

**UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA**



LOURIVAL OLIVEIRA DA SILVA

**Projeto de Aquisição de Dados**

Feira de Santana, 19 de Julho de 2004

**UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA**

LOURIVAL OLIVEIRA DA SILVA

**Projeto de Aquisição de Dados**

Relatório do Problema de Aquisição de Dados apresentado para avaliação da Disciplina de Arquitetura e Organização de Computadores do 2º semestre, do Curso de Engenharia de Computação, da Universidade Estadual de Feira de Santana sob orientação do Prof. Dr. Márcio.

Feira de Santana, 19 de Julho de 2004

## Sumário

Introdução.....	4
Placa-Mãe.....	5
Chipsets.....	5
Ponte Norte.....	5
Ponte Sul.....	5
Transferência de Dados.....	5
Pooling.....	6
Interrupção.....	6
DMA.....	6
Barramentos.....	6
Barramento Externo.....	6
Barramento Interno.....	6
Barramento de dados (data bus).....	6
Barramento de Endereço (adress bus).....	6
Barramento de Controle (control bus).....	7
O Barramento PCI.....	8
A interface PCI.....	8
O Protocolo.....	9
Transferência de Dados.....	10
Arbitração.....	10
Pedido de Interrupção (IRQ).....	10
Arquitetura Proposta.....	11
Placa de Aquisição de Dados.....	12
Conversores.....	13
Faixa dinâmica.....	13
Resolução.....	13
Tempo de conversão.....	13
Digital-Analógico.....	13
Analógico-Digital.....	14
Erros de Conversão.....	15
Erro de linearidade.....	15
Erro de quantização.....	15
Erro de desvio do zero ou offset.....	15
Erro de ganho.....	15
Conclusão.....	16
Referências Bibliográficas.....	17

## **Introdução**

A aplicação da computação para automatização de diversos processos industriais tem crescido de forma significativa, de certo que os benefícios advindos da introdução da computação no processo produtivo tem incrementado a produtividade e reduzido as possibilidades de falhas durante o processo, uma das aplicações que se destacam é a utilização de computadores para monitorar os mais diversos sensores que integram uma determinado processo e a partir das informações obtidas enviar sinais controle para outros dispositivos, ou gerar relatórios técnicos para posterior análise.

Para fazer a interface entre um computador e o “mundo externo” utiliza-se as chamadas placas de aquisição de dados. Que conjugam todos os dispositivos necessários ao interfaceamento entre o PC e os mais diversos dispositivos externos.

Este trabalho discutirá sobre as partes envolvidas neste tipo de interfaceamento, isto inclui as linhas de controle de um Processador, assim como os componentes de uma placa-mãe que fornecem suporte a funções de E/S, e o barramento utilizado para comunicação entre a placa de aquisição e o processador, neste caso o PCI, por fim o presente trabalho discutirá o processo de conversão entre sinais analógicos e digitais, e o projeto básico de uma placa de aquisição.

## Placa-Mãe

A placa-mãe é o coração dos computadores, de forma geral é ela quem define os tipos de processadores que o sistema vai suportar, assim como a quantidade de memória e as interfaces de periféricos, por isso, um bom projeto de placa mãe é essencial para o desempenho e a expansibilidade do sistema. Essas características são determinadas pelos chipsets que compõem as placas-mãe

### Chipsets

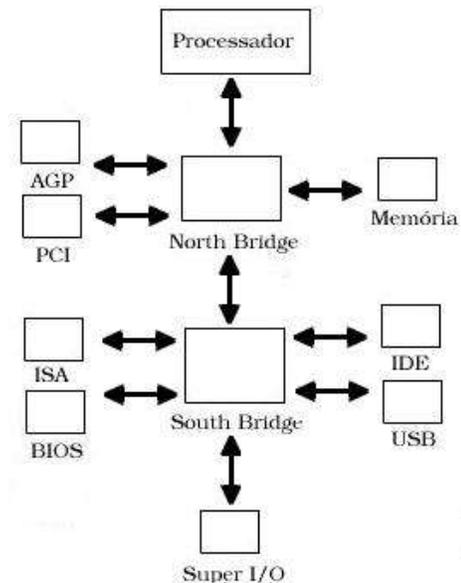
Os chipsets são circuitos integrados que fornecem conectividade ao processador, através dele é que os periféricos se comunicam com a CPU do sistema, eles implementam diversas funções de auxílio ao processador como controlador de DMA, controlador de interrupções, temporizadores, entre outras. Historicamente cada uma dessas funções eram implementadas por um chip dedicado, assim tinha-se um chip para DMA, outro para interrupções, e assim por diante, mas, com os avanços da micro-eletrônica, esses chips foram condensados, e hoje a maioria das placas mães vem equipadas com dois chips para realizar essas funções, estes chips são denominados de ponte-norte (north bridge), e ponte-sul (south bridge), cada um desses chipsets executam um conjunto de funções específicas como detalhado abaixo.

