

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIA E TECNOLOGIA

CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**MÓDULO DE INTERFACE IEEE-488 PARA O
MICROCONTROLADOR MC68HC11**

Fábio Toni Braz

Campina Grande - PB

**MODULO DE INTERFACE IEEE-488 PARA O
MICROCONTROLADOR MC68HC11**

Fábio Toni Braz

Dissertação apresentada à Coordenação dos Cursos de Pós-Graduação
em Engenharia Elétrica da Universidade Federal da Paraíba, em
cumprimento às exigências para a obtenção do grau de Mestre.

Gurdip Singh Deep
Orientador

Raimundo Carlos Silvério Freire
Orientador

Campina Grande - PB
Dezembro de 1993



B794m Braz, Fábio Toni.
Módulo de interface IEEE-488 para o microcontrolador
MC68HC11 / Fábio Toni Braz. - Campina Grande, 1993.
79 f.

Dissertação (Mestrado em Engenharia Elétrica) -
Universidade Federal da Paraíba, Centro de Ciências e
Tecnologia, 1993.

"Orientação : Prof. Dr. Gurdip Singh Deep, Prof. Dr.
Raimundo Carlos Silvério Freire".

Referências.

1. Automação de Sistemas de Instrumentação e Controle.
2. IEEE-488 - Interface 3. Microcontrolador MC68HC11. 4. Dissertação - Engenharia Elétrica. I. Deep, Gurdip Singh. II. Freire, Raimundo Carlos Silvério. III. Universidade Federal da Paraíba - Campina Grande (PB). IV. Título

CDU 004.052.32(043)

**MÓDULO DE INTERFACE IEEE-488 COM O
MICROCONTROLADOR MC68HC11**

Fábio Toni Braz


DISSERTAÇÃO APROVADA EM 20.12.93




GURDIP SINGH DEEP, Ph. D., UFPB
Orientador



RAIMUNDO CARLOS SILVERIO FREIRE, Dr., UFPB
Orientador



ANTÔNIO MARCUS NOGUEIRA LIMA, Dr., UFPB
Componente da Banca



MISAEEL ELIAS DE MORAES, Dr. - Ing., UFPB
Componente da Banca

**CAMPINA GRANDE - PB
DEZEMBRO - 1993**

Aos meus pais Otaviano e Carol,
pelo apoio, amor e dedicação;
à minha esposa Dênia, por
sua paciência e amor;
e ao meu filho Daniel.

AGRADECIMENTOS

Ao término desta dissertação, agradeço a todos que de forma direta ou indireta, auxiliaram em sua execução.

Aos professores Gurdip Singh Deep e Raimundo Carlos Silvério Freire, pela orientação e apoio dedicados ao longo deste trabalho.

Ao professor Antônio Marcus Nogueira Lima, pela orientação e incentivo no desenvolvimento deste trabalho.

Ao professor José Homero Feitosa Cavalcanti, pela orientação no desenvolvimento do "software" do executivo em tempo real.

Ao CNPq pela oportunidade e auxílio concedidos durante o desenvolvimento deste trabalho.

À Universidade Federal da Paraíba e em especial à Coordenação do Mestrado em Engenharia Elétrica.

Aos amigos do laboratório de Instrumentação e em especial a Sérgio Daher, Alberto Willian e José Sérgio que ajudaram nas diversas etapas deste trabalho.

RESUMO

Na automação de sistemas de instrumentação e controle, pode-se usar um barramento com o padrão IEEE-488 para interconectar equipamentos utilizados no sistema. Apresenta-se neste trabalho o desenvolvimento de um módulo para executar as tarefas da interface IEEE-488. Este módulo é baseado no microcontrolador MC68HC11 e executa todas as tarefas da interface, de aquisição de dados, geração de sinais e comandos de dispositivos externos. Foi adaptado um executivo em tempo real para gerenciar a operação do microcontrolador, tornando relativamente simples as modificações no "software". Foram feitas aplicações como digitalizador e como gerador de pulsos.

ABSTRACT

IEEE-488 interface bus is employed in the instrumentation and control systems, to interconnect different equipments employed in these systems. In this dissertation, the development of a module to execute the tasks associated with the IEEE-488 interface is presented. This module is based on MC68HC11 microcontroller. The microcontroller, apart from executing the interface tasks, can also execute the tasks of data acquisition or signal generation. A real time executive has been adapted to supervise the operation of the MC68HC11. The executive makes it relatively easy to modify the software. The proposed module has been used as a digitizer and a pulse generator.

SUMÁRIO

RESUMO	iii
ABSTRACT	iv
SUMÁRIO	v
LISTA DE FIGURAS E TABELAS	viii
LISTA DE ABREVIACÕES E SÍMBOLOS	ix
INTRODUÇÃO	1
CAPÍTULO I	3
CARACTERÍSTICAS DO PADRÃO IEEE-488	3
I.1. EVOLUÇÃO	3
I.2. PRINCIPAIS CARACTERÍSTICAS DO BARRAMENTO IEEE-488	5
I.2.1. Vias do Barramento	7
I.2.1.1. Barramento de Dados	8
I.2.1.2. Vias de "Handshake"	9
I.2.1.3. Vias de Gerenciamento Geral	11
I.2.2. Classificação das Mensagens do Barramento	12
I.2.3. Especificações Mecânicas e Elétricas da Interface IEEE-488	13
I.2.4. Exemplo do Funcionamento das Vias de "Handshake"	14
I.2.5. Repertório de Funções da Interface	18
I.2.6. Circuitos Integrados Dedicados para Interface IEEE-488	19
CAPÍTULO II	22
MÓDULO DE INTERFACE IEEE-488 BASEADO NO	
MICROCONTROLADOR MC68HC11	22
II.1. MÓDULO DE INTERFACE PARA O PADRÃO IEEE-488	22
II.2. REQUISITOS TÉCNICOS DO MÓDULO DE INTERFACE	23
II.3. MICROCONTROLADOR MC68HC11	23

CAPÍTULO III	27
EXECUTIVO EM TEMPO REAL DE GERENCIAMENTO	27
III.1. EXECUTIVO EM TEMPO REAL	27
III.1.1. Executivo em Tempo Real do Módulo de Interface	28
III.2. IMPLEMENTAÇÃO DAS TAREFAS DA INTERFACE IEEE-488	32
III.2.1. Notação dos Diagramas de Estados da Interface IEEE-488	32
III.2.1.1. Diagramas de estados da função RL ("remote/local" - remoto/local)	33
III.2.1.2. Diagramas de estados da função AH ("acceptor handshake" - receptor de "handshake")	34
III.2.1.3. Diagramas de estados da função SH ("source handshake" - fonte de "handshake")	36
III.2.1.4. Diagramas de estados da função L ("listen" - ouvinte)	38
III.2.1.5. Diagramas de estados da função T ("talker" - locutor)	39
III.2.1.6. Diagramas de estado da função DC ("device clear" - limpa instrumento)	41
III.2.2. Registradores de Supervisão do Barramento	42
III.2.3. Tarefas da Interface IEEE-488	43
III.2.3.1. Tarefas 1, 2 e 3	43
III.2.3.2. Tarefas 4	43
III.2.3.3. Tarefas 5	44
III.2.3.4. Tarefas 6, 7 e 8	45
III.2.3.5. Tarefas 9, 10 e 11	46
III.2.3.6. Tarefa 12	47
III.2.3.7. Tarefa 13	47
III.2.3.8. Tarefas 14, 15 e 16	48
III.2.4 DIMENSIONAMENTO DE MEMÓRIA PARA ESTE MÓDULO	49
CAPÍTULO IV	51
APLICAÇÕES	51
IV.1. APLICAÇÕES EM INSTRUMENTOS DE MEDIDA	51
IV.1.1. Digitalizador com o Módulo IEEE-488	51
IV.1.2. Gerador de Pulsos	53
IV.2. MODIFICAÇÕES NO EXECUTIVO	54
IV.2.1. Introdução de Novas Tarefas no Executivo em Tempo Real	55

