

EXPERIÊNCIA 1:
FAMILIARIZAÇÃO COM A PLACA EXPERIMENTAL DE
MICROCOMPUTADOR

Autor: Prof. Dr. Carlos Eduardo Cugnasca

Versão: 08/2001

1. OBJETIVO

Esta experiência tem como objetivo a familiarização com o microprocessador MC 68010, da Motorola, bem como os recursos existentes para a sua utilização neste laboratório.

2. INTRODUÇÃO

De forma a permitir uma familiarização adequada com o microprocessador e seus recursos, planejou-se a execução desta experiência em duas etapas:

- 1ª. Etapa:
 - apresentação do microprocessador e suas características fundamentais;
 - apresentação de seus manuais descritivos e literatura complementar;
 - apresentação do ambiente de utilização do microprocessador;
 - execução de programas de demonstração.
- 2ª. Etapa:
 - utilização dos principais recursos vistos na 1ª Etapa, através do teste de programas desenvolvidos pelos alunos.

Complementando esta apostila serão utilizados manuais, livros e outros materiais citados na bibliografia.

3. O MICROPROCESSADOR MC 68010

O MC 68010 é um microprocessador de 16/32 bits desenvolvido pela empresa americana Motorola, constituindo-se como uma extensão do microprocessador MC 68000. As informações básicas a respeito desse microprocessador podem ser encontradas em [1], que é um manual que descreve vários microprocessadores com características comuns da família do MC 68000.

Em sua seção 1, este manual orienta o leitor a obter as informações de seu interesse. Para este curso é suficiente apenas a familiarização com o microprocessador MC 68010.

Em um primeiro estudo, o aluno deve se concentrar apenas nos aspectos fundamentais do componente, tais como as suas características básicas, arquitetura, organização interna e conjunto de instruções, apresentados nas seções de 1 a 5 de [1].

Em particular, a programação do microprocessador deve ser objeto de um rápido aprendizado por parte do aluno, pois isso lhe permitirá utilizar imediatamente o componente, através dos recursos de apoio existentes no laboratório. Futuramente está planejada a familiarização com os seus demais recursos.

Por se tratar de um microprocessador poderoso, o MC 68010 apresenta recursos flexíveis de programação. Entretanto, pode-se desenvolver muitos programas utilizando-se uma pequena parte desses recursos, ou seja, as instruções e modos de endereçamento convencionais, deixando o restante para ser mais bem conhecido e utilizado quando o aluno já tiver alguma desenvoltura no uso do microprocessador. Complementando a parte de programação, tem-se [2] e [3].

4. A PLACA EXPERIMENTAL DO MC 68010

Para viabilizar uma rápida familiarização e uso do MC 68010 foi desenvolvida uma Placa Experimental, denominada *MD68k* (ou "*Kit*"), que utiliza uma série de componentes básicos que compõem a família do microprocessador, e outros recursos que permitem a sua conexão com placas ou circuitos externos.

Com o intuito de agilizar a interação do usuário com a Placa Experimental, nela foi incorporado o "Programa Monitor", e prevista a sua conexão com um "Terminal de Vídeo". Esses recursos são de fundamental importância prática, pois viabilizam o carregamento, a execução e a depuração de programas.

As características principais da Placa Experimental são detalhadas nos próximos itens.

4.1. Descrição da Placa Experimental MD68k

A Placa Experimental MD68k contém os componentes básicos para a constituição de um microcomputador de 16/32 bits. Todos os componentes deste sistema estão dispostos em uma única

placa de circuito impresso, especialmente projetada para aplicações didáticas e de desenvolvimento de protótipos.

Além de possuir espaços reservados para a expansão de memória, a placa já dispõe de alguns periféricos básicos, como temporizadores, portas de entrada e saída, e canais de comunicação serial.

Para permitir a interconexão de outros circuitos e periféricos, existe um conector com os principais sinais do sistema (também conhecido por barramento), no qual poderão ser interligadas placas comercialmente disponíveis ou projetadas pelo usuário. A descrição completa da Placa Experimental MD68k se encontra em [4], itens 1 e 2, e apêndices A a H.

4.2. Programa Monitor

A Placa Experimental dispõe de um *Programa Monitor* residente que pode ser comparado, do ponto de vista funcional, ao sistema operacional de um microcomputador. Esse programa faz uso de um terminal de vídeo externo à Placa Experimental, através de uma interface de comunicação serial.

