DMA Verify Cycle
0	1	DMA Read Cycle
1	0	DMA Write Cycle
1	1	Illegal

Figure 10. TC Register for Memory Mapped I/O Only

SYSTEM APPLICATION EXAMPLES

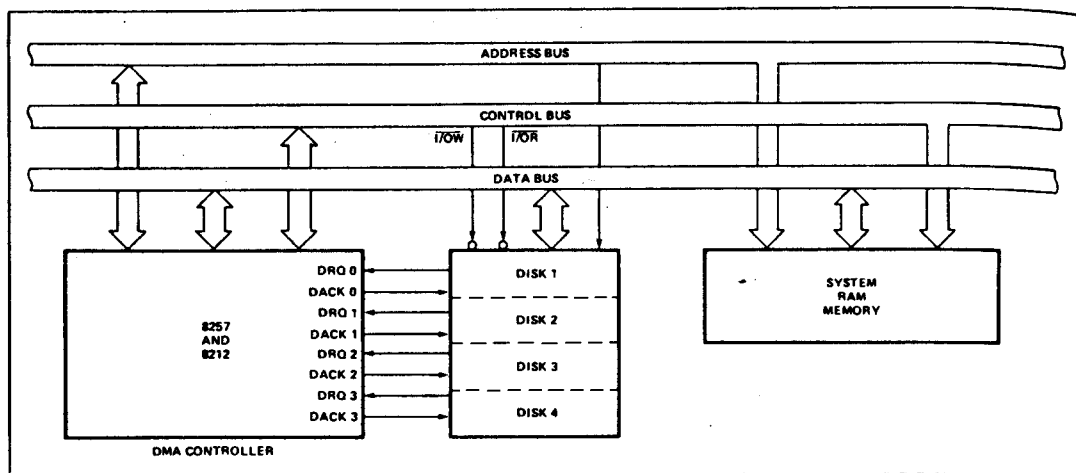


Figure 11. Floppy Disk Controller (4 Drives)

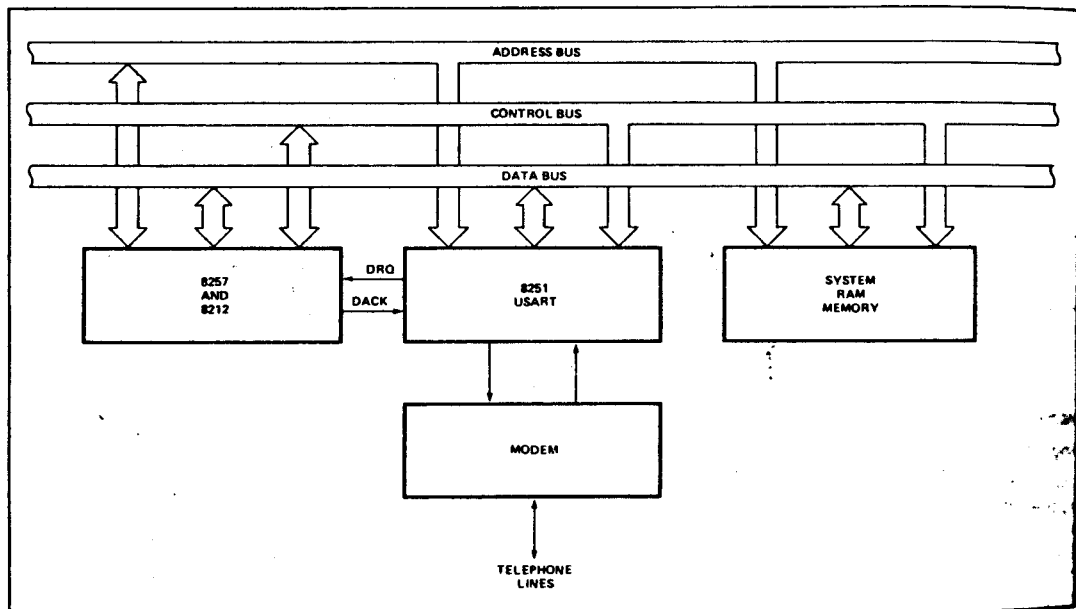
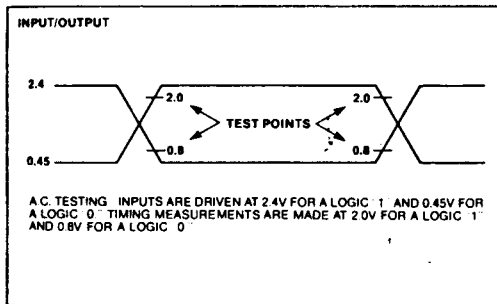