IV.2.2. Troca dos Endereços de Locutor e Ouvinte do Módulo de Interface IEEE-488	56
IV.2.3. Novos Comandos dos Sub-Sistemas do Módulo de Interface IEEE-488	57
CAPÍTULO V	58
CONCLUSÕES	58
ANEXO A - O PROCESSO DE "POLL"	60
ANEXO B - MENSAGENS DE ENDEREÇOS	62
ANEXO C - COMANDOS UNIVERSAIS	64
ANEXO D - COMANDOS DE ENDEREÇAMENTO	66
ANEXO E - COMANDOS SECUNDÁRIOS	68
ANEXO F - COMANDOS DE DESATIVAÇÃO	69
ANEXO G - REPERTÓRIO DAS FUNÇÕES DA INTEFACE IEEE-488	70
ANEXO H - MICROCONTROLADOR MC68HC11E2	76
REFERÊNCIAS BIBLIOGRÁFICAS	77

LISTA DE FIGURAS E TABELAS

Figura 1.1: Barramento IEEE-488.	6
Figura 1.2: Níveis lógicos das 16 vias do barramento da interface.	8
Figura 1.3: Barramento com um locutor e dois ouvintes.	14
Figura 1.4: Diagrama de tempos dos sinais de "handshake".	15
Figura 1.5: Diagrama de fluxo mostrando o "handshake" entre o locutor e os ouvintes.	17
Figura 2.1: Diagrama do módulo para interface IEEE-488 baseado no 68HC11.	25
Figura 3.1: Estados das tarefas.	28
Figura 3.2: Fluxograma do executivo em tempo real.	30
Figura 3.3: Continuação do fluxograma do executivo em tempo real	31
Figura 3.4: Lista de descritores das tarefas desativáveis ou não desativáveis.	32
Figura 3.5: Diagrama de estados da função RL do módulo desenvolvido.	33
Figura 3.6: Diagrama de estados da função AH do módulo de interface.	35
Figura 3.7: Diagrama de estados da função SH do módulo de interface.	37
Figura 3.8: Diagrama de estados da função L do módulo de interface.	39
Figura 3.9: Diagrama de estados da função T do módulo de interface.	40
Figura 3.10: Diagrama de estados da função DC do módulo de interface.	42
Figura 4.1: Disposição dos equipamentos para testes no módulo de interface IEEE-488.	52
Figura 4.2: Disposição dos equipamentos para testes no módulo de interface IEEE-488.	54
Tabela I.1. CI's dedicados a interface IEEE-488.	19
Tabela I.2. CI's dedicados para controle das vias do barramento de dados e de gerenciamento geral.	20
Tabela III.1: Campos dos descritores das tarefas.	29
Tabela III.2: Quantidade de memória e número de instruções por tarefa.	50
Tabela IV.1: Resultados das leituras do módulo de interface e do multímetro digital.	53
Tabela B.1: Códigos para os endereços de ouvinte e locutor.	63
Tabela C.1: Comandos universais.	64
Tabela D.1: Comandos de endereçamento.	66
Tabela F.1: Comandos de desativação.	69

LISTA DE ABREVIACÕES E SÍMBOLOS

ACDS - "Accept Data State" - estado de aceitação de dados.
ACRS - "Acceptor Ready State" - estado de receptor pronto.
AH - "Acceptor Handshake" - receptor de "handshake".
AIDS - "Acceptor Idle State" - estado de recepção desativado.
ANRS - "Acceptor Not Ready State" - estado de receptor não pronto.
ANSI - "American National Standard Institute".
ASCII - "American Standard Code for Information Interchange".
ATN - "Attention" - atenção.
AWNS - "Acceptor Wait for New ciclo State" - estado de espera por um novo ciclo.
C - "Controller" - controlador.
CI's - Circuitos Integrados.
CPU - Unidade de Processamento Central.
DAV - "Data Valid" - dados válidos.
DAB - "Data Byte" - "byte" de dados.
DAC - "Data Accepted" - dado accito.
DIO1...DIO8 - linhas do barramento de dados.
DT - "Device Trigger".
DC - "Device Clear" - limpa instrumento.
DCAS - "Device Clear Active State" - instrumento limpo em estado ativo.
DCIS - "Device Clear Idle State" - instrumento limpo em estado desativado".
DCL - "Device Clear" - instrumento limpo.
EEPROM - "Electrically Erasable PROM" -
EOS - "End Of String" - final de "string".
EOI - "End Or Identify" - fim ou identifique.
END - "End" - fim.
GTL - "Go To Local" - vá para local.
GET - "Group Execute Trigger" -
GPIB - "General Purpose Interface Bus".
HP - "Hewlett Packard".
HP-IB - "Hewlett Packard Interface Bus".
IEEE - "Institute of Electrical and Electronics Engineers".
IEC - "International Electrical Commission".
IFC - "Interface Clear" - inicialização da interface.
IRQ - "Interrupt Request" - pedido de interrupção.

L - "Listen" - ouvinte.
L/LE - "Listen or Extended Listener" - ouvinte ou ouvinte estendido.
LACS - "Listener Active State" - estado de ouvinte ativo.
LDAS - "Listener Addressed State" - estado de ouvinte endereçado.
LIDS - "Listener Idle State" - estado de ouvinte desativado.
LOCS - "Local State" - estado local.
LLO - "Local Lockout" - chave local.
MA - "My Adress" - meu endereço.
MLA - "My Listen Adress" - meu endereço de ouvinte.
MTA "My Talk Adress" - meu endereço de locutor.
nba - "new byte available" - novo "byte" disponível.
NDAC - "Data Not Accepted" - dado não aceito.
NRFD - "Not Ready For Data" - não pronto para dados.
OTA - "Other Talk Adress" - outro endereço de locutor.
pon - "power on" - ligado.
PP - "Parallel Poll" - poll paralelo.
PPC - "Parallel Poll Configure" - paralelo poll configurado.
PPU - "Parallel Poll Unconfigure" - paralelo poll não configurado.
REN - "Remoto Enable" - equipamento em modo remoto.
RFD - "Ready For Data" - pronto para dados.
RAM - memória de acesso aleatório.
RL - "Remote/Local" - remoto ou local.
ROM - "Read Memory Only" - memória somente de leitura.
RTI - "Real Time Interrupt" - interrupção em tempo real.
REMS - "Remote State" - estado remoto.
SH - "Source Handshake" - fonte de "handshake".
SDC - "Selected Device Clear" - seleciona dispositivo limpo.
STRS - "Source Transfer State" - estado de transferência da fonte.
STB - "Status Byte" - "byte" de estado.
SRQ - "Service Request" - solicitação de serviço.
SPE - "Serial Poll Enable" - poll serial habilitado.
SPD - "Serial Poll Disable" - poll serial desabilitado.
SR - "Service Request" - pedido de serviço.
SIDS - "Source Idle State" - fonte em estado desativado.
SGNS - "Source Generate State" - estado de fonte geral.
SDYS - "Source Delay State" - fonte em estado de atraso.

SWNS - "Source Wait for New ciclo State" - estado de espera da fonte por um novo ciclo.

T - "Talker"- locutor.

T/TE - "Talker or Extended Talker" - locutor ou locutor estendido.

TADS - "Talker Address State" - locutor em estado endereçado.

TACS - "Talker Active State" - locutor em estado ativo.

TCT - "Take Control" - tome controle.

TIDS - "Talker Idle State" - locutor em estado desativado.

TTL - lógica transistor-transistor.

UNL - "Unlisten" - não ouvinte.

UNT - "Untalk" - não locutor.

INTRODUÇÃO

Na automação de sistemas de instrumentação e controle é necessário que os instrumentos possam se comunicar entre si e com o equipamento controlador. Para isso, necessita-se de um barramento padrão e uma interface com um protocolo de comunicação, compatível para todos os instrumentos.

Para que os equipamentos fabricados por diferentes empresas possam se comunicar, torna-se necessário uma padronização desse barramento e da interface com relação às características elétricas, operacionais, funcionais e mecânicas. Um desses padrões, largamente utilizado, é o IEEE-488, que é uma evolução da interface HP-IB, desenvolvida em 1965. Este padrão define as funções de locutor, ouvinte e controlador para cada equipamento interligado com a interface IEEE-488 e usa uma estrutura de barramento de dados bit paralelo e byte serial, assíncrono e geralmente bidirecional para a comunicação com os equipamentos interligados [1],[2].

