

UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA

DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

SISTEMA DIGITAL PARA MONITORAMENTO DA VELOCIDADE E
POSIÇÃO ANGULAR DO MOTOR DE ACIONAMENTO
DE UM EIXO DE UMA MÁQUINA FRESADORA *OK*

Autor: Wilson Rodrigues *Lucas 5/92*

ORIENTADOR: Prof. Dr. Jaime *Szajner **

Este exemplar copiado da coleção final da tese
defendida por *Wilson Rodrigues*
Lucas pela Comissão
Julgadora em *20/09 1993*
Jaime Szajner
Orientador

Tese de Mestrado apresentada à Faculdade de Engenharia Elétrica
da UNICAMP como parte dos Requisitos para obtenção do Grau de
Mestre em Engenharia Elétrica

1993



Este trabalho é dedicado à
minha noiva Alessandra, aos
meus pais Geraldo e Esmeralda
e a todas as pessoas que
ajudaram na sua implementação

AGRADECIMENTOS

- à minha família e noiva pela dedicação e incentivos durante todos estes anos
- ao professor Dr. Jaime Szajner pelo ensino e orientação durante a execução deste trabalho.
- aos professores do Departamento de Engenharia de Computação e Automação Industrial que contribuíram para minha formação.
- ao Professor José Raimundo pelo apoio dado durante o projeto.
- ao Valfredo, Víctor, Marcel, Francisco, Sérgio, Ivo, Edson pela companhia e apoio nas diversas etapas de testes.
- ao Gustavo e Alexandre pela dedicação de muitas horas de trabalho.
- aos amigos Carlos e Rudinei pelo apoio e colaboração.

RESUMO

O objetivo deste trabalho foi desenvolver um Controlador Digital para uma Máquina Ferramenta (Fresadora). Dentro desta diretriz implementaram-se dois cartões digitais similares, cuja função é fornecer dados para o Controlador. O software do Controlador foi desenvolvido em um computador PC.

O primeiro cartão digital é acoplado em um dos "slots" do computador e suas funções são executadas por componentes discretos.

O segundo cartão digital foi desenvolvido utilizando um microcontrolador. Este cartão desempenha funções análogas às do primeiro cartão digital. Seu microcontrolador permitiu sintetizar muitas operações lógicas que eram executadas por hardware no primeiro cartão, além de efetuar operações aritméticas. Este "chip" é também responsável pela comunicação serial (RS 232 - C) entre este cartão digital e o computador PC.

Numa aplicação industrial com várias máquinas pode-se acoplar em cada uma delas um cartão digital deste tipo, e o PC ficaria responsável pelo gerenciamento do sistema.

ABSTRACT

The purpose of this work was to develop a Digital Controller for a Tool Machine (Milling Machine). Within this line, two similar digital cards were implemented, whose function is to provide data for the Controller. The software for the Controller was developed on a PC computer.

The first digital card is coupled to one of the computer slots and its functions are performed discreet components.

The second digital card was developed with a microcontroller. This card performs functions similar to the first one. Its microcontroller allowed to synthesize many logic operations executed by hardware in the first digital card. Besides it can execute arithmetic operations. This chip is responsible as well by the serial communication (RS 232 - C) between this digital card and the PC computer.

In an industrial application, each machine can be coupled to one card like the second type, leaving the PC remains responsible by the management of the system.

SUMÁRIO

Introdução	1
1. Apresentação do Sistema	2
1.1 Resumo	2
1.2 Descrição Funcional	2
1.3 Arquitetura da Unidade de Processamento e Gerenciamento	3
1.3.1 Configuração do Sistema	3
1.4 Descrição da Máquina Fresadora	7
1.5 Motores de Corrente Contínua	8
1.6 Codificador Incremental de Posição Angular	9
1.7 Sistema de Transmissão	10
1.8 Unidade de Potência e Controle	13
1.8.1 Módulo de Controle	13
1.8.2 Unidade de Potência	14
2. Descrição do Módulo Digital	16
2.1 Resumo	16
2.2 Análise Estrutural do Cartão IBM PC-XT/AT	16
2.2.1 Características Gerais	17
2.2.2 Diagrama de Blocos	17
2.2.3 Descrição dos Blocos	19
2.2.4 Endereçamento	21
2.2.5 Operação	23
2.3 Software do Cartão Digital	24
2.3.1 Módulo de Inicialização	25
2.3.2 Módulo de Posição	26
2.3.3 Módulo de Velocidade	27
3. Sistema de Velocidade/Posição com Microcontrolador	29
3.1 Principais Características do Microcontrolador	29
3.1.1 Descrição Funcional dos Pinos	30
3.1.2 Estudo dos Blocos do Microcontrolador	31
3.1.3 Descrição do "Reset"	36
3.1.4 Interrupções	37
3.1.5 O "Clock" do Microcontrolador	40
3.1.6 Estudo das Portas de I/O	41

3.2	Descrição do Hardware com Microcontrolador	44
3.2.1	Características Gerais	44
3.2.2	Diagrama de Blocos	44
3.2.3	Descrição dos Blocos	46
3.2.4	Endereçamento	47
3.2.5	Modos de Operação	49
3.3	Software da Unidade Digital com Microcontrolador	50
3.3.1	Módulo de Inicialização da Unidade com Microcontrolador	51
3.3.2	Módulo de Posição da Unidade com Microcontrolador	52
3.3.3	Módulo de Velocidade da Unidade com Microcontrolador	53
3.4	Sistema de Comunicação	55
3.4.1	Modo Assíncrono de Comunicação	55
3.4.2	A Interface Serial do Microcontrolador	56
3.4.3	Modo de Operação e Formato de Dados	56
3.4.4	Taxas de Transmissão	56
3.4.5	Processo de Comunicação	57
4.	Modelagem	59
4.1	Resumo	59
4.2	Considerações Teóricas	59
4.2.1	Método da Resposta em Frequência	59
4.2.2	Acelerômetro Piezoelétrico	61
4.2.3	Estudo da Função de Transferência do Motor DC	63
4.2.3.1	Equações Matemáticas do Circuito Equivalente do Motor	64
4.3	Funções de Transferência dos Módulos	71
4.3.1	Função de Transferência do Módulo MAM	71
4.3.2	Função de Transferência do Módulo de Transmissão	74
4.3.3	Função de Transferência do Conversor D/A	76
4.3.4	Função de Transferência do Codificador Incremental	77
5.	Conclusão	79
5.1	Resumo	79
5.2	Comparação entre os Módulos Digitais	79
5.3	Futuras Implementações	83
	Bibliografia	86

Lista de Figuras

Figura 1.1:	Diagrama de Blocos do Sistema	3
Figura 1.2:	Definição dos Pinos e Sinais do "Slot"	5
Figura 1.3:	Acoplamento entre Motor e Mesa	8
Figura 1.4:	Motor DC com Excitação de Imã Permanente	9
Figura 1.5:	Conector de Saída do Codificador Incremental de Posição Angular	10
Figura 1.6:	Diagrama de um Sistema de Transmissão	11
Figura 1.7:	Sistema de Transmissão por Fuso de Esferas Recirculantes	12
Figura 1.8:	Unidade de Controle da Velocidade	13
Figura 1.9:	Diagrama do Amplificador de Potência com o Circuito de Proteção	15
Figura 2.1:	Diagrama de Blocos do Cartão Digital	18
Figura 2.2:	Cartão dos Conversores D/A	20
Figura 3.1:	Pinagem do Microcontrolador 87C51	30
Figura 3.2:	Diagrama de Blocos do 87C51	31
Figura 3.3:	Mapa da Memória Interna	32
Figura 3.4:	Mapa da Memória Externa	32
Figura 3.5:	"Bits" Endereçáveis na RAM	34
Figura 3.6:	Registros de Funções Especiais	36
Figura 3.7:	Mapa das Interrupções do 87C51	40
Figura 3.8:	"Clock" Externo do 87C51	41
Figura 3.9:	"Clock" Interno para o Microcontrolador	41
Figura 3.10:	Diagrama de Blocos Simplificado de um Pino I/O	42
Figura 3.11:	Expansão de Memória de Dados e Programa	43
Figura 3.12:	Diagrama de Blocos do Sistema	45
Figura 3.13:	Sistema de Transmissão Assíncrono	55
Figura 3.14:	Formato dos Dados no Modo 1	56
Figura 3.15:	Processo de Comunicação	58
Figura 4.1:	Curva do Log-Módulo conjuntamente com a Curva do Ângulo de Fase	61
Figura 4.2:	Um Típico Acelerômetro Piezoelétrico	62
Figura 4.3:	Acelerômetro Típico e Circuito Equivalente	62
Figura 4.4:	Diagrama Esquemático de um Motor DC Controlado por Armadura	63
Figura 4.5:	Circuito Equivalente do Motor (Armadura)	64
Figura 4.6:	Resposta à Entrada Degrau do Motor DC	70
Figura 4.7:	Gráfico do Ganho	72
Figura 4.8:	Gráfico do Ângulo de Fase	73

Figura 4.9: Diagrama de Bloco do Módulo MAM	74
Figura 4.10: Diagrama de Bloco da Função de Transferência do Fuso de Esferas	75
Figura 4.11: Diagrama de Bloco do Módulo de Transmissão	76
Figura 4.12: Diagrama de Bloco da Função de Transferência do Conversor D/A	77
Figura 4.13: Diagrama de Bloco da Função de Transferência do Codificador	78
Figura 5.1: Arranjo de uma PAL	84

Lista de Tabelas

Tabela 1.1:	Fonte de Alimentação Padrão para o PC	3
Tabela 1.2:	Descrição da Porta Serial RS-232C	4
Tabela 2.1:	Formação do Endereço Base do Cartão Digital	21
Tabela 2.2:	Mapeamento dos Endereços dos Circuitos de I/O	22
Tabela 3.1:	Endereçamento dos Bancos de Registradores	33
Tabela 3.2:	Mapeamento dos Endereços dos Circuitos de I/O	48
Tabela 3.3:	Endereço dos Conversores D/A	49

INTRODUÇÃO

A evolução da tecnologia e a industrialização proporcionaram a pesquisa da metodização e a racionalização do trabalho, objetivando o aumento da produção e a redução de custos.

No primeiro estágio deste processo, todos os controles e funções auxiliares, tais como operação da máquina, gerenciamento das peças de trabalho e inspeção foram operadas manualmente. O aumento da produção, só foi possível com o desenvolvimento de máquinas semi-automáticas, sendo a quantidade produzida muito maior, o que implicou em um aumento da inspeção. Num estágio posterior, o processo e o gerenciamento do trabalho foram sendo automatizados, apesar do sistema de verificação ainda ser executado manualmente. No patamar seguinte todas as tarefas são automatizadas, sendo a intervenção do operador necessária somente quando a unidade de inspeção indica que o processo automático requer ajustes ou reinicialização. Mais recentemente, o processo automático de produção é obtido com um sistema de medida de componente incorporado na própria máquina, as correções são executadas imediatamente; e apenas há necessidade da intervenção humana na quebra ou reprogramação da máquina.

Esta automatização está ligada ao desenvolvimento de máquinas cada vez mais versáteis, gerando produtos geometricamente complexos e num intervalo de tempo cada vez menor.

Este processo está exigindo tecnologia mais avançada e atualmente empregam-se no controle numérico dessas máquinas, microcontroladores e microcomputadores que realizam também operações de monitoramento e supervisão dos sistemas.

O resultado de todo este desenvolvimento é a substituição da mão-de-obra humana em vários setores, a produção em larga escala e a globalização do comércio, entre outros.

1. Apresentação do Sistema

1.1 Resumo

As várias unidades que compõem o sistema serão descritas, de forma sucinta, neste capítulo, além de suas funções. A finalidade é fornecer suporte, para o entendimento global do sistema.

1.2 Descrição Funcional

No hardware, temos os seguintes módulos:

- . Computador IBM PC-XT/AT responsável pelo processamento dos dados e gerenciamento do controle
- . Máquina Fresadora composta de duas mesas X, Y e um eixo árvore Z
- . Unidade de Potência usada para o acionamento dos motores DC
- . Motores DC que convertem a energia, fornecida pelos "drivers", em deslocamento da fresa
- . Decodificador Incremental de Posição Angular responsável pela codificação em pulsos do deslocamento angular, efetuado pelo eixo do motor, quando há deslocamento na mesa da máquina.

Na figura 1.1 temos uma descrição em blocos de todo o equipamento.

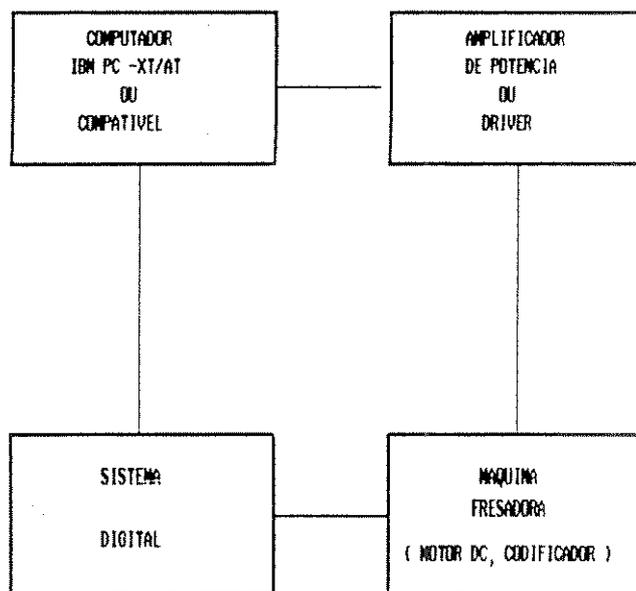


Figura 1.1

Diagrama de Blocos do Sistema

1.3 Arquitetura da Unidade de Processamento e Gerenciamento

Os dados digitais necessários ao processo são gerenciados em um software, desenvolvido em um microcomputador da linha PC.

Este microcomputador possui uma arquitetura baseada em um processador de 16 ou 32 "bits", sendo um projeto aberto que possibilita a conexão de interfaces e dispositivos de modo simples. Esta característica traz vantagens, a nível de software básico, já que a ROM (memória apenas de leitura) pode ser modificada sem que haja mudanças no equipamento original (computador).

1.3.1 Configuração do Sistema

O PC possui um módulo básico, a placa do sistema, e diversos periféricos conectados nela.

A alimentação da Placa Mãe e das que controlam os periféricos é feita através de uma fonte de alimentação, a qual fornece quatro níveis de tensão DC: +5V, -5V, +12V, -12V. Como a maioria dos circuitos integrados do PC necessitam de alimentação padrão TTL (+5V), este nível possui a maior capacidade para fornecimento de corrente. A tabela mostra as características de uma fonte padrão para o PC.

DC(nominal) volts	valor máximo volts	valor mínimo volts	corrente máxima (A)
+ 5	5.25	4.80	7.0
- 5	- 4.60	- 5.50	0.3
+ 12	12.60	11.52	2.0
-12	- 10.92	- 13.20	0.2

Tabela 1.1

Fonte de Alimentação Padrão para o PC

Estes níveis de alimentação estão disponíveis nos "slots" e podem ser utilizados em projeto de interfaceamento.

A entrada de dados é feita através de um teclado, enquanto que mensagens destinadas ao operador são apresentadas em um vídeo padrão CGA ("Color Graphic Adapter").

Das portas de comunicação existentes no PC a usada foi a serial, pois dados armazenados na RAM do microcontrolador são transferidos para o computador, através desta porta. A transmissão pode ser programada numa velocidade que

varia de 300 a 9600 bps ("bits por second"). Opera no padrão RS 232-C, como um dispositivo terminal, DTE ("Data Communication Equipment"). O conector usado é o padrão DB-25P, tipo macho. Somente alguns dos seus pinos são utilizados, de acordo com a descrição da tabela seguinte.

Direção	DTE		DCE
Saída	2.	TD ("Transmit data")	2.
Entrada	3.	RD ("Receive data")	3.
Saída	4.	RTS ("Request to send")	4.
Entrada	5.	CTS ("Clear to send")	5.
Entrada	6.	DSR ("Data set read")	6.
Comum	7.	GND ("Ground", terra)	7.
Entrada	8.	DCD ("Carrier detect")	8.
Saída	20.	DTR ("Data Terminal Ready")	20.
Entrada	22.	RI ("Ring indicator")	22.

Tabela 1.2
Descrição da Porta Serial RS-232C

O armazenamento de informações pode ser efetuado em um diskete (360 "Kbytes") ou num disco rígido, cuja capacidade de armazenamento varia de 20 "Mbytes" até algumas centenas de "Mbytes".

O controle da maioria das funções executadas no PC é efetuada na Placa Mãe. Nela estão o microprocessador, a memória RAM ("Random Access Memory", ou memória volátil), a ROM ("Read Only Memory", memória de leitura), o conector de teclado, a saída de vídeo e do alto falante. Para interfaceamento e controle de periféricos (impressoras, "drivers" e outros), existem cartões digitais conectados em um dos seus 8 "slots". Cinco destes "slots" estão disponíveis ao usuário.

Interface com o Exterior - "Slot"

A capacidade de adaptar novos periféricos ao PC é devido à existência dos 5 "slots" livres de expansão. Esta facilidade permite projetar placas com tarefas específicas de I/O e contar com todo o potencial de processamento do computador. A figura 1.2 ilustra os 62 pinos do "slot" de interfaceamento.

A maioria dos sinais do duto interno do PC estão presentes no duto externo. Será descrita a função de cada um destes pinos, os quais são usados para

projetar cartões digitais.

.Barramento de Dados/Endereços (AD0 - AD7)

Este conjunto é usado para transmitir informação e endereçar a memória e circuitos de I/O, em cada ciclo de máquina. Eles, entretanto, são multiplexados; o "byte" menos significativo do endereço (A0-A7) é apresentado no barramento no início do ciclo e, mais tarde aparece o "byte" do dado (D0-D7). Na arquitetura do PC, a informação do endereço (A0-A7) é guardada num "latch" externo.

.Barramento de Endereços (A8-A15)

São oito linhas que compõem a parte mais significativa do endereço. Não são multiplexadas e permanecem estáveis durante o ciclo de escrita e leitura em memória ou I/O.

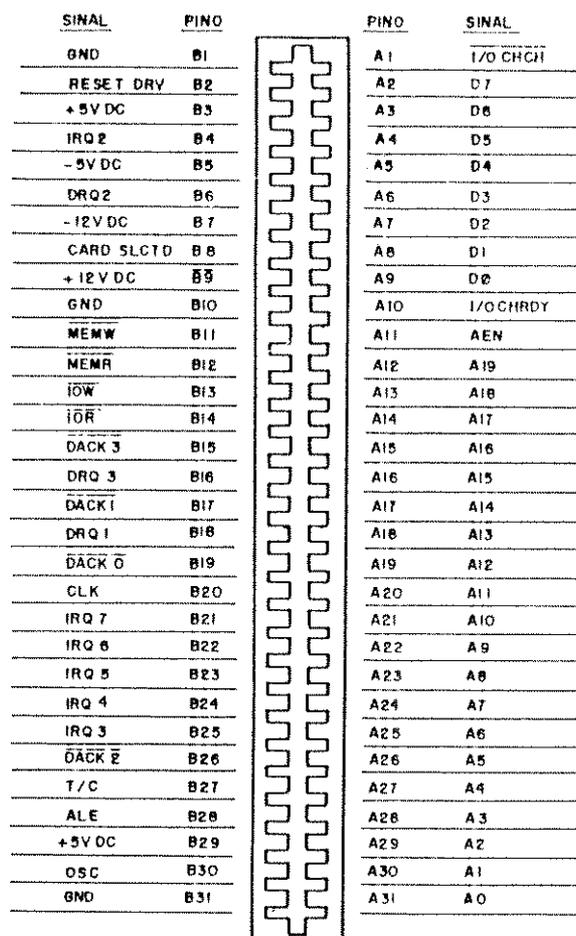


Figura 1.2

Definição dos Pinos e Sinais do "Slot"

.Ready

Esta linha é usada para inserir estados de espera nos

ciclos de escrita e leitura e assim prolongar o comprimento de um ciclo. Este sinal é usado para diminuir a "velocidade" do processador quando o mesmo está acessando uma porta de I/O, ou uma memória que é muito lenta num ciclo executado pelo micro.

.OSC

É um sinal de saída com uma frequência de 14.31818MHz e um período, aproximadamente, de 70 nanossegundos (ns). Possui um "duty cycle" de aproximadamente 50%. Outros sinais de sincronização são gerados a partir dele. A frequência do mesmo pode ser ajustada via um capacitor de "trimming" localizado na Placa Mãe.

.CLK

É derivado do sinal OSC. Obtido pela divisão do sinal OSC por um fator de três, o qual resulta numa frequência, para o mesmo de 4.77MHz. Ele não é simétrico, sendo sua duração em nível alto de 70 ns e em baixo de 140 ns. Sua sincronização com os sinais de controle de R/W (leitura e escrita) sempre ocorre, sendo que o mesmo pode ser usado para gerar estados de espera.

.Reset Drv

Mantido ativo alto durante o "power-on" do sistema e até que todos os níveis tenham alcançado seu nível de operação, então vai a nível baixo, permanecendo assim. Usa-se o mesmo para fornecer um "reset" as Interfaces lógicas de I/O, a fim de colocá-las em um estado conhecido durante sua inicialização.

.Interrupção (IRQ2-IRQ7)

Geram as requisições de interrupção para o processador. Os programas da BIOS inicializam o periférico (8259), que as trata, de tal forma a ter IRQ2 como o sinal de maior prioridade e IRQ7 o de menor. Uma vez que ocorre um pedido de interrupção, o mesmo deve se manter até que o processador o atenda.

.IOR (leitura de I/O)

É usado para indicar que o ciclo iniciado é uma operação em I/O e o endereço no barramento corresponde a um endereço de porta. À porta selecionada, deve responder colocando no barramento de dados o seu endereço, durante este ciclo.

.IOW

Ativo em nível baixo, indica que o barramento de endereços contém um valor de porta de I/O e no barramento de dados uma informação a ser escrita nela. Enquanto este sinal está a nível baixo, o dado permanece válido e deve ser guardado em um "latch" ou registrador de porta.

+.5 Volts DC

Disponível em um dos pinos, fornece uma tensão de 4.75 V

a 5.25 Volts DC.

.+12 Volts DC

Está presente em um pino e apresenta o valor de 11.4 V a 12.6 Volts DC.

