

Arquitectura de Computadores II



3º Ano

Periférico Genérico PG04

João Luís Ferreira Sobral
Departamento do Informática
Universidade do Minho



Junho 2004

Bancada de Ensaios

● Introdução

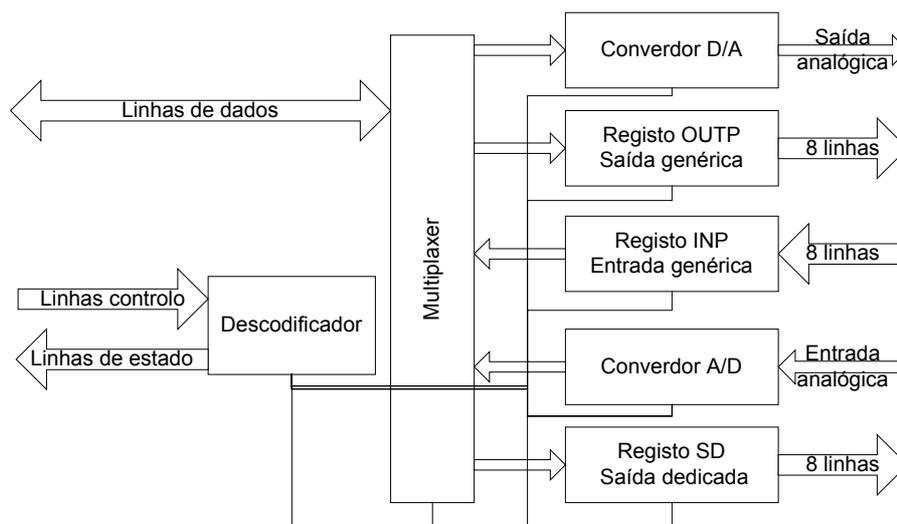
- A bancada de ensaios destina-se a apoiar as aulas laboratoriais de estudo da interface do micro-computador PC compatível com o exterior
- A bancada de ensaios é constituída por dois elementos:
 - **Periférico genérico** - inclui portas de entrada e de saída genéricas, um conversor A/D e um conversor D/A
 - **Painel de comando** - inclui um visor de 7 segmentos, 8 interruptores, 8 *leds*, uma ligação para microfone e uma ligação para um altifalante (ambas com amplificação) e uma fonte de tensão regulável
- Existem versões do periférico genérico, actualmente as duas versões em utilização ambas se ligam à porta paralelas do PC (já existiu uma versão para barramento ISA e esta em desenvolvimento versão para USB):
 - **PG04 / Versão laboratório** – Toda a funcionalidade do PG é implementada por uma CPLD, sendo esta reprogramável directamente através da porta paralela. Esta placa foi desenvolvida no DI em 2004, sendo o PCB fabricado por uma empresa especializada.
 - **PG04 / Versão alunos** – A funcionalidade é implementada através de *buffers* (74HCT244/245) e de uma PAL 16V8. Esta é uma versão de baixo custo essencialmente dirigida para os alunos que pretendam uma placa compatível com as utilizadas no laboratório.
- As duas versões anteriores partilham o mesmo painel de comando, ou seja um mesmo painel de comando pode ser utilizado em qualquer uma das duas versões.
- A ligação entre o periférico genérico e o PC é efectuada através de um cabo. O painel de comando encaixa directamente nos conectores do periférico genérico.

Periférico genérico

● Características

- Uma porta de saída universal de 8 bits (OUTP);
- Uma porta de entrada universal de 8 bits (INP);
- Uma porta de saída dedicada de 8 bits (SD);
- Um conversor Analógico/Digital (A/D) de 8 bits;
- Um conversor Digital/Analógico (D/A) de 8 bits;
- Um módulo programável, com 6 pinos de E/S e um conjunto de ligações pré-definidas (apenas na versão CPLD).