### Ponte Norte

A Ponte Norte é o circuito que integra funções essenciais ao sistema, é este circuito que contém o controlador de memória, assim como a ponte entre o barramento local e o barramento PCI.

### Ponte Sul

Este chipset é responsável pelo controle de acesso aos periféricos do sistema, assim como a controladora da unidade de disquete, porta serial, porta paralela e portas IDE, são mantidas por este chipset bem como o acesso ao barramento USB, ponte PCI-ISA, Teclado, Mouse, UART.



### Transferência de Dados

Existem diversas metodologias para transferência de dados entre o processador e os dispositivos de E/S, as técnicas mais simples como o pooling gozam de uma implementação fácil e direta em degradação à performance, já técnicas mais avançadas como DMA requer circuitos adicionais e

suporte dos dispositivos envolvidos a esta tecnologia, um sistema convencional emprega várias tecnologias como o Pooling, Interrupções e DMA para transferência de dados, abaixo segue uma explicação mais detalhada sobre estas tres formas de transferência de dados.

### **Pooling**

O Pooling ou espera ocupada consiste na requisição de um dados ao dispositivo de E/S e então se faz sucessivas consultas para determinar se o dado solicitado já foi disponibilizado, durante este tipo de transferência o processador fica ocupado até que o dado esteja disponível para consulta/escrita. Dentre as metodologias de transferência de dados o Pooling é a mais trivial, e também a que degrada mais a performance do sistema.

### **Interrupção**

Neste tipo de transferência o processador faz a requisição dos dados então continua seu processamento normalmente, quando os dados solicitados estão disponíveis o dispositivo de E/S notifica o fato ao processador através de uma interrupção, o que faz com que o processador para o que está fazendo e vá tratar os dados que ele requisitou.

### **DMA**

Esta metodologia de transferência consiste na adição de um Circuito integrado especializado em transferência de dados, que coordena todo o processo de transferência de dados deixando o processador livre para executar outras tarefas, assim este CI pode transferir blocos inteiros de dados sem o auxílio do processador que será notificado somente ao final de todo o processo de transferência, ou se algum evento importante para o processador ocorrer.

### **Barramentos**

Oe barramentos podem ser divididos em dois grupos principais:

**Barramento Externo** – Interliga os diversos componentes de um sistema computacional tais como CPU, memória, unidades de Entrada/Saída etc.

**Barramento Interno** – Interliga os diversos componentes de um circuito, tal como as linhas que ligam os registradores dentro do processador.

Além dessa classificação os barramentos podem ser agrupados de acordo como os dados que trafegam por eles:

**Barramento de dados (data bus)** – São os barramentos utilizados para o tráfego de dados ou instruções no sistema.

**Barramento de Endereço (address bus)** – Utilizados pra transportar os endereços dos dados que serão acessados.

**Barramento de Controle (control bus)** – Transferem sinais de controle para os dispositivos, tais como interrupções, sinais de sincronização, etc. Este é o mais complexo dos barramentos do sistema, devido à miscelânea de sinais que trafegam por ele, a tabela abaixo ilustra os sinais do processador Intel Pentium-MMX

Control							
A20M#	AK08	BREQ	AJ01	HIT#	AK06	PRDY	AC05
ADS#	AJ05	BUSCHK#	AL07	HITM#	AL05	PWT	AL03
ADSC#	AM02	CACHE#	U03	HLDA	AJ03	R/S#	AC35
AHOLD	V04	CPUTYP	Q35	HOLD	AB04	RESET	AK20
AP	AK02	D/C#	AK04	IERR#	P04	SCYC	AL17
APCHK#	AE05	D/P#	AE35	IGNNE#	AA35	SMI#	AB34
BE0#	AL09	DP0	D36	INIT	AA33	SMIACT#	AG03
BE1#	AK10	DP1	D30	INTR/LINT0	AD34	TCK	M34
BE2#	AL11	DP2	C25	INV	U05	TDI	N35
BE3#	AK12	DP3	D18	KEN#	W05	TDO	N33
BE4#	AL13	DP4	C07	LOCK#	AH04	TMS	P34
BE5#	AK14	DP5	F06	M/IO#	T04	TRST#	Q33
BE6#	AL15	DP6	F02	NA#	Y05	VCC2DET#	AL01
BE7#	AK16	DP7	N05	NMI/LINT1	AC33	W/R#	AM06
BOFF#	Z04	EADS#	AM04	PCD	AG05	WB/WT#	AA05
BP2	S03	EWBE#	W03	PCHK#	AF04		
BP3	S05	FERR#	Q05	PEN#	Z34		
BRDY#	X04	FLUSH#	AN07	PM0/BP0	Q03		
BRDYC#	Y03	FRCMC#1	Y35	PM1/BP1	R04		