Dessa forma implementou-se uma interface "homem-máquina" que permite a execução de uma série de funções, dentre as quais:

- exame do conteúdo das posições de memória e registradores;
- armazenamento de instruções de programas em memória;
- armazenamento de dados de programas em memória e registradores;
- execução de programas e sua interrupção sob comando, com recursos de depuração;
- recebimento de pacotes de dados através da comunicação serial;
- *assembler* e *disassembler*.

Além dessas funções, o Programa Monitor dispõe de algumas subrotinas de uso geral disponíveis ao usuário. Maiores detalhes sobre o Programa Monitor podem ser encontrados em [4], item 3.

4.3. Interligação da Placa Experimental a um Terminal de Vídeo.

O Terminal de Vídeo é um equipamento já conhecido, tendo sido utilizado em outras disciplinas, como PCS308. Ele se encontra conectado ao *canal serial A* da Placa Experimental (ver [4], item 2.12).

Normalmente, um microcomputador de uso geral (por exemplo, compatível com o modelo PC da IBM), pode se comportar como um Terminal de Vídeo desde que nele se instale um programa que emule tal equipamento, e se utilize um de seus canais de comunicação serial.

Neste curso as duas soluções serão adotadas. Contudo, nas experiências futuras, a presença de um microcomputador será importante também para a geração do código dos programas por meio de montadores e compiladores.

O programa emulador de terminais de vídeo a ser utilizado é apresentado em [5].

4.4. Outro Ambiente de Desenvolvimento

Uma maneira eficiente e poderosa de se desenvolver projetos baseados em microprocessadores envolve o uso de um "ICE - In-Circuit Emulator", que dispõe de muitos recursos para a depuração do hardware e software. Trata-se de um equipamento de alto custo e que requer muitos conhecimentos e preparo para sua utilização.

Basicamente, tal equipamento dispõe de um cabo com conector, que deve ser colocado no lugar do microprocessador, e recursos de hardware e software que se comportam como o microprocessador, mas com maior controle sobre as operações. Devido ao fato de apresentar alto custo e de ser específico para cada microprocessador, na prática tal equipamento é utilizado principalmente por empresas que desenvolvem muitos projetos baseados em um mesmo microprocessador.

5. BIBLIOGRAFIA

- [1] MOTOROLA. M68000 - 8 -/16-/32-Bit Microprocessors User's Manual. N.J., EUA, Prentice-Hall, 6ª edição, 1989.
- [2] MOTOROLA. M68000 - 16/32-Bit Microprocessor Programmer's - Reference Manual. N.J., EUA, Prentice Hall, 4a. edição, 1984. 218p.
- [3] SHIMIZU, T. Programação Assembler para Microprocessadores 68000, 68010, 68020. McGraw-Hill, São Paulo - Brasil, 1987.
- [4] YAGIMA, L.C. & CUGNASCA, C.E. Manual da Placa Experimental do Microprocessador MC 68010. São Paulo, LSD/PCS/EPUSP, 1991.
- [5] Instruções/comandos do Programa XTALK.
- [6] MOTOROLA. M68000 Family Reference. EUA, 1988.

- [7] CRAMER, W. & KANE, G. 68000 Microprocessor Handbook. Osborne McGraw-Hill, Berkeley, CA, EUA, 1986.
- [8] CUGNASCA, C.E. & ZERBINI, R.C. Experiência nº 1 - Familiarização com a Placa Experimental MD68k de Microcomputador. EPUSP, 1989.
- [9] CLEMENTS, A. Microprocessor Systems Design. 2nd edition. ITP, EUA. 1992.
- [10] MOTOROLA. www.motorola.com.

6. PARTE PRÁTICA

6.1. 1ª Etapa

- a) Leia os itens 3, 4 e 4.1 desta apostila, e estude os itens da bibliografia relacionados.
- b) Com a Placa Experimental **desligada**, efetue um reconhecimento dos seus principais componentes, de acordo com o apresentado na literatura.
- c) Leia os itens 4.2 e 4.3 desta apostila, juntamente com os itens citados da literatura.
- d) Colocação da Placa Experimental em Operação:
 - identificar o cabo de alimentação da Placa Experimental e interligá-lo à fonte de alimentação, previamente calibrada para as tensões + 5 Volts, + 12 Volts e - 12 Volts (use um multímetro para verificar os valores exatos das tensões).

Atenção:

Não ligar os equipamentos sem a presença do professor!

A inversão de algum fio danificará seriamente a Placa Experimental.

- identificar o cabo de comunicação do microcomputador que operará como terminal de vídeo, e conectá-lo ao *canal A* da Placa Experimental.
- ligar o microcomputador.
- localizar o *programa emulador de terminal*, de acordo com a orientação de seu professor, e executá-lo.
- configurá-lo, conforme descrito em [5].
- ativar a emulação de Terminal de Vídeo.