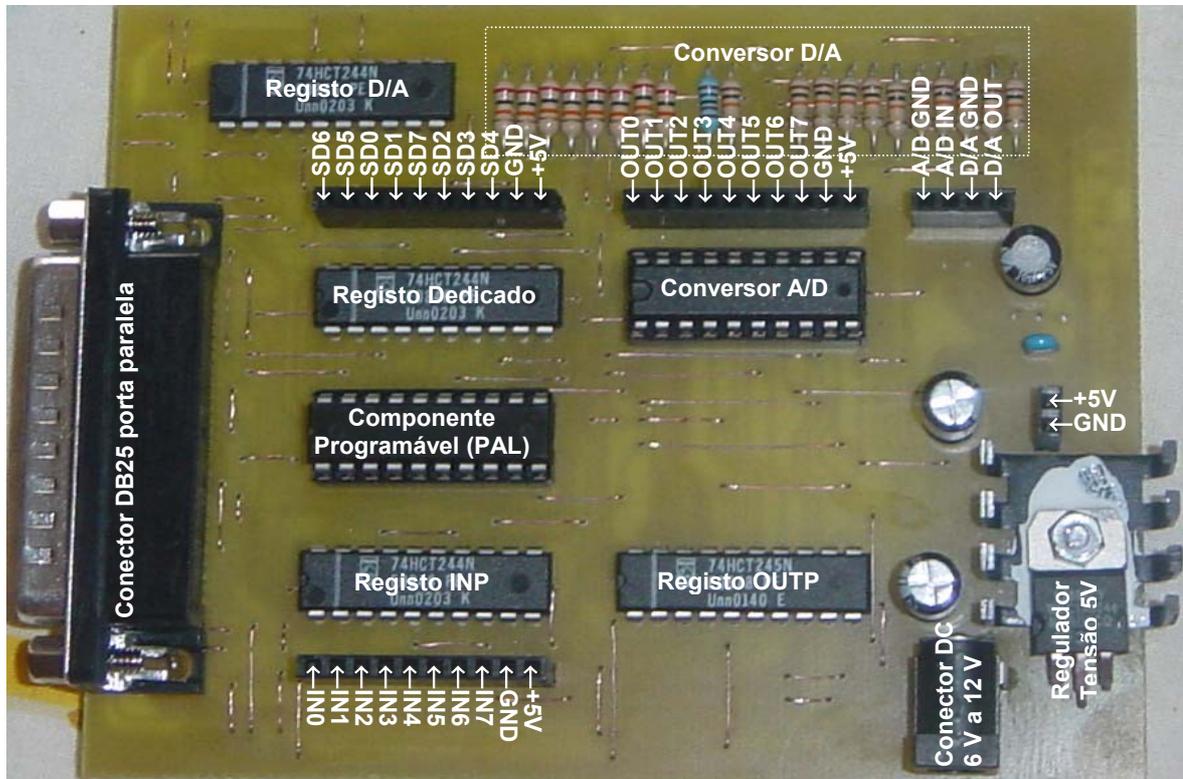
● Diagrama de blocos



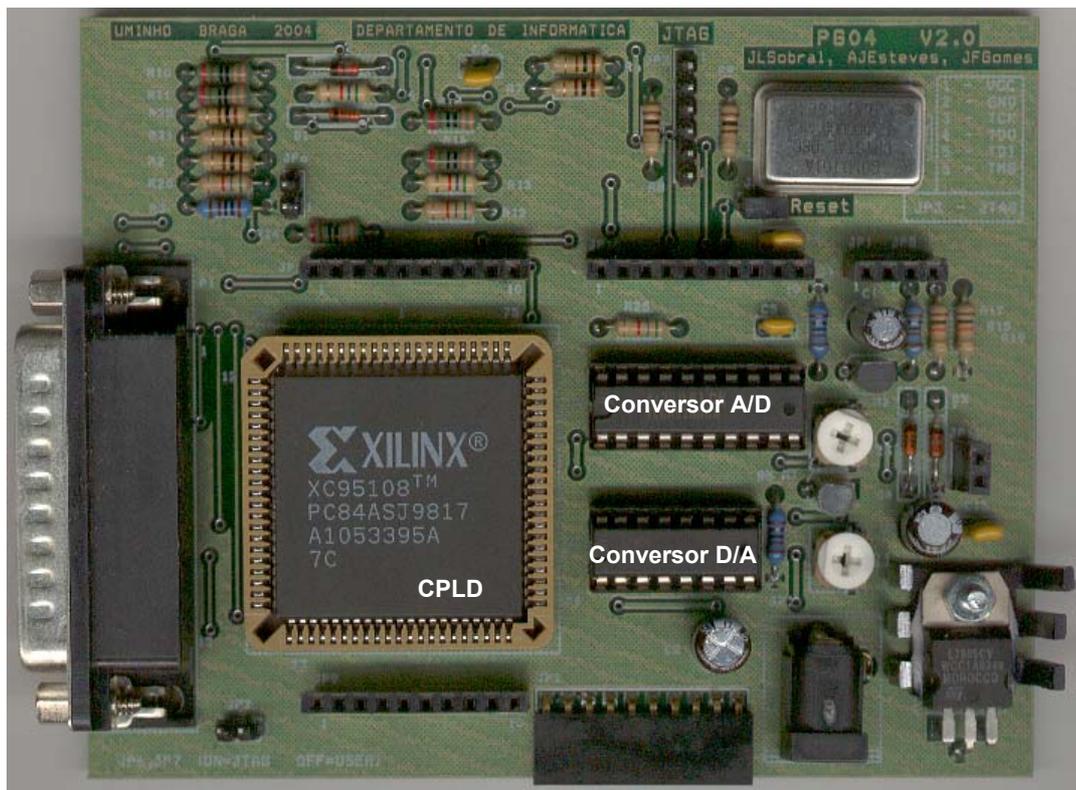
- Os sinais do lado esquerdo são recolhidos da porta paralela. Os sinais do lado direito do diagrama disponibilizados ao Painel de Comando através de conectores.
- Na versão do PG com CPLD, os registos, o decodificador e o multiplexer são