APIC		Clock Control		Dual Processor Private Interface	
PICCLK	H34 (2)	CLK	AK18 (2)	PBGNT#	AD04
PICD0	J33	[BF0]	Y33	PBREQ#	AE03
[DPEN#]		[BF1]	X34	PHIT#	AA03
PICD1	L35	STPCLK#	V34	PHITM#	AC03
[APICEN]					

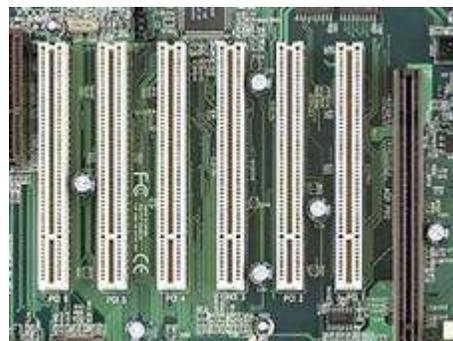
Sinal	Descrição
A20M	Usado para dar acesso à chamada memória HMA, que consiste nos primeiros 64 kB localizados após o endereço 1M. É usado quando o processador opera no Modo Real, ou seja, o mesmo utilizado pelo MS-DOS.
ADS	Address Strobe. Indica que o processador está iniciando um novo ciclo. Os ciclos podem ser de leitura da memória, escrita na memória, leitura de I/O, escrita em I/O, etc.
AP e EADS	Usados para geração e checagem de paridade no barramento de endereços.
BF0 e BF1	Indica o multiplicador usado para definir o clock interno, em função do clock externo (2x, 2,5x, 3x e 3,5x, no caso do Pentium MMX). Processadores como o K6-2 possuem ainda um pino adicional BF2 para a definição de multiplicadores maiores. Esses pinos são ligados aos jumpers da placa de CPU que fazem a programação dos multiplicadores.
BRDY	Indica ao processador se a atual transferência em modo burst já está finalizada ou se devem ser usados wait states.

CLK	Através desta entrada o processador recebe o seu clock externo.
CPUTYP	Usado para distinguir entre processador primário e secundário, em placas equipadas com mais de um processador.
D/C	Data/Code. Através deste sinal o processador indica se está fazendo um acesso a dados ou a instruções de programas.
DP0-DP7	Usados para geração e controle de paridade no barramento de dados. Cada um desses 8 sinais representa a paridade de 8 bits do barramento de dados.
FERR	Floating Point Erros. Usado para indicar a ocorrência de um erro de ponto flutuante, resultante de uma operação matemática inválida feita pela unidade de ponto flutuante, como uma divisão por zero ou raiz quadrada de um número negativo.
HOLD	Por esta entrada o processador recebe uma requisição para que entre em alta impedância (tristate), possibilitando que outro chip tome o controle do barramento. É usado em transferências de DMA e Bus Mastering.
HLDA	HOLD Acknowledge. O processador informa que está entrando em modo de tristate. A partir daí o barramento pode ser usado por outros chips.
M/IO	O processador indica se está acessando uma posição de memória ou uma interface de I/O (entrada e saída).
R/W	O processador indica se está realizando uma operação de leitura ou escrita.
RESET	Ao receber um comando nesta entrada, o processador realiza seu processo de RESET interno. Este sinal deve ser ativado mediante o botão RESET do gabinete, e também quando o computador é ligado (Power-on RESET).

## O Barramento PCI

Este barramento surgiu como uma resposta da Intel à pressão da indústria para padronização dos barramentos no início dos anos 90. O comunicação no barramento PCI é síncrona onde todas as transferências de dados se baseiam num clock do sistema (CLK), nas primeiras especificações do PCI essa taxa era de 33 Mhz, as taxa de 66 Mhz só foi definida a partir da especificação 2.1 de 1995, porém a grande maioria dos computadores modernos só implementam versões de 33 MHz.

<i>Clock</i>	<i>Bits</i>	<i>Taxa de Transferência</i>
33 MHz	32	133 MB/s
33 MHz	64	266 MB/s
66 MHz	32	266 MB/s
66 MHz	64	533 MB/s



### A interface PCI

A interface do PCI é composta pelo seguinte conjunto de 62 pinos na especificação de 32 bits e por 94 pinos na versão de 64 bits, porém muitos destes sinais não precisam ser tratados por todos os dispositivos, assim uma implementação mínima de um dispositivo PCI é composta de apenas 47 pinos caso o dispositivo nunca se torne um master do barramento ou 49 pinos caso este dispositivo seja

projetado para se tornar o master do barramento. Esta pinagem reduzida é uma das grandes vantagens do PCI. Abaixo segue uma descrição dos das vias de um barramento PCI.

**AD0-AD31** – Nestas vias do barramento os dados e os endereços são multiplexados, assim no início de uma transferência, este barramento indica o endereço, e na fase seguinte, os dados.

**C/BE0-C/BE3** – Estes sinais idicam qual o comando que será executado (leitura, escrita, etc.), essa informação é enviada durante a fase de endereço. Já durante a fase de dados cada bit desses quatro sinais indica quais dos quatro bytes do barramento estão ativos, assim é possivle acessar os bytes de dados de forma individual.