.-12 Volts DC

É o complementar do sinal acima e fornece uma tensão de -10.8 V a -13.2 Volts DC.

. Terra

Está presente em três pinos do conector de saída, sendo o sinal de referência para as outras tensões (+5V, +12V, -5V).

1.4 Descrição da Máquina Fresadora

A máquina em questão está inserida no grupo das máquinas ferramentas, e é utilizada na fabricação de peças, objetos de metal ou outros componentes.

Numa visão suscinta, seus principais elementos são:

.duas mesas de alumínio que deslizam na direção X, Y. Em uma delas a peça, a ser trabalhada, é encaixada e presa através de grampos.

.um eixo vertical ou eixo árvore construído do mesmo material que os outros dois, desliza na direção Z. Neste eixo a ferramenta é presa. Possui também um canal que contém fluido refrigerante, e este tem por função diminuir o atrito entre a ferramenta e a peça ou apara, além de esfriar a ferramenta. Deve ser antisséptico, não oxidante, para peças, máquinas e pessoas.

.uma base com furos, cuja função é fixar a estrutura em um local, para evitar a vibração da mesma durante sua operação.

.três motores de corrente contínua, cuja tensão de alimentação é 12 Volts DC. Cada um destes movimenta os eixos (X,Y,Z) nos dois sentidos. Existe um pequeno motor no eixo árvore, com a função de injetar líquido refrigerante.

.três Codificadores Incrementais de Posição Angular, cada um situado em um eixo do motor e girando de forma concêntrica com ele. Este sistema óptico converte o deslocamento angular (giro do eixo do motor) em pulsos digitais, os quais são utilizados para medir a velocidade de giro e o deslocamento linear de cada mesa na direção que está trabalhando.

.um sistema de transmissão cuja função é acoplar o eixo do motor à mesa, para que a mesma possa se deslocar. Na figura 1.3 temos uma visão deste tipo de ligação.

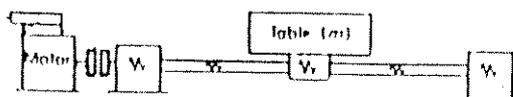


Figura 1.3
Acoplamento entre Motor e Mesa

uma Unidade de Potência e Controle que é responsável pelo fornecimento de energia aos motores DC e controle de variáveis como corrente e velocidade dos mesmos. Tal sistema é responsável pelo giro do motor nos sentidos horário e anti-horário, possibilitando o deslocamento das mesas e do eixo árvore em ambas as direções, direto e inverso. Todo o módulo é colocado em uma estrutura de metal para evitar efeitos como choque, ruídos, ataque de agentes externos (oxidação). Sua ligação aos motores e ao sistema de controle digital é feito através de cabos blindados e conectores para evitar que haja perda de sinal ou mesmo rompimento.

Alguns dos módulos descritos acima serão apresentados nas próximas seções de forma detalhada.

1.5 Motores de Corrente Contínua

São alimentados com uma tensão DC de 12V e empregados em máquinas ferramentas por possuírem as seguintes características:

- .baixo momento de inércia;
- .alto torque de aceleração;
- .grande estabilidade, até em baixas velocidades;
- .pequena constante de tempo, devido à sua construção;
- .grande capacidade de sobrecorrente, por curtos intervalos de tempo.

São motores cujo campo do estator é gerado por ímãs permanentes, o que implica que nenhuma potência é gasta nesta estrutura. O fluxo magnético mantém-se essencialmente constante em todos os níveis de corrente de armadura e assim a sua curva torque-velocidade é linear. A corrente é conduzida para o coletor por escovas especiais de carbono. Isto permite menos flocamento entre elas e assim a vida útil do motor é aumentada.

A armadura é construída de um material isolante, possuindo condutores de cobre e o campo magnético é uniformemente distribuído. A ausência de ferro nela reduz os efeitos de inércia, possibilitando altas taxas de crescimento da corrente a ser aplicada, de tal maneira que a velocidade do motor pode ser rapidamente ajustada em um patamar de velocidade desejado.

O alto torque disponível gera grande quantidade de calor, o qual é dissipado para o meio externo por um eficiente sistema de troca de calor, impedindo assim prejuízos às características do motor.

Velocidades de algumas centenas de rpm (rotações por minuto) podem ser conseguidas com uma diminuição do torque. A figura 1.4 ilustra sua estrutura interna.

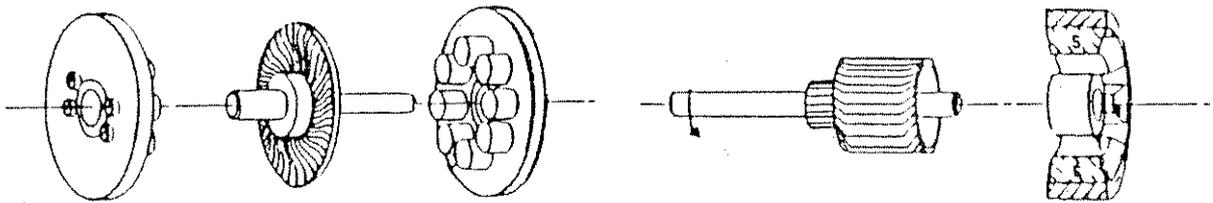


Figura 1.4

Motor DC com Excitação de Ímã Permanente

Devido às características mencionadas anteriormente, este tipo de motor, usado em máquinas ferramentas, possui tamanho e peso reduzidos, o que facilita não só seu manuseio como também o seu transporte, além de causarem esforços menores nas estruturas onde estão alojados.

1.6 Codificador Incremental de Posição Angular

Em sua estrutura básica, é formado por uma fonte de luz, um sensor óptico e um disco com várias raias opacas e transparentes. É encaixado no eixo do motor e gira concentricamente com o mesmo. Conforme o eixo gira, o disco se movimenta pelas raias, fazendo com que o sensor óptico (foto transistor) registre esta ação. Este movimento gera uma seqüência de pulsos que será posteriormente tratada pelo tacômetro digital (cartão).

O sistema fica encapsulado em um disco de metal que o protege. As saídas do equipamento são sinais de baixa potência, 20 a 40 mV. Conectado a cada um

destes sinais existe um amplificador que transforma cada saída num sinal TTL. Em ambientes ruidosos, como por exemplo chão de fábrica, o ruído das máquinas pode induzir sinais espúrios na saída do sensor ou do amplificador. Para evitar este tipo de problema, toda estrutura é blindada.

Um outro problema que pode afetar a contagem é a concentricidade entre o Codificador e o eixo do motor, já que os dois devem girar sem folgas.

Os pulsos gerados são usados para monitorar digitalmente a velocidade do motor e o deslocamento da mesa da máquina.

Cada uma das saídas é ligada a um dos pinos de um conector como é visto a seguir. Apresentamos uma descrição de cada um deles.

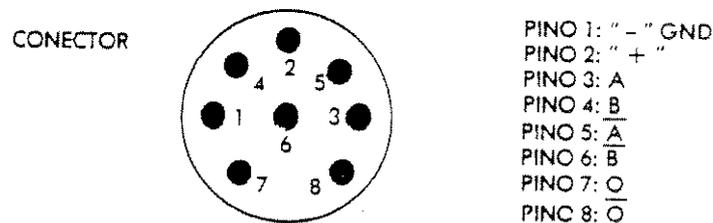


Figura 1.5

Conector de Saída do Codificador Incremental de Posição Angular

.GND

Sendo o terra do sistema, deve ser ligado numa linha correspondente de baixíssimo ruído, já que os níveis de tensão são TTL.

.VCC

Corresponde a alimentação do conjunto.

.sinais A e B

São duas saídas defasadas de noventa graus uma da outra. Pela análise de qual delas está adiantada em relação a outra, pode-se determinar o sentido de giro do motor . Qualquer uma pode ser usada para indicar o deslocamento angular que houve, em um certo intervalo de tempo.

.sinais \overline{A} e \overline{B}

São complementares dos sinais A e B.

.sinais O e \overline{O}

Estão defasados de 180° . Indicam quando o eixo completou uma volta.

1.7 Sistema de Transmissão

A função dos elementos de transmissão mecânica, em um controle de

posição, é transmitir o movimento recebido do motor proporcionalmente a peça de trabalho. O fluxo de transmissão para a conversão do torque do motor em um movimento linear passa através de muitos componentes individuais, os quais comportam-se como sistemas massa-mola. Estas características conduzem a vibrações quando há excitações. Estas vibrações entre a peça de trabalho e a ferramenta pode gerar:

- .erros de geometria;
- .acabamento de superfícies pobre;
- .incremento no desgaste da ferramenta e da máquina.

Na fig. 1.6 temos um diagrama esquemático de um sistema de transmissão.

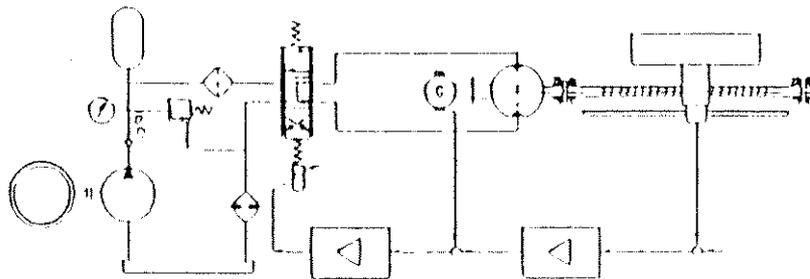


Figura 1.6

Diagrama de um Sistema de Transmissão

Alguns elementos que fazem parte deste sistema são:

- .caixa de engrenagens;
- .fuso de esferas e rosca;
- .eixos;
- .rolamentos;
- .fluidos.

O posicionamento de certos elementos no sistema e sua geometria contribuem para gerar desajustes. Dentre eles podemos citar:

- .erros de alinhamento de mancais;
- .erros de alinhamento e paralelismo nos guias da mesa.

Um tipo de vibração que pode surgir é causado por forças de excitação as quais variam com respeito ao tempo. Estas incluem erros de aceleração e auto excitação (ressonância) as quais afetam o deslocamento da mesa.

Com relação aos efeitos de vibração, pontos de baixa frequência têm grande influência nos erros de transmissão dinâmicos. Para simplificação do comportamento dinâmico dos elementos de transmissão, o mesmo pode ser aproximado por um sistema de vibração com uma relação de tempo de segunda ordem, onde o ponto de ressonância mais baixo do sistema é a sua frequência natural [99].

Grandes massas e flexibilidade excessivas nos elementos são responsáveis por modos de vibração em baixas frequências.

Nas máquinas ferramentas desenvolveu-se uma estrutura de transmissão que procura eliminar ou atenuar todos estes tipos de problemas apresentados na discussão acima. Este sistema é conhecido como transmissão por fuso de esferas recirculantes e apresenta as seguintes características:

- .alta eficiência ($\eta = 0.95$ a 0.99);[99]
- .grande resistência, por ser feito de metal;
- .alta rigidez;
- .capacidade de pré-carga;
- .baixo coeficiente de atrito ($\mu = 0.01$ a 0.02);[99]
- .pequena dissipação de calor;
- .mínimo desgaste

A seguir temos uma visão de sua estrutura.

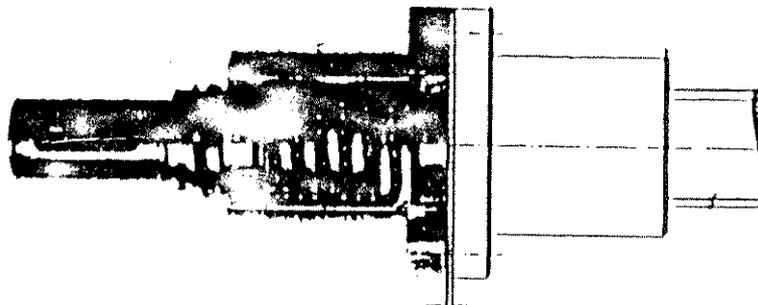


Figura 1.7

Sistema de Transmissão por Fuso de Esferas Recirculantes

A porca do fuso tem um canal interno, onde os rolamentos deslizam, além de possuir um lubrificante para diminuir o atrito. Conforme o parafuso gira, a mesa executa um deslocamento linear em uma direção direta ou inversa.

1.8 Unidade de Potência e Controle

A máquina ferramenta controlada numericamente permite que o motor opere nos quatro quadrantes de velocidade, ou seja:

- .giro do motor no sentido horário;
- .giro do motor no sentido anti-horário;
- .frenagem do motor no sentido horário;
- .frenagem do motor no sentido anti-horário;

O movimento do motor em um dos sentidos, bem como sua frenagem, é realizada através de um módulo de controle analógico/digital. O módulo de potência por sua vez, é responsável pelo fornecimento de energia aos motores da máquina. Um diagrama em blocos desta unidade é mostrado na figura 1.8.

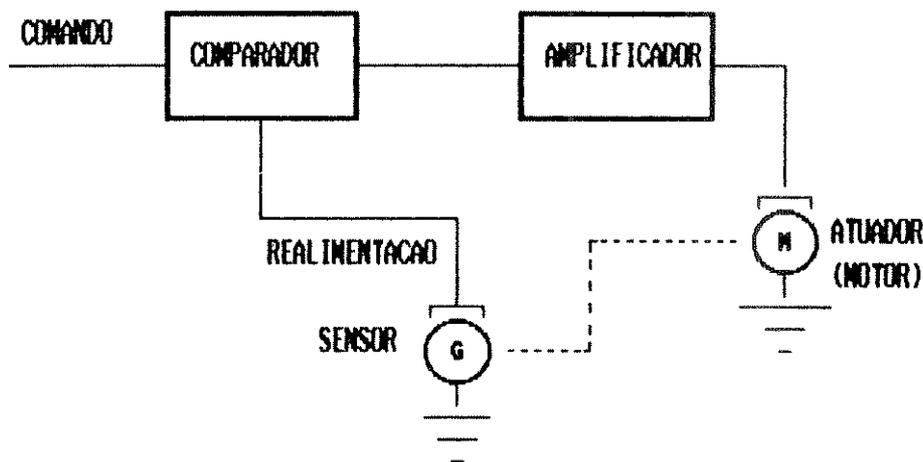


Figura 1.8

Unidade de Controle da Velocidade

1.8.1 Módulo de Controle

O motor de corrente contínua é um transdutor eletromecânico que converte a tensão contínua, aplicada nos seus terminais, em movimento mecânico do seu eixo, ou seja, ele gira a uma certa velocidade angular, proporcional à tensão.

A entrada do sistema é uma tensão de referência que se traduz em uma dada velocidade. O amplificador ao ser excitado com essa tensão, aciona o motor que começa a girar com uma certa velocidade. Esta é convertida em pulsos digitais através de um Codificador Incremental de Posição Angular (ver Seção 1.6). Estes pulsos gerados na sua saída são processados em um cartão (tacômetro digital (ver Cap. 2 e 3)), o qual fornece indiretamente uma medida da velocidade angular. Com este dado e uma nova referência, o computador faz a subtração dos dois valores e gera um sinal de erro, ou seja, um valor digital que será transformado em uma tensão de referência analógica, através de um conversor D/A.

Assim, a unidade de controle opera da seguinte maneira:

.quando o sistema está desativado (motor parado), e aplica-se na sua entrada um sinal, com uma certa amplitude, a tensão de erro é grande, pois o eixo do motor está parado. Este sinal faz com que haja movimento do motor, e conseqüentemente a tensão de erro começa a diminuir proporcionalmente ao aumento de velocidade. Esta tenderá a se elevar rapidamente, até ultrapassar o valor de referência, resultando num valor de erro negativo. Neste ponto, há uma diminuição da tensão de alimentação do motor e a velocidade passa novamente pela referência. O ciclo se repetirá para cada novo valor a ser ajustado ao motor.

.o sistema permanece estabilizado, pois há uma "igualdade" entre o valor real e a referência. Esta situação ocorre somente em pouquíssimas situações pois qualquer perturbação pode alterar um dos valores.

.caso ocorra alguma perturbação no movimento do motor, a sua velocidade pode aumentar ou diminuir, causando um desequilíbrio entre os valores de tensão (real e referência). O resultado é um movimento mais rápido ou mais lento, que logo deverá cessar e o motor deverá voltar a girar em um certo patamar de velocidade pré-estabelecido.

1.8.2 Unidade de Potência

A unidade de controle fornece à de potência uma tensão DC de referência. Este sinal é amplificado na sua característica de corrente, a fim de alimentar o motor. Assim o circuito de entrada deste módulo é um amplificador com uma impedância de entrada alta, para evitar carregamento, e uma impedância de saída baixa.

Os elementos fornecedores de corrente são transistores de potência, os quais trabalham em sua região não linear para evitar uma grande dissipação de calor e o uso de dissipadores enormes. Portanto, funcionam como chaves e desta

forma a energia dissipada é muito menor, pois este fenômeno só ocorre na passagem entre a condução e o corte e vice-versa.

A tensão que chega a estes transistores é uma tensão recortada sendo fornecida por um circuito PWM. Através desta técnica, a corrente fornecida pelos transistores é controlada pela largura do pulso de tensão, em cada ciclo de serviço.

A estrutura básica de um PWM é composta de um gerador de rampa, um sinal de erro e um comparador. A cada período é gerada uma onda triangular (referência) e esta é comparada com a tensão de erro, fornecida pela unidade de controle. Enquanto a tensão de referência for menor que a do erro na saída do comparador temos uma tensão baixa, quando a situação se inverte a tensão será alta. Portanto, a amplitude da tensão de saída do comparador determinará o tempo de chaveamento entre a condução e o corte dos transistores.

Qualquer diminuição da velocidade provocada, por exemplo, por um aumento de carga fará aumentar o erro de velocidade detetado. A largura do pulso, responsável pela condução do transistor, se torna maior e há um aumento do valor médio da tensão fornecida ao motor, permitindo que este volte a girar na velocidade estabelecida.

Este aumento de carga (torque) é limitado pela corrente do motor. Quanto maior a carga, mais consumo de corrente há e o aquecimento interno se eleva, podendo provocar danos permanentes ao motor. Para evitar este sobreaquecimento são usados sensores de corrente, como por exemplo os de efeito Hall, cuja função é limitar a corrente do motor.

Um esquema de todo o conjunto é visto na figura 1.9.

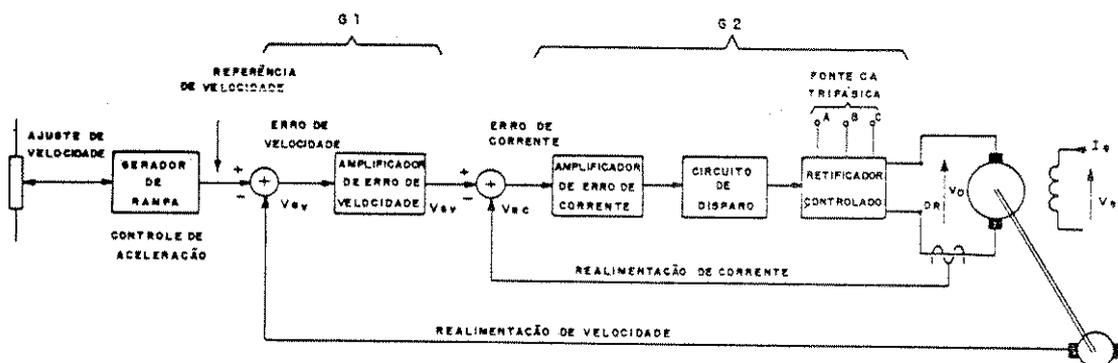


Figura 1.9

Diagrama do Amplificador de Potência com o Circuito de Proteção

2. Descrição do Módulo Digital

2.1 Resumo

Este módulo apresenta duas versões:

.o primeiro cartão digital foi desenvolvido para ser instalado no "slot" do computador do tipo IBM PC/XT-AT, e usa os recursos disponíveis deste barramento.

.uma segunda versão contém um microcontrolador da família 8051, o 87C51.

Qualquer um dos sistemas tem a função de monitorar a velocidade do motor DC e a posição da mesa da máquina fresadora. Acoplado a um destes dois módulos existe uma unidade que fornece tensões analógicas usadas no amplificador de potência (conversores D/A).

Neste capítulo apresentaremos o cartão digital, que está conectado ao "slot" do PC, e executa os processos de velocidade e posição do sistema. No próximo passaremos a descrever a unidade com microcontrolador.

2.2 Análise Estrutural do Cartão digital

Com a finalidade de facilitar o entendimento e programação deste cartão digital, serão detalhados os vários módulos que o compõe.

Esta análise permitirá a compreensão dos algoritmos, apresentados numa fase posterior. Este sistema é formado de duas unidades principais:

.a primeira recebe pulsos vindos do Codificador Incremental, acoplado ao eixo do motor, pré-processa esta informação e a envia ao microcomputador. Através de operações matemáticas executadas sobre os dados contidos na informação, obtém-se o valor da velocidade angular do eixo, e o deslocamento linear executado pela mesa da máquina.

. a outra unidade fornece, a partir de sinais de controle, tensões analógicas que serão usadas pelo amplificador de potência ("driver"). Como consequência, o eixo do motor girará a uma velocidade proporcional a esta tensão. Esta rotação será transferida à unidade de transmissão que transformará o deslocamento angular em linear, efetuado pela mesa. (ver Seção 1.7)

2.2.1 Características Gerais

o cartão digital, utilizado na primeira versão, deste sistema, para a montagem do hardware é compatível com os barramentos do computador IBM PC/XT-AT. Para que o mesmo possa operar deve ser encaixado em um dos seus "slots".

o deslocamento linear efetuado pela mesa é obtido em intervalos regulares de tempo. O seu valor é processado em uma rotina de posição que compõe o software básico.

os sinais analógicos são gerados através de conversores digitais/analógicos de 10 "bits", e sua tensão de saída varia de -10 V a +10 V, o que permite o motor girar no sentido horário e anti-horário.

há um cabo "flat-cable" utilizado para fazer a comunicação entre o Codificador e este sistema. Por ele passam informações digitais que são usadas no processamento.

2.2.2 Diagrama de Blocos

Uma das maneiras de se visualizar cada módulo e facilitar a compreensão da função que ele desempenha é através de um diagrama de blocos. Utilizando esta ferramenta, temos na figura 2.1 a representação do cartão digital, em diagrama de blocos.