implementados pela CPLD. Na versão com PAL o decodificador e o multiplexer são implementados pela PAL e os registos por *chips* dedicados (74hct244 e 74hct245).
- Em ambas as versões os conversores A/D e D/A são implementados por componentes externos.

Periférico genérico

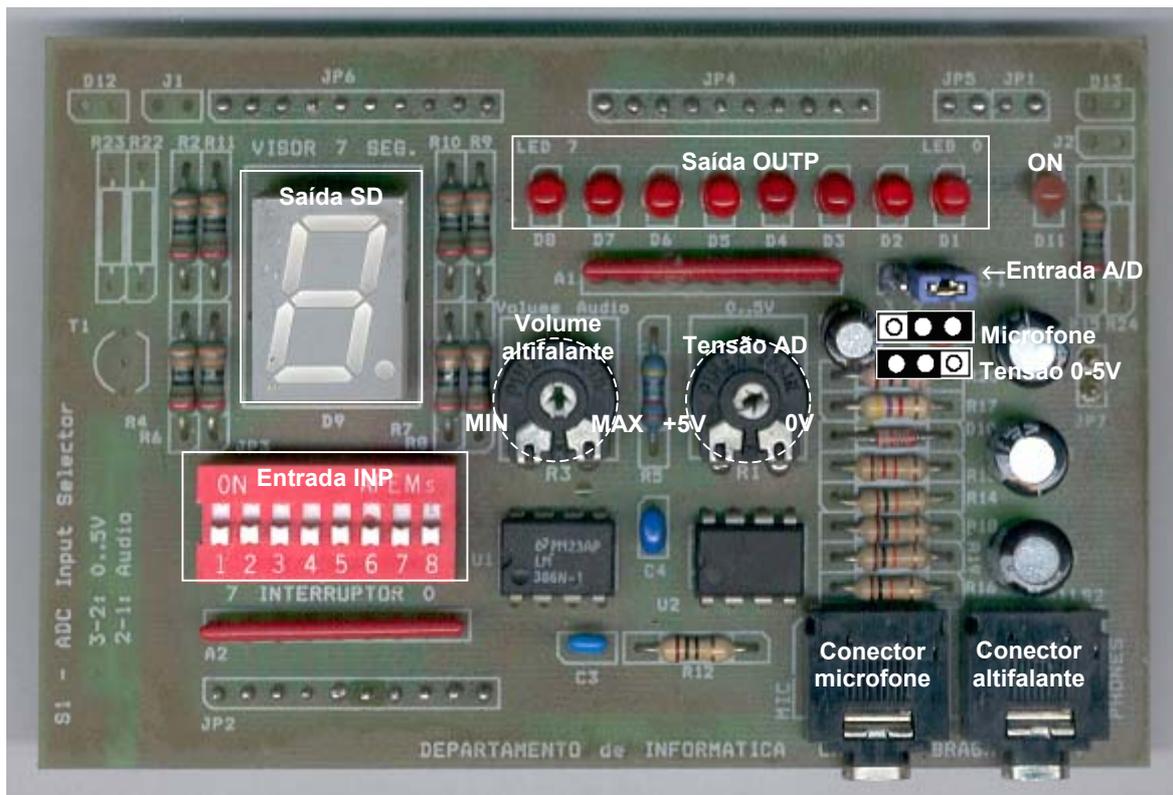
- PG04 v2 (Versão alunos)