**FRAME** – Este sinal é ativado pelo Mestre do Barramento para dar início a um ciclo de transferência.

**IRDY** - Initiator Ready. Indica que o Master está pronto para ler ou enviar dados. Quando este sinal não é ativado, o Target irá esperar tantos wait states quanto forem necessários.

**TRDY** - Target Ready. Indica que o Target está pronto para receber dados (escrita) ou que o dado lido já está disponível (leitura). Quando este sinal não é ativado, o Master irá gerar tantos wait states quando forem necessários.

**DEVSEL** - Ativado pelo Target quando reconhece o seu endereço. Desta forma o Master pode saber se o dispositivo Target está ativo ou presente no barramento.

**REQ** - Requisição enviada ao Bus Arbitrer, para que o dispositivo se torne Bus Master. Cada dispositivo tem seu próprio sinal REQ.

**GNT** - Grant. Através deste sinal o Bus Arbitrer indica ao dispositivo solicitante que o barramento está liberado, permitindo assim que se torne Bus Master. Cada dispositivo tem seu próprio sinal GNT.

**INTA, INTB, INTC, INTD** - São linhas de interrupção a serem usadas pelos dispositivos PCI. Cada dispositivo e cada slot é ligado a um desses sinais, que podem ser compartilhados, ou seja, uma mesma linha INT pode ser usada por mais de um slot. O padrão PCI prevê o compartilhamento de interrupções.

**AD32-AD63** - Continuação do barramento de dados e endereços nos slots PCI de 64 bits.

**C/BE4-C/BE7** - Continuação do barramento de comando e habilitação de bytes nos slots PCI de 64 bits.

**REQ64** - Requisição de transferência de 64 bits.

**ACK64** - Indica que o Target está apto a realizar transferência de 64.

**CLK** - *Clock* este é o trem de pulso para todas as tranferências do PCI.

**RST** – *Reset* é o sinal que instrui o dispositivo a assumir um estado inicial, reiniciando todos os registradores ou qualquer componente de estado para o estado padrão, este sinal normalmente é enviado para o PCI quando o computador inicia.

**PERR** - *Parity Error* é utilizado para reportar erros de paridade na transmissão dos dados.

**SERR** - *System Error* indica erros do sistema, como erro de paridade no endereço ou qualquer outr tipo de erro fatal.

## **O Protocolo**

A comunicação dos dispositivos PCI é síncrona, toda ela baseada no sinal CLK, execto alguns sinais como de interrupção (INT) e reset (RST).

## Transferência de Dados

Um transferência de dados consiste em uma fase de endereço seguido de uma ou mais fases de dados, uma operação de I/O padrão tem somente uma fase de endereço e uma fase de dados, já transferências para memória, quando se transfere blocos de dados a operação consiste em uma fase de endereço seguida de múltiplas fases de dados escrevendo/lendo blocos contíguos de uma única vez. Tanto o alvo como o iniciador podem finalizar uma sequência de transferência a qualquer momento, o dispositivo alvo faz isso desabilitando o sinal FRAME após enviar o último dados solicitado, já o iniciador ou master pode ativar o sinal STOP que sinaliza para o alvo o final da transferência.

## Arbitração

O acesso ao barramento PCI é controlado por um árbitro central, cada dispositivo PCI, faz uma requisição de acesso ao barramento através do sinal REQ, o árbitro PCI então verifica quem dos dispositivos que necessitam usar o barramento deve obter o acesso então ativa o sinal GNT para o dispositivo.

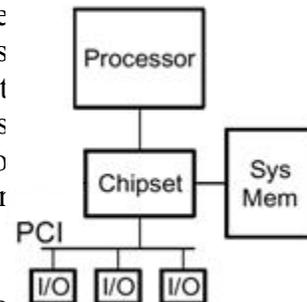
## Pedido de Interrupção (IRQ)

Para que os diversos dispositivos do sistema possam ter acesso ao processador eles se utilizam dos chamados pedidos de interrupções ou IRQ, cuja função é permitir que os vários dispositivos do micro possam interromper o processamento corrente para que eventos importantes possam ser tratados e continuar o seu trabalho após atendê-los. Cada dispositivo deve ter um IRQ único, caso contrário teremos um conflito de IRQ, neste caso, o processador não saberá qual dispositivo o está chamando, o que pode levar a um mal funcionamento dos dispositivos envolvidos. Num PC convencional há 256 códigos de interrupção, associada a cada código há o endereço da rotina que irá tratar tal interrupção, a tabela de vetores de interrupção é armazenada no primeiro 1K da RAM do sistema, cada entrada nesta tabela ocupa 4 bytes de memória, que contém o endereço da rotina que irá tratar a interrupção (2 bytes) e o endereço do segmento da rotina (2 bytes)