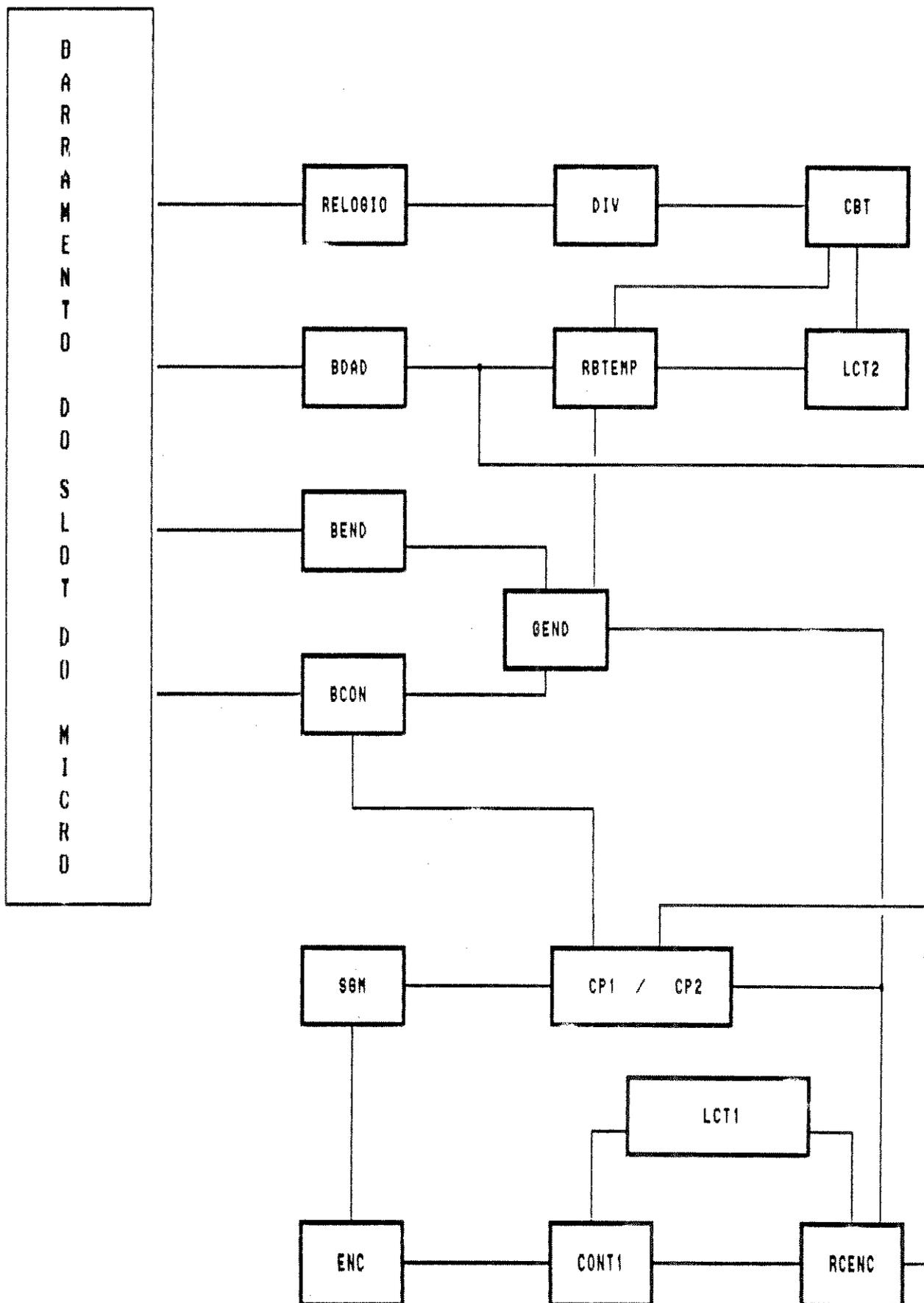


Figura 2.1
Diagrama de Blocos do Cartão Digital

2.2.3 Descrição dos Blocos

.Relógio

Fornece um sinal de "clock" da ordem de 14.33 MHz, o qual pode ser usado para sincronizar as operações que envolvem vários sinais.

.BDAD ("buffer" de dados)

Este módulo tem a função de "bufferizar" as linhas de dados.

.BEND ("buffer" de endereços)

Este bloco é usado como "buffer" nas linhas de endereço, vindas do "slot" do microcomputador.

.BCON ("buffer" de controle)

Serve como "buffer" para os sinais de controle.

.GEND

Codifica os endereços de I/O.

.DIV (divisor de frequência)

Divide por dois o sinal do relógio, sendo a sua frequência usada nos contadores da base de tempo.

.ENC (Codificador Incremental de Posição Angular)

O sinal gerado em suas saídas permite, após o seu processamento, a obtenção de informações a respeito dos entes físicos, velocidade e posição.

.SGM (sentido de giro do motor)

Permite decodificar o sentido de giro do eixo do motor, a qualquer instante.

.CONT1 (contador um)

Conta os pulsos vindos do Codificador. Quando a operação é completada, gera um sinal que fará com que uma nova amostragem seja iniciada.

.LCT1 (lógica de controle um)

Gera os sinais de controle para o contador que processa os pulsos vindos do Codificador Incremental de Posição Angular.

.RCENC ("latch" do contador do Codificador Incremental de Posição Angular)

Armazena o valor acumulado no contador do Codificador, durante uma amostragem do processo.

.CBT (contador da base de tempo)

Recebe os pulsos vindos do módulo divisor de frequência (DIV). O número de pulsos processados por ele gera a base de tempo, para um certo número de pulsos que é registrado no contador um.

.LCT2 (lógica de controle dois)

Sua função é gerar os sinais de controle para o módulo da base de tempo.

.RBTEMP (registro da base de tempo)

Armazena a contagem efetuada pelos contadores da base de tempo, à cada ciclo da operação.

.CP1 (contador de posição um)

Conta os pulsos vindos do Codificador, quando o motor está girando no sentido horário.

.CP2 (contador de posição dois)

Opera de forma análoga ao contador de posição um, só que registra os pulsos quando o motor está girando no sentido anti-horário.

.BDA (banco de conversores D/A)

Fornece os sinais analógicos de tensão para o "driver" do motor da máquina fresadora (ver figura abaixo).

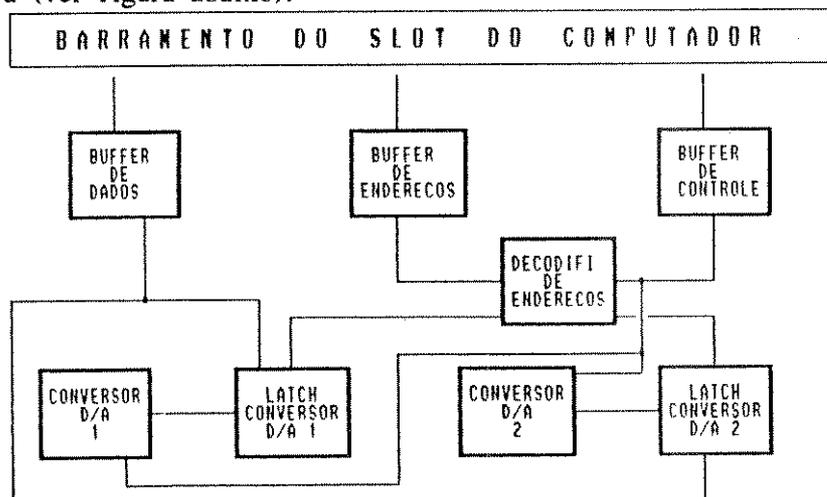


Figura 2.2

Cartão dos Conversores D/A

2.2.4 Endereçamento

No cartão, existem 3 dispositivos mapeados em I/O. São eles:

- .registro do contador do Codificador
- .registro do contador da base de tempo
- .registros dos contadores de posição

O endereço do cartão é formado, conforme a tabela abaixo:

bit de endereço	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
valor do bit	1	1	D7	D6	D5	D4	D3	D2	D1	D0

Tabela 2.1

Formação do Endereço Base do Cartão Digital

Os "bits" D7 a D0 são os valores carregados no barramento de dados. Analisando as faixas de endereços de I/O livres no PC, podemos perceber que uma das regiões disponíveis é a que vai de 300 (hexa) a 31F (hexa). Assim, o endereço base deste cartão é o 300 (hexa), e todos os outros circuitos de I/O que estão montados sobre este cartão, possuem endereços a partir deste valor. Na tabela 2.2 temos o mapeamento destes dispositivos e se são configurados como circuitos de leitura ou escrita, vistos pela operação do microcomputador.

Estes endereços serão usados em rotinas que executarão algumas funções como:

- .configuração de componentes;
- .inicialização de registros;
- .leitura de dados digitais;
- .geração de sinais de controle;

O endereço base destes "chips" é descrito acima, enquanto que seus "offsets" são apresentados na tabela seguinte.

Módulo	Offset	Operação L/E	Função
RCE	00	L/E	LATCH CONTENC
RMO	04	L/E	REG. OPER.
RCP/RCBT	08	L/E	REGISTRO CONTADOR 0
	09	L/E	REGISTRO CONTADOR 1
	0A	L/E	REGISTRO BASE DE TEMPO
	0B	E	REGISTRO CONTROLE

Tabela 2.2
Mapeamento dos Endereços dos Circuitos de I/O

onde:

- .RCBT - registro do contador da base de tempo
- .RCE - registro do contador do Codificador
- .RCP - registros dos contadores de posição
- .RMO - registro de modo de operação

Tanto os registros ligados à base de tempo, quanto os de posição são configurados através de uma palavra de controle carregada no seu registro de controle. Esta operação é efetuada na inicialização do software através de uma rotina de configuração. Em seguida são armazenados os valores desejados nestes registros para que o sistema possa começar a operar. Esta operação também é efetuada no "latch" ligado ao Codificador Incremental de Posição Angular e ao RMO.

Sempre que um novo ciclo se iniciar estes registros são carregados com seus valores correspondentes para que o processo recomece novamente.

Quanto ao RMO este não foi descrito separadamente como um módulo, pois ele faz parte da lógica que controla os contadores do Codificador Incremental de Posição Angular e da Base de Tempo.

2.2.5 Operação

Este cartão digital, referente ao módulo de velocidade, possui três modos de operação que são:

.MODO 1

É aquele no qual a janela de amostragem da velocidade, está ajustada para um valor de velocidade que é menor que o do motor. Como consequência o tempo de amostragem da janela é inferior ao mínimo tempo para que o sistema possa coletar uma amostra de velocidade. Neste caso, há um "auto-resetamento" efetuado pelo hardware, e o dado de velocidade não é considerado. O sistema se autoconfigura, o que implica um tempo de janela maior, e o processo reinicia.

.MODO 2

O motor não gerou pulsos, durante a exposição da janela de velocidade. Então a janela é repetida várias vezes, no tempo, até que se consiga medir a velocidade angular do eixo do motor, ou então até que esta operação não possa ser mais efetuada. Após ocorrer uma das operações descritas acima, o processo se reinicia.

.MODO 3

O tamanho da janela de amostragem da velocidade é coerente com a velocidade angular do motor. Os dados referentes à posição são armazenados e depois transferidos ao microcomputador para que o mesmo calcule o seu valor. Assim que a informação de velocidade está nos registros, o sistema já está começando uma nova operação.

Quanto ao módulo de posição, os pulsos armazenados nos seus registros são lidos em intervalos de tempo fixo, mas independentes da operação de velocidade.

2.3 Software do Cartão Digital

Descreveremos através de algoritmos as rotinas que formam a estrutura do software. Nos algoritmos detalham-se cada uma das tarefas a ser implementada, permitindo o fácil entendimento das linhas de codificação do programa. Para facilitar a implementação do software, em uma linguagem de alto nível, o mesmo foi dividido em três módulos que são:

- .módulo de inicialização
- .módulo de posição
- .módulo de velocidade

Para a codificação do software, foi utilizado o TURBO C da Borland, versão 2.0. Após sua codificação, compilação e "linkagem" executou-se os testes de validação.

A nível de interface homem-máquina há um "menu" onde o usuário fornece as informações necessárias para que o sistema possa operar. Após estas informações terem sido fornecidas ao software, a unidade digital começa a operar.

2.3.1 Módulo de Inicialização

Neste módulo são configurados e inicializados os registros de I/O.

Algoritmo

- .programa o registro de controle dos contadores de posição com a palavra de controle
- .lê seu "status" e verifica se está programado corretamente, senão programa-o novamente
- .programa o registro de controle, do contador da base de tempo, e verifica se está programado corretamente
- .carrega os contadores de posição com valor apropriado
- .carrega o contador da base de tempo
- .zera o contador do Codificador e programa-o para modo "UP"
- .libera a operação dos "clocks" destes contadores , através da programação do RMO (registro de modo de operação)

2.3.2 Módulo de Posição

Sua função é monitorar o deslocamento linear efetuado pela mesa da máquina fresadora, sempre que houver movimento do motor.

Algoritmo

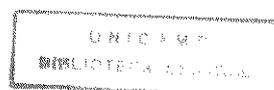
- .lê o "byte" menos significativo do valor armazenado no registro do contador de posição zero e guarda-o numa posição de memória
- .lê o "byte" mais significativo do valor contido no registro do contador de posição zero e coloca-o numa posição subsequente
- .reinicializa este contador com valor apropriado
- .lê o "byte" menos significativo do valor armazenado no registro do contador de posição um e guarda-o numa posição subsequente, de memória
- .lê o "byte" mais significativo do valor contido no registro do contador de posição um e coloca-o numa posição subsequente
- .reinicializa este contador com valor apropriado
- .executa operações matemáticas com estes valores e obtem o deslocamento real efetuado, pela mesa da máquina fresadora
- .este valor é somado aos deslocamentos que já haviam sido efetuados pela mesa, até aquele instante
- .guarda o resultado em uma posição de memória de dados

2.3.3 Módulo de Velocidade

É o responsável pelo gerenciamento da função de velocidade do sistema, a cada novo ciclo.

Algoritmo

```
.o contador da base de tempo chegou a zero?
  então
    .valor do contador do Codificador Incremental de Posição
      Angular é transferido para seu registro
    .reprograma-o para o modo de contagem "DOWN"
  senão
    .sistema fica esperando que o contador da base de tempo
      chegue à condição assinalada acima
.o contador do Codificador Incremental de Posição Angular chegou a
  zero antes que o da base de tempo tivesse zerado novamente?
  então
    .a parte menos significativa do dado, que está no
      registro da base de tempo, é transferida para uma
      posição de memória
    .a parte mais significativa do dado, que está no registro
      da base de tempo, é transferida para uma posição de
      memória subsequente
    .executa o cálculo para obtenção do valor da
      velocidade de giro do eixo do motor
    .reinicializa os contadores para uma nova coleta de
      informação referente à velocidade
  senão
    .contador da base de tempo zerou novamente, antes que
      o do Codificador Incremental de Posição Angular?
      então
        .valor do registro do contador do Codificador
          é diferente de zero?
          então
            .reinicializa os contadores
            .recomeça o processo
          senão
            .carrega no contador do Codificador o
```



valor unitário

.espera que um dos contadores
chegue a zero e repete os
processos acima

senão

.fica esperando que um dos contadores chegue a
zero

3. Sistema de Velocidade/Posição com Microcontrolador

Com a finalidade de desenvolver um sistema semelhante ao anterior no que concerne às suas funções, mas que ao mesmo tempo fosse capaz de se autogerenciar, implementou-se um novo cartão que possui como elemento central um microcontrolador, o 87C51. Este componente pertence à família dos microcontroladores 8051, sendo versátil para este tipo de aplicação.

Este componente permitiu a sintetização de várias operações lógicas, que eram executadas por componentes discretos no cartão anterior. Estas passam a ser efetuadas devido à sua estrutura de interrupção.

Para entendermos a concepção deste "novo projeto" serão abordados de forma sucinta as principais características deste componente. Em seguida discutiremos o projeto a nível de hardware/software e para finalizar apresentaremos o sistema de comunicação utilizado pelo microcontrolador.

3.1 Principais Características do Microcontrolador

O 87C51 é um microcontrolador rápido, com "clocks" de 8, 10 e 12 MHz, e suas características de hardware e software permitem usá-lo como um poderoso controlador, com até 64 "Kbytes" de memória de programa e 64 "Kbytes" de dados, isoladamente.

Suas características de hardware incluem capacidade de expansão de memória de programa (inicialmente em 4 "Kbytes" interna), para até 64 "Kbytes" (60 "Kbytes" externa) e da memória de dados (inicialmente em 128 "bytes" internos) para até 64 "Kbytes" (65408 "bytes" externa); quatro portas de I/O de 8 "bits" cada, "bits" esses individualmente endereçáveis; interrupção com estrutura "nesting" (nível de prioridade) com cinco linhas físicas selecionáveis e níveis de prioridade a escolher; 2 temporizadores (contadores de 16 "bits" programáveis).

Possui também um oscilador interno e facilidade de usar uma das portas como porta de funções alternativas, que incluem as interrupções, as entradas dos "timers/counters" (temporizadores/contadores), pinos para controle de memória externa e principalmente para canal serial, usado para comunicação.

Suas facilidades de software permitem expandir as memórias até os limites citados, fazer complexas operações aritméticas e lógicas (multiplicação, deslocamentos e permutas de "bits"), trabalhar com bancos de registros nominais, e inclusive operar com até 218 "bits" individualmente endereçáveis para teste, operações booleanas e "set/reset".

3.1.1 Descrição Funcional dos Pinos

A figura 3.1 ilustra a configuração dos pinos.

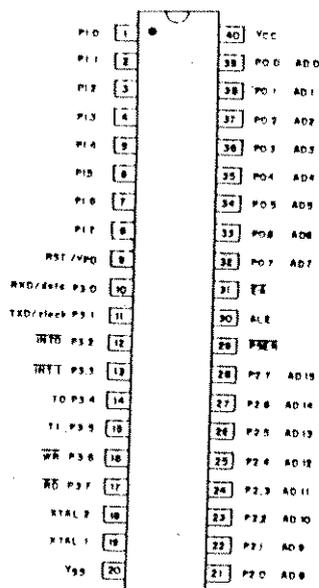


Figura 3.1
Pinagem do Microcontrolador 87C51

A função que cada pino desempenha é descrita a seguir.

Pinos 1 a 8 - São endereçáveis como porta bidirecional de 8 "bits", podendo ter seus pinos individualmente endereçáveis. Podem alimentar ou drenar uma carga TTL.

Pino 9 - Através dele pode-se "resetar" o sistema pela aplicação de um nível lógico "1" TTL, durante dois ou mais ciclos de máquina. Serve também para o modo "standby" para a RAM interna.

Pinos 10 a 17 - Porta bidirecional de 8 "bits".

Pino 18 - Saída do amplificador inversor do oscilador e entrada para o gerador de "clock" interno. Se for usado "clock" externo, serve como entrada para o mesmo.

Pino 19 - Entrada do amplificador inversor do oscilador.

Pino 20 - Conexão de terra.

Pinos 21 a 28 - Porta bidirecional de 8 "bits", usada como saída ou entrada da parte mais significativa do endereço na expansão de memória.

Pino 29 - Saída para habilitação do programa externo. Funciona como "strobe" da memória de programa externa.

Pino 30 - Entrada de seleção de memórias.

Pinos 32 a 39 - Porta de 8 "bits" bidirecional. Durante operações com memória externa, fornece multiplexados o "byte" menos significativo do endereço e o dado.

Pino 40 - Conexão de alimentação.

3.1.2 Estudo dos Blocos do Microcontrolador

Na figura 3.2 temos um diagrama de blocos do microcontrolador utilizado.

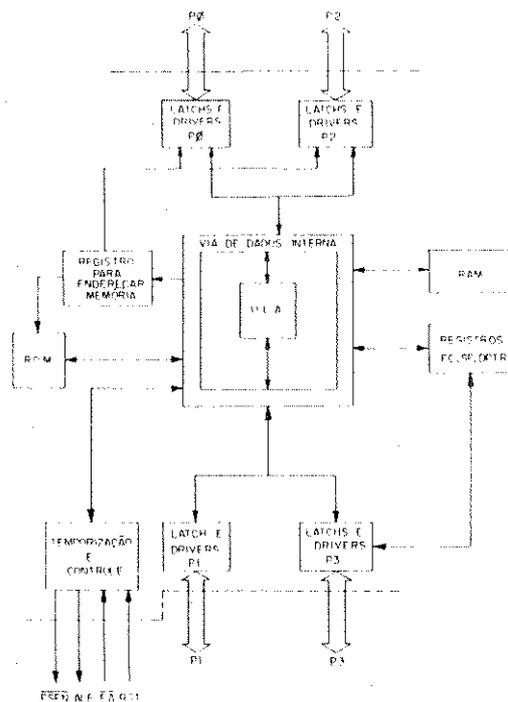


Figura 3.2

Diagrama de Blocos do 87C51

Organização da Memória

A memória está dividida conforme o mapa abaixo.

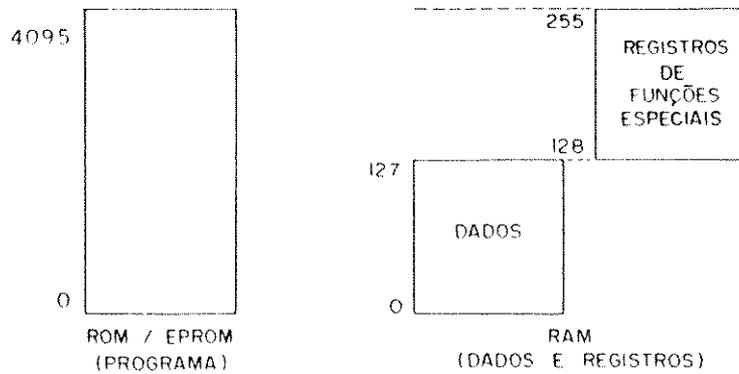


Figura 3.3

Mapa da Memória Interna

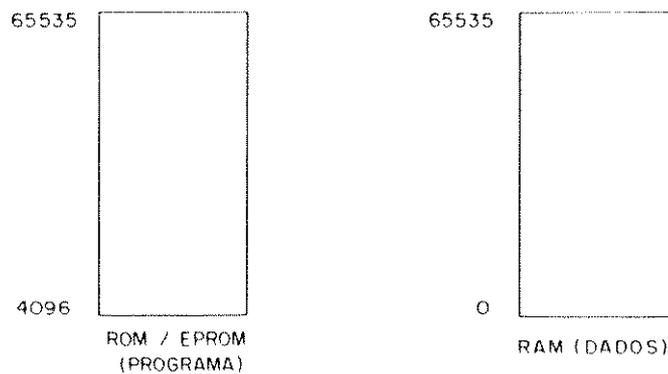


Figura 3.4

Mapa da Memória Externa

Este microcontrolador pode endereçar até 4096 posições de memória de programa interna, e o restante, até completar 64 "Kbytes", em memória externa.