Figure 12. High-Speed Communication Controller

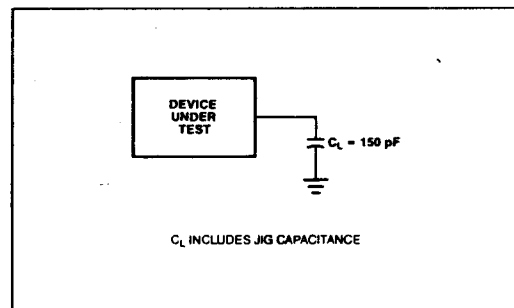


8257/8257-5

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. TESTING LOAD CIRCUIT



Tracking Parameters

Signals labeled as Tracking Parameters (footnotes 1 and 5-7 under A.C. Specifications) are signals that follow similar paths through the silicon die. The propagation speed of these signals varies in the manufacturing process but the relationship between all these parameters is constant. The variation is less than or equal to 50 ns.

Suppose the following timing equation is being evaluated,

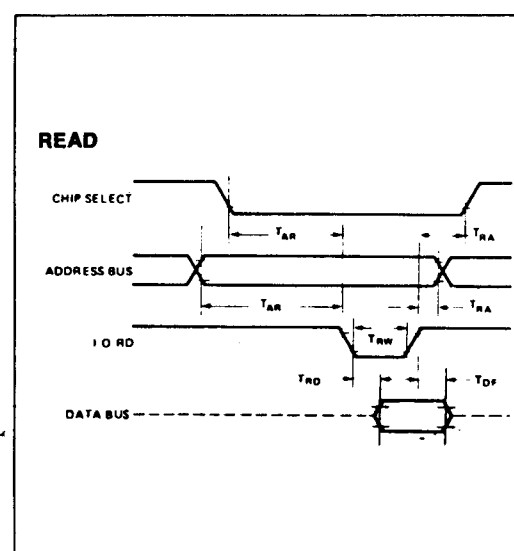
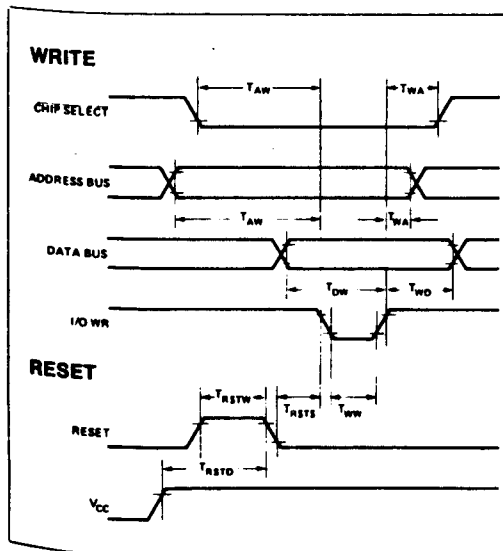
$$T_{A(MIN)} + T_{B(MAX)} \leq 150 \text{ ns}$$

and only minimum specifications exist for T_A and T_B . If $T_{A(MIN)}$ is used, and if T_A and T_B are tracking parameters, $T_{B(MAX)}$ can be taken as $T_{B(MIN)} + 50 \text{ ns}$.