<i>Código</i>	<i>Atribuição Comum</i>	<i>Código</i>	<i>Atribuição Comum</i>
<b>IRQ 0</b>	Usado pela placa mãe	<b>IRQ 8</b>	Relógio de tempo real
<b>IRQ 1</b>	Teclado	<b>IRQ 9</b>	Placa de Vídeo (algumas placas não utilizam)
<b>IRQ 2</b>	Usado pela placa mãe	<b>IRQ 10</b>	Controladora SCSI
<b>IRQ 3</b>	Porta serial 1 (Com2 e Com 4)	<b>IRQ 11</b>	Disponível
<b>IRQ 4</b>	Porta Serial 2 (Com1 e Com 3)	<b>IRQ 12</b>	Conector USB
<b>IRQ 5</b>	Placa de Som	<b>IRQ 13</b>	Coprocessador Aritmético
<b>IRQ 6</b>	Unidade de Disquetes	<b>IRQ 14</b>	Controladora IDE Primária
<b>IRQ 7</b>	LPT 1 (porta da impressora)	<b>IRQ 15</b>	Controladora IDE Secundária

## Arquitetura Proposta

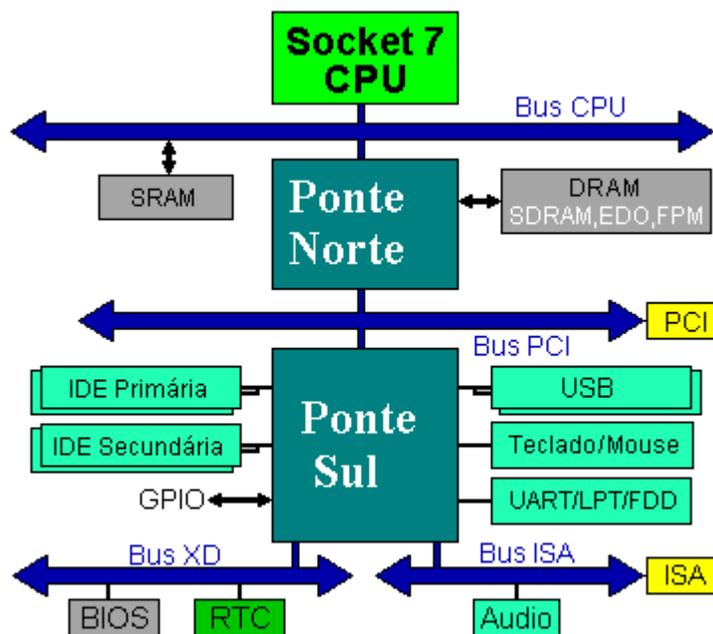
Para resolução do problema apresentado a proposta de placa-mãe se baseia numa solução conservadora que não contempla grandes expansões futuras, assim restringiu-se os slots de expansão deixando de lado o slot AGP, apesar deste oferecer performance excelentes para aplicações gráficas tal capacidade não é um requisito essencial neste tipo de aplicação específico, assim o sistema se limita a utilizar placas de vídeo PCI, que por apresentar uma performance somente razoável tem sua oferta .



O sistema é composto de dois chipsets especiais, a ponte norte como supra citado interliga a memória e o processador. Os periféricos mais lentos são controlados pela ponte sul que é ligada ao processador através da ponte norte, os discos rígidos são ligados aos canais IDEs, e a placa de vídeo como mencionado ocupa um dos slots PCI. É recomendável que o sistema tenha no mínimo 4 slots PCI já que dois ficarão ocupados (P. de Vídeo e Placa de Aquisição), assim ficarão outros dois livres onde se poderá adicionar um Modem ou mesmo outra placa de aquisição.

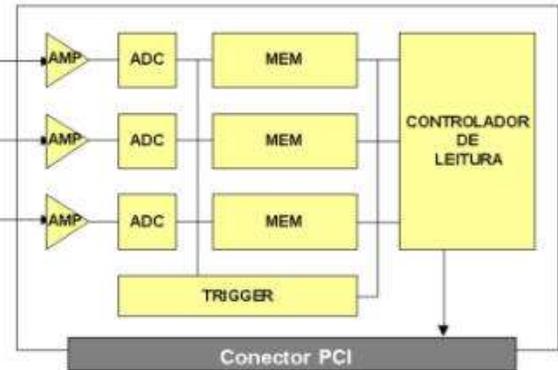
O suporte a USB é opcional mas devido à crescente implementação dos mais diversos dispositivos para essa interface seria interessante adicioná-lo ao sistema.

A figura abaixo ilustra a arquitetura proposta nos seus detalhes mais latentes.



## Placa de Aquisição de Dados

As placas de aquisição de dados são responsáveis pela interface entre o “mundo externo” e o computador, há diversas opções de placas de aquisição e a opção por um modelo ou outro se baseia no número de interfaces que cada uma possui, na quantidade de dados que será transportada, e diversos outros fatores. A interface PCI se mostra bastante aconselhável para utilização neste tipo de placa, pois associa alta performance e um protocolo de transferência relativamente simples.