Quanto aos dados, tem-se 256 "bytes" de memória RAM, sendo que os primeiros 128 "bytes" são de memória ao usuário, e os 128 "bytes" restantes são os Registros de Funções Especiais (SFR), de uso do sistema. Pode endereçar até 64 "Kbytes" de memória de dados externa, cujo endereço vai de 0000 até FFFF (hexa).

Os registros de funções especiais incluem locações referentes às portas de I/O, ponteiros do sistema, registros de interrupção, registros da porta serial, dos temporizadores e registros aritméticos. Também pode endereçar "bits" isoladamente tanto na RAM como nos SFR.

A parte "baixa" da memória está dividida em regiões distintas, que

incluem quatro bancos de registradores de 8 "bytes" cada, uma região de 16 "bytes", cujos "bits" são individualmente endereçáveis, e o restante de memória comum, endereçada "byte" a "byte".

.Banco de Registradores

São simples posições de memória RAM, que permitem sua alocação pelo nome dado a cada registro, além do endereçamento pela posição de memória.

A seleção de um certo banco se dá pelo valor dos "bits" RS2 e RS1, conforme tabela a seguir.

7F	Apenas Bytes Endereçáveis
30	
20	BIT e BYTE Endereçáveis
1F	R7 } BANCO 3 R0
18	
17	R7 } BANCO 2 R0
10	
0F	R7 } BANCO 1 R0
08	
07	R7 } BANCO 0 R0
00	

↑
Endereço (RAM interna)

Tabela 3.1

Endereçamento dos Bancos de Registradores

."Bits" Endereçáveis na RAM

Tem-se 16 "bytes" na RAM cujos "bits" são individualmente endereçáveis pela CPU, conforme pode-se observar na próxima figura.

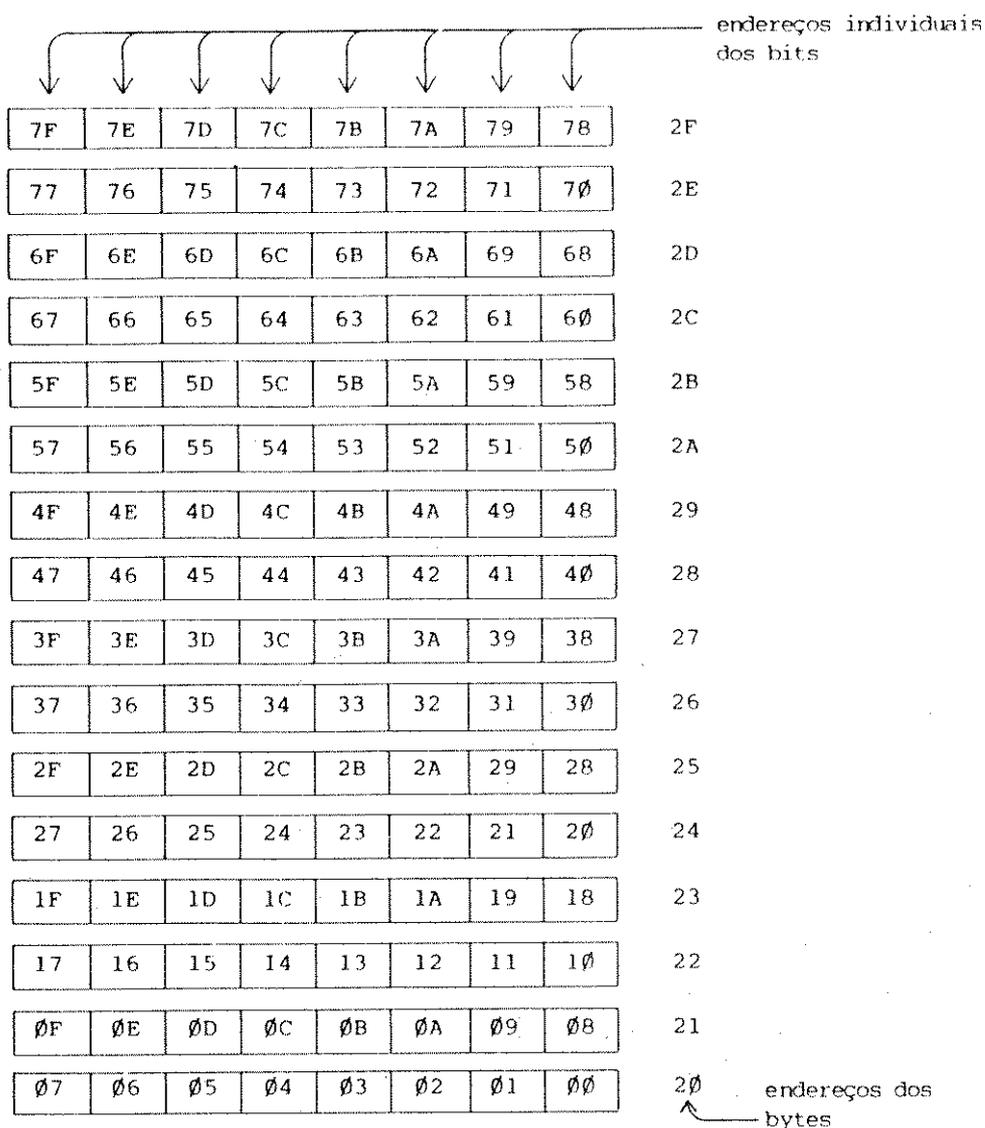


Figura 3.5
"Bits" Endereçáveis na RAM

Existem instruções específicas para o manuseio de "bits", nos quais escreve-se diretamente o endereço do "bit" desejado.

.Registros de Funções Especiais

Têm-se alguns SFR que possuem "bits" endereçáveis, sendo que alguns deles possuem inclusive um nome mnemônico, para facilitar o desenvolvimento de software em compiladores.

Em seguida ao nome de cada registro, tem-se seu endereço em hexadecimal na RAM:

.P0(80), P1(90), P2(A0) e P3(B0)

São posições da RAM que contêm os dados das quatro portas

de I/O do micro, caso as mesmas sejam usadas como tal.

Uma operação de escrita num desses registros altera o conteúdo presente na saída do "chip", e uma operação de leitura dos mesmos coloca o estado presente nos pinos dentro destes registros. Essa operação é possível devido aos "latches" de cada porta, que são ativos na leitura ou escrita destas.

.SP(81)

Corresponde ao "Stack Pointer", que indica o último endereço de armazenamento na pilha.

.TH1(8D), TL1(8B), TH0(8C) e TL0(8A)

Registros de Controle e de Modo de Operação do "Timer/Counter". Nestes registros efetua-se a programação de como eles operarão.

.PCON(87)

Este registro permite adaptar o "chip" para situações em que não há processamento, mas não se deseja perder o conteúdo das memórias internas, como pode ocorrer numa falha de alimentação.

.SCON(98) e SBUF(99)

No SCON, colocam-se os "bits" que permitem trabalhar com a porta de comunicação serial conforme se deseja e o SBUF é o registro do qual a porta armazena o dado recebido ou escreve-se o dado a ser transmitido.

.IEC(A8) e IPC(B8)

Permitem escolher qual ou quais interrupções serão habilitadas (ou desabilitadas) e qual a prioridade (maior ou menor) de cada uma.

.PSW (D0)

O "Program Status Word" (palavra de "status" da CPU) contém "bits" que indicam quais as ocorrências da ALU na última operação lógica e aritmética, além de indicar qual banco de registradores foi acessado pela última vez.

.ACC(E0) e B(F0)

Equivalem ao acumulador e ao registro B.

.DPH(83) e DPL(82)

Estes dois registradores podem ser referenciados como um registro de 16 "bits", chamado DPTR. Usado quando se deseja fazer endereçamento

Indireto para a leitura de constantes armazenadas na memória de programa, para ler ou escrever variáveis na memória externa de dados e desvio para memória de programa com capacidade, até 64 "Kbytes".

Tem-se na figura 3.6, os registros e seus "bits" com seus respectivos endereços.

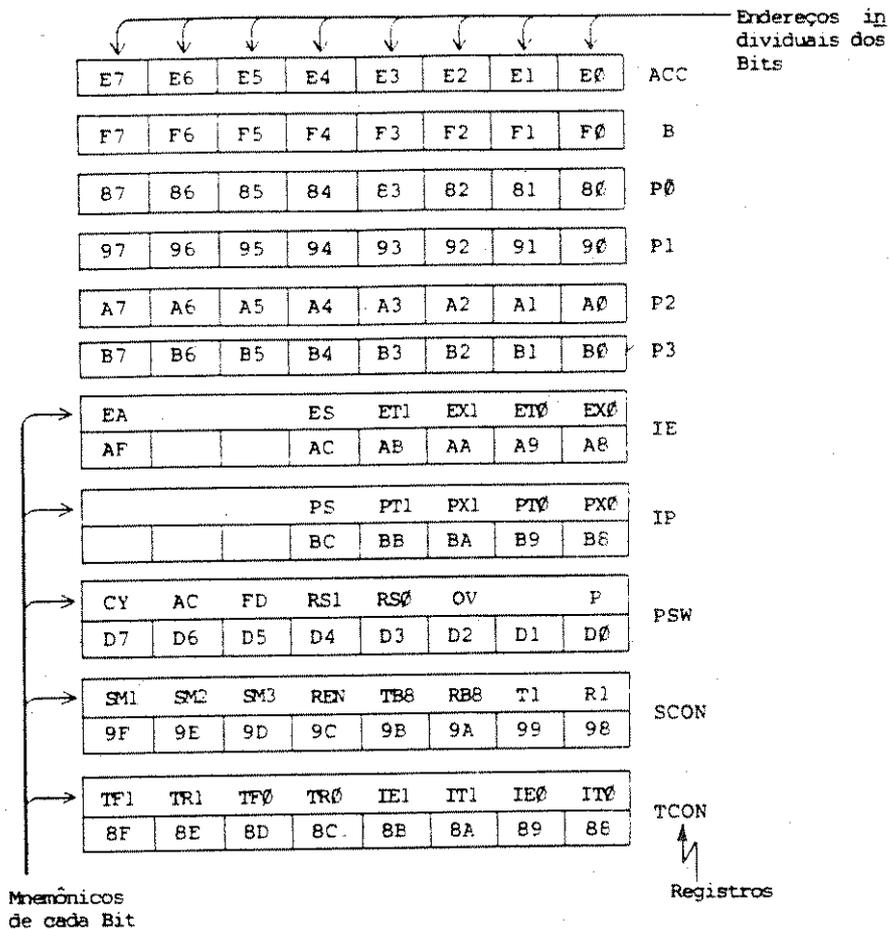


Figura 3.6
Registros de Funções Especiais

3.1.3 Descrição do "Reset"

Embora o "Reset" não seja uma interrupção, muitas vezes é assim chamado devido à semelhança de suas ações, já que o mesmo interrompe o processamento e reinicializa o sistema.

No microcontrolador utilizado, é ativo quando o pino RST permanece em nível alto por dois ou mais ciclos de máquina. Neste caso, a CPU executa um "Reset" interno, que consiste em preencher seus registros internos com valores

pré-determinados ou seja:

.o PC, o Acumulador, o registro B, os Flags e todos os registros dos temporizadores/contadores são zerados;

.no "Stack Pointer" (SP) é colocado o valor 07H;

.as portas P0 a P3 terão o valor FFH;

.o buffer serial (SBUF) estará com o conteúdo indeterminado e o registro de controle da porta serial (SCON) possuirá o valor 00H;

.o registro PCON possuirá o valor binário 0XXXXXXX (onde X pode ser 0 ou 1);

.finalmente, os registros de controle de interrupção (IP e IE) terão ambos valor binário indeterminado.

Quanto ao software gravado na EPROM interna este é afetado pelo "Reset", ao religar o sistema o microcontrolador irá executar o software a partir da sua primeira instrução.

3.1.4 Interrupções

A interrupção é uma das mais importantes ferramentas nos sistemas de controle com microcontrolador, pois é o processo pelo qual a execução de um programa é interrompido (ou não), de acordo com as necessidades de eventos aleatórios externos ou internos.

Existem três fontes de interrupção:

.interrupção por software (instrução);

.interrupção por periférico externo;

.interrupção por ("timer/counter", porta serial).

Uma vez que a CPU atende à interrupção solicitada, o programa que estava sendo executado é interrompido e passa a ser executado a rotina de tratamento da interrupção.

A sua vantagem está na simplificação do hardware e do software, já que não é necessário que o sistema fique monitorando o funcionamento de certos dispositivos periféricos. Muitas vezes as entradas de interrupção (ou a entrada, se for única), são as principais vias de controle do sistema.

.Mascaramento

Sua função é a de impedir, geralmente por software, que uma

certa interrupção seja atendida. Existem sistemas que possuem interrupções não mascaráveis, ou seja, jamais podem ser desabilitadas por software.

.Prioridades

No caso do sistema poder atender mais de uma interrupção, deve-se ter uma tabela de seqüência de prioridades, para que o microcontrolador saiba como agir no caso de duas ou mais interrupções chegarem simultaneamente.

.Interrupção vetorizada e não vetorizada

As interrupções vetorizadas são aquelas que possuem o vetor de interrupção (ou endereço de início da interrupção) pré-estabelecido, e não podem ser mudadas pelo usuário. O vetor já está definido pelo fabricante.

As não vetorizadas são aquelas que os endereços de desvios do programa são escolhidos pelo programador.

.Tipos de ativação das interrupções

Existem três maneiras pelas quais um sistema pode reconhecer um sinal lógico:

- .pelo nível (alto ou baixo);
- .pela borda de subida ou descida;
- .pela soma da borda (subida ou descida) e um nível correspondente.

.Estudo das interrupções no 87C51

O 87C51 pode ser interrompido de cinco maneiras:

- .pela interrupção externa $\overline{\text{INT0}}$;
- .pela interrupção externa $\overline{\text{INT1}}$;
- .pelo "timer/counter" interno TIMER0;
- .pelo "timer/counter" interno TIMER1;
- .pelo canal de comunicação serial.

O canal serial e as interrupções externas usam os seguintes pinos da porta três:

- .Porta 3, pino 0 = receptor de dados seriais;
- .Porta 3, pino 1 = transmissor de dados seriais;
- .Porta 3, pino 2 = entrada de interrupção externa $\overline{\text{INT0}}$;

.Porta 3, pino 3 = entrada de interrupção externa $\overline{\text{INT1}}$;

Dessa forma, se usarmos algumas dessas interrupções, deve-se tratar a porta três apenas como porta de I/O de "bit" endereçável. (ver Seção 3.1.6)

No 87C51 cada interrupção pode ser individualmente habilitada ou não, e pode-se também desabilitar todas de uma única vez. As interrupções podem ter apenas dois níveis de prioridade, e dessa forma, se duas ou mais ocorrem ao mesmo tempo, a de maior prioridade será atendida primeiro.

Este microcontrolador possui um processo de prioridade conhecido como "Nesting", pelo qual uma interrupção de prioridade maior interrompe outra de prioridade menor, mesmo que tenha chegado primeiro. Neste caso, a CPU atende a de maior prioridade e ao fim desta, volta e continua o processamento daquela de menor prioridade.

Se a interrupção que chegar for de nível igual ou menor à que já está sendo atendida, ficará aguardando o fim desta para ser servida. Quando atende uma interrupção, o processador salva no "Stack" o valor do PC, mas não salva o acumulador e o registro "Program Status Word" (PSW). Os mesmos devem ser salvos e devolvidos pela rotina de interrupção, se assim for necessário.

Não só estes registros devem ser armazenados na pilha, mas todos aqueles que forem usados em algum cálculo, ou para armazenar algum valor durante o processamento da rotina de interrupção.

.Endereços de Desvio para cada Interrupção

Cada interrupção é desviada para um certo endereço fixo, endereço este no qual começa a rotina de interrupção. São os seguintes os endereços de desvio:

$\overline{\text{INT0}}$ -> 03H;
.TIMER/COUNTER0 -> 0BH;
 $\overline{\text{INT1}}$ -> 13H;
.TIMER/COUNTER1 -> 1BH;
.Canal Serial -> 23H;

Como foi observado anteriormente, o "Reset" inicia em 00 (hexa) e desta forma, por exemplo, se forem usados todas as interrupções, tem-se que usar os 3 "bytes" disponíveis para o "Reset", num desvio incondicional para outra parte da memória, onde realmente teria início o programa.

As interrupções $\overline{\text{INT0}}$, TIMER/COUNTER0, $\overline{\text{INT1}}$, TIMER/COUNTER1 possuem apenas oito "bytes" disponíveis antes de se sobreporem com a próxima, e desta

forma, provavelmente, deve-se efetuar um desvio para outra parte da memória, sendo a única exceção a interrupção do Canal Serial, que é a última e não sofre problema de sobreposição.

Tem-se na figura abaixo, um mapa parcial da memória de programa com destaque para os endereços de desvio de cada interrupção.

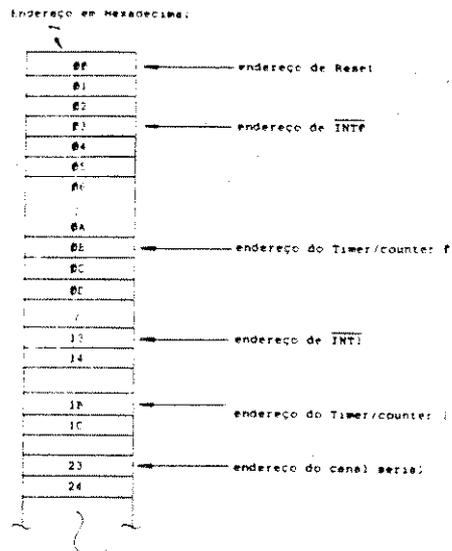


Figura 3.7
Mapa das Interrupções do 87C51

3.1.5 O "Clock" do Microcontrolador

Esta CPU possui internamente um circuito oscilador com um inversor linear de estágio simples.

Têm-se duas possibilidades distintas para o "clock" do sistema, sendo uma externa e outra interna. Para o uso de fonte de "clock" externa, deve-se configurar o pino XTAL1 e injetar o sinal em XTAL2, que dessa forma não atuará no gerador interno, e irá diretamente para o sistema de controle e temporização do "chip". Este sistema pode ser visto na figura 3.8.

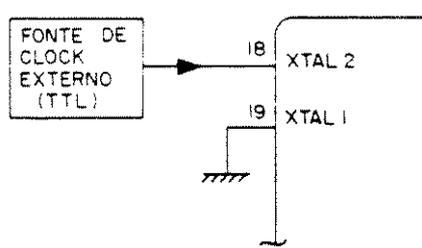


Figura 3.8
"Clock" Externo do 87C51

Para usar o "clock" gerado pelo oscilador interno, deve-se conectar entre os pinos XTAL1 e XTAL2, um cristal ou um filtro cerâmico com a frequência desejada e dois capacitores de realimentação, conforme a figura 3.9.

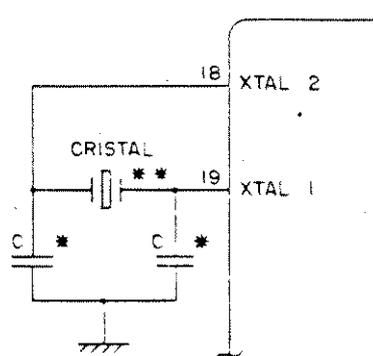


Figura 3.9
"Clock" Interno para o Microcontrolador

3.1.6 Estudo das Portas de I/O

Este microcontrolador possui 32 linhas para I/O, que podem ser individualmente endereçáveis como portas de 8 "bits" cada. Todas as linhas podem ser usadas como I/O, dependendo da operação desejada (leitura ou escrita).

Princípios de Funcionamento e Utilização

O uso das linhas de I/O é simples e necessita apenas que se escreva no pino desejado (ou na porta desejada) o valor a ser enviado para o

mundo externo, ou no caso da leitura, que se preencha os registros internos com o dado presente na porta, no instante de leitura. Se o valor presente numa porta se alterar, esta mudança será sentida somente quando for efetuada uma leitura nesta porta.

Este fato ocorre porque cada linha é na, realidade, um "flip-flop" tipo D, ou seja, um "latch" de um "bit".

Já no caso de leitura, o que se tem é um "buffer" "tri-state", que no instante da leitura deixa passar o "bit" da linha para a via interna, travando-a depois.

Em seguida temos um bloco simples de uma linha de I/O.

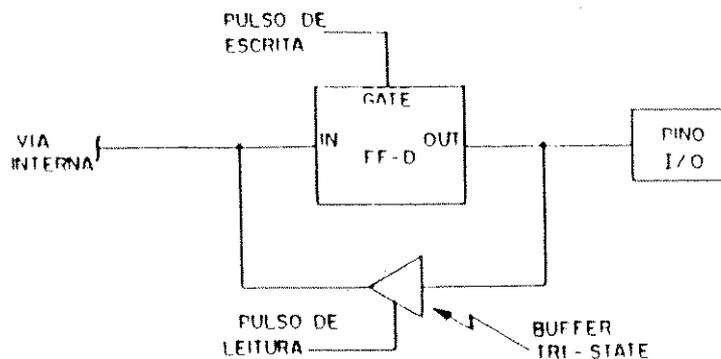


Figura 3.10

Diagrama em Blocos Simplificado de um Pino de I/O

O acesso a uma destas linhas de I/O pode ser efetuado por várias instruções de software.

As portas 1, 2 e 3 possuem "pull-ups" internos, ou seja resistores ligados à alimentação. Esta característica faz com que seus pinos estejam sempre num estado definido. A porta zero não possui esta característica, e quando configurada para entrada, terá seu nível flutuando na ausência de nível fixo de entrada.

Uso das Portas de I/O para a expansão de Memória de Dados e Programa

As portas zero e dois possuem como função alternativa endereçar e acessar memória externa, para que este "chip" possa trabalhar com a faixa de memória que necessita.

