

UM MÉTODO DE TRANSFERÊNCIA DE DADOS DE UM PROCESSO PARA MEMÓRIA DE UM COMPUTADOR PC-XT VIA CANAL DE DMA

239906

Remy Eskinazi Sant'Anna
Afonso H. Costa e Silva
Hugo Guerra Vasconcelos

RESUMO

Freqüentemente, a transferência de dados para a memória de um computador PC-XT compatível deve ser realizada de maneira rápida e direta, de tal forma que os dados transferidos estejam disponíveis na memória do computador e possam ser processados imediatamente ou conduzidos para um dispositivo de armazenamento de dados em massa tal como um disco flexível ou rígido. Este tipo de transferência é particularmente interessante quando algum dispositivo externo ao microcomputador opera em tempo real fazendo a aquisição de dados, necessitando transferir o dado imediatamente ou após o armazenamento em um buffer interno, como ocorre freqüentemente com dispositivos microcontroladores que armazenam em sua memória interna uma quantidade de dados recolhida do processo.

Remy Eskinazi Sant'Anna – Engenheiro Eletricista, modalidade Eletrônica, mestrando do Departamento de Eletrônica e Sistemas da UFPE, – professor coordenador dos cursos de Eletrônica e Telecomunicações da Escola Técnica Federal de Pernambuco.

Afonso H. Costa e Silva – Engenheiro Eletricista, modalidade Eletrônica, mestrando do Departamento de Eletrônica e Sistemas da UFPE.

Hugo Guerra Vasconcelos – Professor responsável pelo Laboratório de Eletrônica Aplicada, DES-UFPE; Docteur Ingenieur ENSEEHT – Toulouse - França.

1. INTRODUÇÃO

Este artigo se propõe a descrever uma experiência onde um microcontrolador da família 8051 faz interconexão com o canal 1 da controladora de DMA, disponível para aplicações no barramento de um computador PC-XT. São abordados quatro pontos:

1. resumo de operação da controladora de DMA 8237 e hardware de DMA do PC-XT;

2. descrição do hardware externo microcontrolado;
3. software de programação do microcontrolador 8751;
4. software de programação da controladora de DMA 8237.

A figura 1 a seguir ilustra o sistema de transferência de dados proposto em diagrama de blocos mostrando também a sinalização envolvida.