Uma placa de aquisição normalmente é composta por:

**Entradas e Saídas** - (analógicas e/ou digitais de acordo com a aplicação);

**Conversores AD** – Transformam os sinais analógicos em digitais, para que estes possam ser processados pela CPU;

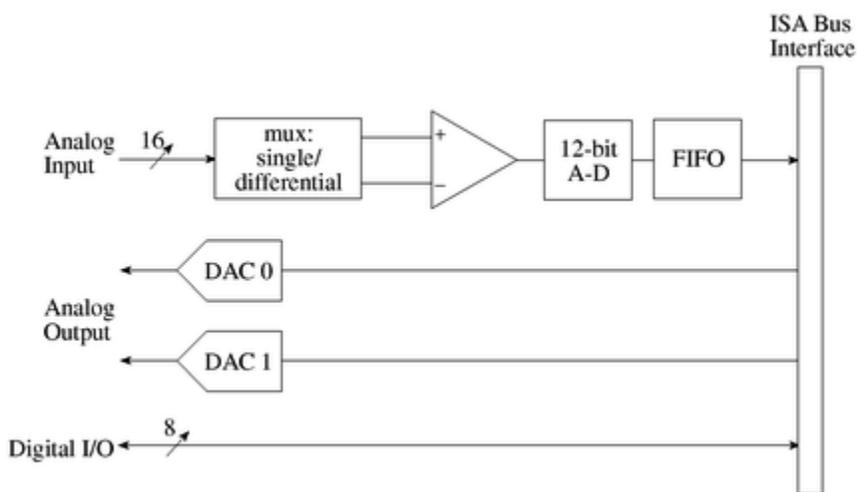
**Conversores DA** – Transformam os sinais digitais em analógicos, quando a saída precisar ser representada desta forma;

**Trigger** - que são “monitores” que ficam aguardando a ocorrência de determinado evento;

**Buffer (FIFO)** - que armazenam os dados coletados temporariamente, em uma fila onde o primeiro a entrar é o primeiro a sair (First In-First Out);

**Lógica da Interface** - circuitos especiais que implementam o protocolo PCI ou ISA.

A figura abaixo ilustra uma placa de aquisição com os componentes descritos.



## Conversores

Os computadores convencionais só trabalham com sinais digitais, devido a essa limitação é necessário que se converta todas as entradas analógicas para uma representação digital que possa ser processada pelo computador, de forma analógica as informações geradas pelo computador devem ser convertidas para sinais analógicos quando a saída precisar estar neste formato. Esta tradução é feita por conversores AD e DA que serão discutidos abaixo. Alguns pontos devem ser observados durante a definição de um conversor:

**Faixa dinâmica:** é o intervalo de sinal de entrada definido como válido para o sinal analógico normalmente este valor se encontra na faixa de 0,1 a 10V.

**Resolução:** Indica a precisão com que os dados podem ser convertidos, sendo definida pelo número de bits do conversor, quanto maior este número mais fiel é o conversor.

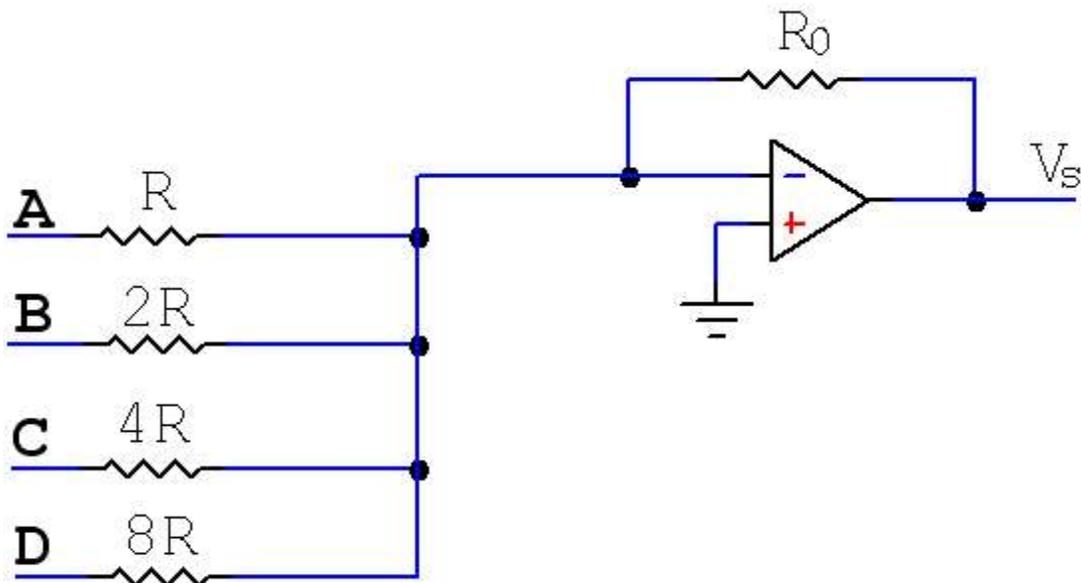
**Tempo de conversão:** refere-se ao intervalo de tempo necessário para que se obtenha o sinal de saída (analógico ou digital), a partir do momento em que a entrada foi fornecida (analógico ou digital). A resolução do conversor afeta significativamente o tempo de conversão.