$$T_{A(MIN)} + (T_{B(MIN)} + 50 \text{ ns}) \leq 150 \text{ ns}$$

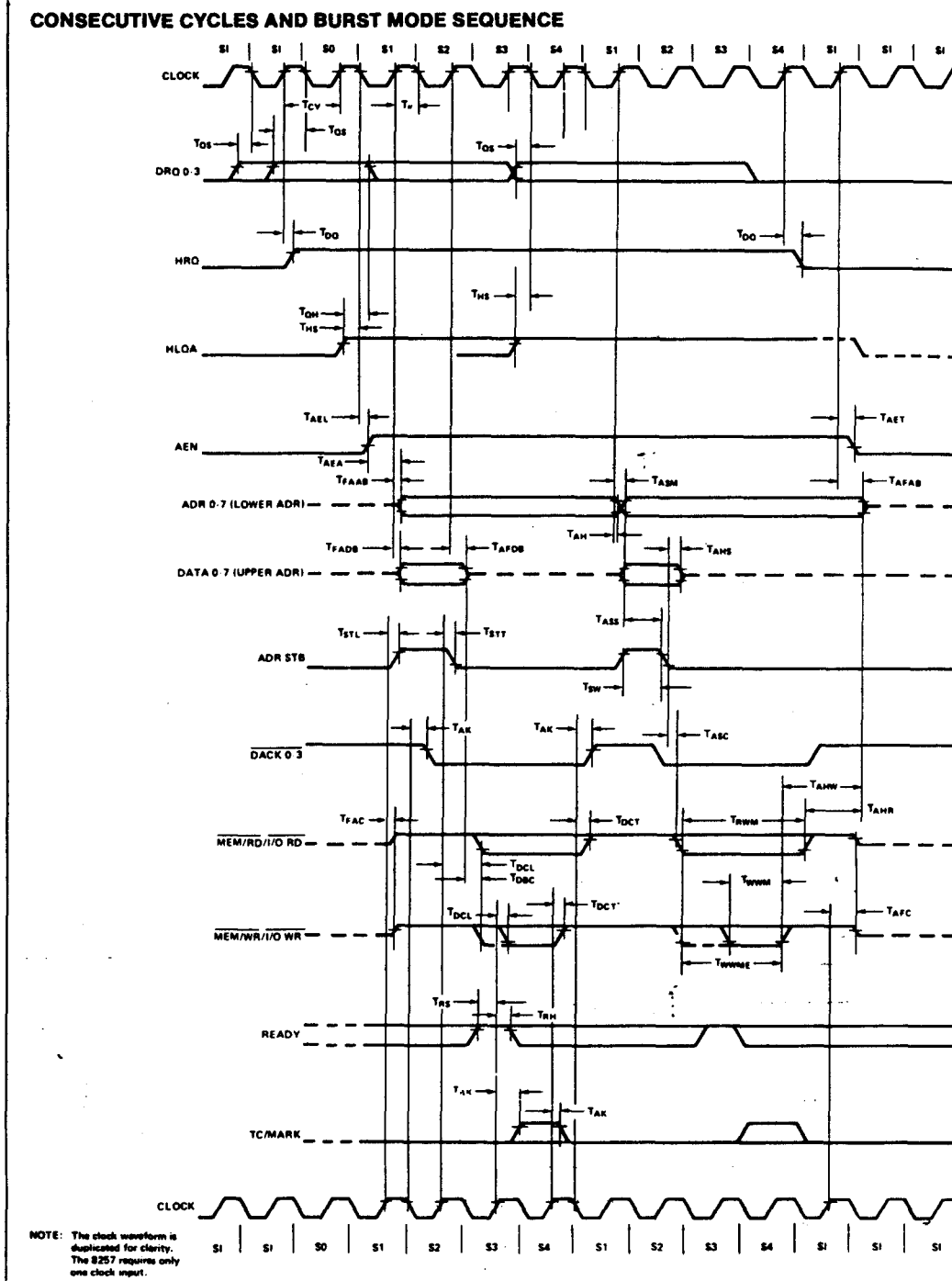
*If T_A and T_B are tracking parameters

WAVEFORMS—PERIPHERAL MODE





CONSECUTIVE CYCLES AND BURST MODE SEQUENCE

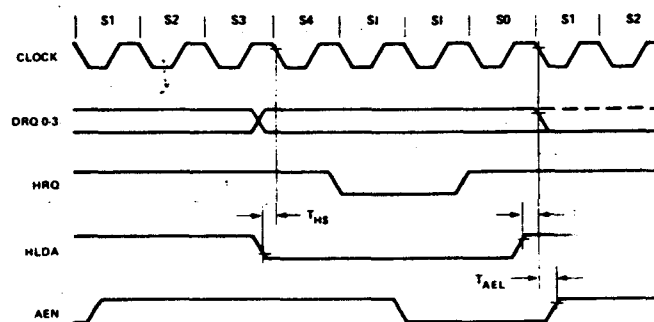




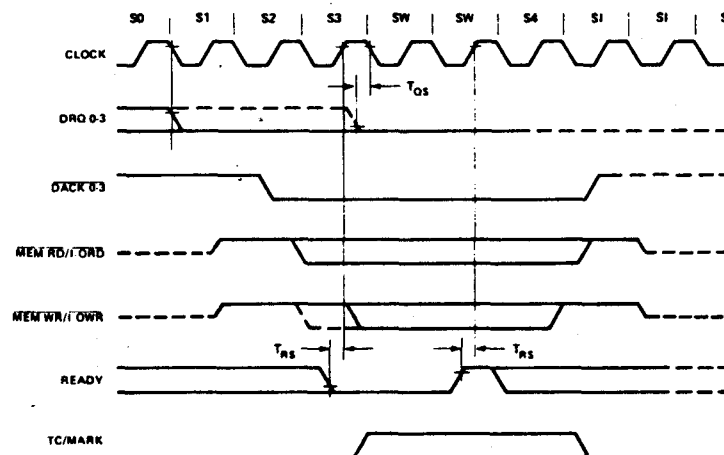
8257/8257-5

WAVEFORMS (Continued)