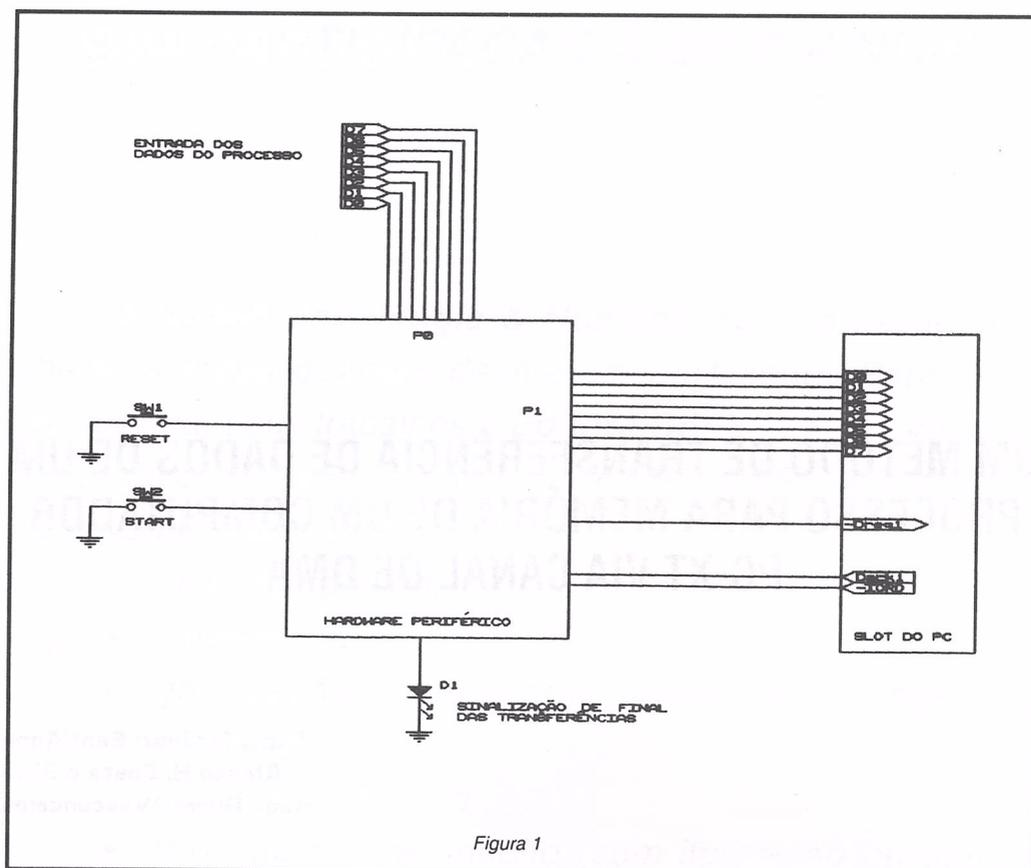


Figura 1

2. SISTEMA DE DMA DE UM COMPUTADOR PC-XT

O sistema de DMA de um computador PC-XT é baseado na controladora de DMA da intel 8237. Esta controladora possui quatro canais que são em sua maioria utilizados em operações internas ao microcomputador. Em resumo, temos a seguinte configuração:

Canal # 0 - utilizada na geração de endereços do processo de refreshamento das RAMs dinâmicas;

Canal # 1 - disponível para aplicações (não usada internamente);

Canal # 2 - utilizada para transferência de dados entre memória e disco flexível;

Canal # 3 - utilizada para transferências de dados entre memória e disco rígido.

Durante a operação de inicialização, cada canal é programado para operar conforme a descrição acima. Apenas o canal 1 de DMA fica disponível para aplicações do usuário.

A controladora 8237 possui 27 registradores internos que necessitam ser inicializados de modo a definir uma operação funcional para cada

um destes canais. Nesta aplicação são usados os seguintes registradores para configuração do canal # 1 de DMA:

REGISTRADOR DE MODO - Tem por função definir um modo funcional para um determinado canal de DMA. Tendo neste caso a seguinte configuração para o canal # 1:

- a) **escrita em memória** - permite que o dado a ser transferido seja escrito na posição de memória endereçada pela controladora;
- b) **auto-inicialização habilitada** - permite que os registradores internos base e corrente sejam reinicializados após o final das transferências, para os seus conteúdos iniciais;
- c) **endereços incrementados a cada transferência** - após cada transferência os registradores corrente podem ser incrementados ou decrementados dependendo desta programação. Neste caso os endereços de transferência são incrementados para haver uma relação seqüencial direta entre dado transferido e o endereço correspondente;

d) modo de transferência único - este modo de transferência permite que apenas um byte seja transferido por cada solicitação de operação de DMA.

REGISTRADOR DE ENDEREÇO BASE - Tem por função armazenar o endereço inicial de memória para onde os dados são transferidos. Se a auto-inicialização estiver habilitada, quando for gerado um sinal \overline{EOP} (end of process), na 8237 este endereço será recarregado no registrador de endereço corrente. Nesta aplicação este registrador será inicializado com o endereço 0000H da página de memória 03H.

O sinal \overline{EOP} é gerado externamente e pode antecipar o término das transferências e restabelecer os registradores corrente caso a auto-inicialização esteja programada. O segmento 03H de memória foi escolhido para esta experiência porque se trata de uma região livre, não sendo utilizada pelo sistema operacional. Não podemos utilizar, por exemplo, o segmento 00H de memória por ter importância fundamental para o funcionamento do computador, pois ali são armazenados os vetores de interrupção entre outros dados importantes.

REGISTRADOR CONTADOR BASE - Tem por função armazenar o número de transferências de DMA a serem realizadas. Se a auto-inicialização estiver habilitada, quando for gerado um sinal \overline{EOP} este valor será recarregado no registrador contador corrente para que um novo ciclo de transferências possa ser realizado. Nesta aplicação este registrador será inicializado com o valor FFFFH, ou seja, uma página inteira de memória (64Kb) será preenchida com dados provenientes do sistema externo. Este registrador é decrementado a cada transferência.