Instrumentos de medição e controle baseados em microprocessadores foram desenvolvidos, incorporando este padrão de barramento. Neles, as tarefas associadas à medição, controle e à interface eram feitas pelo microprocessador. Com a evolução dos sistemas de instrumentação, as tarefas de medição e controle desenvolvidas em "hardware" foram, cada vez mais, sendo transferidas para serem feitas em "software", sobrecarregando assim o microprocessador.

Surgiram então os circuitos integrados dedicados para executar as tarefas da interface IEEE-488, produzidos por diferentes empresas, com arquitetura voltada para as suas linhas de circuitos integrados, mas respeitando as normas do padrão IEEE-488.

Com o desenvolvimento dos microprocessadores, cada vez mais poderosos e rápidos, e com o surgimento dos microcontroladores, pode-se voltar a se pensar em instrumentos onde todas as tarefas são feitas por um único circuito integrado, o microcontrolador. Nele, os diversos sub-sistemas internos (conversor A/D e D/A, temporizador, comunicação serial, etc) são programados e fornecem dados para a CPU, mas funcionam independente desta, que pode estar fazendo outro processamento qualquer, como as tarefas da interface IEEE-488.

Neste trabalho apresenta-se o projeto e implementação de um módulo, desenvolvido para executar as tarefas associadas a uma interface compatível com o padrão IEEE-488, deixando-se também a possibilidade de se executar as tarefas de medição e controle com equipamentos modestos bem como a de controlador da interface. Um digitalizador e um gerador de funções foram implementados com esse módulo.

Para que este módulo possa responder em tempo real aos comandos originados no controlador de barramento (geralmente residente no microcomputador), foi desenvolvido um executivo em tempo real para supervisionar a operação do microcontrolador e facilitar o desenvolvimento do "software" de aplicação, bem como para facilitar modificações posteriores nas características do módulo para adaptá-lo a novas aplicações.

Este trabalho foi dividido em 5 capítulos: no primeiro é mostrado uma evolução histórica da interface IEEE-488, assim como de suas principais características e de alguns circuitos integrados dedicados; no segundo, mostra-se o "hardware" do módulo desenvolvido; no terceiro, o "software" e no quarto, aplicações desse módulo.

Apresenta-se a seguir uma breve descrição da norma IEEE-488 e as principais características de circuitos integrados dedicados para implementação da interface IEEE-488.

CAPÍTULO I

CARACTERÍSTICAS DO PADRÃO IEEE-488

Apresenta-se neste capítulo a evolução histórica do barramento IEEE-488, suas principais características e alguns circuitos integrados dedicados a implementação deste barramento.

1.1. EVOLUÇÃO

Em 1965 a Hewlett-Packard projetou a interface HP-IB ("Hewlett Packard Interface Bus") para conectar sua linha de instrumentos programáveis, baseados em microprocessadores, a seus computadores. Por causa de sua alta taxa de transferência (até 1 "Mbytes"/segundo), este barramento de interface foi rapidamente ganhando popularidade [2].

Ele continua sendo aperfeiçoado e ganhando novas denominações, tendo uma evolução histórica, até o presente, da seguinte forma:

- em setembro de 1965, a empresa Hewlett Packard idealiza uma interface para interconectar seus instrumentos de laboratório;

- em março de 1972, é formado um comitê para estudar a proposta da HP;

-em setembro de 1974, o IEC ("International Electrical Commission") aprova a proposta de padronização do barramento;

-em abril de 1975, o IEEE publica o padrão IEEE-488;

-em janeiro de 1976, a ANSI ("American National Standard Institute") publica o padrão MC1.1;

-em novembro de 1978, o IEEE revisa o padrão IEEE-488;

-em junho de 1980, é publicado o padrão IEC-625;

-em dezembro de 1981, é publicado o padrão IEEE-728. (códigos e formatos);

-em junho de 1987, o IEEE revisa o padrão IEEE-488 e surge o IEEE-488.1;

-em junho de 1987, é publicado o padrão IEEE-488.2. (códigos, formatos, protocolos e comandos).

Atualmente, este barramento de interface é padronizado mundialmente como:

- 1- IEEE-488.1-1987 (padrão original IEEE);
- 2- ANSI MC1.1 (idêntico);
- 3- IEC 625-1 (idêntico ao padrão original exceto pelo conector);
- 4- B.S. 6146 (padrão Inglês, idêntico ao IEC-625);
- 5- IEEE-488.2-1987 (códigos, formatos, protocolos e comandos comuns).

Esta interface é internacionalmente conhecida como [1],[3]:

- 1- HP-IB - "Hewlett Packard Interface Bus";
- 2- GPIB - "General Purpose Interface Bus";

3- barramento IEEE-488;

4- IEC -625.

As características elétricas, operacionais, funcionais e mecânicas do barramento de interface são definidas pelo padrão IEEE-488. Apresenta-se a seguir, as principais características deste padrão, em especial as que foram objeto de implementação no módulo desenvolvido.

I.2. PRINCIPAIS CARACTERÍSTICAS DO BARRAMENTO IEEE-488

Para organizar e controlar o fluxo de informações entre os instrumentos interligados a um barramento IEEE-488 e dependendo de sua natureza operacional, cada instrumento pode ter até 3 funções, mas só pode usar uma delas a cada momento (figura 1.1) [1]-[8]. Estas funções são as seguintes:

1- o instrumento que está apto a receber mensagens ou comandos de outros instrumentos conectados no barramento [1]-[8] é chamado comumente de **ouvinte**, exemplos: impressoras, fontes programáveis de tensão, geradores de sinal, etc;

2- o instrumento que está apto a enviar mensagens ou comandos para outros instrumentos conectados no barramento [1]-[8] é chamado comumente de **locutor**, exemplos: gravadores, voltímetros, contadores, etc, e;

3- o instrumento que tem a capacidade de endereçar outros instrumentos e programá-los como ouvintes ou locutor (incluindo ele mesmo), ou enviar mensagens para gerar ações específicas [1]-[8] é comumente chamado de **controlador**. Em geral, um controlador é uma placa inserida em um "slot" de um microcomputador. Entretanto, qualquer equipamento ligado ao barramento pode ter a função de controle dentro de sua interface IEEE - 488.

A comunicação entre os instrumentos no barramento IEEE-488 é organizada a partir de regras bem definidas:

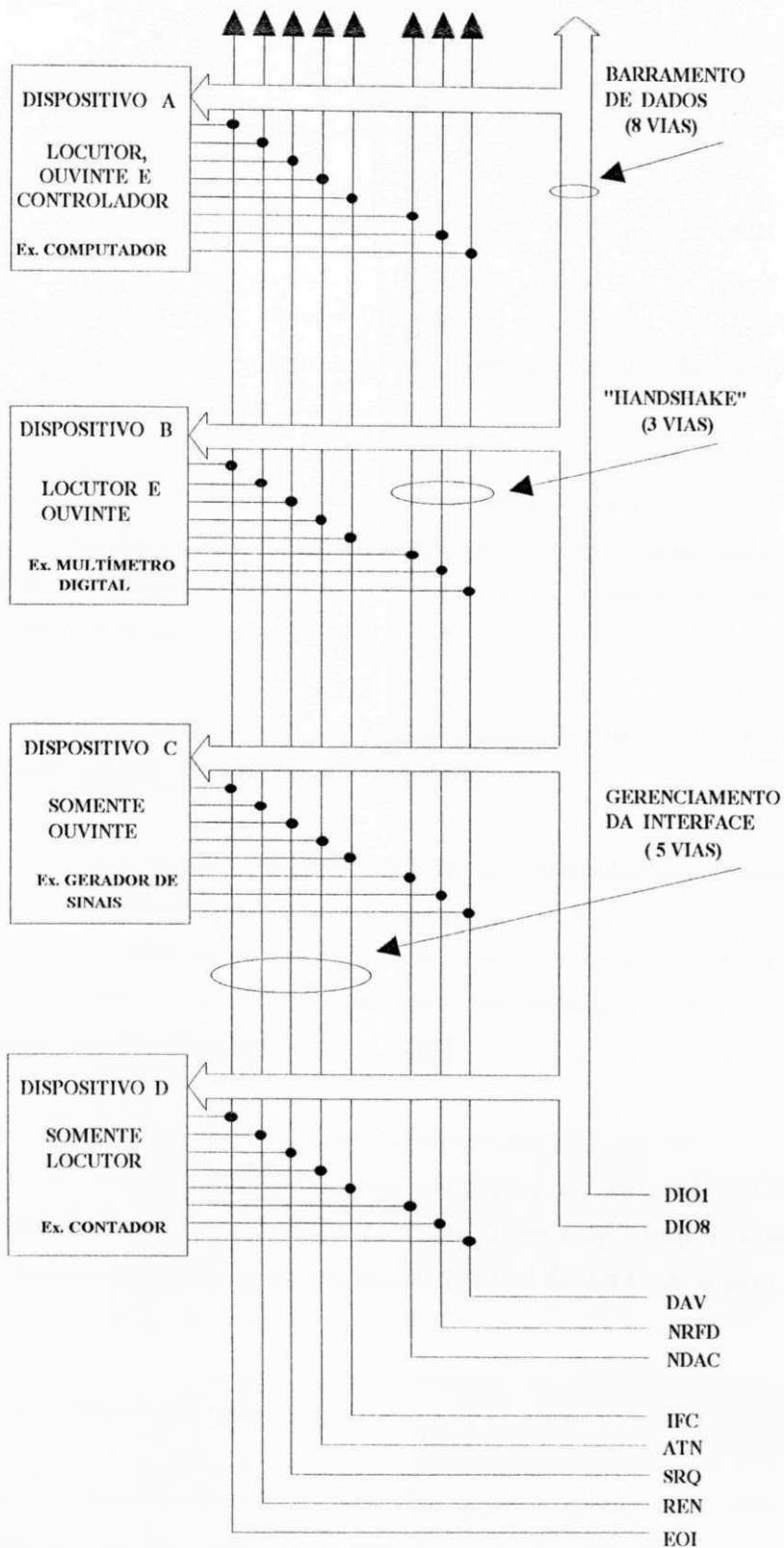


Figura 1.1: Barramento IEEE-488.

1- somente um instrumento, em determinado instante, pode ser o locutor (transmitir dados ou comandos pelo barramento), enquanto vários instrumentos podem ouvir (receber comandos ou dados) simultaneamente [3],[4];

2- no barramento deve existir apenas um controlador ativo por vez [3],[4];

3- a taxa de transmissão de uma informação é automaticamente adaptada para a velocidade do instrumento cujo processamento de informação é mais lento [3],[4];

4- desde que um controlador passe informações (tal como instruções de programação que dependem do instrumento) regularmente para o barramento, ele deve ter uma função de locutor em adição a sua função de controlador. Geralmente, ele também tem uma função de ouvinte, para receber informações de outros instrumentos ligados ao barramento para ele mesmo [3],[4];

5- o controlador é o único instrumento que pode ativar as vias IFC ("interface clear") e REN ("remote enable") [3],[4];

6- é possível passar o controle do sistema de um controlador para outro [3],[4];

7- cada dispositivo tem seu próprio endereço e antes da transferência dos dados ser iniciada entre dois instrumentos determinados, ambos têm que ser endereçados e programados pelo controlador [3],[4];

8- um sistema sem controlador é também possível. Tal sistema pode consistir de um locutor e um ou mais ouvintes. Isto deve ser feito pela colocação destes instrumentos pelo controle local para "talk only" ou "listen only". Depois deste endereçamento manual, a transferência dos dados é somente possível de um locutor para os ouvintes [3],[4].

1.2.1. Vias do Barramento

O barramento utiliza 24 vias, entre as quais, 8 são vias de aterramento e 16 são vias da interface, com lógica negativa, e níveis TTL (figura 1.2). Estas 16 vias podem ser subdivididas em três grupos, cada uma com diferentes funções [1]-[7],[9], como mostra a figura 1.1:

- 8 vias usadas para transferência de dados [1]-[9], denominadas como DIO1...DIO8;

- 3 vias usadas para controle da transferência dos "bytes" de dados [1]-[9], denominadas como DAV, NRFD, NDAC (vias de "handshake");

- 5 vias usadas para gerenciamento geral das mensagens da interface [1]-[9], denominadas como ATN, IFC, SRQ, REN, EOI.

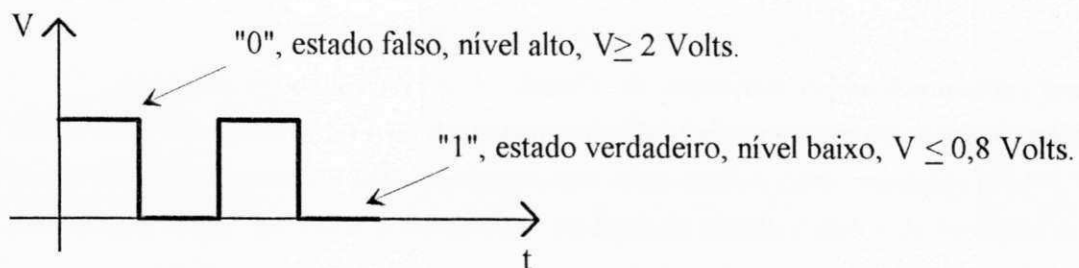


Figura 1.2: Níveis lógicos das 16 vias do barramento da interface.

I.2.1.1. Barramento de Dados

As 8 vias de entrada/saída (DIO1...DIO8) são usadas para enviar "bytes" de dados, endereços, instruções de programação, "bytes" de estado e comandos especiais do barramento [1]-[4].

Um "byte" de dados consiste de 8 bits paralelos, que são transferidos sobre o barramento como "byte" serial e bit paralelo, bidirecional e assíncrono. Uma mensagem completa pode consistir de um ou mais "bytes" em série com taxa máxima de transferência de 1 "Mbyte"/s [1]-[4].

Os termos bit paralelo e "byte" serial são expressões que indicam como as mensagens se apresentam, nas vias de dados, no processo de comunicação. O locutor coloca simultaneamente, nas vias de dados, 8 bits que formam o "byte" (bit paralelo). Tão logo seja confirmada a leitura do "byte" por todos os ouvintes, o locutor o substitui pelo próximo "byte" e o ciclo se repete enquanto houver "bytes" a serem transmitidos ("byte" serial).

A comunicação é assíncrona porque cada instrumento ouvinte ligado ao barramento pode ter velocidade de leitura diferente de um outro ouvinte e também da velocidade de escrita do locutor. Para possibilitar a comunicação em tal ambiente, o IEEE-488 provê um mecanismo de gerenciamento de transferência de dados chamado "handshake" a três fios.

No barramento IEEE - 488, pode-se transmitir dados (modo de dados) ou comandos (modo de comandos). Uma das vias de gerenciamento geral da interface (ATN, "Attention") é usada para selecionar um desses modos [1]-[4]:

-no modo de dados (ATN = "falso"), as instruções de programação ("byte" de dados) são transmitidas através do barramento de dados por um dispositivo endereçado como locutor para um ou mais equipamentos endereçados como ouvintes [1]-[4]. Como exemplo de "byte" de dados pode-se citar: códigos de estado e dados de medição, e;

-no modo de comandos (ATN = "verdadeiro"), o barramento de dados é usado para endereçar o instrumento como locutor ou ouvinte e para transmitir comandos especiais do barramento para programar um ou mais equipamentos [1]-[4]. Todas estas mensagens são enviadas pelo dispositivo controlador.

Geralmente, o código ASCII é usado para dados e mensagens. Este código cobre praticamente todas as figuras, letras e sinais necessários à comunicação. O oitavo bit é disponível para verificação de paridade, se necessário [1]-[4].

Cada transferência de um "byte" de dados sobre o barramento é sincronizada pela ação de três vias para controlar a transferência, como é mostrado a seguir.

As vias DIO1...DIO8 devem usar saídas em coletor aberto ou três estados.