### Digital-Analógico

A conversão de sinais digitais para sinais analógicos se dá através da seguinte fórmula:

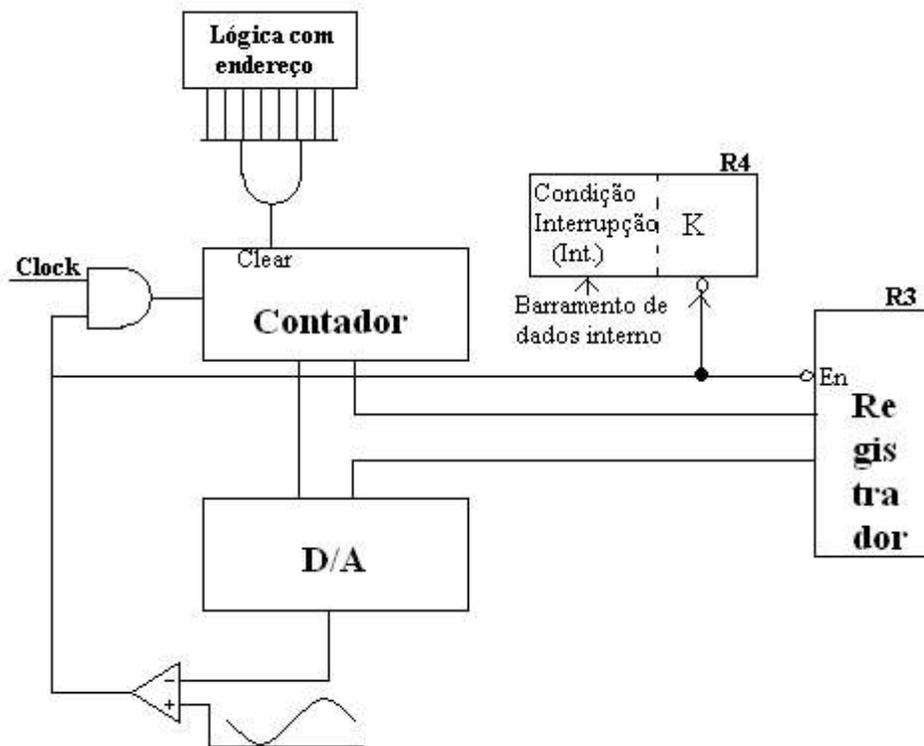
$$V_s = \frac{-R_0}{R} \left( \frac{A}{1} + \frac{B}{2} + \frac{C}{4} + \frac{D}{8} \right)$$

Aqui A, B, C e D são as entradas binárias que serão convertidas, cada resistor R tem um valor associado que atribui pesos a cada valor binário, a atribuição se dá através da seguinte lógica: o valor do resistor do bit mais significativo deve ter metade do valor do resistor do seu antecessor, da mesma forma o resistor do próximo bit terá metade do valor do seu antecessor, por fim estes valores são amplificados através de um amplificador operacional, que gera a saída analógica correspondente ao sinal de entrada.



### Analógico-Digital

Existem muitas metodologias para se converter uma sinal analógico em um sinal digital um desses métodos consiste na associação de um conversor DA, um comparador, e um contador, toda vez que o valor do sinal de analógico muda o contador é resetado e incrementado até que o valor da conversão de Digital pra analógico se iguale ao valor da entrada, a performance deste tipo de conversão é aceitável, especialmente quando se leva em conta a sua fácil implementação.



## Erros de Conversão

Abaixo há uma breve descrição dos erros de conversão mais frequentes:

**Erro de linearidade:** expressa o desvio do resultado de conversão de uma reta ideal. É expresso em  $\pm$  uma porcentagem do valor total ou em número de bits.