A porta dois fornece a parte mais significativa da via de endereços e a porta zero fornece multiplexados no tempo, a parte menos significativa do

endereço e os dados. O pino 3.0 fornece o sinal ALE, que habilita o "latch" de endereços para manter a parte menos significativa do endereço.

Os outros sinais de controle existentes são o \overline{RD} , \overline{WR} e \overline{PSEN} . Este último habilita a saída da memória de programa de forma que é ativo quando a CPU deseja ler programa de memória externa. Neste instante, os sinais \overline{RD} e \overline{WR} estão em nível lógico "1". Quando a CPU deseja ler dados, os sinais \overline{PSEN} e \overline{WR} vão a nível "1" e o sinal \overline{RD} vai a nível "0", e para a escrita de dados, o sinal \overline{WR} vai a zero e \overline{PSEN} e \overline{RD} vão a "1".

A seguir, temos um diagrama simplificado de como é um sistema com expansão de memória de dados e de programa, com memórias de dados e programa de 32 "Kbytes" cada.

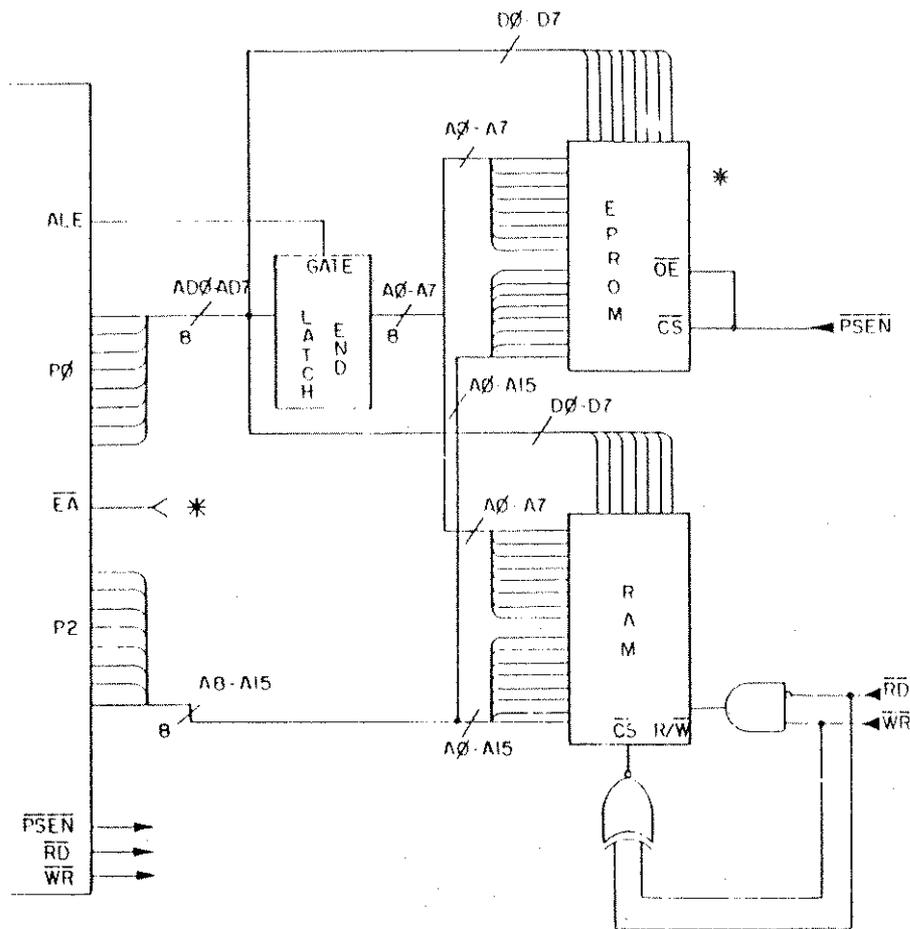


Figura 3.11

Expansão de Memória de Dados e Programa

3.2 Descrição do Hardware com Microcontrolador

Este sistema foi desenvolvido com um microcontrolador da família 8051 e possui uma estrutura de hardware mais compacta que o outro cartão digital. Como resultado, há um aumento na parte de programação. Para entendermos a estrutura dos algoritmos, começaremos apresentando e definindo os módulos físicos do sistema.

3.2.1 Características Gerais

São as seguintes as funções executadas pelo sistema:

.o módulo de posição é formado por dois contadores de 16 "bits", permitindo uma contagem até o valor 65535. Cada um deles conta os pulsos vindos do Codificador, dependendo do sentido de giro do motor. O valor armazenado é transferido ao microcontrolador, para posterior processamento do deslocamento efetuado pela mesa da máquina.

.o módulo de velocidade é composto por dois contadores, um conta os pulsos do Codificador Incremental e o outro totaliza os pulsos vindos de uma base de tempo (contador da base de tempo). A partir dos valores contidos nesses contadores, se processa o cálculo da velocidade angular do eixo do motor.

.o módulo do conversor D/A converte uma palavra digital, fornecida pelo computador, em um valor de tensão analógica que será fornecida ao "driver" de potência da máquina. Compõe-se de 10 "bits" e cada "step" de tensão é calculado como sendo $(1/1024) * (\text{tensão de fundo de escala})$, para modo unipolar.

Os dois primeiros módulos foram confeccionados em uma placa, sendo o último módulo (conversor D/A) montado em uma placa separada. Todo o conjunto possui três fontes de tensão:

.+5V para os CIs TTL e o microcontrolador

.+12V/-12V para o módulo de comunicação e conversor D/A

3.2.2 Diagrama de Blocos

Abaixo temos a representação do conjunto através de um diagrama de blocos.

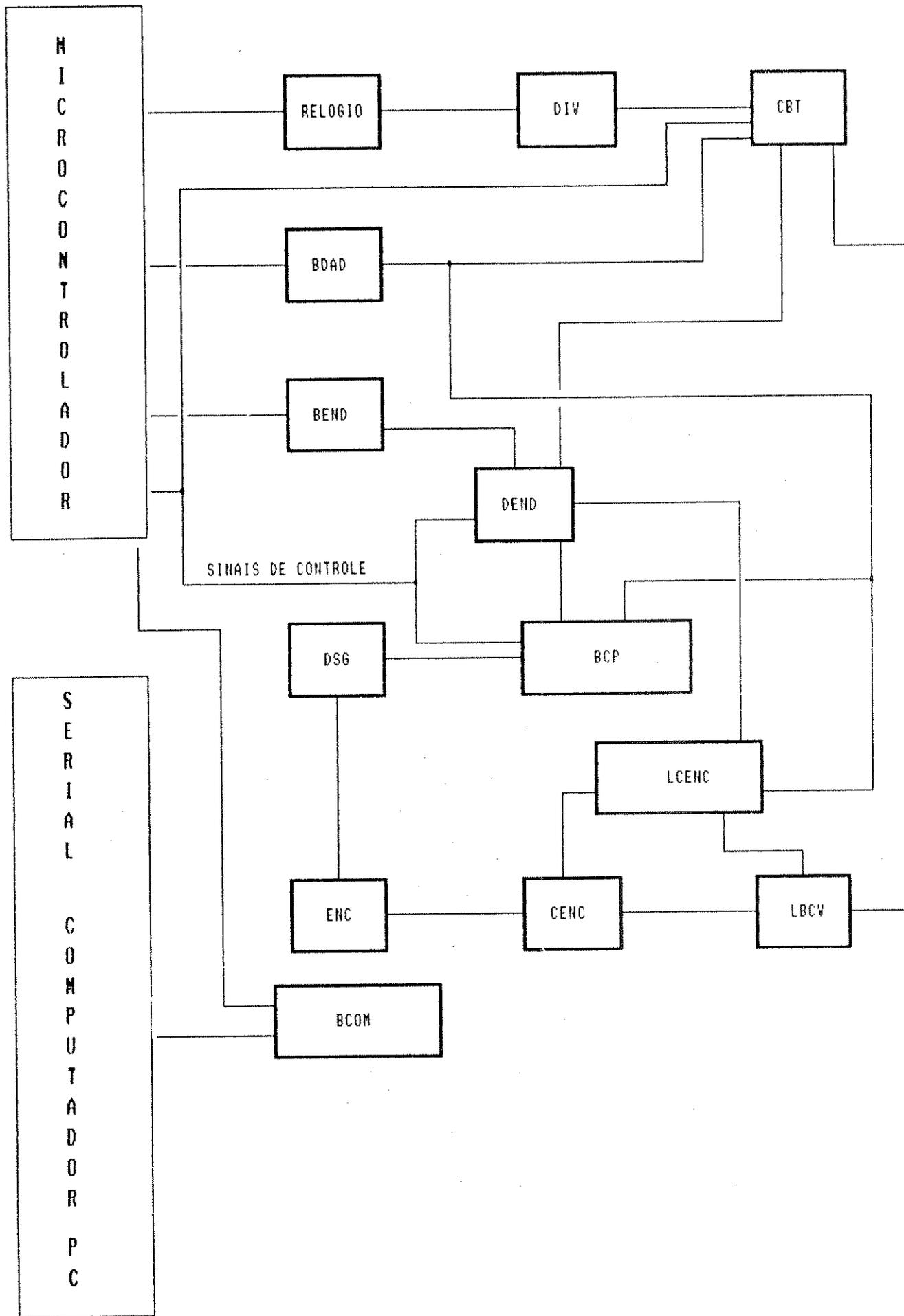


Figura 3.12
Diagrama de Blocos do Sistema

3.2.3 Descrição dos Blocos

.Relógio

É um circuito TTL que fornece na sua saída um "clock" da ordem da frequência do cristal. É usado pelo microcontrolador para temporização de suas operações. Este sinal, após ser dividido por dois, é usado no contador da base de tempo.

."Buffer" de Dados/Endereços (BDAD/BEND)

Pelo fato do microcontrolador poder alimentar poucas cargas TTL, utilizam-se "buffers" nas suas linhas de dados e endereços. Sua operação é transparente ao sistema, servindo apenas para aumentar a capacidade de corrente destas linhas.

.ENC (Codificador Incremental de Posição Angular)

Fornece os sinais que serão usados nas entradas dos contadores de posição e velocidade. Possui ainda uma saída que indica quando o eixo do motor executou uma volta.

.Decodificador de Endereços (DEND)

Ele decodifica os endereços de I/O, permitindo que um periférico seja acessado para uma leitura ou escrita, pela CPU.

.Decodificador de Sentido de Giro (DSG)

Sua lógica ativa apenas um dos contadores de posição, a cada pulso gerado do Codificador, ou seja, se o motor estiver girando no sentido horário o contador ligado a esta linha estará contando, caso contrário estará o outro.

.Banco de Contadores da Posição (BCP)

Formado por dois contadores de 16 "bits". Cada um deles registra o deslocamento angular efetuado pelo eixo do motor, em um sentido ou outro. Este valor é armazenado em registros que podem ser acessados a qualquer instante.

. Banco de Contadores da Velocidade

Compõe-se de dois contadores:

.o contador do Codificador (CENC) que processa os pulsos vindos do Codificador deste elemento. Pode ser programado no modo "up" ou "down".

. o contador da base de tempo (CBT), que registra os pulsos

vindos do divisor de frequência, do relógio. Funciona como um contador "down" de 16 "bits".

. "Latch" do contador do Codificador (LCENC)

Armazena o valor contido no contador do Codificador e depois o transfere para o microcontrolador, através de uma operação de leitura de I/O.

.Lógica dos Bancos de Contadores da Velocidade (LBCV)

Executa as seguintes tarefas:

.sincronização entre os contadores (contador do Codificador e da base de tempo);

.controle de suas funções;

.requisição de pedido de interrupção ao microcontrolador;

.CPU

É a responsável pelo controle de todo o hardware que compõe esta estrutura . As suas entradas para interrupção (ver Seção 3.1.4), permitem o desenvolvimento de um "firmware" (software básico) simples e compacto. Possui um sistema de "Reset" (ver Seção 3.1.3), que reinicializa o sistema. Também é através de algumas de suas linhas que ocorre a comunicação (ver Seção 3.4), entre o microcomputador e o cartão digital que contém este componente.

.Banco de conversores D/A

Geram uma tensão de referência para o amplificador de potência.

.Bloco de Comunicação

É o responsável pela comunicação entre o computador e o microcontrolador.

3.2.4 Endereçamento

Neste sistema, temos alguns periféricos descritos abaixo:

. "Latch" do contador do Codificador;

.Registrador de Modo de Operação;

.Contadores de Posição/Contador da Base de Tempo;

Todos os endereços gerados são mapeados em I/O, ao invés de memória. Na

tabela seguinte encontram-se os endereços dos periféricos.

Módulo	Offset	Operação L/E	Função
RCP/RCBT	00	L/E	REG. CONT0
	01	L/E	REG. CONT1
	02	L/E	REG. BASTEMP
	03	L/E	REG. CONTR
RMO	04	E	REG. OPER.
LCE1	08	E	ESCREVE VALOR NO CE
LCE2	0C	L	LÊ VALOR DO CE

Tabela 3.2

Mapeamento dos Endereços dos Circuitos de I/O

onde:

- .CP/CBT -> contadores de posição e base de tempo;
- .CE -> contador do Codificador;
- .RMO -> registro de modo de operação;
- .LCE1 -> "latch" um do contador do Codificador;
- .LCE2 -> "latch" dois do contador do Codificador;

Os conversores D/A não são especificados na tabela acima, por terem sido montados em uma placa independente. Esta placa é encaixada no "slot" do PC, e o endereço do primeiro conversor é (202)H e do segundo (203)H, conforme tabela 3.3.

bit de endereço	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
valor do bit	1	0	0	0	0	0	0	0	1	0
	1	0	0	0	0	0	0	0	1	1

Tabela 3.3
Endereço dos Conversores D/A

3.2.5 Modos de Operação

Neste item será descrito como o módulo de velocidade opera.

Quando o "driver" alimenta o motor, com uma tensão analógica, este começa a girar com uma velocidade proporcional a esta tensão. Girando concentricamente ao eixo do motor está o Codificador que fornece a cada deslocamento angular do eixo um pulso que é recebido pelo contador do Codificador. Durante cada operação de cálculo da velocidade angular do eixo do motor, podemos ter três situações:

.a janela aberta, para amostragem da velocidade, é pequena e conseqüentemente não se consegue medir o valor da velocidade angular do eixo do motor. O sistema se auto-reinicializa e abre uma nova janela para fazer a medida.

.o tamanho da janela é coerente com a velocidade angular do eixo do motor. Os valores armazenados, nos registros dos contadores, são transferidos para o microcontrolador e o processo recomeça. Os dados da velocidade vão para o microcomputador, onde se processa o cálculo da mesma.

.o tamanho da janela não é suficiente mas pode ainda ser aumentado. Esta operação é executada pelo sistema e novamente tenta-se medir este valor. Se o tempo de exposição não for ainda suficiente esta amostra é perdida e parte-se para uma nova janela; caso contrário, obtém-se o valor desejado da velocidade.

3.3 Software da Unidade Digital com Microcontrolador

Por ser um sistema desenvolvido em torno de um microcontrolador utilizou-se a sua própria linguagem ("assembly") para a codificação dos algoritmos que envolvem os processos de velocidade e posição. Se olharmos, de forma global para o software, perceberemos que o mesmo apresenta três unidades básicas que são:

- .inicialização;
- .módulo de posição;
- .módulo de velocidade.

Na inicialização, o hardware é configurado a fim de executar determinadas tarefas que são necessárias nos módulos de velocidade e posição. As principais funções implementadas são:

- .programação dos periféricos;
- .inicialização das funções do microcontrolador.

No módulo de posição obtêm-se os dados que serão usados para cálculo do deslocamento linear da mesa da máquina. A nível de tarefas implementadas temos:

- .leitura dos registros dos contadores de posição;
- .escrita de valores nestes contadores.

O último módulo executa o processo da velocidade. Nele temos implementado:

- .controle deste processo;
- .reprogramação de registros e "latches" de contadores;
- .leitura de dados;
- .finalização/reinício de uma nova amostragem.

Toda a codificação dos algoritmos foi implementada em linguagem de máquina ("assembly" do 87C51), o que tornou o software mais compacto em termos de memória de programa e de dados.

Começaremos apresentando agora os algoritmos que compõem cada um destes módulos e que será a base para a codificação do software.

3.3.1 Módulo de Inicialização da Unidade com Microcontrolador

Neste módulo são configurados e inicializados os registros de I/O.

Algoritmo

- .programar "timer/counter" da CPU
- .programar as interrupções
- .programar o nível de prioridade
- .programar o canal serial
- .zerar a memória interna
- .programar os contadores de posição
- .programar o contador da base de tempo
- .programar o contador do Codificador
- .fazer verificação do status destes contadores, se for incorreta reprogramá-los
- .carregar valor inicial nos mesmos
- .programar o RMO para os processos terem início

3.3.2 Módulo de Posição da Unidade com Microcontrolador

Executa o processo que tem a função de determinar o deslocamento linear efetuado pela mesa da máquina.

Algoritmo

```
.o usuário (computador) solicitou o envio dos dados armazenados nos
registros dos contadores de posição?
    então
        .lê os valores dos registros
        .guarda-os em unidades de armazenamento
        .reinicializa estes contadores
    senão
        .fica esperando o pedido
.microcontrolador faz um pedido para transmissão
.envia os dados para o usuário, após o pedido ter sido aceito
```

3.3.3 Módulo de Velocidade da Unidade com Microcontrolador

Gerencia o processo de velocidade.

Algoritmo

```
.começou a janela de amostragem?  
    então  
        .lê valor do registro de velocidade (contador do Codificador)  
        .armazena-o numa posição de memória  
        .reprograma o contador do Codificador para modo "DOWN"  
    senão  
        .espera o início da janela  
.tamanho da janela é compatível com a velocidade angular do eixo do motor?  
    então  
        .lê o registro do contador da base de tempo  
        .guarda-o em uma posição de memória subsequente  
        .reinicializa os contadores da base de tempo e do Codificador  
        .reprograma o contador do Codificador  
        .recomeça o processo  
        .transmite estes dados ao usuário (computador) quando este os solicitar  
    senão  
        .pode-se aumentar o tamanho da janela?  
            então  
                .executa-se esta tarefa  
                .houve um estouro no tempo total do processo?  
                    então  
                        .reinicializa os contadores da base de tempo e do Codificador  
                        .reprograma o contador do Codificador  
                        .recomeça o processo  
                    senão  
                        .lê o registro da base de tempo  
                        .guarda-o em posição de memória  
                        .reinicializa o sistema para uma nova operação  
                        .transmite os dados ao usuário, após
```

pedido de solicitação

senão

.reinicializa os contadores

.processo recomeça

3.4 Sistema de Comunicação

Quando há troca de informações entre o computador e um cartão digital esta pode ser efetuada basicamente de duas maneiras:

comunicação paralela

comunicação serial

Na paralela, envia-se um caracter de cada vez, por intermédio de um cabo que une os dois sistemas. É implementada quando a distância de transmissão é pequena.

Na serial cada "bit" é transmitido de forma sequencial em uma taxa que depende do hardware.

3.4.1 Modo Assíncrono de Comunicação

Neste modo de comunicação, não existe a necessidade de gerar-se um sinal de sincronismo. Aqui cada caractere é transmitido individualmente, e para cada um, temos "bits" de partida ou início de transmissão ("start bit") e "bits" de fim de transmissão ("stop bit").

Na figura seguinte temos um esquema de transmissão.

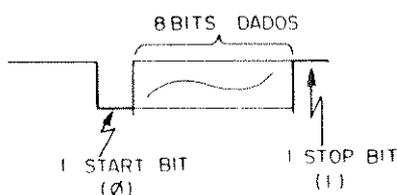


Figura 3.13

Sistema de Transmissão Assíncrono

O "start bit" é reconhecido pela transição do nível presente na linha de "1" para "0". Neste instante, o "clock" interno do sistema efetua uma varredura da linha de tempos em tempos para detectar o nível na mesma, nível esse que será associado a cada "bit" de forma conveniente. Ao reconhecer o oitavo "bit", o sistema fica esperando o "stop bit", que é a transição de "0" para "1", ou a permanência em nível 1, se já estava neste estado.

Neste ponto, o sistema entra em repouso e fica na espera de um novo "start bit", para iniciar a recepção de um novo caracter.

3.4.2 A Interface Serial do Microcontrolador

O sistema pode receber e transmitir informações, sendo estas operações controladas por um registro especial, o SBUF ("serial buffer" - "buffer" serial). Uma escrita no mesmo implica em automática transmissão do dado escrito, assim como uma certa informação que chegue no seu pino de recepção, implicará na automática operação de recepção por parte do sistema, independentemente do controle do usuário (desde que o canal serial esteja habilitado).

Fisicamente tem-se dois registros com o mesmo nome SBUF, sendo que um é para recepção e o outro para transmissão. O reconhecimento é feito pelo sistema através das instruções que acessarão o mesmo. Se for instrução de escrita, o registro de transmissão será alterado, e se for instrução de leitura, o dado será captado no registro de recepção.

A transmissão inicia-se assim que um dado é escrito no registro SBUF. A recepção começa quando o "bit" REN do registro SCON estiver setado e detectar-se o "start bit".

3.4.3 Modo de Operação e Formato de Dados

Tem-se quatro possíveis modos de operação, chamados modos 0, 1, 2, 3. Descreveremos o modo 1, o qual foi implementado neste cartão digital.

Nele o pino de recepção é o RXD e o de transmissão é o TXD. Em cada pacote são transmitidos (recebidos 10 "bits"), sendo um "start bit" (nível 0), seguido de 8 "bits" de dados e o "stop bit" (nível 1). Neste modo a taxa de transmissão é variável. Temos um exemplo abaixo do formato dos dados neste modo.