1.2.1.2. Vias de "Handshake"

A transferência de mensagens é sincronizada pela função de "handshake" (aperto de mão) ou controle de transferência dos "bytes" de dados. Cada "byte" enviado através das oito vias do barramento da interface está no modo de dados ou no modo de comandos

e é acompanhado pela ação de três vias de "handshake" denominadas: DAV, NRFD e NDAC [1]-[9].

Durante a transferência do "byte", o acionamento da via DAV na interface IEEE-488 é realizado pelo instrumento locutor e o acionamento das vias NRFD e NDAC é realizado pelo instrumento ouvinte. A função associada a cada uma das vias de "handshake" é descrita a seguir [1]-[4].

NRFD ("Not Ready for Data" - Não pronto para dados) - indica quando um equipamento está ou não pronto para receber um "byte" de mensagem. Esta via é acionada por todos os instrumentos ouvintes, quando estão prontos para receber comandos e mensagens de dados. Esta via é monitorada pelo locutor [1]-[4].

NDAC ("Data Not Accepted" - Dado não aceito) - indica quando um equipamento aceitou ou não um "byte" de mensagem. Esta via é acionada por todos os equipamentos ouvintes quando aceitam comandos e mensagens e é monitorada pelo locutor [1]-[4].

DAV ("Data Valid" - Dados válidos) - quando ativa (nível baixo), indica que o "byte" presente nas vias DIO1-DIO8 é válido. O controlador ou locutor aciona a via DAV quando envia comandos, e o locutor aciona a via DAV quando envia mensagens ou dados. Esta via é monitorada pelo ouvinte [1]-[4].

É importante que um locutor não envie uma mensagem quando um ouvinte endereçado não possa recebê-lo (NRFD = nível baixo). Contudo, um locutor não pode operar mais rápido do que o mais lento dos ouvintes endereçados. Isto é garantido pelas vias NRFD e NDAC dos instrumentos ouvintes que estão conectadas numa configuração "AND" com coletor aberto. A mensagem sobre o barramento pode portanto ser alta (estado falso) somente se todas as saídas dos instrumentos conectadas àquelas vias estão em nível alto. Se apenas um dos instrumentos não está pronto para aceitar dados, então NRFD = nível baixo ("verdadeiro"). Igualmente, NDAC = nível baixo ("verdadeiro") se apenas um dos instrumentos ainda não aceitou os dados [1],[3]. A lógica equivalente para as vias NRFD e NDAC, do barramento, pode ser representada como:

$$\text{NRFD} = \text{nrfd}(1) \cdot \text{nrfd}(2) \cdot \text{nrfd}(3) \cdot \text{nrfd}(4) \cdot \dots \cdot \text{nrfd}(n) \quad (1)$$

e,
$$\text{NDAC} = \text{ndac}(1) \cdot \text{ndac}(2) \cdot \text{ndac}(3) \cdot \text{ndac}(4) \cdot \dots \cdot \text{ndac}(n) \quad (2)$$

onde nrfd(n) e ndac(n) são os estados das vias nrfd e ndac, respectivamente, do instrumento n.

As vias NRFD e NDAC devem usar saídas em coletor aberto e a via DAV deve usar saída em coletor aberto ou três estados.

Deve-se observar a nomenclatura utilizada para as vias NRFD e NDAC:

1- NRFD falso é idêntico a RFD verdadeiro, e;

2- NDAC falso é idêntico a DAC verdadeiro.

1.2.1.3. Vias de Gerenciamento Geral

Cinco vias, cada uma contendo uma função específica, são usadas para controlar o fluxo ordenado de informação entre o controlador e os outros instrumentos, como mostrado na figura 1.1:

-**ATN** ("Attention" - Atenção) - esta via é usada pelo controlador para selecionar o modo de dados (ATN = "falso") ou modo de comandos (ATN= "verdadeiro") [1]-[4];

-**REN** ("Remoto Enable" - Equipamento em modo remoto) - esta via é ativada (REN = "verdadeiro") somente pelo controlador do sistema, para mudar o controle do instrumento endereçado, de local para remoto, através da interface. Esse instrumento endereçado como ouvinte ou locutor pode voltar para o modo local por um comando "return to local" (retorno para local) enviado pelo controlador, porém, este comando não afeta o estado da via REN [1]-[4];

-**SRQ** ("Service Request" - Solicitação de serviço) - esta via é ativada por um ou mais instrumentos para solicitar atenção do controlador pelo envio (SRQ = "verdadeiro"), o controlador pode então interromper a seqüência de eventos que estava executando para atender ao equipamento que enviou SRQ. O controlador deve então realizar uma seqüência de "poll" serial dos instrumentos para determinar quem solicitou atenção, (anexo A) [1]-[4],[7], e;

-EOI ("End or Identify" - Fim ou identifique) - esta via tem duas funções que dependem da linha ATN. No modo de dados (ATN = "falso"), um locutor pode usar esta via para indicar o fim de uma transferência de vários "bytes" (EOI = "verdadeiro"). No modo de comandos (ATN = "verdadeiro") o controlador usa esta via (EOI = "verdadeiro") para executar uma sequência de "poll" paralelo [1]-[4],[7], (anexo A).

-IFC ("Interface Clear" - Inicialização da interface) - esta via é usada pelo controlador para inicializar todos os instrumentos ligados ao barramento (IFC = "verdadeiro"). Todos os locutores e ouvintes são desativados após a linha IFC ter sido ativada. Se existir mais de um controlador dentro do sistema, o controle do barramento é dado de volta ao instrumento (controlador do sistema) que gerou a mensagem IFC [1]-[4];

A via SRQ deve usar saída em coletor aberto e as vias REN, EOI, IFC e ATN devem usar saídas em coletor aberto ou três estados.

1.2.2. Classificação das Mensagens do Barramento

A comunicação entre instrumentos interligados por um barramento IEEE-488 é feita a partir de mensagens. As vias de controle são usadas para iniciar esta comunicação, preparando os equipamentos para receber mensagens codificadas e enviadas sobre as vias de dados. Essas mensagens podem ser classificadas em:

1- mensagens de endereços - quando o controlador quer ativar um determinado equipamento como locutor ou ouvinte, ele ativa a via ATN e envia o endereço do equipamento pelas vias de dados [1]-[4]. A tabela dos endereços válidos esta listada no anexo B;

2- comandos universais - fazem com que todos os instrumentos ligados ao barramento executem com a ação solicitada. Estes comandos, apresentados com mais detalhes no anexo C, são: DCL, LLO, SPE, SPD e PPU [1]-[4];

3- comandos de endereçamento - apresentados no anexo D, estes comandos (SDC, GTL, GET, TCT e PPC) são idênticos aos comandos universais, porém atuam somente no instrumento endereçado [1]-[4];

4- comandos secundários - são usados para selecionar partes de um certo instrumento ou um instrumento de um grupo com o mesmo endereço [1]-[4]. Mais detalhes desses comandos estão apresentados no anexo E;

5- comandos de desativação - são usados para desabilitar instrumentos endereçados como locutor ou ouvinte (ver anexo F) [1]-[4], e;

6- comandos específicos do instrumento - são mensagens que devem ser enviadas pelo controlador para solicitar tarefas específicas do instrumento endereçado. Os códigos relativos a essas mensagens são definidas pelos fabricantes dos instrumentos em questão [1],[3],[4].

1.2.3. Especificações Mecânicas e Elétricas da Interface IEEE-488

A norma IEEE-488 define também especificações mecânicas e elétricas dos cabos e sinais da interface:

1- o comprimento máximo acumulado por cabo é de 20 metros [1]-[7];

2- o comprimento máximo do cabo entre dois equipamentos é de 2 metros [1]-[7];

3- o número máximo de instrumentos ligados ao barramento é 15 (incluindo o controlador) [1]-[7];

4- os sinais operam em níveis TTL com lógica negativa, o "1" lógico corresponde ao estado verdadeiro, nível baixo e, $V \leq 0,8$ volts e o "0" lógico corresponde ao estado falso, nível alto e, $V \geq 2,0$ volts [1]-[7], e;

5- a velocidade máxima de transferência de dados e comandos entre instrumentos é de 1 Mb/sec. (distâncias limites de 1 a 10 metros) [1]-[7].