CONTROL OVERRIDE SEQUENCE



NOT READY SEQUENCE





8257/8257-5

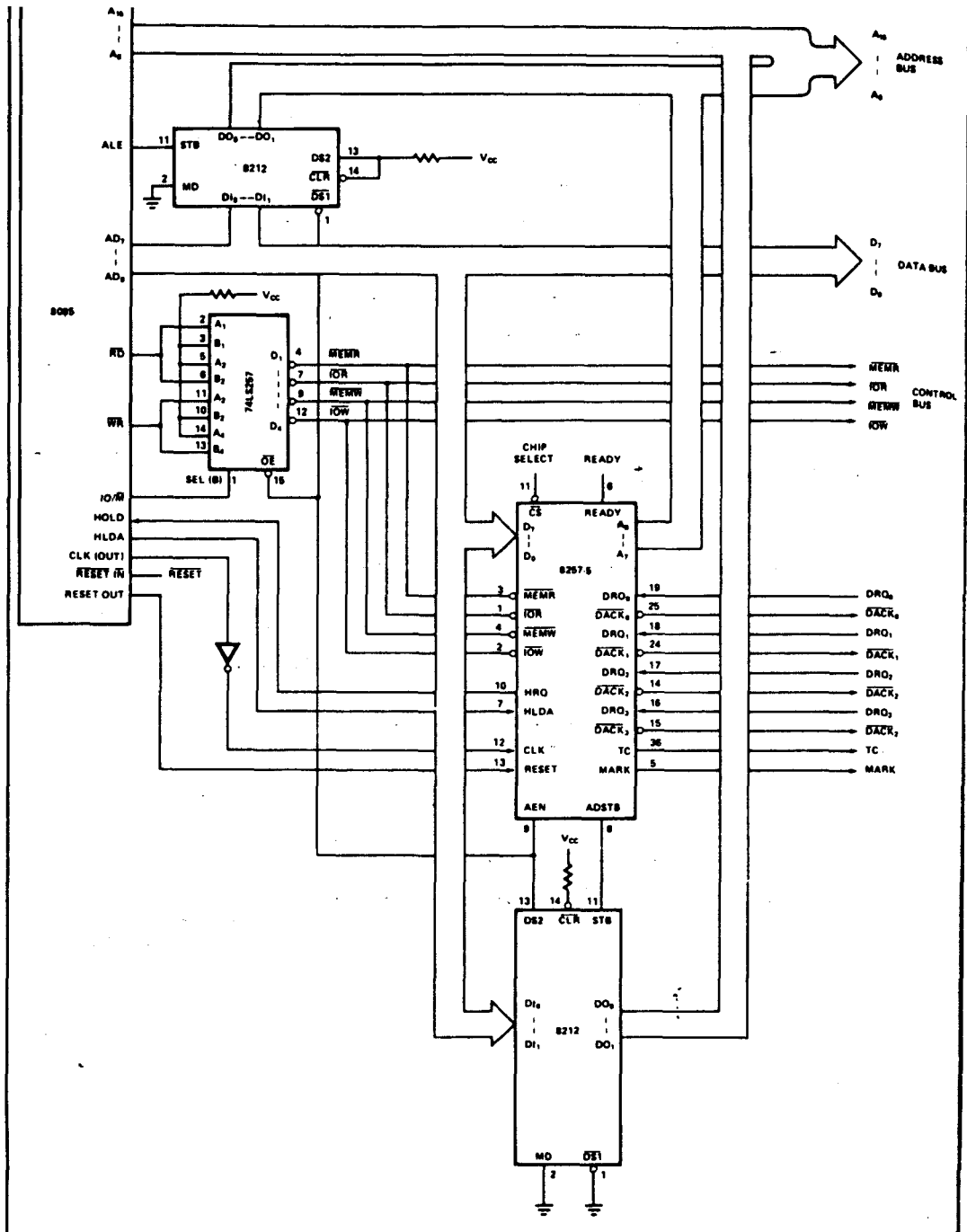


Figure 13. Detailed System Interface Schematic



ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias 0°C to 70°C
Storage Temperature -65°C to +150°C
Voltage on Any Pin
With Respect to Ground -0.5V to +7V
Power Dissipation 1 Watt

**NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS

(8257: $T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5.0\text{V} \pm 5\%$, GND = 0V)
(8257-5: $T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5.0\text{V} \pm 10\%$, GND = 0V)

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	Volts	
V_{IH}	Input High Voltage	2.0	$V_{CC} + .5$	Volts	
V_{OL}	Output Low Voltage		0.45	Volts	$I_{OL} = 1.6\text{ mA}$
V_{OH}	Output High Voltage	2.4	V_{CC}	Volts	$I_{OH} = -150\mu\text{A}$ for AB, DB and AEN $I_{OH} = -80\mu\text{A}$ for others
V_{HH}	HRQ Output High Voltage	3.3	V_{CC}	Volts	$I_{OH} = -80\mu\text{A}$
I_{CC}	V_{CC} Current Drain		120	mA	
I_{IL}	Input Leakage		± 10	μA	$0\text{V} \leq V_{IN} \leq V_{CC}$
