

Universidade do Minho
Departamento de Informática

Arquitectura de Computadores II
LESI

Bancada de Ensaios PG01/UM

Abril/2001

João Luís Sobral, Abril 2001
António Pina, Paulo Alves, Abril 2000

Índice

1 Introdução.....	2
2 Expansão do Barramento.....	3
3 Sistema de Desenvolvimento CID700.....	3
4 Periférico Genérico PG01.....	4
5 Painel de Comando.....	6
Exercício 1: Portas de Entrada e de Saída.....	7
Exercício 2: Portas de Controlo/Estado.....	8
Exercício 3: Conversores A/D e D/A.....	9
Exercício 4: E/S através de Posições de Memória.....	10
Exercício 5: Desvio de Interrupções.....	11

1 Introdução

A Bancada de Ensaios destina-se a explorar as facilidades de expansão do barramento dos PCs, utilização de portas de Entrada/Saída, sistema de interrupções e programação do controlador de DMA para transferência de dados directa entre periféricos e memória.

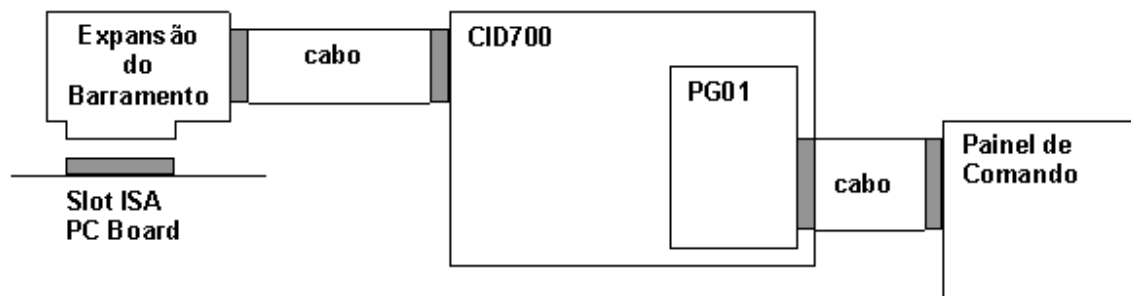


Figura 1: Diagrama de blocos da Bancada de Ensaios

A Bancada de Ensaios é composta por:

- Expansão do Barramento
- Sistema de Desenvolvimento CID700
- Periférico Genérico PG01
- Painel de Comando

Para cada um dos módulos é apresentado de seguida a sua finalidade e características.

2 Expansão do Barramento

O acesso, por parte de periféricos, ao barramento de um PC é feito através das *slots* (conectores para encaixe de placas de circuito impresso) existentes na placa mãe do PC. A sua acessibilidade é reduzida, pelo que é necessário desligar electricamente o computador, abrir a caixa e encaixar/desencaixar a placa e ligar de novo o computador. Quando se trata de desenvolvimento protótipos este processo torna-se penoso ao longo das diversas fases de teste, detecção de anomalias e correcção.

A placa de Expansão do Barramento destina-se unicamente a colocar fora da caixa do computador todas as linhas do barramento isolando no entanto os sinais eléctricos de modo a que erros no exterior não deteriore os componentes do interior do PC. Isto é feito recorrendo a tampões que não modificam o comportamento lógico dos sinais mas que tem a particularidade de colocar atrasos na propagação dos sinais. Normalmente estes atrasos não são significativos.

Esta placa encaixa directamente num conector ISA (8 bits) de um PC e através de um cabo estabelece a ligação entre o Sistema de Desenvolvimento e o barramento do PC.

3 Sistema de Desenvolvimento CID700

Este equipamento foi construído com o intuito de auxiliar o analista/arquitecto de sistemas a conhecer o barramento do PC e a desenvolver circuitos de interface para usos variados.

Consiste de uma fonte de alimentação comutada e várias placas de ensaios onde se podem encaixar directamente circuitos integrados e interligá-los por fio aos sinais do barramento.

No âmbito desta disciplina, é a funcionalidade e capacidade de reutilização de componentes já construídos que interessa analisar. Pormenores relativos ao desenho e projecto de sistemas não se enquadram nos objectivos imediatos da disciplina.

Assim, este sistema serve apenas de base a uma placa desenvolvida na Universidade do Minho – PG01 – e que recolhe deste todos os sinais eléctricos (alimentação e linhas do barramento).