- PG04 v2 (Versão do laboratório)



Painel de Comando



● Características

- **8 visualizadores** vulgarmente conhecidos por *leds*
- **8 interruptores**
- **visor de 7-segmentos** com ponto decimal.
- fonte de tensão regulável de 0V a +5V.
- microfone ligado a um pré-amplificador;
- amplificador de saída para ligar a um altifalante com controlo de volume por um **potenciómetro**;
- **comutador** que selecciona a entrada analógica entre a fonte de tensão e o pré-amplificador do microfone;

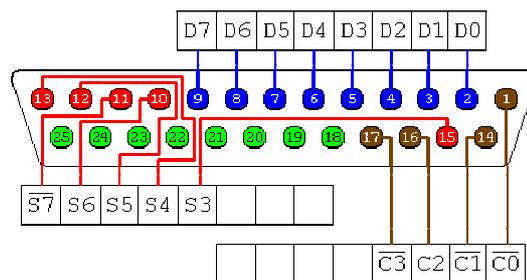
● Ligação periférico – painel de comando

Periférico genérico	Painel de comando
OUTP	<i>leds</i>
INP	interruptores
SD	visor 7 segmentos
Entrada analógica	fonte de tensão/microfone
Saída analógica	amplificador

Porta Paralela

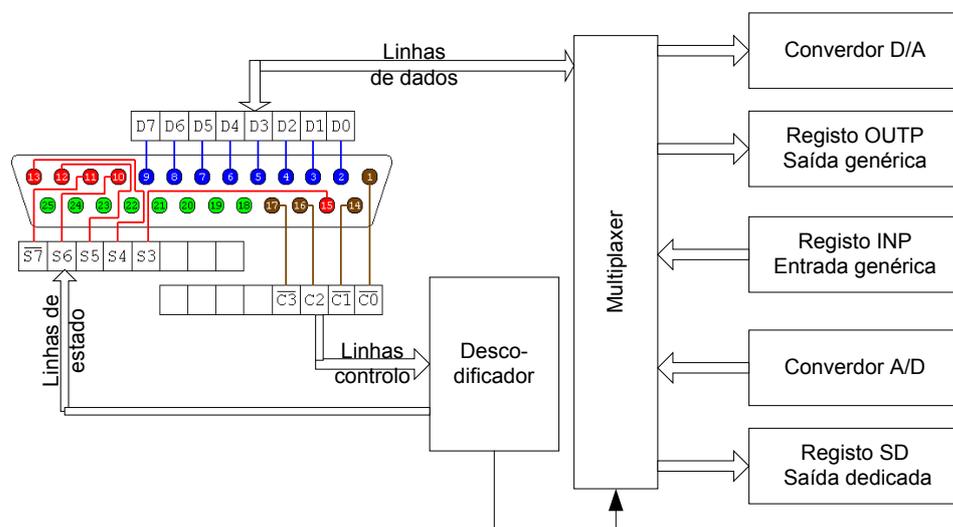
• Modo de funcionamento

- A porta paralela é constituída por 3 tipos de sinais:
 1. Linhas de dados: utilizadas para enviar e receber dados
 2. Linhas de estado: utilizadas para determinar o estado do periférico
 3. Linhas de controlo: utilizadas para controlar o modo de funcionamento do periférico
- Os sinais nos pinos da porta paralela são controlados directamente através da escrita e leitura de 3 registos, designados, respectivamente por registo de dados (D), estado (S) e de controlo (C)
- A figura seguinte faz a correspondência entre os pinos (DB25) da porta paralela e os três registos da porta :



• Ligação Periférico Genérico – Porta Paralela

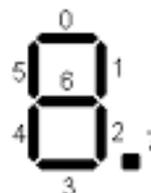
- Os três tipos de sinais utilizados pelo PG são recolhidos directamente dos sinais da porta paralela:



Periférico genérico PG04 (Porta Paralela)

● Modo de funcionamento

1. Indicar no registo de controlo da porta paralela (bits 5-0) a funcionalidade pretendida
 2. Escrever ou ler na porta de dados (ou estado) os valores pretendidos
- A porta paralela deve ser colocada em modo SPP bidireccional. Para tal deve ser escrito o valor 0x35 no endereço (0x378+0x402)
 - A conversão AD é iniciada enviando o valor 0x02 para a porta de comando, seguido de 0x00. O fim da conversão é assinalado em 0x379.
 - Esquema de bits do visor de 7 segmentos:



● Portas de E/S

Endereço E/S	LEITURA	ESCRITA
0x378	DADOS: Valor na entrada INP (interruptores) Valor do conversor A/D	DADOS: Valor a colocar na saída OUTP (leds) Valor a colocar no visor de 7-segmentos Valor para o conversor D/A
0x379	ESTADO: Bit 6 activo a '1' indica o fim da conversão A/D	