I_{OFL}	Output Leakage During Float		± 10	μA	$0.45\text{V} \leq V_{OUT} \leq V_{CC}$

CAPACITANCE ($T_A = 25^\circ\text{C}$; $V_{CC} = \text{GND} = 0\text{V}$)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	$f_c = 1\text{MHz}$
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to GND



8257/8257-5

A.C. CHARACTERISTICS—PERIPHERAL (SLAVE) MODE

(8257: $T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5.0\text{V} \pm 5\%$, $\text{GND} = 0\text{V}$)

(8257-5: $T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5.0\text{V} \pm 10\%$, $\text{GND} = 0\text{V}$)

8080 Bus Parameters

READ CYCLE

Symbol	Parameter	8257		8257-5		Unit	Test Conditions
		Min.	Max.	Min.	Max.		
T_{AR}	Adr or $\overline{\text{CS}}$ Setup to $\overline{\text{RD}}$	0		0		ns	
T_{RA}	Adr or $\overline{\text{CS}}$ Hold from $\overline{\text{RD}}$	0		0		ns	
T_{RD}	Data Access from $\overline{\text{RD}}$	0	300	0	220	ns	
T_{DF}	DB→Float Delay from $\overline{\text{RD}}$	20	150	20	120	ns	
T_{RR}	$\overline{\text{RD}}$ Width	250		250		ns	