REGISTRADOR DE MÁSCARA DE DMA - Tem a finalidade de mascarar os pedidos de DMA solicitados à controladora, habilitando ou não o reconhecimento dos sinais Dreq. Nesta experiência o canal 1 de DMA deve ser habilitado para operar pelo programa de transferência uma vez que durante o processo de inicialização do PC este canal fica desabilitado.

Uma configuração típica de DMA em um PC é mostrada na figura 2 onde o circuito possui um registrador especial, o registrador de página de DMA, 741s670 que é um chip com quatro registradores de 4 bits, cada um armazenando um valor binário que corresponde às linhas de endereço A16 a A19 de forma a endereçar a página de memória que será acessada durante esta operação. A utilização deste chip se faz necessária devido ao fato da controladora de DMA gerar apenas 16 linhas de endereço.

Outros dois chips utilizados neste circuito são o 741s244 e o 741s373. O primeiro recebe as linhas A0 a A7 geradas pela controladora de DMA e o segundo recebe as linhas D0 a D7 provenientes da controladora que geram as linhas de endereço superiores A8 a A15.

Portanto, durante o ciclo ativo de DMA estes chips geram as linhas de endereços A0 a A19 necessárias para fazer o endereçamento do sistema. A CPU 8088 não fica em estado de hold, mas em estado de wait enquanto que os buffers de barramento ficam em tri-state. Para programar o registrador do 741s670 correspondente ao canal #1 deve ser notado que a seleção de um determinado registrador é realizada através da decodificação das linhas Dack3 e Dack2. Como a controladora é programada para ter o nível ativo 0 em uma linha Dack, para haver o acesso ao registrador correspondente ao canal 1 deve-se programar o registrador 3 de tal modo que quando Dack1 = 0 (Dack3 = Dack2 = 1), pela decodificação, o registrador 3 será acessado. Pode ser notado que durante uma operação de refrescamento de memória realizada pelo canal #0, também será gerado o mesmo código (Dack0 = 0, Dack3 = Dack2 = 1), entretanto as linhas de endereço A19 a A16 não são utilizadas para fins de refrescamento de memória e qualquer nível lógico presente nestas linhas durante esta operação terá efeito transparente. O sinal DMAPG é gerado por uma lógica de decodificação e tem a finalidade de habilitar a operação de escrita (programação) nos registradores internos do 741s670.