0x37a		COMANDO: 0x00 – Desactiva entradas e saídas 0x21 – Liga entrada de dados a INP (interr.) 0x02 – Inicia conversão A/D 0x23 – Liga entrada ao conversor A/D 0x04 – Liga saída de dados a D/A 0x05 – Liga saída de dados a SD (visor) 0x07 – Liga saída de dados a OUTP (leds)

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 com periféricos em Linguagem C

Exercício

Considere as seguintes ligações do PG para o exterior:

- **registo de saída** (outp)
 - 8 leds
 - bit a 1 - acende
 - bit a 0 - apaga
- **registo de entrada** (inp)
 - 8 interruptores
 - ligado - bit a 1
 - desligado - bit a 0

Escreva os programas que realizam as seguintes operações:

- a) Mostrar o valores dos interruptores, da esquerda para a direita: 1) no ecrã 2) nos *leds*.
- b) 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 da técnica de sondagem

Exercício

Considere que às ligações do PG (no exercício anterior) juntamos:

- **visor**
 - 7 segmentos + ponto
 - bit a 1 - acende traço
 - bit a 0 - apaga traço
- **conversor A/D**
 - escala de 0 a 5 Volt
 - cada unidade corresponde a aproximadamente 0,02 Volt
- **entrada de tensão**
 - 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.
- 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 PG possui uma escala de 0V a +5V, correspondendo cada unidade do valor lido a cerca de 0,02V.

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

Usando o periférico PG 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: A qualidade do som não é muito elevada, uma vez que o conversor é apenas de 8 bits e se trata de um circuito bastante simplificado.

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.

Nos modos gráficos com cores verdadeira (24 bits) cada 3 células de memória corresponde ao valor utilizado por um pixel, respectivamente a quantidade de vermelho, verde e azul (RGB).

Em aplicações Windows o acesso à memória vídeo está reservado aos *device-drivers*, sendo o endereço da memória vídeo, utilizada pelo controlador para gerar a imagem, está acessível a partir do endereço pre-definido.

Usando o periférico PG 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: E/S Controlado por 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
}
```

- 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 0x60 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.
- Adapte o programa anterior ao uso da técnica de interrupção.
- 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