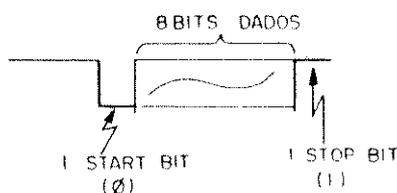


Figura 3.14

Formato dos Dados no Modo 1

3.4.4 Taxas de Transmissão

Para o modo 1, a taxa de transmissão é fornecida pelo "timer/counter1", onde cada ordem da mesma é gerada ao ocorrer o "overflow" deste contador. Neste

caso, uma fórmula para cálculo desta taxa é:

$$\text{taxa} = \frac{2^{\text{SMOD}}}{32} * (\text{taxa de overflow do timer}_1) \quad (1)$$

O "timer/counter" é usado como um temporizador recarregável automaticamente, e neste caso tem-se a seguinte fórmula:

$$\text{taxa} = \frac{2^{\text{SMOD}}}{32} * \frac{f_{\text{clock}}}{12 * (256 - \text{TH}_1)} \quad (2)$$

onde:

f_{clock} é a frequência do relógio do microcontrolador

TH_1 é a parte mais significativa do registro do contador um

3.4.5 Processo de Comunicação

Este será implementado entre o microcontrolador 87C51 e um computador da linha IBM PC-XT/AT. O computador assumirá a função de mestre, ou seja, ele será o responsável pelo gerenciamento da troca de informações. Também o pedido de envio de informações sempre será de sua iniciativa. Quando estas forem solicitadas, as mesmas já devem estar no "buffer" de transmissão do microcontrolador para serem transmitidas.

Antes que esta operação seja efetuada, o mestre efetuará uma pergunta e aguardará um dos caracteres de reconhecimento positivo ou negativo (ACK ou NACK, respectivamente).

A cada mensagem que o microcontrolador enviar para o computador este solicitará uma confirmação de recebida ou não.

Basicamente o processo de comunicação seguirá as seguintes regras:

o mestre faz uma transmissão, pedindo um dado ou checando se a linha está funcionando corretamente. O escravo terá de responder uma cadeia válida pois se isto não ocorrer o mestre terá condições de detectar falha no sistema de comunicação, ou problemas no hardware do microcontrolador.

haverá um "time-out" que determinará o tempo de resposta. Se a resposta não ocorrer dentro deste tempo, será indicada uma queda de comunicação.

Todo o processo será paralisado, caso as tentativas de recuperação da transmissão não tenham sucesso.

Abaixo, temos uma figura que mostra estas regras do processo de transmissão.

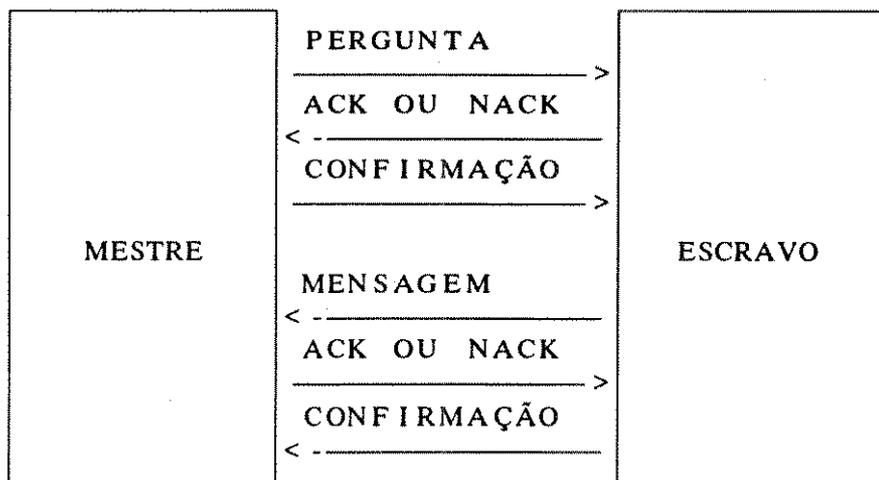


Figura 3.15
Processo de Comunicação

4. MODELAGEM

4.1 RESUMO

Na primeira etapa serão descritos alguns tópicos da Teoria de Controle, e estes conceitos nos fornecerão o embasamento teórico necessário para análise das estruturas que compõem o sistema. Na outra fase mostraremos como obter as funções de transferência e uma representação em diagrama de blocos dos vários módulos que formam o conjunto, usando os conceitos apresentados na parte teórica. Todo este desenvolvimento matemático visa fornecer informações para o futuro módulo de controle.

4.2 CONSIDERAÇÕES TEÓRICAS

4.2.1 Método da Resposta em Frequência

O termo "resposta em frequência" significa a resposta em regime estacionário de um sistema com entrada senoidal. Varia-se a frequência do sinal de entrada de um certo intervalo e estuda-se a resposta em frequência do sistema considerado.

Para um sistema linear, invariante no tempo, com entrada $x(t)$ senoidal, saída $y(t)$ e função de transferência $G(s)$, temos a equação de saída:

$$y(t) = X |G(j\omega)| \sin(\omega t + \varphi) \quad (1)$$

ou

$$y(t) = Y \sin(\omega t + \varphi) \quad (2)$$

com $Y = X |G(j\omega)|$

Logo verifica-se que um sistema linear, estável, invariante no tempo, e sujeito a uma entrada senoidal possui, em regime estacionário, uma saída senoidal com a mesma frequência da entrada. Porém a amplitude e o ângulo de fase da saída, em geral, serão diferentes daqueles da entrada. De fato, a amplitude da saída é dada pelo produto da amplitude da entrada por $|G(j\omega)|$, enquanto o ângulo de fase da saída difere daquele da entrada de um valor $\varphi = \text{ângulo}(G(j\omega))$.

Para entradas senoidais temos:

$$|G(j\omega)| = \frac{|Y(j\omega)|}{|X(j\omega)|} \quad (3)$$

A expressão acima representa a relação de amplitude entre a saída senoidal e a entrada senoidal.

$$\angle G(j\omega) = \frac{\angle Y(j\omega)}{\angle X(j\omega)} \quad (4)$$

Já a equação (4) é a defasagem da saída senoidal em relação à entrada senoidal.

A função de transferência senoidal $G(j\omega)$, a relação entre $Y(j\omega)$ e $X(j\omega)$, são funções complexas e podem ser representadas pelo módulo e ângulo de fase, tendo a frequência como variável ou parâmetro (um ângulo de fase negativo é denominado atraso de fase e um positivo é chamado avanço de fase). A função de transferência senoidal de qualquer sistema linear é obtida substituindo-se (s) por $(j\omega)$ na função de transferência do sistema. Para caracterizar completamente um sistema linear no domínio da frequência, deve-se especificar tanto a relação de amplitude como o ângulo de fase como função da frequência. Há três representações comumente utilizadas de funções de transferência senoidais. Elas são:

- .gráfico logarítmico ou diagrama de Bode
- .gráfico polar
- .gráfico do log-módulo versus fase

Uma função de transferência senoidal pode ser representada por dois gráficos separados, um fornecendo o módulo versus frequência e o outro o ângulo de fase versus frequência. Um gráfico logarítmico ou diagrama de Bode consiste neste dois gráficos, plotados em escala logarítmica.

A representação logarítmica é útil pelo fato de que mostra tanto as características de baixa frequência como aquelas de alta frequência, para a função de transferência considerada, em um único diagrama. A expansão da faixa de baixa frequência utilizando uma escala logarítmica para a frequência é muito vantajosa desde que as características de baixa frequência são mais importantes em sistemas práticos.

Um exemplo destas curvas são mostradas na figura 4.1.

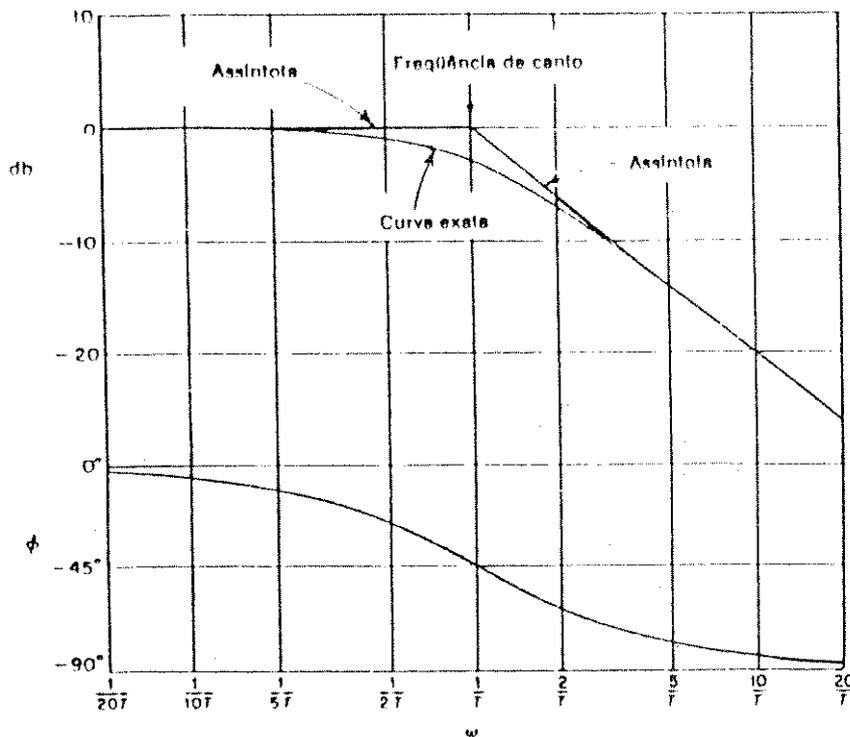


Figura 4.1

Curva do Log-Módulo conjuntamente com a Curva do Ângulo de Fase

Assim a determinação experimental de uma função de transferência pode ser realizada de modo simples se os dados de resposta em frequência estão representados na forma de um gráfico logarítmico.

4.2.2 Acelerômetro Piezoelétrico

Frequentemente, é necessário caracterizar a performance de um sistema mecânico em termos de massa e aceleração. Isto é particularmente verdadeiro em situações onde é impossível determinar a natureza do mecanismo do sistema ou para medir as forças aplicadas por partes constituintes do sistema.

Em aplicações onde as acelerações dinâmicas são medidas, o acelerômetro piezoelétrico tem sido usado porque não necessita de fontes de potência externa.

Um acelerômetro piezoelétrico para testes de vibrações é mostrado na figura seguinte.

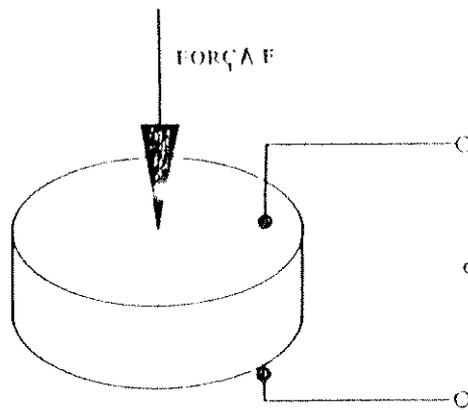


Figura 4.2
Um Típico Acelerômetro Piezoelétrico

Ele consiste de um material cristalino que gera uma carga q , quando uma força é aplicada sobre o mesmo. Através da incorporação de uma massa em contato direto com o cristal, um transdutor de aceleração é produzido. Quando uma variação de aceleração é aplicada ao conjunto cristal-massa, o cristal experimenta uma variação de força ($F=Ma$), causando uma variação de carga elétrica, ou seja:

$$a=dF=d(Ma) \quad (5)$$

onde d é a constante piezoelétrica.

Um acelerômetro piezoelétrico é um transdutor que converte força dinâmica (vibração) em energia elétrica. Na próxima figura temos um exemplo do que ocorre eletricamente e mecanicamente com este tipo de transdutor, quando uma força é aplicada sobre ele.

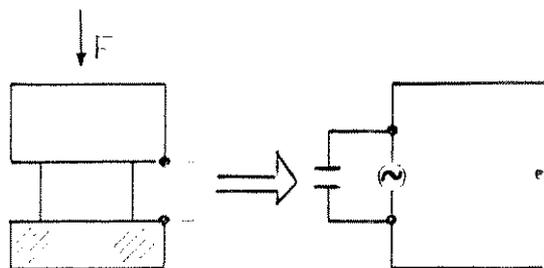


Figura 4.3
Acelerômetro Típico e Circuito Equivalente

Uma propriedade dos materiais piezolétricos é gerar cargas elétricas (q) quando uma força aparece sobre o mesmo. Uma carga gerada desta maneira é uma função da constante piezolétrica (d) e da força aplicada ($q=dF$). No caso de um acelerômetro, a força (F) é o produto da massa (M) e da aceleração aplicada (a), onde $F=Ma$. Assim, chegamos a

$$q=d(Ma) \quad (6)$$

Sabemos que aumentando a aceleração, aumenta-se a força de pressão e mais cargas serão geradas aumentando-se a tensão de saída nos seus terminais. Logo a tensão gerada nos terminais do transdutor é proporcional à aceleração.

4.2.3 Estudo da Função de Transferência do Motor DC

Um motor DC é empregado em um sistema de controle quando é exigida uma quantidade apreciável de potência no eixo.

São controlados por armaduras com campo fixo ou controlados por campo com corrente de armadura fixa. Por exemplo, em sistemas de servos emprega-se um campo por ímã permanente, e o sinal de controle é aplicado aos terminais de armadura. Um exemplo deste tipo de motor é mostrado em seguida.

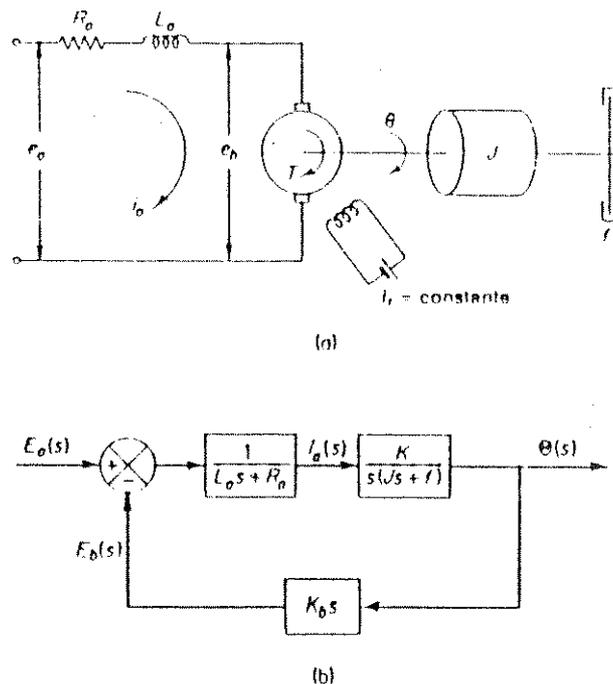


Figura 4.4

Diagrama Esquemático de um Motor DC Controlado por Armadura

Neste sistema temos:

R_a = resistência do enrolamento da armadura, (Ω)

L_a = indutância do enrolamento de armadura, (H)

I_a = corrente de armadura do enrolamento, (A)

I_f = corrente de campo, (A)

E_a = tensão aplicada na armadura, (V)

E_g = força contra-eletromotriz, (V)

K_e = constante elétrica do motor, (V/Krpm)

K_t = constante de torque do motor, (V/rpm)

ω = velocidade angular do eixo do motor, (rpm)

θ = deslocamento angular do eixo do motor, (rad)

T_m = torque fornecido pelo motor, (Nm)

J = momento de inércia equivalente do motor e da carga, referida ao eixo do motor, (Kgm^2)

f = coeficiente de atrito-viscoso equivalente do motor e da carga, referida ao eixo do motor, (Kgm/rad/s)

4.2.3.1 Equações Matemáticas do Circuito Equivalente do Motor

O circuito elétrico equivalente do motor é apresentado abaixo.

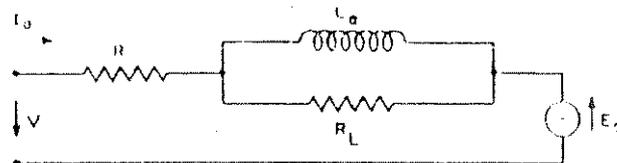


Figura 4.5

Circuito Equivalente do Motor (Armadura)

A explicação física deste modelo é que R_L representa as perdas no circuito magnético. Como R_L é usualmente muito maior que R_a , o efeito de R_L na operação do motor é insignificante.

Logo é possível ignorar esta resistência em aplicações práticas de servomotores, e aproximar o circuito equivalente do motor por R_a , L_a e E_g . Seja E_a a tensão aplicada na armadura do motor e I_a a sua corrente. A relação entre estas variáveis é dada por:

$$E_a = \frac{L_a d(I_a)}{dt} + R_a I_a + E_g \quad (7)$$

onde E_g é proporcional a velocidade do motor, ω .

$$E_g = K_e \omega \quad (8)$$

As equações acima podem ser combinadas de forma a obtermos:

$$E_a = \frac{L_a d(I_a)}{dt} + R_a I_a + K_e \omega \quad (9)$$

Esta equação é conhecida como equação elétrica do motor DC.

Sendo o campo fixo, a corrente produzida gera um torque proporcional da forma:

$$T_g = K_t I_a \quad (10)$$

Será denotado o momento de inércia do motor por J_m (Kgm^2) e T_f (Nm), representa o torque de fricção constante no motor. Também serão denotados todos os torques de atrito viscoso e outros torques que são proporcionais a velocidade por $D\omega$. Então, o torque oposto no motor, T_m é dado por:

$$T_m = T_f + D\omega \quad (11)$$

Devido ao acoplamento entre o motor e a carga temos um momento de inércia denotado por J_1 (Kgm^2) e um torque oposto de carga T_1 (Nm). A relação entre os torques e velocidade apresenta-se na forma:

$$T_g = (J_m + J_1) \frac{d\omega}{dt} + D\omega + T_f + T_1 \quad (12)$$

Esta equação é conhecida como equação dinâmica do motor.

As equações (9), (10) e (12) fornecem um modelo para o motor que

descreve a relação entre suas variáveis. Quando o mesmo é usado como um componente em um sistema, é desejável descrevê-lo pela função de transferência entre a voltagem do motor e sua velocidade. Para esta proposta assume-se $T_i = 0$ e $T_f = 0$, desde que nenhum deles afeta a função de transferência. Se agora aplicar-se Laplace as equações do motor, obtém-se:

$$Ea(s) = (sLa + Ra) Ia(s) + K_e \omega(s) \quad (13)$$

$$Tg(s) = K_t Ia(s) \quad (14)$$

$$Tg(s) = (Jm + J_l) s\omega(s) + D\omega(s) \quad (15)$$

Sendo o momento total de inércia, $J(Nm)$, dado por

$$J = Jm + J_l \quad (16)$$

Se agora combinar-se (14) e (15), chega-se a expressão para a corrente:

$$Ia(s) = \frac{1}{K_t} (sJ + D) \omega(s) \quad (17)$$

Substituindo a equação (17) em (13) tem-se:

$$Ea(s) = \frac{1}{K_t} (sLa + Ra) (sJ + D) \omega(s) + K_e \omega(s) \quad (18)$$

e a correspondente função de transferência do motor é:

$$Gm(s) = \frac{\omega(s)}{Ea(s)} = \frac{k_t}{(sLa + Ra) (sJ + D) + k_e K_t} \quad (19)$$

A função de transferência tem dois pólos, os quais são negativos e reais, nos casos práticos. Pode-se então escrevê-la da forma:

$$Gm(s) = \frac{K_t}{JLa (s-p_1) (s-p_2)} \quad (20)$$

onde p_1 e p_2 são os pólos da função de transferência. Os pólos são raízes da equação característica:

$$s^2 LaJ + s(LaD + RaJ) + RaD + K_e K_t = 0 \quad (21)$$

Sendo o fator $D = 0$, chega-se à relação:

$$G_m(s) = \frac{\omega(s)}{E_a(s)} = \frac{K_t}{s^2 LaJ + s RaJ + K_e K_t} \quad (22)$$

Os pólos, neste caso, são raízes da equação característica:

$$s^2 LaJ + s RaJ + K_e K_t = 0 \quad (23)$$

onde:

$$p_1 = \frac{-RaJ + \sqrt{(RaJ)^2 - 4LaJK_e K_t}}{2LaJ} \quad (24)$$

$$p_2 = \frac{-RaJ - \sqrt{(RaJ)^2 - 4LaJK_e K_t}}{2LaJ} \quad (25)$$

Como a indutância La na prática é pequena, pode-se usar a aproximação:

$$\sqrt{1 - x} \cong 1 - \frac{x}{2} \quad (26)$$

o que é válido para valores pequenos de x .

A aproximação nos permite chegar a:

$$\begin{aligned} \sqrt{(RaJ)^2 - 4LaJK_e K_t} &= RaJ \sqrt{1 - \frac{4LaK_e K_t}{(RaJ)^2}} \\ &= RaJ \left(1 - \frac{2LaK_e K_t}{(RaJ)^2} \right) \quad (27) \end{aligned}$$

Quando a equação (27) é substituída nas equações (24) e (25), obtêm-se as seguintes relações:

$$P_1 = \frac{-RaJ + RaJ \left(1 - \frac{2LaK_e K_t}{(RaJ)^2} \right)}{2LaJ}$$

$$P_1 = -\frac{K_e K_t}{RaJ} \quad (28)$$

$$P_2 = \frac{-RaJ - RaJ \left(1 - \frac{2LaK_e K_t}{(RaJ)^2} \right)}{2LaJ}$$

$$P_2 = -\frac{Ra}{La} \quad (29)$$

Considerando as equações (28) e (29), pode-se reescrever a função de transferência da seguinte forma:

$$G_m(s) = \frac{K_t / (LaJ)}{\left(s + \frac{K_e K_t}{RaJ} \right) \left(s + \frac{Ra}{La} \right)} \quad (30)$$

ou

$$G_m(s) = \frac{1/K_e}{(s\tau_m + 1)(s\tau_e + 1)} \quad (31)$$

onde:

$$\tau_m = \frac{RaJ}{K_e K_t} \quad (32)$$

é a constante mecânica, e

$$\tau_e = \frac{L_a}{R_a} \quad (33)$$

a constante elétrica

A função de transferência do motor pela equação (31) é de segunda ordem e possui dois pólos. Cada pólo representa atraso, e no caso do motor temos dois. Um deles é representado pela indutância da bobina do motor (L_a) que gera o atraso para a passagem da corrente i_a , quando se aplica uma tensão aos seus terminais. O outro é o momento de inércia J , que representa o peso e as dimensões da armadura do motor, que dificultam o movimento mecânico do mesmo.

O atraso mecânico possui, devido à construção física do motor, valor muito maior que o atraso elétrico; e é da ordem de 10 vezes, ou seja:

$$\tau_m > 10 \tau_e \quad (34)$$

Da relação acima pode-se desprezar a indutância do motor L_a , para eliminar o atraso elétrico. Assim, chega-se a:

$$G_m(s) = \frac{\omega(s)}{E_a(s)} = \frac{1 / K_e}{\left(\frac{s R_a J}{K_e K_t} + 1 \right)}$$
$$G_m(s) = \frac{\omega(s)}{E_a(s)} = \frac{K_t}{s R_a J + K_e K_t} \quad (35)$$

Observa-se que esta equação é de primeira ordem, com características de um filtro passa-baixas [11] e por isso diz-se que o motor é um integrador [12]. Este dado informa que se for aplicada na entrada do motor uma tensão em forma de degrau, sua resposta será exponencial, tendendo ao valor de regime. O gráfico desenhado na página seguinte, ilustra a resposta transitória do motor DC à entrada degrau.