WRITE CYCLE

Symbol	Parameter	8257		8257-5		Unit	Test Conditions
		Min.	Max.	Min.	Max.		
T_{AW}	Adr Setup to $\overline{\text{WR}}$	20		20		ns	
T_{WA}	Adr Hold from $\overline{\text{WR}}$	0		0		ns	
T_{DW}	Data Setup to $\overline{\text{WR}}$	200		200		ns	
T_{WD}	Data Hold from $\overline{\text{WR}}$	0		0		ns	
T_{WW}	$\overline{\text{WR}}$ Width	200		200		ns	

OTHER TIMING

Symbol	Parameter	8257		8257-5		Unit	Test Conditions
		Min.	Max.	Min.	Max.		
T_{RSTW}	Reset Pulse Width	300		300		ns	
T_{RSTD}	Power Supply↑ (V_{CC}) Setup to Reset↓	500		500		μs	
T_r	Signal Rise Time		20		20	ns	
T_f	Signal Fall Time		20		20	ns	
T_{RSTS}	Reset to First $\overline{\text{I/O}}\overline{\text{WR}}$	2		2		t_{CY}	

A.C. CHARACTERISTICS—DMA (MASTER) MODE

(8257: $T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5.0\text{V} \pm 5\%$, $\text{GND} = 0\text{V}$)

(8257-5: $T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5.0\text{V} \pm 10\%$, $\text{GND} = 0\text{V}$)

TIMING REQUIREMENTS

Symbol	Parameter	8257		8257-5		Unit
		Min.	Max.	Min.	Max.	
T_{CY}	Cycle Time (Period)	0.320	4	0.320	4	μs
T_ϕ	Clock Active (High)	120	$.8T_{CY}$	80	$.8T_{CY}$	ns
T_{OS}	$\overline{\text{DRQ}}$ Setup to CLK (S1, S4)	120		120		ns
T_{OH}	$\overline{\text{DRQ}}$ Hold from HLDA (1)	0		0		ns
T_{HS}	HLDA or I Setup to CLK (S1, S4)	100		100		ns
T_{RS}	READY Setup Time to CLK (S3, Sw)	30		30		ns
T_{RH}	READY Hold Time from CLK (S3, Sw)	20		20		ns



8257/8257-5

A.C. CHARACTERISTICS—DMA (MASTER) MODE(8257: $T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5.0\text{V} \pm 5\%$, $\text{GND} = 0\text{V}$)(8257-5: $T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5.0\text{V} \pm 10\%$, $\text{GND} = 0\text{V}$)**• TIMING RESPONSES**