1.2.4. Exemplo do Funcionamento das Vias de "Handshake"

Um exemplo do funcionamento das vias de "handshake" em uma transferência de dados ou comandos é mostrado nas figuras 1.3 (barramento IEEE-488), 1.4 (diagrama de tempos) e 1.5 (fluxograma). Três equipamentos estão presentes no barramento e são endereçados, um como locutor e dois como ouvintes. Alguns "bytes" de dados são enviados do locutor para os ouvintes (no modo de dados).

Nas figuras 1.3 e 1.4, os sinais de "handshake" do primeiro ouvinte foram chamados de nrfd(1) e ndac(1) e do segundo ouvinte de nrfd(2) e ndac(2). Nos tempos $t_0 - t_{10}$ dos vários eventos, indicados nas figura 1.4 e 1.5, são feitas as seguintes ações:

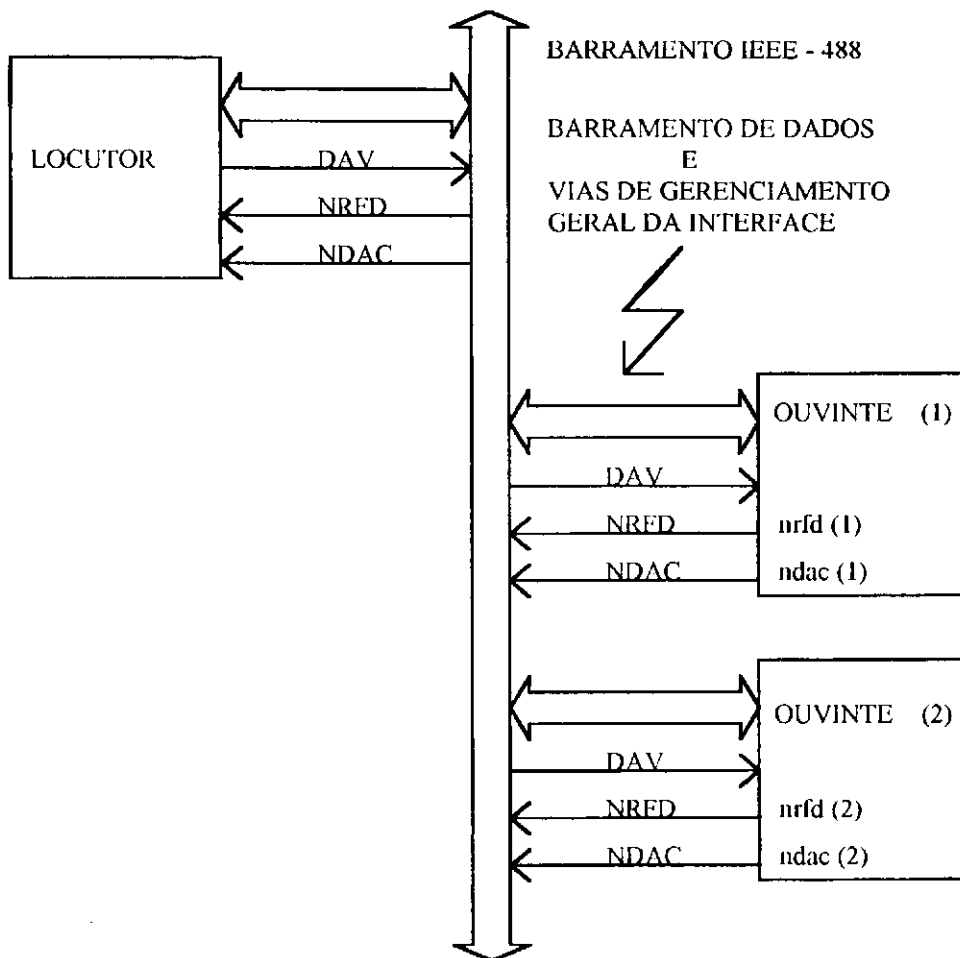


Figura 1.3: Barramento com um locutor e dois ouvintes.

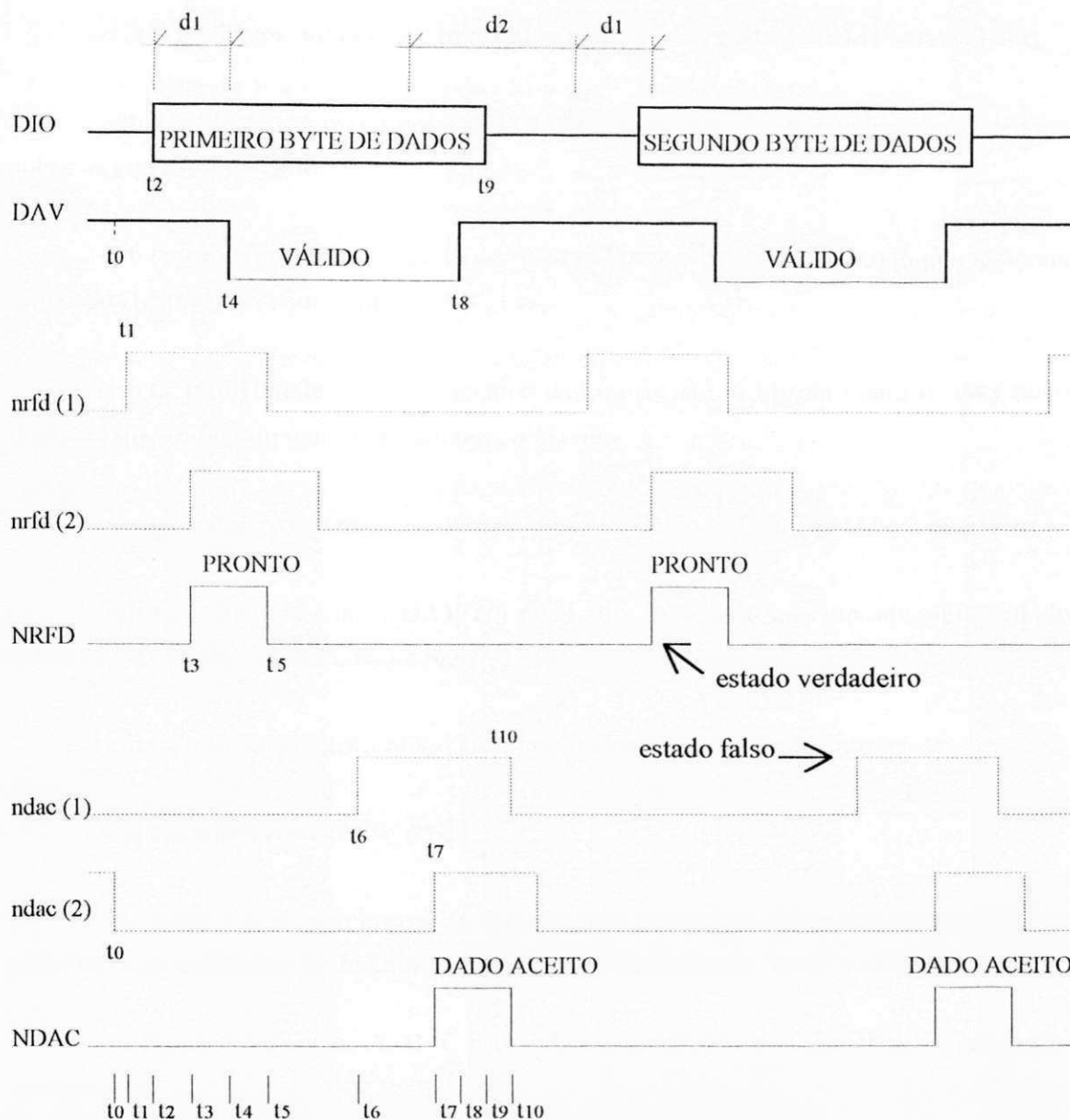


Figura 1.4: Diagrama de tempos dos sinais de "handshake".