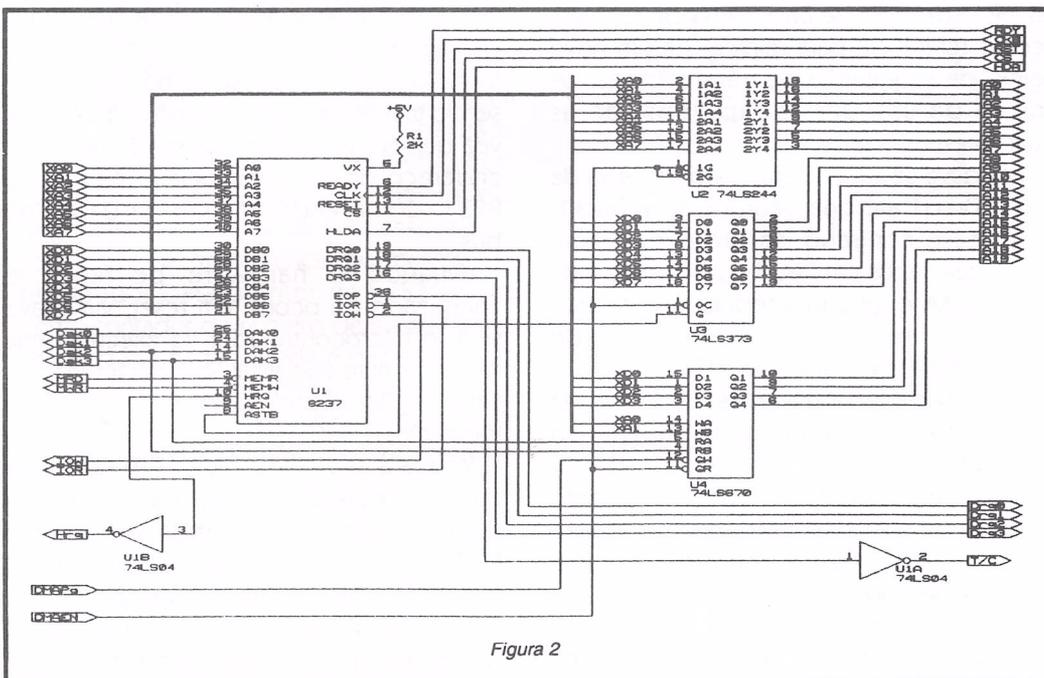


Figura 2

3. HARDWARE PARA AQUISIÇÃO DE DADOS EXTERNOS

O sistema externo deve ser capaz de gerar o sinal de Dreq em intervalos de tempo regulares e precisos. Necessita também reconhecer o sinal

de início de ciclo de DMA (Dack). Neste hardware, um microcontrolador 8751 recolhe os dados do processo através do pórtilo P0 e os envia para o pórtilo P1 onde ficam guardados e presentes na entrada do buffer 3-state 741s244. Este circuito pode ser visualizado na figura 3.

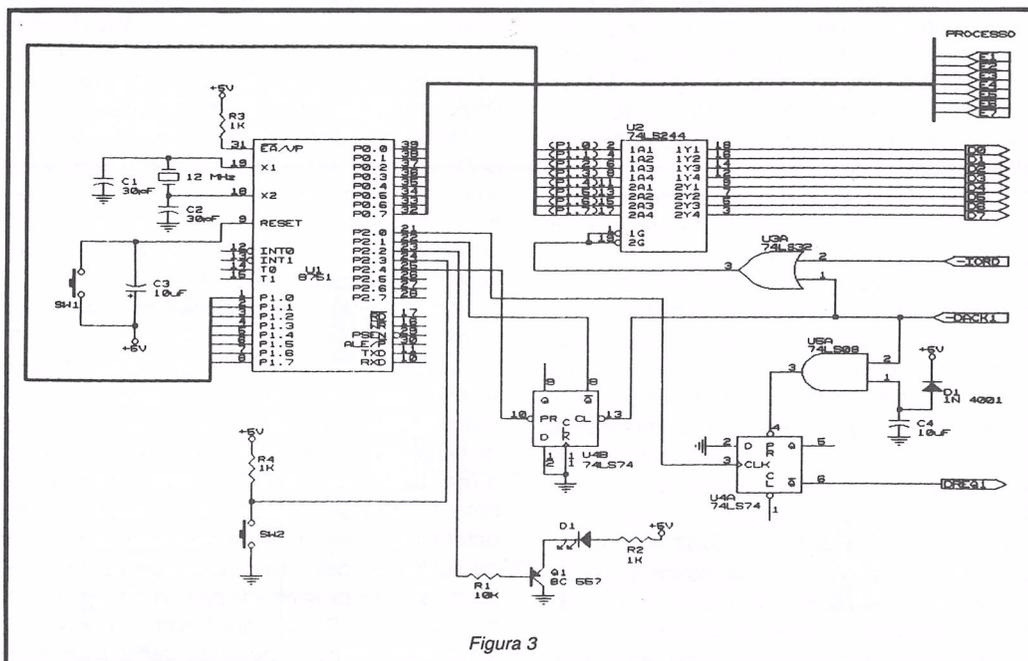


Figura 3

A transição positiva da linha P2.0 atua como um clock para o flip-flop 741s74 (u4a) de modo a setar a sua saída \bar{Q} que vai diretamente para a linha Dreq1, presente no barramento do PC. Este flip-flop só será desativado (Dreq1 = 0) quando for gerado pela controladora de DMA o sinal de reconhecimento de ciclo de DMA Dack1, que atua na porta and (u5a) que seta o flip-flop ($\bar{Q} = 0$). O sinal Dack1 também atua no preset do flip-flop u4b fazendo com que a sua saída \bar{Q} (P2.1) fique em nível 0. Este é o sinal para que o microcontrolador reconheça o início de ciclo de DMA, servindo também como referência para reinício de um novo ciclo de solicitação de transferências. A razão de uso destes flip-flops são as seguintes:

- deve-se armazenar a condição ativa de Dreq1 até que ocorra o sinal Dack1, caso contrário, se Dreq1 continuar ativo após um Dack1 um novo ciclo de DMA é gerado imediatamente. Isto sempre ocorre no modo de transferência única da 8237.
- devido ao fato do sinal Dack ser extremamente rápido em relação ao ciclo de máquina do microcontrolador é necessário fazer o armazenamento deste sinal para leitura segura por parte da 8751, caso contrário este sinal poderá nunca ser reconhecido.

Um circuito de reset-ao-ligar formado por D1 e C4 é ligado a outra entrada da porta

and. Este circuito tem a função de garantir o nível 0 em Dreq1 durante o transitório de energização do hardware, evitando uma solicitação falsa de transferência. O circuito de teste foi alimentado pela própria tensão disponível no slot do PC evitando assim transitórios indesejáveis durante a energização.

A movimentação do dado presente na entrada do buffer 741s244 só será efetivada quando o sinal $\overline{\text{IORD}}$ gerado pela controladora 8237 juntamente com o sinal Dack1 estiverem ativos. Estes sinais atuam na porta or (u3a) e a sua saída habilita a passagem dos dados pelo buffer. Desta forma o dado fica presente no barramento sendo transferido para a memória do PC uma vez que neste instante está sendo gerado o endereço de memória pela controladora 8237 não havendo conflito de dados no bus.

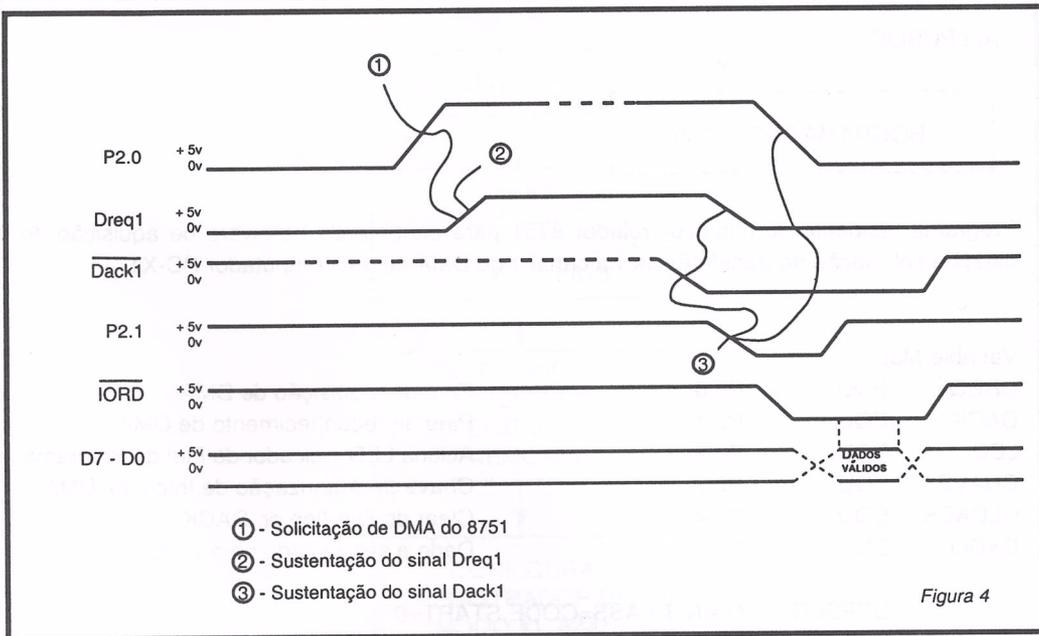
Quando o hardware periférico é energizado ou ocorre um reset, via chave sw1, o microcontrolador irá executar uma rotina de auto-inicialização configurando os seus registradores internos. A efetiva transferência apenas é realizada quando for acionada a chave sw2 (P2.3 = 0) que serve como autorização para início das mesmas, e sua conclusão é sinalizada através da linha P2.2 = 0 de modo a acionar o led D1, indicando o término de transferências.