4 Periférico Genérico PG01

Esta placa que encaixa directamente no sistema de desenvolvimento CID700 destina-se a apoiar as aulas laboratoriais de estudo do interface do micro-computador IBM PC compatível com diversos periféricos.

Tal como o nome indica, comporta-se como um periférico vulgar possuindo:

- Uma porta de saída universal de 8 bits;
- Uma porta de entrada universal de 8 bits;
- Uma porta de saída dedicada a um visor de 7 segmentos com ponto decimal;
- Um conversor Analógico/Digital (A/D) de 8 bits;
- Um conversor Digital/Analógico (D/A) de 8 bits;
- Um módulo programável, com 11 pinos de E/S e um conjunto de ligações ao barramento, pré-definidas, de modo a poder utilizar a interrupção IRQ3 e o canal 3 de DMA.

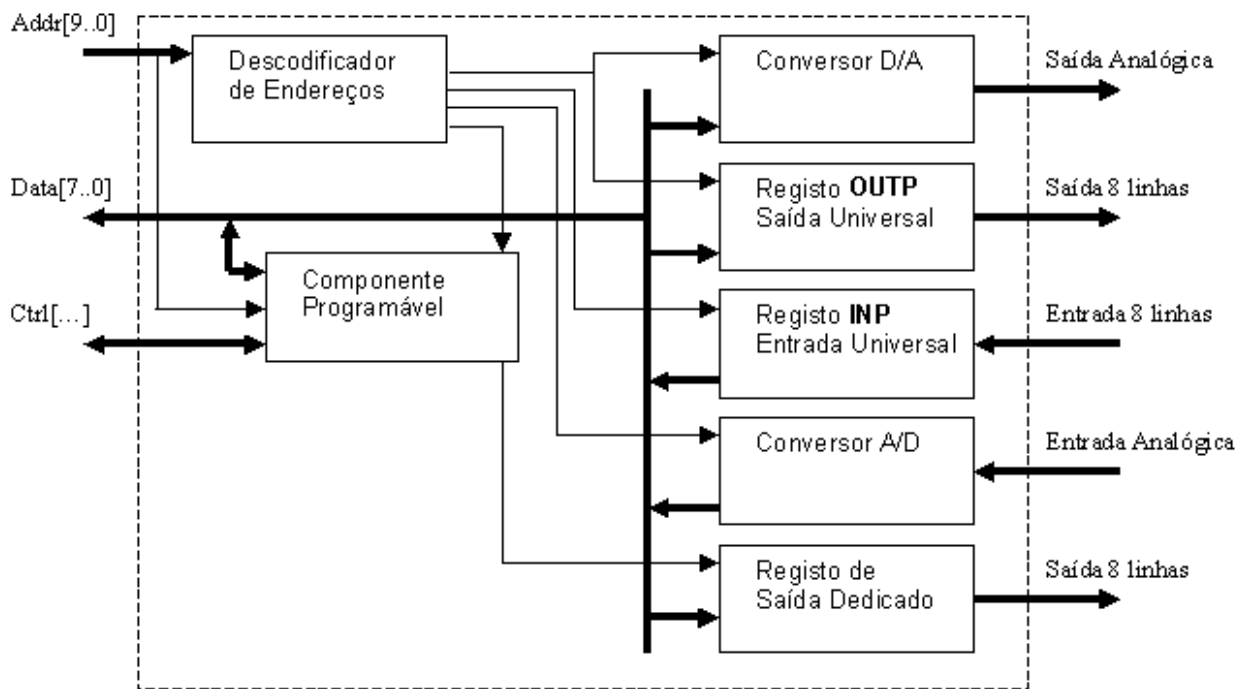


Figura 2: Diagrama de blocos do PG01

Os sinais do lado esquerdo do diagrama da figura 2 e as linhas de alimentação (não representadas), são recolhidos do sistema de desenvolvimento CID700. Os sinais do lado direito do diagrama são ligados por cabo ao Painel de Comando.

Os endereços da área de E/S atribuídos a cada um dos elementos do PG01 (registos e conversores) estão definidos na tabela 1.