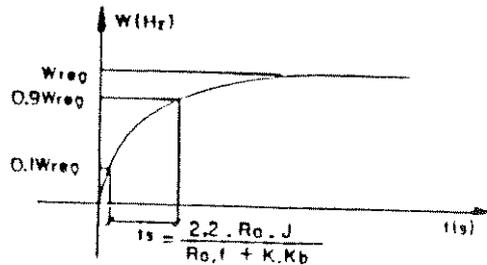


Figura 4.6
Resposta a Entrada Degrau do Motor DC

Fazendo-se $s = 0$, na equação (35) tem-se o ganho do motor em regime permanente, ou seja, obtém-se o valor que ao ser multiplicado pela tensão de entrada resulta na rotação de saída do motor em rotação por segundo (Hz), resultando em:

$$G_m(s = 0) = \frac{K_t}{K_e K_t} = \frac{1}{K_e} \quad (36)$$

Pode-se observar que a função de transferência fornece informações a respeito do funcionamento global do motor. Além de dizer qual a velocidade final do motor, em relação à tensão aplicada ($G_m(s=0)$), ela diz como é a resposta transitória e qual o tempo de duração desta transição.

Agora, estudando o motor como variável de saída, a posição do seu eixo apresenta a seguinte função de transferência.

$$\Theta(t) = \int \omega(t) dt \quad (37)$$

$$\Theta(s) = \frac{1}{s} \omega(s) \quad (38)$$

$$F\Theta(s) = \frac{\omega(s)}{sEa(s)} \quad (39)$$

$$F\Theta(s) = \frac{K_t}{s(RaJs + K_e K_t)} \quad (40)$$

A equação (40) representa a função de transferência do motor quando a saída é a posição do eixo e a entrada é a tensão de armadura. Existe um pólo

em ($s=0$), e o outro pode ser obtido igualando-se o polinômio característico de $F\Theta(s)$ a zero.

4.3 Funções de Transferência dos Módulos

4.3.1 Função de Transferência do Módulo MAM

O conjunto chamado MAM é composto pelo motor, amplificador e a mesa da máquina. Através de um levantamento prático e do embasamento teórico apresentado anteriormente, obteve-se a sua função de transferência.

Usando-se um gerador de sinais, aplicou-se na entrada deste módulo uma senóide de tensão 2Vpp e variou-se a frequência desde 2Hz até algumas dezenas de Hertz. Observou-se, preliminarmente, que em baixas frequências a mesa oscilava conforme a frequência da senóide alterava-se. Este deslocamento era registrado num acelerômetro (ver Seção 4.2.2) Por possuir uma saída pequena foi acoplado ao mesmo um amplificador para aumentar o sinal de sua saída. Após a varredura de frequência obteve-se uma tabela de pontos entre a entrada (frequência da senóide) e a saída (valor registrado no acelerômetro). Com estes valores, chegou-se à Resposta em Frequência (ver Seção 4.2.1) deste módulo, a qual está representada num gráfico de ganho e outro de fase.

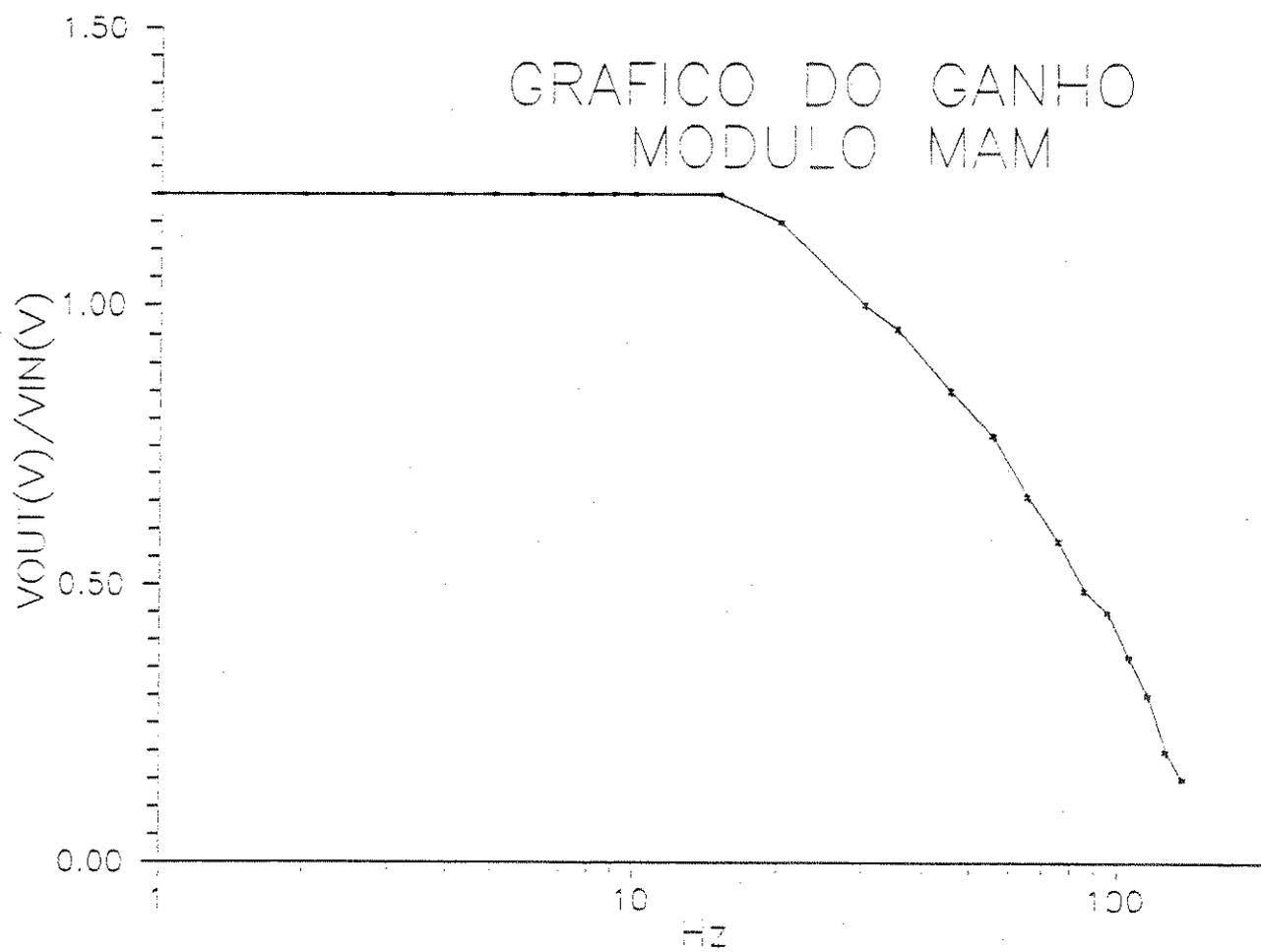


Figura 4.7
Gráfico do Ganho

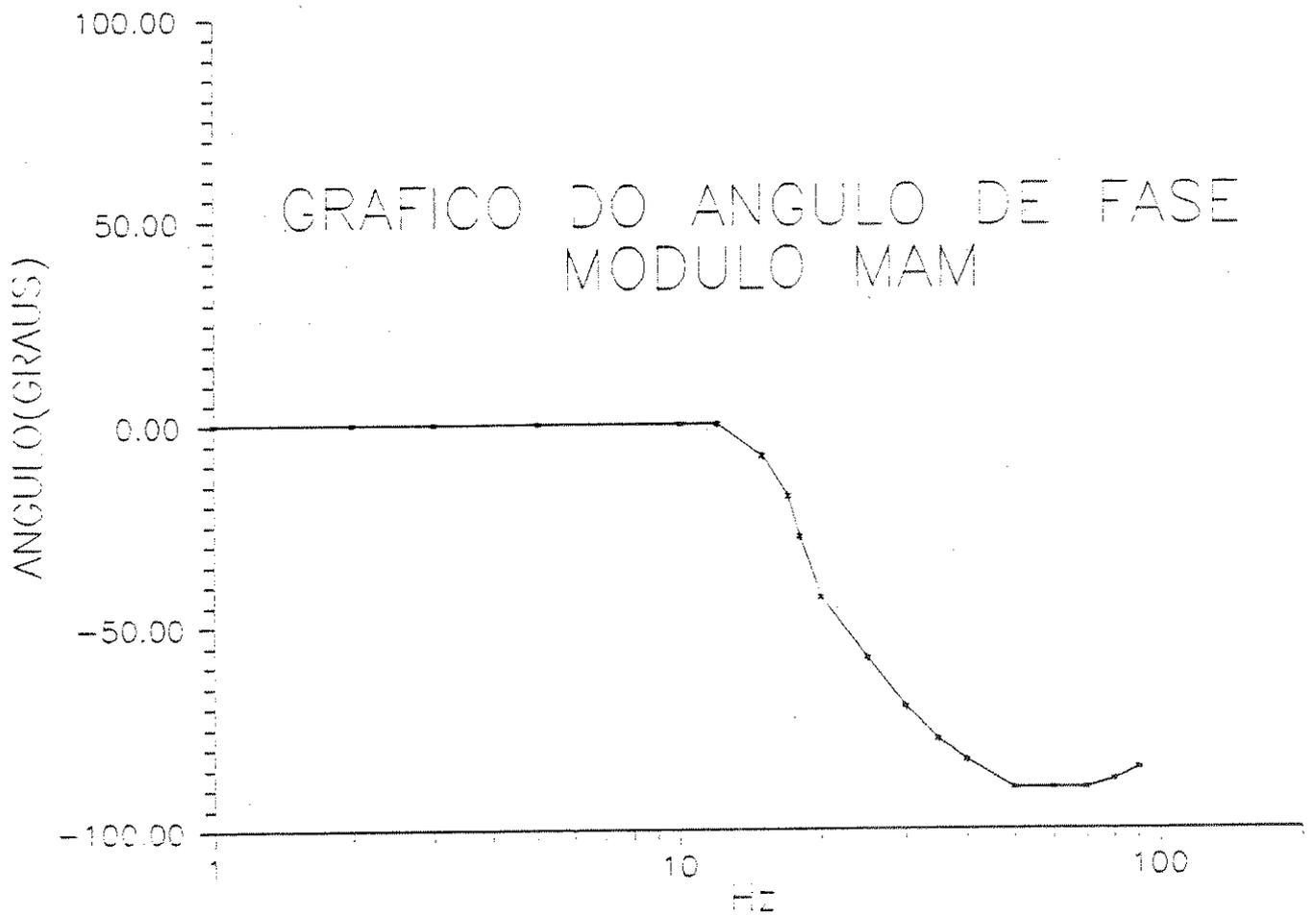


Figura 4.8
Gráfico do Ângulo de Fase

Como primeira conclusão observa-se que o mesmo não apresenta nenhuma frequência de ressonância, dentro do espectro de frequência analisado. Também pode-se notar que conforme a frequência aumenta, o conjunto tem menos tempo de responder aos estímulos externos e conseqüentemente a oscilação da mesa vai reduzindo exponencialmente.

Com relação aos gráficos, pode-se perceber que este módulo pode ser aproximado por um sistema de primeira ordem, tendo um ganho dc de 400 rad/s/V e uma frequência de corte da ordem de 20Hz. Um diagrama de bloco que o representa é mostrado em seguida.

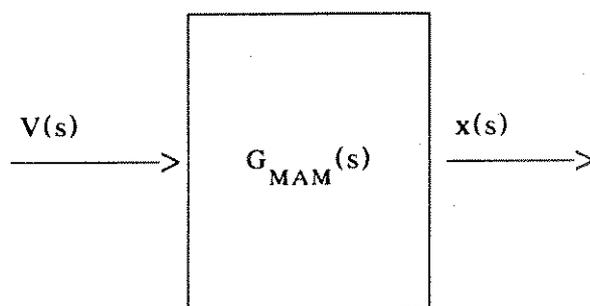


Figura 4.9
Diagrama de Bloco do Módulo MAM

onde:

$$G_{MAM}(s) = \frac{V(s)}{x(s)}$$

$$G_{MAM}(s) = \frac{400}{1 + 0.008s} \quad (41)$$

sendo $V(s)$ é a tensão de alimentação do Amplificador
 $x(s)$ é o deslocamento linear da mesa, dado em milímetros

4.3.2 Função de Transferência do Módulo de Transmissão

A construção mecânica da transmissão da máquina por fuso de esferas e sua aplicação em baixas solicitações de torque nos permite desprezar as não linearidades devido ao atrito de escorregamento e às folgas nos enrolamentos. Portanto, o módulo de transmissão pode ser modelado como uma constante, a qual é representada pelo passo do fuso de esferas recirculantes. (ver Seção 1.7)

O passo deste fuso determina o deslocamento linear da mesa.

Usando-se instrumentos apropriados mediu-se este valor (5mm). Uma representação, em um diagrama de bloco, da função de transferência do fuso é vista na figura 4.10.

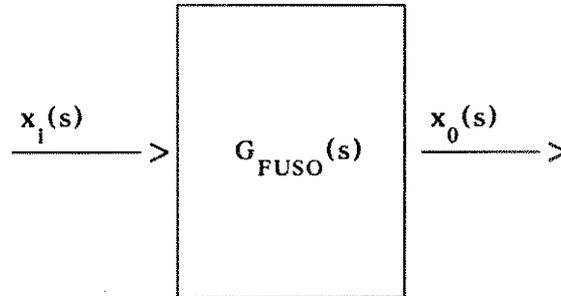


Figura 4.10

Diagrama de Bloco da Função de Transferência do Fuso de Esferas

onde:

$$G_{\text{FUSO}}(s) = \frac{x_o(s)}{x_i(s)}$$

$$G_{\text{FUSO}}(s) = 5 \text{ mm} \quad (42)$$

sendo x_o e x_i deslocamentos lineares, dado em milímetros.

Além do fuso, o sistema de transmissão é composto de outros elementos que transformam o deslocamento angular executado pelo eixo em movimento linear, efetuado pela mesa da máquina fresadora. O sistema como um todo apresenta a seguinte função de transferência:

$$G_{\text{ST}}(s) = \frac{x(s)}{\Theta(s)}$$

sendo $x(s)$ um deslocamento linear e $\Theta(s)$ um deslocamento angular.

$$G_{\text{ST}}(s) = \frac{1}{2\pi} G_{\text{FUSO}} \quad (43)$$

$$G_{\text{ST}}(s) = \frac{5}{2\pi} \text{ mm/rad}$$

A próxima figura mostra um diagrama de bloco da função de transferência do módulo de transmissão

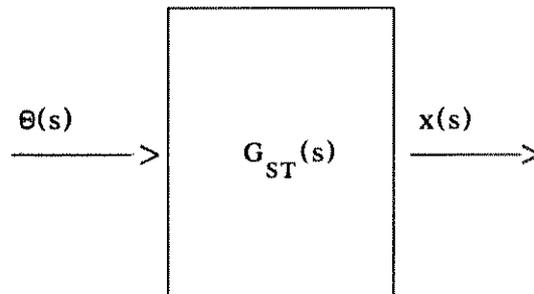


Figura 4.11

Diagrama de Bloco do Módulo de Transmissão

4.3.3 Função de Transferência do Conversor D/A

A interface entre o sistema digital (computador) e o analógico (amplificador) é executada pelo conversor D/A. Através de uma palavra digital colocada na sua entrada, obtemos na saída uma tensão analógica.

O conversor utilizado é de 10 "bits" e sua saída varia de -10V a +10V. Para este tipo de componente o número de "steps" de tensão, que ele fornece, pode ser obtido da seguinte maneira:

$$N = 2^B \quad (44)$$

onde: N é o número de níveis que o conversor fornece

B é o número de "bits" do conversor

Com um conversor de 10 "bits" temos:

$$N = 2^{10}$$

$$N = 1024 \text{ ("steps")}$$

Sua saída analógica pode variar de -10V a +10V, logo seu range (R) é:

$$R = V(+) - V(-) \quad (45)$$

sendo:

V(+) a tensão positiva máxima de saída do conversor

V(-) a tensão negativa máxima de saída do conversor

$$R = 10 - (-10)$$

$$R = 20 \text{ V}$$

Para calcularmos a função de transferência deste módulo, basta identificar a sua entrada e saída, em termos de valores. A entrada é uma palavra digital de 0 a 1023, enquanto a saída é um valor analógico de -10V a +10V. Logo a sua função de transferência é da forma:

$$G_{D/A}(s) = \frac{R(s)}{N(s)} \quad (46)$$

com:

$$R(s) = 20 \text{ V}$$

$$N(s) = 1024$$

$$G_{D/A}(s) = \frac{20}{1024} \text{ V/bit}$$

O diagrama em bloco do conversor D/A é apresentado na página seguinte.

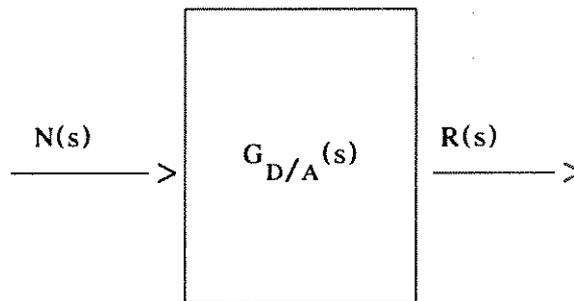


Figura 4.12

Diagrama de Bloco da Função de Transferência do Conversor D/A

4.3.4 Função de Transferência do Codificador Incremental

Para completar, analisaremos o Codificador. Como descrito anteriormente, o seu circuito gera pulsos digitais, dado um incremento angular do eixo do motor. No caso do Codificador, o pulso de saída é quadrado e a sua tensão TTL.

Para calcularmos a sua função de transferência, basta lembrar que na sua entrada tem-se uma frequência do motor em Hz e ele fornece na saída também uma frequência. Logo a função de transferência do circuito é o número de raias N do disco óptico, pois a frequência do sinal de saída é N vezes maior que a frequência de rotação do eixo do motor.

$$G_E(s) = \frac{f_0(s)}{f_i(s)} \quad (47)$$

onde:

$f_0(s)$ é a frequência de saída do sinal do Codificador

$f_i(s)$ é a frequência do eixo do motor em Hz

como $f_0(s) = N f_i(s)$ temos:

$$G_E(s) = N$$

Temos na página seguinte a sua representação.

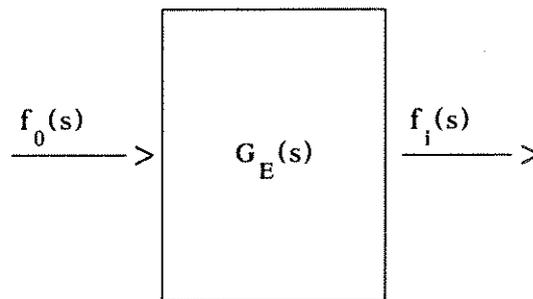


Figura 4.13

Diagrama de Bloco da Função de Transferência do Codificador

5. Conclusão

5.1 Resumo

O objetivo desta pesquisa foi projetar um sistema digital, visando atender às necessidades de controle, para uma máquina fresadora. Dentro desta visão, procuramos relatar este trabalho e as peculiaridades de cada uma das unidades que formam toda a estrutura, além das características de hardware que serão úteis para o desenvolvimento do módulo de controle.

Nesta última etapa, faremos uma comparação entre os dois cartões digitais e como poderão ser alterados.

5.2 Comparação entre os Módulos Digitais

Se observarmos o primeiro cartão digital e o outro cartão com microcontrolador notaremos algumas diferenças em termos de concepção de projetos que influenciarão no comportamento de cada um deles. Dentre elas destacamos:

- .interligação com o computador;
- .os módulos de hardware;
- .software básico;
- .confecção.

Interligação

O 1º cartão digital foi concebido para operar em conjunto com o computador, ou seja, o mesmo só desempenhará suas funções, caso esteja inserido em um "slot" de um computador da linha IBM PC ou compatível. Isto significa que operações de leitura e escrita em periféricos ocorrem quando, sinais existentes no "slot" do micro são ativados ou não, mediante comandos do software. Também a temporização dos mesmos é controlada pelo processador do computador, que determina como uma operação, ligada ao software, será executada.

Este tipo de ligação (cartão conectado ao "slot") possibilita comunicação, entre o computador e o cartão de forma simples, já que o fluxo de dados entre os dois sistemas ocorre através do barramento de dados, existente no "slot" do PC. A cada leitura e escrita são transmitidos 8 "bits", já que o barramento de dados dos periféricos existentes no cartão, pode receber e transmitir 1 "byte" em cada operação.

A unidade com microcontrolador permite uma maior independência em

relação ao computador, isto porque operações de escrita e leitura, por exemplo, em periféricos são efetuadas por sinais provenientes do microcontrolador. Este componente possui também uma área de memória que pode armazenar uma quantidade de dados, antes que a mesma seja transmitida para o PC.

Esta ligação mais tênue torna o sistema mais flexível, já que muitas operações que no sistema anterior eram executadas pelo processador do microcomputador, passaram a ser efetuadas pelo microcontrolador. Por outro lado, a comunicação tornar-se mais complexa, já que não existe uma ligação direta entre o barramento de dados das duas unidades de processamento. A transmissão não é mais em unidades de 8 "bits", sendo agora efetuada "bit" a "bit" (modo serial).

Um software de comunicação teve que ser desenvolvido e implantado para que os sistemas possam transmitir e receber informações, de maneira correta. Caso haja alguma queda no sistema de comunicação, a mesma deve ser sanada imediatamente.

Módulos de Hardware

No cartão que não possui o microcontrolador, muitos dos sinais vindos do "slot" devem ser ajustados às necessidades do hardware deste cartão. Em consequência temos um aumento no número de componentes que fazem parte dos módulos ligados à lógica. A implementação do mesmo é mais complexa e pode haver um número maior de falhas por queima ou mau funcionamento de um deles.