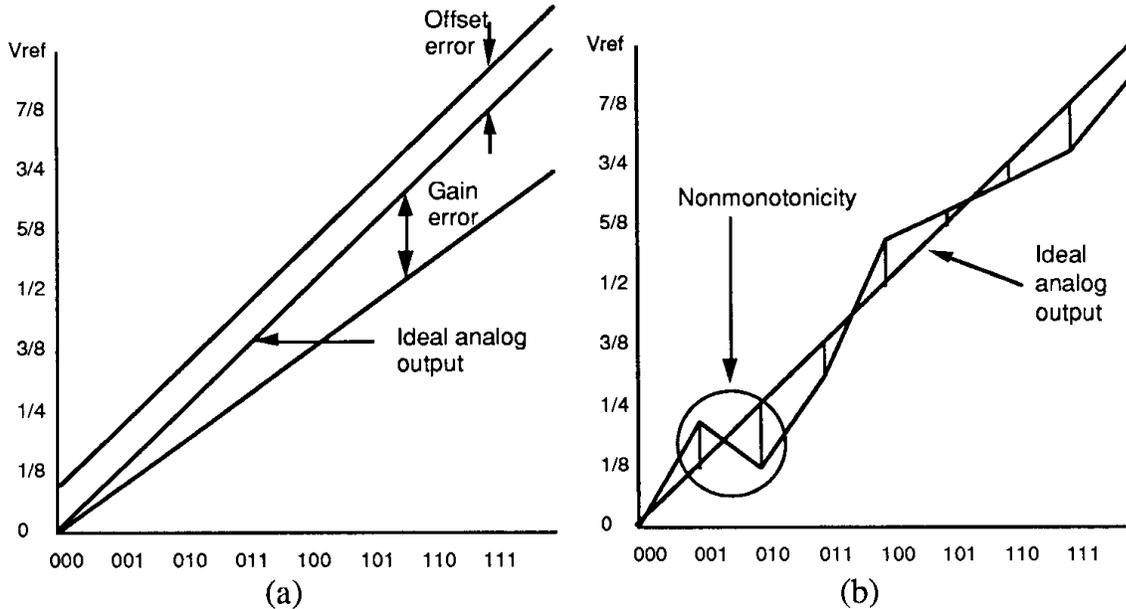
Exemplo) Um erro de linearidade de  $\pm 0,4\%$  equivale a uma linearidade de  $\pm 1$  bit num conversor de 8 bits.

**Erro de quantização:** É um erro intrínseco ao processo de conversão, na qual os valores que não podem ser representados com a precisão requerida são arredondados para o valor mais próximo. Este tipo de erro é contornado com o aumento da resolução do conversor empregado.

**Erro de desvio do zero ou offset:** Este tipo de erro se dá pela adição de um deslocamento a todos os valores na faixa de conversão.

**Erro de ganho:** É caracterizado por um deslocamento gradual entre os valores ideais e o sinal fornecido pelo conversor.

A figura abaixo ilustra alguma destas ocorrências.



**Figure 3.7** Digital-to-analog converter characteristics. (a) Gain and offset errors. (b) Monotonicity errors.

## Conclusão

A integração de computadores com dispositivos externos pode ser implementada através das diversas interfaces que compõem os computadores modernos, a interface PCI se mostrou bastante flexível e poderosa, sua flexibilidade e simplicidade implementação foram fundamentais para indicação deste tipo de barramento para o projeto proposto. Outro importante ponto forte do PCI é a implementação de um árbitro de barramento, que ajuda a promover uma melhor utilização do barramento pelos dispositivos de forma relativamente autônoma em relação à CPU. A utilização de componentes com suporte a DMA em detrimento a métodos menos sofisticados de transmissão de dados pode incrementar significativamente a performance geral do sistema, assim sempre que possível deve-se empregar tal tecnologia.

Por fim a conversão AD e DA apesar de ter se tornado uma constante nos sistemas modernos não é tão simples o quanto pode sugerir o seu emprego acentuado, há diversos problemas que podem “assombrar” tal processo como foi demonstrado, porém muito deles são contornáveis e/ou minimizados com o emprego das técnicas corretas.

Apesar do esforço feito para compilar os dados deste trabalho há muitos outros detalhes a serem explorados, como por exemplo um estudo mais profundo da conversão de sinais AD e DA, bem como detalhes de como se dá exatamente o acesso a esses valores depois de convertidos pela interface PCI.

Em alguns outros pontos houve certa omissão, como na questão das interrupções, de fato sua discussão está incompleta, a discussão sobre transferência de dados está um tanto quanto resumida, por fim a discussão sobre arbitração do barramento poderia traçar um paralelo entre os outros métodos de implementação propostos, e não se limitar ao estudo do PCI.

## Referências Bibliográficas

PATTERSON, David A. e HENNESSY, John L., *Organização e projeto de computadores a interface hardware/software*, Segunda edição. Rio de Janeiro: LTC, 2000.

STALLINGS, Willian, *Computer Organization and Architecture Design for Performance*, Sixth Edition, Prentice Hall, 2003.

TANENBAUM, Andrew. S., *Organização Estruturada de Computadores*, Quarta Edição. Rio de Janeiro: Prentice-Hall do Brasil, 1999.

WEBER, Raul Fernando, *Fundamentos de Arquitetura de Computadores*, 2ª edição. Porto Alegre: Editora Sagra Luzzatto, 2001.

### Websites

[www.intel.com](http://www.intel.com)

<http://www.laercio.com.br/>

### Materiais Técnicos

*PCI SIG: PCI Local Bus Specification Revision 2.2 December 18, 1998*

*Intel Architecture Software Developer's Manual Volume 1: Basic Architecture*

*Intel: Understanding the Interrupt Control Unit of the 80C186EC/80C188EC Processor*