A figura 4 a seguir ilustra um diagrama de tempo que representa a sinalização

necessária para a transferência de um byte para a memória do PC. São mostrados os

sinais utilizados pelo hardware periférico em sincronismo com os sinais de DMA.

DIAGRAMA DE TEMPO PARA TRANSFERÊNCIA DE DADOS



4 - SOFTWARE RESIDENTE NO MICROCONTROLADOR

Este software executa basicamente os seguintes passos:

- programação do timer interno;
- reconhecimento de autorização para início das transferências;
- envio do dado para o buffer 3-state;
- ativar o sinal Dreq1;
- reconhecimento do sinal $\overline{\text{Dack1}}$;
- início de novo ciclo de DMA;
- finalização das transferências.

O timer interno será responsável pela marcação de tempo para que periodicamente sejam feitas as transferências dos dados para a memória do PC. Nesta experiência o timer 0 do 8751 é inicializado para operar no modo 2 (timer de 8 bits com re-load) de modo que a cada 20µs seja feita uma leitura do dado do processo e uma solicitação de transferência do mesmo. O registrador data pointer da 8751 servirá como contador do número de transferências sendo que a cada uma ele é decrementado, finalizando as transferências quando o seu valor for zero. O timer ficará temporizando até que haja o transbordo do seu contador. Enquanto isto não ocorre, o programa fica em loop testando o bit de timer flag 0 (que não irá gerar interrupção, pois não estarão habilitadas). Deve ser enfatizado que o projetista pode programar qualquer intervalo de tempo para a aquisição do dado externo desde que este tempo não seja inferior ao

tempo necessário para execução das instruções do programa pelo microcontrolador, antes deste testar a condição do timer flag 0.

As figuras 5 e 6 mostram respectivamente o fluxograma e a listagem do programa residente no microcontrolador.

FLUXOGRAMA DO PROGRAMA RESIDENTE NO MICROCONTROLADOR

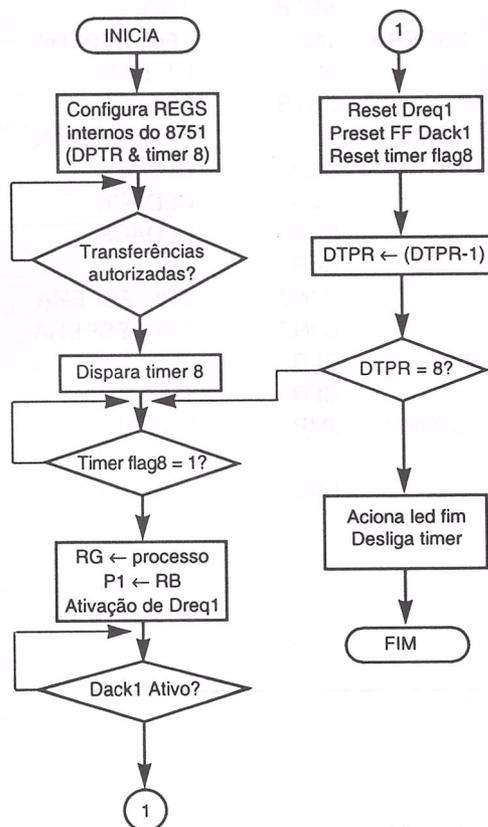


Figura 5

* ALLPUBLIC

PROGRAMA TESTEDMA

Programa residente no microcontrolador 8751 para controle do hardware de aquisição de dados e solicitação de transferência via canal 1 de DMA de um computador PC-XT.