-em t_0 , o locutor ativa DAV = nível alto (dado não válido). Os ouvintes ativam NRFD = nível baixo (não pronto para dados) e NDAC = nível baixo (dado não aceito);

-em t_1 , o primeiro ouvinte está pronto para aceitar o "byte" de dados;

-em t_2 , o locutor coloca um "byte" de dados nas vias DIO, com DAV em nível alto;

- em t_3 , todos os ouvintes estão prontos para receber dados; NRFD em nível alto;
- em t_4 , o locutor reage colocando DAV "verdadeiro" para indicar que o dado sobre as vias DIO é válido;
- em t_5 , o ouvinte sae do estado de "pronto" para receber dados, enquanto esperam que todos respondam com "dado aceito";
- em t_6 , o ouvinte mais rápido aceita o dado, mas não fica pronto para receber novo dado até que o ouvinte mais lento também o aceite;
- em t_7 , todos os ouvintes aceitam o "byte" de dados e a via NDAC em nível alto;
- em t_8 , o locutor coloca DAV em nível alto, indicando aos ouvintes que o dado sobre as vias DIO1 ... DIO8, não é mais válido;
- em t_9 , o locutor coloca novo "byte" de dados sobre o barramento, e;
- em t_{10} , um novo ciclo se repete.

Na figura 1.4, d_1 é o atraso do locutor para permitir o ajuste dos dados sobre as vias DIO e d_2 é o atraso do locutor para permitir a mudança do "byte" sobre as vias DIO.

Na figura 1.5 as setas A, B, C e D indicam condições necessárias à comunicação, quais sejam:

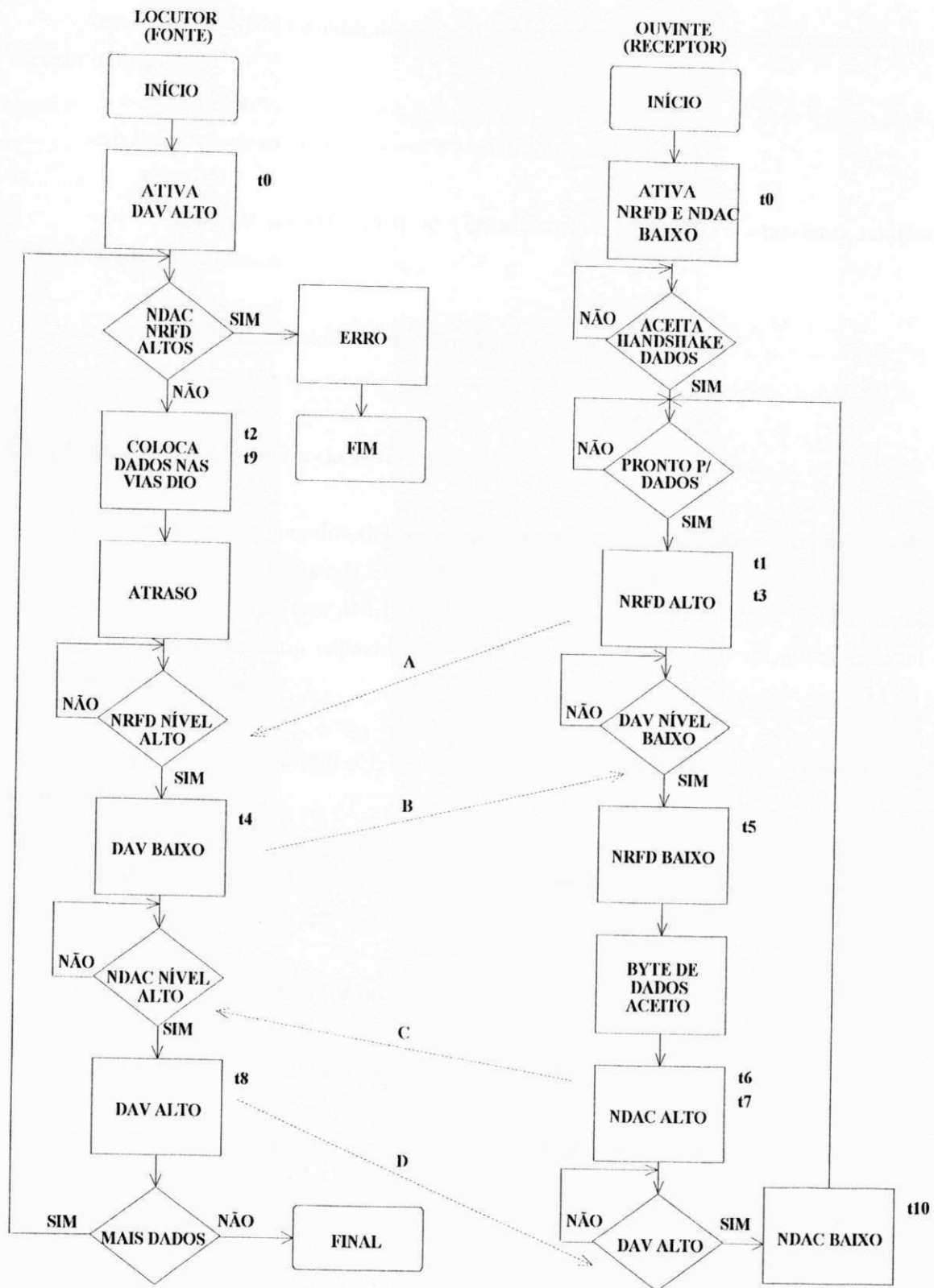


Figura 1.5: Diagrama de fluxo mostrando o "handshake" entre o locutor e os ouvintes.

-em A, a via NRFD é falsa somente quando todos os ouvintes estão prontos para receber dados;

-em B, o dado é válido e pode ser transferido;

-em C, o estado da via NDAC é verdadeiro até que todos os ouvintes tenham aceito o "byte" de dados, e;

-em D, o dado não é mais válido, depois do tempo t_g .

I.2.5. Repertório de Funções da Interface

As operações e comandos descritos até o momento, representam o repertório de funções permitidas pela interface IEEE-488. Nem todas as funções estão presentes em um dado instrumento com interface IEEE-488. Esta norma define 10 funções, apresentadas no anexo G, que englobam a capacidade de um instrumento executar ou ativar este ou aquele comando, a saber:

1- SH ("Source Handshake"), controle do locutor para sincronizar a transferência de comandos e dados;

2- AH ("Acceptor Handshake"), controle dos ouvintes para sincronizar a recepção de dados e comandos;

3- T/TE ("Talker"), locutor ou locutor estendido;

4- L/LE ("Listen"), ouvinte ou ouvinte estendido;

5- SR ("Service Request"), possibilidade de solicitar serviço ao controlador;

6- RL ("Remote/Local");

7- PP ("Parallel Poll");

8- DC ("Device Clear");

9- DT ("Device Trigger"), e;

10- C ("Controller") [1],[3],[7],[10].

1.2.6. Circuitos Integrados Dedicados para Interface IEEE-488

Circuitos integrados dedicados foram desenvolvidos, por diferentes fabricantes, incorporando este padrão de interface, com arquiteturas voltadas para as suas linhas de microprocessadores. Esses circuitos dedicados vieram liberar o microprocessador para executar outras tarefas, já que eles se encarregaram das funções da interface. Entretanto, de um fabricante para outro, estes CI's apresentam características peculiares às suas aplicações, como mostra a Tabela I.1 [3],[4],[7],[9],[11]-[15].

Com a fabricação dos CI's para controle da interface, observou-se a necessidade de fabricar acopladores bidirecionais ("transceivers") com o objetivo de conectar os pinos de entrada/saída dos CI's dedicados ao barramento IEEE - 488, para controle da direção das linhas: barramento de dados e gerenciamento geral da interface; e manter as características elétricas do padrão original (Tabela I.2) [3],[4],[7],[10]-[18].

Tabela I.1. CI's dedicados a interface IEEE-488.

FABRICANTE	CI's	FUNÇÕES QUE PODEM EXECUTAR
MOTOROLA	MC68488	SH,AH,L/LE,T/TE,RL,DC,DT,PP,SR
TEXAS INSTRUMENTS	TMS 9914A	SH,AH,L/LE,T/TE,RL,DC,DT,PP,SR,C
INTEL	8291/8292	SH,AH,L/LE,T/TE,RL,DC,DT,PP,SR,C
NEC	7210 TLC	SH,AH,L/LE,T/TE,RL,DC,DT,PP,SR,C
RTC	HEF 4738	SH,AH,L/LE,T/TE,RL,DC,DT,PP,SR
FAIRCHILD	96LS488	SH,AH,L/LE,T/TE,RL,DC,DT,PP,SR

Tabela I.2. CI's dedicados para controle das vias do barramento de dados e de gerenciamento geral, onde: 1- barramento de dados; 2- vias de gerenciamento geral (sem função de controlador), e; 3- vias de gerenciamento geral (com função de controlador).