Symbol	Parameter	8257		8257-5		Unit
		Min.	Max.	Min.	Max.	
T_{DQ}	$\text{HRQ}\uparrow$ or $\downarrow$ Delay from $\text{CLK}\uparrow$ (S1, S4) (measured at 2.0V)		160		160	ns
T_{DQ1}	$\text{HRQ}\uparrow$ or $\downarrow$ Delay from $\text{CLK}\uparrow$ (S1, S4) (measured at 3.3V) ^[3]		250		250	ns
T_{AEL}	$\text{AEN}\uparrow$ Delay from $\text{CLK}\downarrow$ (S1)		300		300	ns
T_{AET}	$\text{AEN}\downarrow$ Delay from $\text{CLK}\uparrow$ (S1)		200		200	ns
T_{AEA}	Adr (AB) (Active) Delay from $\text{AEN}\uparrow$ (S1) ^[1]	20		20		ns
T_{FAAB}	Adr (AB) (Active) Delay from $\text{CLK}\uparrow$ (S1) ^[2]		250		250	ns
T_{FAB}	Adr (AB) (Float) Delay from $\text{CLK}\uparrow$ (S1) ^[2]		150		150	ns
T_{ASM}	Adr (AB) (Stable) Delay from $\text{CLK}\uparrow$ (S1) ^[2]		250		250	ns
T_{AH}	Adr (AB) (Stable) Hold from $\text{CLK}\uparrow$ (S1) ^[2]	$T_{ASM} - 50$		$T_{ASM} - 50$		ns
T_{AHR}	Adr (AB) (Valid) Hold from $\text{RD}\uparrow$ (S1, S1) ^[1]	60		60		ns
T_{AHW}	Adr (AB) (Valid) Hold from $\text{Wr}\uparrow$ (S1, S1) ^[1]	300		300		ns
T_{FADB}	Adr (DB) (Active) Delay from $\text{CLK}\uparrow$ (S1) ^[2]		300		300	ns
T_{AFDB}	Adr (DB) (Float) Delay from $\text{CLK}\uparrow$ (S2) ^[2]	$T_{STT} + 20$	250	$T_{STT} + 20$	170	ns
T_{ASS}	Adr (DB) Setup to Adr Stb $\downarrow$ (S1-S2) ^[1]	100		100		ns
T_{AHS}	Adr (DB) (Valid) Hold from Adr Stb $\downarrow$ (S2) ^[1]	20		20		ns
T_{STL}	Adr Stb $\uparrow$ Delay from $\text{CLK}\uparrow$ (S1)		200		200	ns
T_{STT}	Adr Stb $\downarrow$ Delay from $\text{CLK}\uparrow$ (S2)		140		140	ns
T_{SW}	Adr Stb Width (S1-S2) ^[1]	$T_{CY} - 100$		$T_{CY} - 100$		ns
T_{ASC}	$\text{Rd}\downarrow$ or $\text{Wr}(\text{Ext})\downarrow$ Delay from Adr Stb $\downarrow$ (S2) ^[1]	70		70		ns
T_{DBC}	$\text{RD}\downarrow$ or $\text{WR}(\text{Ext})\downarrow$ Delay from Adr (DB) (Float) (S2) ^[1]	20		20		ns
T_{AK}	$\text{DACK}\uparrow$ or $\downarrow$ Delay from $\text{CLK}\downarrow$ (S2, S1) and $\text{TC}/\text{Mark}\uparrow$ Delay from $\text{CLK}\uparrow$ (S3) and $\text{TC}/\text{Mark}\downarrow$ Delay from $\text{CLK}\uparrow$ (S4) ^[4]		250		250	ns
T_{DCL}	$\text{RD}\downarrow$ or $\text{Wr}(\text{Ext})\downarrow$ Delay from $\text{CLK}\uparrow$ (S2) and $\text{Wr}\downarrow$ Delay from $\text{CLK}\uparrow$ (S3) ^[2,5]		200		200	ns
T_{DCT}	$\text{Rd}\uparrow$ Delay from $\text{CLK}\downarrow$ (S1, S1) and $\text{Wr}\uparrow$ Delay from $\text{CLK}\uparrow$ (S4) ^[2,6]		200		200	ns
T_{FAC}	Rd or Wr (Active) from $\text{CLK}\uparrow$ (S1) ^[2]		300		300	ns
T_{AFC}	Rd or Wr (Active) from $\text{CLK}\uparrow$ (S1) ^[2]		150		150	ns
T_{RWM}	Rd Width (S2-S1 or S1) ^[1]	$2T_{CY} + T_{\theta} - 50$		$2T_{CY} + T_{\theta} - 50$		ns
T_{WWM}	Wr Width (S3-S4) ^[1]	$T_{CY} - 50$		$T_{CY} - 50$		ns
T_{WWE}	$\text{WR}(\text{Ext})$ Width (S2-S4) ^[1]	$2T_{CY} - 50$		$2T_{CY} - 50$		ns

NOTES:1. Tracking Parameter.
2. Load = + 50 pF.3. Load = $V_{OH} = 3.3\text{V}$.
4. $\Delta T_{AK} < 50\text{ ns}$.5. $\Delta T_{DCL} < 50\text{ ns}$.
6. $\Delta T_{OCT} < 50\text{ ns}$.