Variable Map

DREQ	EQU	P2.0	:	Pino de requisição de DMA
DACK	EQU	P2.1	:	Pino de reconhecimento de DMA
LED	EQU	P2.2	:	Aciona LED indicador de Fim de Programa
CHAVE	EQU	P2.3	:	Chave de Autorização de Início de DMA
CLDACK	EQU	P2.4	:	Clear do Flip-flop do DACK
DADO	EQU	O5AH	:	Dado a ser enviado para o PC

DEFSEG MAIN, CLASS=CODE, START=0
SEG MAIN

INÍCIO:	CLR	DREQ	:	Zera bit de DMA REQUEST
	CLR	CLDACK	:	Reseta Bit de DACK
	SETB	CLDACK	:	
	MOV	IE, #0H	:	Desabilita Interrupções
	MOV	DPTR, #OFFFH	:	Inicializa DPTR com 255 bytes
	MOV	TMOD, #0F2H	:	Inicializa TIMER 0 no modo 2
	MOV	TH0, #0ECH	:	Carrega TIMER 0 para 20 us
	MOV	TLO, TH0	:	
TESTEI:	JB	CHAVE, TESTEI	:	Aguarda autorização Início DMA
	SETB	TRO	:	Dispara Timer 0
ESPERA:	JNB	TFO, ESPERA	:	Espera Timer 0 Overflow
	MOV	P1, DADO	:	Habilita dado 74LS244
	SETB	DREQ	:	Requisição de DMA
ACKNL:	JB	DACK, ACKNL	:	Espera reconhecimento de DMA
	CLR	DREQ	:	Fim de requisição de DMA
	CLR	CLDACK	:	Reseta Bit de DACK
	SETB	CLDACK	:	
	CLR	TFO	:	Zera Timer Flag
	DJNZ	DPL, ESPERA	:	
	DJNZ	DPH, ESPERA	:	
FIM:	CLR	LED	:	Sinaliza fim DMA
	CLR	TRO	:	Para contagem
LOOP1:	JMP	LOOP1	:	FIM PROGRAMA
		END		

Figura 6

FLUXOGRAMA PARA PROGRAMAÇÃO DA CONTROLADORA DE DMA - 8237



Figura 7

comment +★★★★ PROGRAMA CONFIG_8237 ★★★★★

Programa para configuração do canal #1 do controlador de DMA – 8237 no PC-XT.

data: 03-11-91

configura os seguintes registradores do 8237:

- Registrador de endereço base #1
- Registrador de contagem base #1
- Registrador de máscara de DMA #1
- Registrador de modo #1

Será utilizado o segmento 03 de memória para armazenamento dos dados transferidos do processo via ports 0 e 1 de um microcontrolador 8751 +

```
r modo equ 00BH
r mask equ 00AH
r counter equ 003H
r end_base equ 002H
r seg equ 083H
seg_data equ 003H
```

código segment

assume cs: código, ds: código, es: código, ss: código

org 0100H

config proc near

```
mov AL,55H : configura registrador de modo #1
out rmodo,AL
```

```
mov AL,0FFH : configura registrador contador #1
out rcounter,AL
mov AL,0FFH
out rcounter,AL
```

```
mov AL,00H : configura registrador de endereço base #1
out rend_base,AL
mov AL,00H
out rend_base,AL
```

```
mov AL,seg_data : configura segmento para canal #1
out rseg,AL
```

```
mov AL,01H : configura registrador de máscara #1
out rmask,AL
```

```
mov AX,4C00H
int 21H
```

config endp

código ends

end config

BIBLIOGRAFIA

- The 8086 Family User's Manual - Intel corporation - 1989.
- SAB 8051 - User's Manual - Siemens - 1981.
- HOLZNER, Steven - Linguagem Assembly Avançada - Ed. McGraw-Hill - 1990.
- Microsoft Macro Assembler 5.1 - Programmer's Guide - Microsoft - 1987.
- SALVADOR, Raphael M. - IBM PC-XT - Hardware -Ed.Erica - 1988.
- The TTL Data Book - Texas Instruments - 1981.
- Manual de Manutenção em Laboratório - I-7000PC-XT - Itautec-1987.
- SANTOS, Jeremias - Programando em Assembler 8086/8088-IBM-PC-Ed. McGraw-Hill - 1989.
- PEREIRA JÚNIOR, Vidal. - Microcontrolador 8051 - Hardware e Software - Ed. McGraw-Hill - 1990.