Endereço E/S	LEITURA	ESCRITA
300h	ESTADO: Bit0 activo a '0' Final da conversão A/D	DADOS: Valor a colocar no visor de 7-segmentos
304h	DADOS: Lê o valor do conversor A/D	COMANDO: Bit0 activo a '1' Inicia conversão A/D
308h		DADOS: Valor a colocar na saída OUTP Valor para o conversor D/A
30Ch	DADOS: Lê o valor na entrada INP	

Tabela 1: Mapa de Endereços E/S do PG01

O conversor A/D é comandado pelo bit0 do endereço 304h. Logo que é colocado o valor '1' neste bit é iniciada uma nova conversão. Quando o conversor acaba de efectuar uma conversão assiná-la com um '0' no bit0 do endereço 300h. Simultaneamente, activa a linha IRQ3. Normalmente, o pedido de interrupção nº 3 está mascarado pelo que o processador ignora este pedido. Se se pretender proceder à recolha de valores do conversor pela técnica de interrupção em vez de sondagem, basta desmascarar o pedido de interrupção nº 3 do controlador de interrupções.

O PG01 foi desenvolvido na Universidade do Minho e tem sido objecto de revisão, redefinição e reimplementação ao longo do tempo. Sugestões são bem vindas e podem vir a fazer parte da próxima revisão.

5 Painel de Comando

Este equipamento consiste basicamente num conjunto de elementos terminais – comandos manipuláveis pelo utilizador juntamente com um visor de 7-segmentos e um conjunto de visualizadores (LED's) tal como a figura abaixo ilustra:

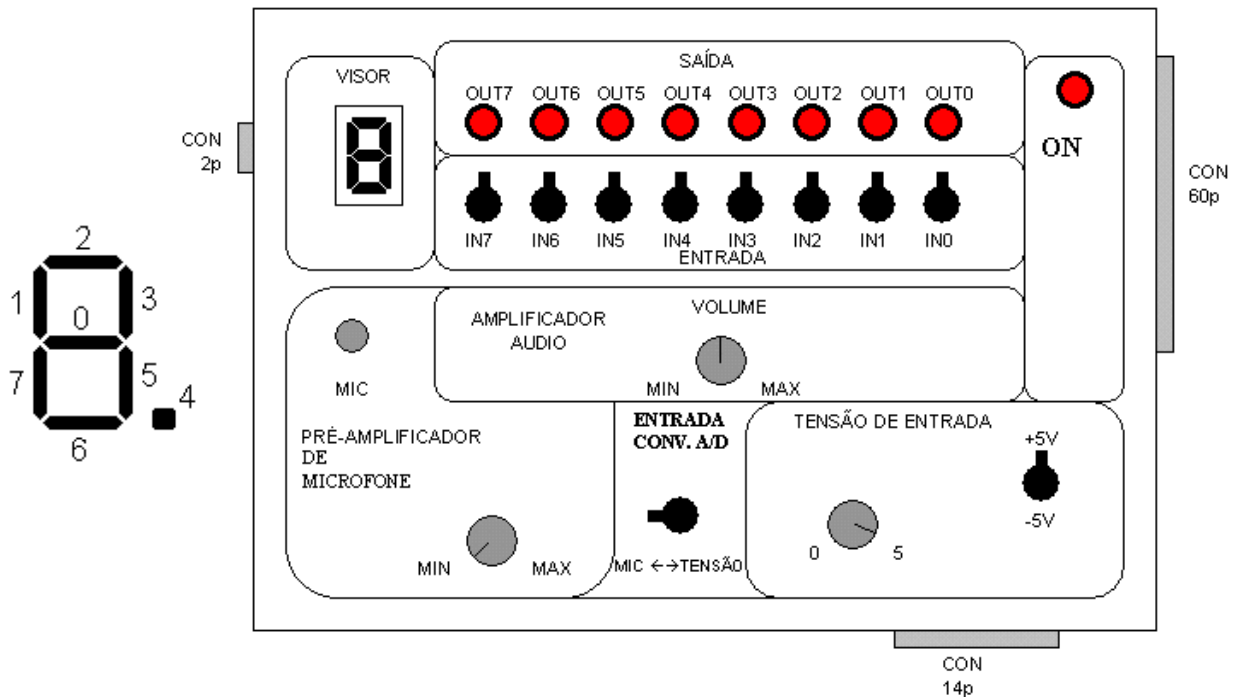


Figura 2: Painel de Comando

Elementos:

- Fonte de tensão regulável de $-5V$ a $0V$ e de $0V$ a $+5V$. Este elemento possui dois comandos: um **potenciômetro** para variar a tensão e um **comutador** para seleccionar tensão positiva ($0V$ a $+5V$) ou tensão negativa ($-5V$ a $0V$);
- Microfone ligado a um pré-amplificador com ganho variável, regulável por um **potenciômetro**;
- Amplificador de saída para ligar a um altifalante com controlo de volume por um **potenciômetro**;
- Um **comutador** que selecciona a entrada analógica entre a fonte de tensão e o pré-amplificador do microfone;
- Um conjunto de **8 visualizadores** vulgarmente conhecidos por *leds*.
- Um conjunto de **8 interruptores**
- Um **visor de 7-segmentos** com ponto decimal.