- ligar a fonte de alimentação da Placa Experimental, e pressionar o botão "RESET", observando a mensagem enviada à tela.
- e) Exercite o uso dos comandos básicos do Programa Monitor: *exame e alteração de posições de memória e registradores*, conforme descrito no capítulo 3 de [4].
- f) Carregue e execute cada um dos programas - exemplo apresentados no apêndice I. Procure exercitar o uso de comandos de execução passo-a-passo de programas e de colocação de "break-points". Para diversos programas, será necessária a prévia colocação de valores convenientes em registradores e posições de memória, e posteriormente verificar suas alterações como efeito da execução dos programas.
- g) Utilize o botão "ABORT", e constate a diferença entre ele e o botão "RESET".

6.2. 2ª Etapa

- a) Elaborar um programa para gerar uma forma de onda quadrada, em uma porta de saída do microcomputador.
- b) Para programar a porta de saída, seguir a orientação do apêndice II.
- c) Escrever o programa em mnemônicos, com comentários.
- d) Carregar o programa na memória da Placa Experimental e testá-lo.
- e) Utilizar o osciloscópio para ver a forma de onda. Anotar os resultados.
- f) Alterar o programa para variar o período da forma de onda.

Observações:

- 1) *A frequência de operação do microprocessador na Placa experimental é de 6,144 Mhz.*
- 2) *A placa Experimental possui o número de "wait states" configurável por meio de "jumpers". Obtenha esse número por inspeção antes de efetuar os cálculos.*
- 3) *Maiores informações podem ser obtidas em [4].*

APÊNDICE I - PROGRAMAS-EXEMPLO

1) LIMPEZA DE REGISTRADORES

- preencha a tabela abaixo, gerando os códigos e endereços (em hexadecimal) para o programa, considerando como endereço inicial 1000H.
- carregue o programa na memória.
- coloque valores diferentes de 0 nos registradores D0 a D2.
- execute uma-a-uma as instruções de 1 a 3, observando os resultados.
- carregue o valor 1100H em A0, e um valor diferente de zero nas posições de memória de 1100H a 110FH.
- execute a instrução 4 e verifique os resultados.
- carregue o valor 1101H em A0 e execute novamente a instrução 4. Analise os resultados.
- carregue o valor 1101H em A0, um valor qualquer na posição de memória 1101H e execute a instrução 5.
- carregue o valor 1106H em A1 e A2, e um valor diferente de zero nas posições de memória de 1100H a 110FH. Execute uma-a-uma as instruções 6 e 7.
- determine o valor que A0 deve receber para que as posições 1104H e 1105H sejam limpas, executando-se a instruções 8.
- idem, para D0 e A0, na instrução 9.
- execute as instruções 10,11, e 12, comentando os resultados.

nº	endereço	código	mnemônico	operandos	comentários
1	H		CLR.B	D0	
2	H		CLR.W	D1	
3	H		CLR.L	D2	
4	H		CLR.W	(A0)	
5	H		CLR.B	(A0)	
6	H		CLR.W	-(A1)	
7	H		CLR.W	(A2)+	
8	H		CLR.W	\$5(A0)	
9	H		CLR.W	\$7(A0,D0.W)	
10	H		CLR.W	\$1102	
11	H		CLR.W	\$110000	
12	H		CLR.W	\$001100	

2) SUBROTINA DE LIMPEZA DE UMA REGIÃO DE MEMÓRIA

A subrotina **LIMP** coloca zeros em uma região de memória, desde o endereço carregado em A3, até o endereço carregado em A4, sendo $(A4) \geq (A3)$. Seguindo a metodologia utilizada no exemplo anterior, carregar e testar a subrotina, escolhendo uma região de memória conveniente. Chamá-la de um programa principal fictício.

Obs.: não se esqueça de substituir os labels do programa por endereços da memória

	BSR	LIMP
	NOP	
	.	
	.	
	.	
LIMP	CLR.B	(A3)+
	CMPA.L	A4,A3
	BLS	LIMP
	RTS	

3) OPERAÇÕES DE SOMA - EXEMPLO

Executar as instruções abaixo e descrever os resultados.

MOVE.W	#\$0100,D0
ADD.W	#\$0100,D0
ADD.B	#\$FE,D0

4) SUBROTINA DE SOMA DE DADOS SEM SINAL

A subrotina abaixo soma 5 dados tipo word, armazenados seqüencialmente na memória, a partir do endereço carregado em A0. O resultado (long word) é armazenado em D0.