Também o número de ligações se torna muito maior, pois cada novo CI precisa no mínimo de uma ligação de alimentação e uma de terra para que possa funcionar corretamente. Este aumento não é infinito, sendo limitado a capacidade de corrente que cada fonte de alimentação do "slot" pode fornecer, e ao espaço físico.

O único módulo que não apresenta este tipo de problema é o de comunicação, já que o número de componentes que fazem parte desta estrutura é pequeno, devido à sua lógica ser extremamente simples, o que implica em um número reduzido de CIs.

No sistema com microcontrolador, o seu hardware possui uma estrutura bem mais simples, pois existe uma CPU totalmente dedicada aos processos de velocidade e posição, logo muitas das funções são executadas via software. Este fator é preponderante para termos uma lógica bem mais simples e com menor número de ligações.

Utiliza-se ainda o recurso da interrupção, disponível nesta CPU. Ela possibilita a substituição de vários componentes por um único, já que a lógica de ativação de operações que a CPU deve executar é feita pelo sinal de interrupção. Apesar desta ferramenta também poder ser utilizada no sistema

anterior, na execução de várias operações, torna-se inadequado seu emprego, devido à BIOS do computador, já haver configurado o controlador de interrupções para que sempre tenha como prioridades mais altas duas interrupções:

.interrupção de "refresh"

.interrupção do relógio

Estas interrupções ocorrem de forma não previsível em relação ao processo de velocidade e conseqüentemente poderia atrasar a execução do serviço de rotina solicitada por algum periférico do cartão ligado ao processo. Dependendo da operação poder-se-ia ter perdas de contagem de pulsos e resultados não precisos.

Olhando novamente para o cartão microcontrolado, este apresenta complicações no sistema de transmissão. São necessários componentes para que esta função possa ser executada. Alguns deles operam em níveis de tensão que não são os mesmos dos componentes TTL. Logo temos outras fontes de tensão, que devem possuir uma capacidade de corrente suficiente para alimentar estes circuitos.

Há ainda um componente especial que é o microcontrolador, 87C51. Ele é o responsável por todo o gerenciamento e funcionamento do hardware e do software dos processos, juntamente com os outros módulos. Qualquer mau funcionamento em uma de suas funções pode afetar o funcionamento do sistema como um todo.

Software Básico

O primeiro cartão digital é visto como uma interface externa, pelo computador. Neste aspecto é necessário que o mesmo possua um endereço base onde possa operar. A partir dele, todos os periféricos existentes nele possuem um endereço, que é a soma deste com um valor chamado "offset". Este tipo de mapeamento é em I/O, já que cada circuito é endereçado através de uma "porta" de entrada/saída. O computador possui algumas faixas de endereçamento que estão livres para serem usadas, portanto foi necessário encontrar uma faixa para operação do mesmo.

Por possuir um barramento de 8 "bits", a transmissão dos dados ocorre de forma transparente; não sendo necessário executar nenhuma manipulação nos dados que são trocados entre as duas unidades. Assim é necessário poucas operações para a manipulação dos mesmos.

Quanto ao software este foi codificado em uma linguagem de alto nível, fornecendo um arquivo "fonte". Após ser compilado e "linkado" resultou em um arquivo executável. Muitos dos comandos são codificados em micro instruções pelo computador para que o mesmo possa entendê-las. Conseqüentemente a área de

memória de programa se torna muito maior do que se o mesmo tivesse sido implementado em "assembly" do processador do computador. Assim ele rodaria apenas nos computadores compatíveis com este tipo de "assembly", tornando o software não portátil. Se esta codificação tivesse sido feita em "assembly" se perderia a portabilidade do software.

Na segunda unidade temos algumas características que diferem das apresentadas acima. O sistema não apresenta memória externa, o que implica numa área de dados muito pequena.

Os periféricos são mapeados em endereços de I/O, porém diferentes dos decodificados no cartão IBM PC-XT/AT. Estes são gerados pelo microcontrolador, conforme a implementação do software e do hardware. Se em futuras aplicações for necessário o uso de memória externa, o endereço atual dos mesmos terá que ser alterado para uma faixa conveniente, e conseqüentemente o programa gravado em EPROM terá que ser alterado.

Optou-se pelo desenvolvimento do software utilizando o "assembler" do microcontrolador, devido à área de memória.

A conseqüência imediata é uma implementação muito mais complexa que a desenvolvida no cartão IBM. Isto ocorre porque a codificação de qualquer comando utilizando-se este tipo de linguagem, exige mais código de programa

A nível de arquivo este ocupa uma área de memória bem menor que o primeiro sistema, logo as áreas de EPROM e RAM ocupadas serão bem menores, o que possibilitará o desenvolvimento de novos módulos neste software sem que haja necessidade de se colocar memória externa.

A depuração dele, nos testes de campo é muito difícil, já que não se possui condições de monitorar o que ocorre em certos pontos do programa. Para se evitar erros de lógica, ou outros que por ventura possam ocorrer, todo o sistema foi depurado utilizando um simulador.

Neste foram feitos vários testes procurando-se verificar e corrigir os módulos de programação que não estavam de acordo com as especificações. Todo o processo foi executado antes que os testes de campo fossem implementados.

No primeiro sistema, esta etapa foi feita de forma menos rigorosa, pois a qualquer instante poder-se-ia verificar o "status" do software e como ele estava se comportando.

Confecção

A primeira unidade, por utilizar os sinais do computador para operar, apresenta uma topologia muito mais complexa que a segunda. Como resultado o número de ligações, de componentes e a área ocupada é muito maior que o segundo

Se a mesma for implementada em um circuito impresso, ter-se-á que

usar um software com vários recursos que possam minimizar o espaço ocupado e tornar o "layout" final o mais compacto possível.

Por ser projetada para operar no "slot" do computador, o seu sistema de comunicação é simples e não necessita de nenhum conector e cabos blindados para implementar a parte física da comunicação.

Na outra unidade, a existência de uma unidade de processamento faz com que tenhamos um hardware menor, já que muitas das operações que deveriam ser efetuadas por periféricos passam a ser executadas por software. Assim o tamanho da placa, que conterá os componentes do sistema, será menor e muito menos densa que a anterior.

Em contraste, a comunicação entre este sistema e o computador, implicará na existência de conectores e cabos blindados para que a mesma possa ser efetuada.

Módulo da Velocidade

Para finalizar temos uma observação sobre o Módulo da Velocidade.

Foi sugerido a apresentação da velocidade angular do eixo do motor no vídeo conforme a máquina movimentava-se. Implementou-se uma rotina para efetuar esta tarefa, porém quando se executava esta rotina degradava-se o sistema operacional DOS. Isto ocorreu porque há uma concorrência entre os processos de enviar a informação da velocidade angular do eixo do motor para o vídeo e receber dados pela porta serial. Devido ao DOS não ser reentrante, essa concorrência causa uma degradação ao sistema operacional (DOS) e não é possível ver tal informação no vídeo.

Para solucionar este problema foi armazenado a informação da velocidade angular do eixo do motor em um vetor. Após a máquina terminar seu deslocamento este vetor é transferido para um arquivo. Usa-se um software gráfico para plotar o perfil da velocidade angular do eixo do motor, com os dados armazenados neste vetor, para um deslocamento efetuado pela mesa da máquina.

5.3 Futuras Implementações

Na discussão anterior observamos que um dos problemas existentes nas duas unidades é o número de CIs existentes nos módulos de lógica que geram placas maiores e mais densas, devido ao maior número de ligações existentes. Uma das possíveis soluções seria o uso das chamadas PALs.

Uma PAL ("Programmable Array Logic") é um componente que pode substituir os sistemas em funções lógicas sequenciais e combinacionais. Eles apresentam baixo custo, proporcionam maior confiabilidade ao sistema como um todo, reduzem o consumo de energia e a área de circuito impresso, além de permitir proteção contra

cópias.

A família destes dispositivos possui um arranjo de portas lógicas "AND" programáveis, tendo as saídas fixas com arranjos de portas lógicas "OR". Uma outra configuração possui na saída um "flip-flop" tipo D que armazena o resultado de uma operação lógica, executada internamente na PAL. Na figura 5.1, temos um possível esquema de uma das configurações destes componentes.

Ainda existem outras características como:

.alta velocidade;

.facilidade de programação, por programadores padrão de EPROM. A descrição da lógica pode ser implementada em alto nível, utilizando-se softwares como PALASM, ABEL.

.fusível de segurança.

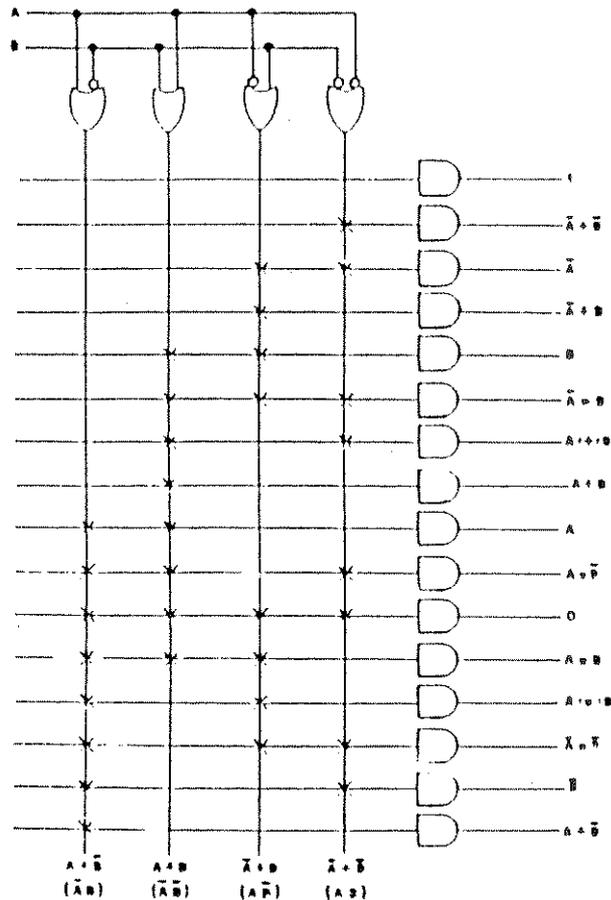


Figura 5.1

Arranjo de uma PAL

O primeiro cartão digital também poderia possuir uma RAM externa que armazenaria os dados vindos dos periféricos e depois os transmitiria para o

computador quando este os solicitasse. Este recurso possibilitaria uma maior flexibilidade na operação do sistema.

Uma outra implementação que reduziria ainda mais os componentes, seria a utilização de GALs. Estes componentes permitem substituir não só os módulos de lógica mas também contadores, registradores e outros componentes.

A primeira tarefa seria determinar as equações que regem o funcionamento de cada um dos módulos. De posse delas, teríamos que minimizá-las, usando uma ferramenta de software. Em seguida as mesmas seriam geradas, no software que trabalha junto com estas GALs, e simuladas para corrigir as falhas que houvesse. Em seguida seriam implementadas nestes componentes e estes seriam montados nas placas dos sistemas que executarão os processos de velocidade, posição e comunicação, se houvesse.

BIBLIOGRAFIA

- [1] ALLOCCA, J. A., "Transducers: Theory and Application", Prentice-Hall Company, 1984.
- [2] ALMEIDA, J. L. A. de, "Eletrônica de Potência", Érica, 1986.
- [3] ASTROM, K. J., "Computer Controlled Systems", Prentice-Hall, Englewood Cliffs, 1984.
- [4] AUBRY, J. F., P. Fister, G. H., Louis J. P., and El-Hefnawy, A. A., "Minimal Microcomputer Hardware for Direct Digital Control of a DC Motor", Control and Computer, vol 9, nº 3, 1981, 41-44.
- [5] AZEVEDO, J. B. Jr., "TTL/CMOS: teoria e aplicaçãoem circuitos digitais", vol 1, Érica, 1984.
- [6] AZEVEDO, J. B. Jr., "TTL/CMOS: teoria e aplicaçãoem circuitos digitais", vol 2, Érica, 1984.
- [7] BEHAR, M., "Máquinas Ferramentas", Hemus- Livraria e Editora Ltda., 1975.
- [8] BENTLEY, J. P., "Principles of Measurement Systems", John Wiley & Sons, 1989.
- [9] BENTO, C. R., "Sistemas de Controle: Teoria e Projetos", Érica, 1989.
- [10] BIRD, M. & SCHMIDT, R. "Practical Digital Electronics - An Introductory Course", vol 2, Hewlitt Packard Company - HP.
- [11] BLUKIS, J. & Baker, M., "Practical Digital Eletronics - An Introductory Course", vol 1, Hewlett Packard Company - HP.
- [12] BOWES, S. R. and T.DAVIS, "Microprocessor - based Development System for PWM Variable - Speed Drives", IEE Proceedings, vol 132, Pt B, nº 1, January 1985.
- [13] BOWLER, P., "Power Transistor in variable Speed Drives", IEE Eletronics & Power, 730-736, October 1978.

- [14] BRITO, A. A. S., "Manual do Programador PC: Hardware & Software", Érica, 1990.
- [15] C-51 Cros-Compile Kit for 8051 Microcontroller Development, Archimedes Software, 1987.
- [16] CALDWELL, S. H., "Switching Circuits and Logical Design", John Wiley & Sons, 1982.
- [17] CAMILO D., J. Baptista, T. Yabu-Uti, Yano, "Circuitos Lógicos: teoria e laboratório", Ciência e Tecnologia, 1984.
- [18] CAMPBELL, J., "THE RS-232 Solution", Sybex Computer Books, 1984.
- [19] CASSELL, D. A., "Microcomputers and Modern Control Engineering", Reston Publication Company, 1983.
- [20] CASTRUCCI, P. B. L. e L. BATISTA, "Controle Linear: Método Básico", Edgard Blucher, 1980.
- [21] CASTRUCCI, P. e R. M. SALES, "Controle Digital", Edgard Blucher Ltda., 1990.
- [22] CETINKUNT, S. and W. J. BOOK, " Performance Limitations of Joint Variable - Feedback Controllers Due to Manipulator Structural Flexibility", IEEE Transactions on Robotics and Automation, vol 6, nº 2, April, 1990, 219-231.
- [23] CHEN, W., "Passive and Active Filters", John Wiley & Sons, 1986.
- [24] CHU, Y., "Digital Computer Fundamentals", McGraw-Hill.
- [25] CMOS Integrated Circuits, Harris Semiconductor.
- [26] CRAING, J. J., "Introduction to Robotics", Addison-Wesley Publishing Company, 1986.
- [27] DAVIS, S. R., "Turbo C - The Art of Advanced Program Design, Optimization and Debugging", M&T Books, 1988.

- [28] D'AZZO, J. J., "Linear Control System: Analysis and Design", McGraw-Hill, 1981.
- [29] "DDC Introduces New Selsyn Indicator", *Electronics*, vol 4, nº 54, February, 1981.
- [30] Designing with TTL Integrated Circuits, Texas Instruments.
- [31] DEWAN S. B., and A. Straughen, "Power Semiconductor Circuits", John Wiley & Sons, 1975.
- [32] DC Motors, Speed Controls, Servo Systems, an Engineering Handbook, Electro-Craft Corporation, 1973.
- [33] DOEBELIN, E. O., "Measurement Systems Application and Design", McGraw-Hill, 1975.
- [34] DORADLA, S. R., "Control Schemes for Saled State DC Series Motor Drives", Unpublished PhD Tesis, Queen's University, Kingston, Canadá, 1975.
- [35] EDWARDS, F. H., "The Principles of Switching Circuits", MIT Press, 1984.
- [36] EGGBRECHT, L. G., "Interfacing to the IBM Pesrsonal Computer", Howard W. Sams, 1983.
- [37] FALDELLA, E. and A. Toniclli, "A Microprocessor - Controlled DC Motor Drive", *Control and Computers*, vol 10, nº 1, 1982, 22-26.
- [38] FRANKLIN, G. F. and J. D. POWEL, "Digital Control of Dynamic Systems", Addison Wesley, 1980.
- [39] GERRY, J. P., "A Comparison of PID Control Algorithms", *Control Engineering*, 102-105, March 1978.
- [40] GOMES, C., Jorge, J. C. S., Pimenta,F., "Desenvolvimento Estruturado de Sistemas", Ebras, 1987.
- [41] HERCEG, E. E., "Handbook of Measurement and Control", Schaevitz Engineering, 1980.

- [42] HOUPIS, C. H. and G. B. LAMONT, "Digital Control Systems Theory, Hardware, Software", McGraw-Hill, 1985.
- [43] IBM Corporation "Technical Reference: Personal Computer XT and Portable Personal Computer", Ed. IBM Corporation, 1986.
- [44] IBM, "IBM PC Technical Reference Manual", 1983.
- [45] IDOETA, F. e G. Capuano, "Elementos de Eletrônica Digital", Érica, 1984.
- [46] "Inside the IBM-PC", Prentice-Hall, 1983.
- [47] Integrated Circuits, Philips, 1982.
- [48] INTEL Corporation "The 8086 Family User's Manual", Ed. Intel Corp., 1979.
- [49] INTEL, "IAPX 86/88 User's Manual", 1983.
- [50] JAMSA, K., "Turbo C - Programmer's Library", McGraw-Hill, 1988.
- [51] JURY, E. I., "Sampled-Data Control Systems", John Wiley & Sons, 1977.
- [52] KAILATH, T., "Linear Systems", Prentice-Hall, 1980.
- [53] KAYA, A. and T. J. SHCIB, "Tunning of PID Controls of Different Structures", Control Engineering, 62-65, July 1988.
- [54] KERNIGHAN, B. W. and Ritchie, D. M., "C a Linguagem de Programação", Campus, 1986.
- [55] KLING, G. Jr., "Introduction to the Methodology of Switching Circuits", Van Nostrand, 1980.
- [56] LANCASTER, D., "CMOS COOKBOOK", Howard W. Sams & Cv, 1981.
- [57] LANCASTER, D., "TTL COOKBOOK", Howard W. Sams & Cv, 1982.
- [58] LANDER, C. W., "Power Electronics", McGraw-Hill, 1981.

- [59] LAUB, A. J. "Numerical Linear Algebra: Aspects of Control Design Computations", IEEE Transactions on Automatic Control, vol AC - 30, nº 2, February 1985, 97-108.
- [60] Linear and Interface Integrated Circuits, Motorola, 1985.
- [61] Linear Semiconductor, National Semiconductor, 1981.
- [62] LION, K. S., "Elements of Electrical and Electronic Instrumentation", McGraw-Hill, 1975.
- [63] MACHADO, A., "O Comando Numérico Aplicado às Máquinas Ferramentas", Ícome Editora Ltda., 1986.
- [64] Manual Palasm, Monolithic Memories, 1987.
- [65] MESQUITA, T. J. M., "Linguagem C, Teoria & Programas", Érica, 1988.
- [66] Microcontroller Handbook, Advanced Micro Devices, 1988.
- [67] MILLES, A. R., "Assembly IBM PC", Ebras, 1987.
- [68] MOLER, C., Little, J. and Bangert, S., "PC-MATLAB for MS-DOS Personal Computers", the Math Works, 1987.
- [69] MORGAN, C. L. and Waite M., "8086/8088-16-Bit Microprocessor Primes", McGraw-Hill, 1988.
- [70] NEUBERT, H. K. P., "Instrument Transducers: An Introduction to their Performance and Design", 1975, p.56.
- [71] NEUBERT, H. K. P., "Instrument Transducers: An Introduction to their Performance and Design", 1975, p. 258-265.
- [72] OGATA, K., "Modern Control Engineering", Prentice-Hall, Englewood Cliffs, 1970.
- [73] OSBORNE, A., "Microprocessadores Conceitos Básicos", vol. 1 McGraw-Hill, 1984.

- [74] PAL Device Data Book, Monolithic Memories, 1988.
- [75] PERMAN, Richard A., "Power Eetronics: Solid State Motor Control", Reston, 1978.
- [76] POLLACK, S. H., "Microprocessor Programming and Application to DC Motor Control", McGraw-Hill, 1985.
- [77] Power Device Data, Motorola, 1978.
- [78] Power Semiconductors and their Applications, Conference Records, IEE London.
- [79] RAY, R., "Basic Digital Eletronics", Tab Books.
- [80] SACHS, J., "IBM PC e Compatíveis: Guia do Usuário", McGraw-Hill, 1986.
- [81] SAMS, H. W., "Modern Computer Concepts", McGraw-Hill, 1982.
- [82] SANTOS, J. P. and Raymundi, E. Jr., "Programando em Assembler 8086/8088", McGraw-Hill, 1989.
- [83] SEN, P. C., "Thyristor DC Drives", John Wiley & Sons, 1981.
- [84] SHILDT, H., "Linguagem C - Guia do Usuário", McGraw-Hill, 1987.
- [85] SHINNO, H., ITO, Y. and HASHIZUME, H., "A Decision - Making Methodology for Basic Layout Design of Machine Tools", JSME International Journal, series III, vol 34, nº 2, 1991, 290 - 294.
- [86] STOUT, D. F. and M. KAUFMAN, "Handbook of Operational Amplifier Circuit Design", McGraw-Hill, 1976.
- [87] Synchros Conversation Handbook, 3rd ed., ILC Data Device Corporation, Bohemia, 1982.
- [88] TAUB, H. e D. Schilling, "Digital Integrated Eletronics", McGraw-Hill, 1983.
- [89] The TTL Data Book, vol 1, Texas Instruments, 1985.

- [90] The TTL Data Book, vol 2, Texas Instruments, 1985.
- [91] TOKHEIN, R. L., "Digital Principles", McGraw-Hill, 1980.
- [92] TOMPKINS, W. L. and J. G. WEBSTER, "Interfacing Sensors to the IBM PC", Prentice-Hall, Englewood Cliffs, 1988.
- [93] TRUXAL, J. G., "Automatic Feedback Control System Synthesis", McGraw-Hill, 1955.
- [94] TTL Data Book, Itaotec, 1984.
- [95] TTL Data Book, National Semiconductor, 1976.
- [96] "Tutsim, User's Manual IBM PC Computers", Applied, 1986.
- [97] WECK, M., "Handbook of Machine Tools", vol 1, Wiley Heyden Ltda., 1984.
- [98] WECK, M., "Handbook of Machine Tools", vol 2, Wiley Heyden Ltda., 1984.
- [99] WECK, M., "Handbook of Machine Tools", vol 3, Wiley Heyden Ltda., 1984.
- [100] WECK, M., "Handbook of Machine Tools", vol 4, Wiley Heyden Ltda., 1984.