A saída OUTP do PG01 é ligada às linhas do conjunto de visualizadores. Assim, o bit0 corresponde ao Led0, o bit1 ao Led1 e assim sucessivamente.

A entrada INP do PG01 é ligada às linhas do conjunto de interruptores. Assim, o bit0 corresponde ao Sw0, o bit1 ao Sw1 e assim sucessivamente.

A saída analógica do PG01 é ligada directamente à entrada do amplificador no Painel de Comando. A entrada analógica do PG01 é ligada através dum comutador, ou à fonte de tensão variável, ou, à saída do pré-amplificador do microfone.

A saída do registo dedicado do PG01 é ligada às linhas do visor. A correspondência entre os bits do registo e os segmentos do visor é apresentada na figura 2.

Exercício 1: Portas de Entrada e de Saída

Objectivos

- Acesso ao espaço de endereços de entrada e saída
- Programação básica de um dispositivo periférico
- Interface em Linguagem C

Exercício

Considere as seguintes ligações do PG01/UM para o exterior:

- **registo de saída**
 - 8 leds
 - L7..L0
 - bit a 1 - acende
 - bit a 0 - apaga
- **registo de entrada**
 - 8 interruptores
 - I7..I0
 - ligado - bit 1
 - desligado - bit 0

Escreva os programas que realizam as seguintes operações:

- Mostrar o valores dos interruptores, da esquerda para a direita: 1) no écran 2) nos *leds*.
- Acender e apagar todos os *leds*: 1) ao mesmo tempo 2) um a um, da esquerda para a direita e vice-versa.

Exercício 2: Portas de Controlo/Estado

Objectivos

- Fazer a distinção entre os diversos tipos de registos num periférico
- Comandar um periférico através de técnicas de sondagem

Exercício

Considere que às ligações do PG01/UM (no exercício anterior) juntamos:

- **visor**
 - 7 segmentos (ver fig [14](#))
 - bit a 1 - acende *led*
 - bit a 0 - apaga *led*
- **conversor A/D**
 - escala de -5 a 5 Volt
 - cada unidade 0,04 Volt
- **entrada de tensão**
 - pino conversor (ANA-IN)
 - potenciómetro
 - variação tensão

Utilize as técnicas de controlo de periféricos por sondagem para realizar as seguintes operações:

- a) Dar início à conversão A/D e mostrar o valor obtido no ecrã.
- b) Repetir o exercício, mas desta vez, mostrando a parte inteira do valor no visor de sete segmentos. Use o ponto para assinalar valores negativos.
- c) Determinar quantas conversões A/D são realizadas por segundo.

Exercício 3: Conversores A/D e D/A.

Objectivos

- Familiarização com as portas de controlo/estado e as de transferência de dados
- Construção de programas para comando de um periférico, através da técnica de sondagem (pooling).

Exercício

O conversor A/D é usado para converter sinais analógicos (valores contínuos) para sinais no formato digital (valores discretos), para posterior tratamento pelo computador. O conversor usado no PG01/UM possui uma escala de -5V a +5V, correspondendo cada unidade do valor lido a cerca de 0,04V.

O conversor D/A é usado para converter valores numéricos num sinal analógico. O conversor D/A usado no PG01/UM converte um valor numérico entre 0 e 255 num sinal analógico entre -5V e +5V.

Usando o periférico PG01/UM juntamente com o Painel de Comando escreva um programa, em linguagem C, para realizar as seguintes operações:

- a) Digitalização de um som, captado pelo microfone.
- b) Reprodução do som, previamente digitalizado, no altifalante.

Nota 1: O ganho do pré-amplificador do microfone não é muito elevado. Para obter resultados visíveis (i. e., audíveis) deverá colocar o ganho máximo do pré-amplificador do microfone e falar perto deste quando estiver a usar o conversor A/D.