SOMA	MOVE.B	#5,D2
	CLR.L	D0
	CLR.L	D1
LOOP	MOVE.W	(A0)+,D1
	ADD.L	D1,D0
	SUBQ.B	#1,D2
	BNE	LOOP
	RTS	

5) SUBROTINA DE SOMA DE DADOS COM SINAL

A subrotina do exemplo anterior pode ser alterada para levar em conta o sinal dos operandos:

```

SOMASI  MOVE.B    #5,D2
        CLR.L    D0
LOOP    MOVE.W    (A0)+,D1
        EXT.L    D1
        ADD.L    D1,D0
        SUBQ.B   #1,D2
        BNE     LOOP
        RTS

```

6) CÁLCULO DO QUADRADO DA DIFERENÇA

A subrotina abaixo calcula a expressão $C = (A - B)** 2$, onde $0 \leq B \leq A \leq 7FFFH$. D0 contém B, D1 contém A, e D2 contém C

```

QUAD    MOVE.W    D1,D2
        SUB.W    D0,D2
        MULU    D2,D2
        RTS

```

7) PESQUISA EM FAIXA

O trecho de programa abaixo determina se o número X esta na faixa 0-1000H. Caso ele não esteja, o programa deve desviar para o endereço ERRO. A faixa de valores de X é 0-7FFFH, e o seu valor esta armazenado nas posições de memória 1100H e 1101H.

```

        MOVE.W   $1100,D0
        BMI     ERRO
        CMPL.W  #1000,D0
        BHI     ERRO
        .
ERRO    .
        .

```

8) MOVIMENTAÇÃO DE BLOCOS - EXEMPLO

Colocar valores conhecidos e convenientes em An e Dn, e executar cada uma das instruções abaixo, analisando os resultados. Para a última instrução, colocar em A0 o valor 1100H.

```

MOVEM    D0/D5/A0-A2,-(A4)
MOVEM    (A6)+,A0-A6/D0-D7
MOVEM    A1/A5/D3,$1100
MOVEP.L  D0,2(A0)

```

9) SUBROTINA DE MOVIMENTAÇÃO DE BLOCOS

A subrotina abaixo movimenta um bloco de dados armazenados a partir de um endereço-fonte (A0), para um endereço-destino (A1). D0 contém o número de long words a ser movimentado.

```

MOBL     MOVE.L    (A0)+,(A1)+
          SUBQ.W    #1,D0
          BNE      MOBL
          RTS

```

10) SOMA DE CADEIAS

O trecho de programa abaixo soma duas cadeias de palavras, apontadas por A0(+2) e por A1(+2). A soma é armazenada em A1. A2 contém o endereço do fim dos dados da cadeia. Observe a utilidade em se ter os flags X e C separados.

```

          ANDL.B    #$EF,SR
LOOP     ADDX.W    -(A0),-(A1)
          CMPA.L    A2,A1
          BHI      LOOP
          .
          .

```

Observações:

O programa monitor não aceita a primeira instrução deste programa na forma ANDL.B #\$EF,CCR (exclusiva do MC 68010), pois somente reconhece instruções do MC 68000.

Ao testar o seu programa, execute-o passo a passo (comando T – trace).

Caso você queira disparar o programa (por exemplo, G1000), executando-o por inteiro, terá problemas, pois o retorno ao programa monitor não é automático. Nesse caso serão executadas “instruções” contidas nas posições subseqüentes da memória, podendo causar diversos tipos de erros.

APÊNDICE II - PROGRAMAÇÃO DA PORTA PARALELA

É apresentada a seguir, a programação de uma das portas de saída da Placa Experimental, para uso na 2ª Parte da experiência. Maiores detalhes sobre o seu funcionamento podem ser obtidos na literatura, e serão vistos também em experiências futuras.

O desenho e número dos pinos do conector da porta de saída são apresentados em [4].

- **Programação do canal 2, porta A:**

- deve ser feita apenas uma vez, no início do programa.
- firma típica:

MOVE.B	#\$00,\$60042	seleciona o "control register" da porta A
MOVE.B	#\$FF,\$60040	programa a porta A como saída
MOVE.B	#\$3C,\$60042	seleciona o registrador de dados para a porta A

- **Envio de dados para o canal 2, porta A:**

- após a programação, sempre que se desejar enviar dados para a porta A, executar uma instrução do tipo:

MOVE.B	_____, \$60040	envia dados para a porta A
	_____	diversas opções possíveis

onde o dado a ser enviado pode ser especificado de diversas formas, como por exemplo:

MOVE.B	#\$01,\$60040	envia o valor 1 para o bit zero da porta, e o valor 0 para os demais
---------------	----------------------	---