Nota 2: A velocidade de reprodução do som (i.é., o número de amostras por segundo) deve ser idêntica à da digitalização. Para implementar esta característica pode utilizar o conversor A/D para definir a cadência reprodução dos valores digitalizados.

Exercício 4: E/S através de Posições de Memória

Objectivos

- Familiarização com as portas de transferência de dados mapeadas em memória.
- Construção de programas para comando de um periférico, através da técnica de sondagem (pooling).

Exercício

O controlador gráfico do PC possibilita a geração de imagens em modo gráfico, onde o tom de cada pixel da imagem é controlado através de um endereço de memória. A linha dos computadores pessoais IBM PC compatíveis permite a geração de imagens em vários modos gráficos (CGA, EGA, VGA, XGA, etc.). Cada modo gráfico possibilita uma resolução (i.é., número de pixels) horizontal e vertical, bem como um número limitado de cores para cada pixel. Por exemplo, o modo VGA possui uma resolução de 640x480, onde cada pixel está associado a um bit (imagem a preto e branco) ou a 16 bits, o que permite a utilização de 64Kcores.

Embora a resolução e o número de bits utilizados para cada pixel possa ser definido directamente em registos da placa gráfica, tal é uma tarefa complexa, sendo preferível utilizar os modos gráficos pré-definidos. Estes modos gráficos podem ser programados através de chamadas a rotinas do sistema, acedidas por interrupções. Nomeadamente, a interrupção 10h permite seleccionar o modo gráfico. Neste exercício será utilizado o modo 13h, o qual possui uma resolução de 320x200 com 8 bits por pixel. O código que permite seleccionar este modo é o seguinte (o qual pode ser introduzido directamente num programa em C):

```
asm {  
    MOV AX, 0x13  
    INT 0x10  
}
```

Neste modo gráfico, cada célula de memória corresponde ao valor utilizado por um pixel. Em aplicações MS-DOS, o endereço da memória vídeo, utilizada pelo controlador para gerar a imagem, está acessível a partir do endereço 0xA0000. O acesso a estes endereços de E/S pode ser realizado através de um apontador, declarado em C da seguinte forma:

```
unsigned char *ptr = (unsigned char far *) 0xA0000000L;  
ptr [10] = 20; // atribui ao pixel (10,0) o tom 20
```

Usando o periférico PG01/UM juntamente com o Painel de Comando escreva um programa, em linguagem C, para ilustrar graficamente a forma de onda de um som, captado pelo microfone.

Exercício 5: Desvio de Interrupções.

Objectivos

- Construção de programas de E/S através da técnica de interrupção.

Exercício

Os sistemas de operação usam interrupções para controlar o teclado. Em MSDOS sempre que é premida uma tecla é accionada a interrupção 09h. Uma rotina, instalada especialmente para atendimento daquela interrupção, é usada para obter informação sobre as teclas premidas, com salvaguarda para uso posterior. Um conjunto adicional de funções são usadas para posteriormente analisar e tratar a informação recolhida.

Uma rotina de atendimento a uma interrupção pode ser instalada através do seguinte esqueleto:

```
void interrupt (* RotinaAntiga)(); // apontador para a rotina antiga

void interrupt RotinaNova() {      // rotina a instalar
    ...                            // código da rotina nova
    (*RotinaAntiga)();             // chamada à rotina antiga
}

void main {
    RotinaAntiga = getvect(<<n°int>>); // endereço da rotina antiga
    disable();                       // inibe interrupções
    setvect(<<n°int>>,RotinaNova);    // instala a nova rotina
    enable();
    keep(0,0x1000);                 // termina o programa, ficando residente
}
```

- a) Escreva um programa, em linguagem C, usando a técnica de sondagem, que faça reflectir nos Leds a acção sobre as teclas direccionais (< e >). A função *kbhit()* permite identificar se foi premida uma tecla, enquanto a porta de E/S 0x64 contém o ASCII da última tecla premida. Considere que no conjunto de Leds do Painel de Comando se encontra sempre um e um só Led aceso. Considere no início o Led4 aceso. Pressionando na tecla “>” deve passar a acender o led à esquerda (direita) do actualmente aceso - até chegar ao extremo em que permanecerá o mesmo aceso.
- b) Adapte o programa anterior ao uso da técnica de interrupção.
- c) Adapte o programa do Exercício 3 para utilizar a técnica de interrupção. Note que a linha de interrupção IRQ 3 está associada ao vector de interrupção 0x0B.