FABRICANTES	CI's	UTILIZAÇÃO	VIAS DE I/O
TEXAS INSTRUMENTS	SN75160A	1	8
	SN75161A	2	8
	SN75162A	3	8
MOTOROLA	MC3440A	1,2	4
	MC3441A	1,2	4
	MC3443A	1,2	4
	MC3446	1,2	4
	MC3447	1,2	8
	MC3448	1,2	4
NATIONAL	DS75160A	1	8
	DS75161A	2	8
	DS75162A	3	8
	DS3666	1,2,3	4

No projeto dos CI's dedicados, todas as tarefas da interface relacionadas no padrão foram consideradas para implementação, considerando os diagramas de estados do anexo G. Normalmente, estes CI's são usados em conjunto com um microprocessador para gerenciamento das tarefas do protocolo IEEE - 488, com o objetivo de evitar sobrecarga a este último. Desta forma, com a utilização de interrupções, o microprocessador pode responder e atender mais rápido aos comandos da interface e o barramento não necessita ser continuamente monitorado pelo microprocessador [11].

A comunicação entre o microprocessador e o CI dedicado é feita a partir de registradores mapeados em memória do microprocessador, entre os quais alguns são de leitura e outros de escrita. O número exato desses registradores depende de cada circuito dedicado. Por exemplo, no TMS9914A existem 13 registradores dos quais 6 são de leitura e 7 de escrita. No MC68488 existem 15 registradores dos quais 8 são de leitura e 7 de escrita. Geralmente, os registradores dos CI's dedicados são endereçados de acordo com os três bits menos significativos da via de endereços do microprocessador [4],[7],[11].

Um instrumento com interface IEEE-488, têm um endereço específico que pode ser modificado por chaves externas. Na maioria dos casos, estas chaves são conectadas a registradores internos dos CI's dedicados através das vias de dados em "buffers" de 3 estados. Isto permite ao microprocessador ler seu endereço no registrador de endereço

dos CI's dedicados para identificá-lo entre os instrumentos ligados ao barramento. Os CI's dedicados geram uma interrupção "My Adress" (MA) e entram em um estado de endereçamento quando seu endereço é detectado sobre o barramento de dados da interface [11].

As ações de gerenciamento entre o microprocessador e os CI's específicos, relacionados a norma IEEE-488, estão na forma de "software" em EEPROM, ROM ou RAM ou em "hardware" (registradores internos tanto no microprocessador como dos CI's dedicados, interrupções RTI e IRQ, decodificadores, etc) [11].

Apresenta-se no próximo capítulo implementação de um módulo de interface com o padrão IEEE-488, baseado no microcontrolador MC68HC11.

CAPÍTULO II

MÓDULO DE INTERFACE IEEE-488 BASEADO NO MICROCONTROLADOR MC68HC11

Neste capítulo, apresenta-se o projeto e implementação de uma placa, baseada no microcontrolador MC68HC11, desenvolvida para executar as tarefas associadas a um interface compatível com o padrão IEEE-488, deixando-se também a possibilidade de se executar as tarefas de medição e controle com equipamentos modestos, bem como a de controlador da interface.

II.1. MÓDULO DE INTERFACE PARA O PADRÃO IEEE-488

Os instrumentos com o padrão IEEE-488 sofreram uma evolução, onde o microprocessador que realizava as tarefas associadas à medição, controle do processo e do barramento foi sendo cada vez mais sobrecarregado com novas tarefas. Para diminuir a sobrecarga de tarefas dos microprocessadores, foram desenvolvidos, na década de setenta, os circuitos integrados dedicados à interface IEEE-488.

Os microprocessadores também tiveram uma evolução bastante acentuada, surgindo os microcontroladores, que incorporam diversos sub-sistemas internos (conversor A/D e D/A, temporizador, comunicação serial, etc) programados pela CPU, mas que funcionam independentemente desta. Para tornar os instrumentos mais compactos, pode-se voltar a se fazer todas as tarefas por um único circuito integrado: o microcontrolador.

II.2. REQUISITOS TÉCNICOS DO MÓDULO DE INTERFACE

O módulo de interface deve ter linhas compatíveis com o padrão IEEE-488 e permitir adaptações futuras, especialmente em relação as funções específicas dos instrumentos dos quais fará parte. Ele deverá ter as funções de locutor e de ouvinte, com a possibilidade de se implementar a função de controlador, com mudanças mínimas de "hardware" e "software".

II.3. MICROCONTROLADOR MC68HC11

O módulo apresentado neste trabalho (figura 2.1), para executar as tarefas da interface IEEE-488, é baseado no microcontrolador MC68HC11E2 de 8 bits (anexo II), que usa tecnologia HCMOS [19], e tem como principais características:

- 1- dois "kbytes" de EEPROM, localizados inicialmente entre os endereços \$F800 e \$FFFF, mas podem ser relocados para outros endereços;
- 2- um temporizador de 16 bits, com três entradas de captura, quatro saídas de comparação e uma entrada/saída associada a um acumulador de pulsos;
- 3- um conversor A/D por aproximação sucessiva de 8 bits, oito canais multiplexados e "sample hold". Uma conversão se completa após 32 ciclos de relógio que corresponde a 16 microsegundos para uma frequência de 2 MHz;
- 4- 256 "bytes" de RAM, localizados entre os endereços \$0000 e \$00FF, correspondendo à página zero;
- 5- diversas interrupções por "hardware" e por "software", entre elas: \overline{IRQ} , \overline{XIRQ} e RTI - "Real Time Interrupt";
- 6- CPU de 8 bits com 64 registradores internos, além dos acumuladores usuais: A, B, X, Y, SP, PC e CCR;

7- um sub-sistema para comunicação serial assíncrona e outra síncrona, e;

8- cinco "ports" (A, B, C, D e E) de propósito geral, que podem também ter funções específicas associados a um dos sub-sistemas do microcontrolador. O "port" A, (3 vias de entrada, 4 vias de saída e 1 de entrada/saída), é conectado ao temporizador, o "port" D a comunicação serial, o "port" E (8 vias) ao conversor A/D e os "ports" B (8 vias) e C (8 vias) para endereços e dados [19].

O microcontrolador pode operar em 4 modos distintos escolhidos pela programação dos pinos MODA e MODB. No módulo desenvolvido, ele opera no modo multiplexado expandido e uma PRU (MC68HC24) é usada para restaurar os "ports" A e B do microcontrolador e simular o modo "single chip" [19].

Como explicado no capítulo 1, um equipamento com interface IEEE-488 necessita de 16 vias bidirecionais, se ele tem as funções de controlador, locutor e ouvinte [1]-[4]. Uma PIA MC6821 ("Peripheral Interface Adapter"), ligada ao barramento de dados do microcontrolador através do decodificador de endereços da placa de desenvolvimento, fornece essas 16 vias (figura 2.1). A configuração funcional dessa PIA é programada pelo microcontrolador, de modo que cada uma das 16 vias ligadas ao barramento podem ser programadas como entrada ou saída [20].

A PIA MC6821 tem as seguintes características:

1- barramento de dados bidirecional (D0-D7), que permite a transferência de dados entre o microprocessador e a PIA;

2- três pinos "chip select" (CS0, CS1, e CS2), que são pinos de entrada, usados para selecioná-la ou não;

3- pino de leitura/escrita, permitindo ao microcontrolador o controle da direção da transferência dos dados sobre o barramento;

4- pino de "reset", usado para inicializar todos os registros internos;

5- registradores RS0 e RS1, usados para seleccionar os seus registradores internos e, em conjunto com os registradores de controle interno, para seleccionar um registrador específico para leitura ou escrita, e;

6- dois registradores de controle: A e B; dois registradores de direção de dados: A e B e dois registradores bidirecionais para interligação com os periféricos A e B [20].

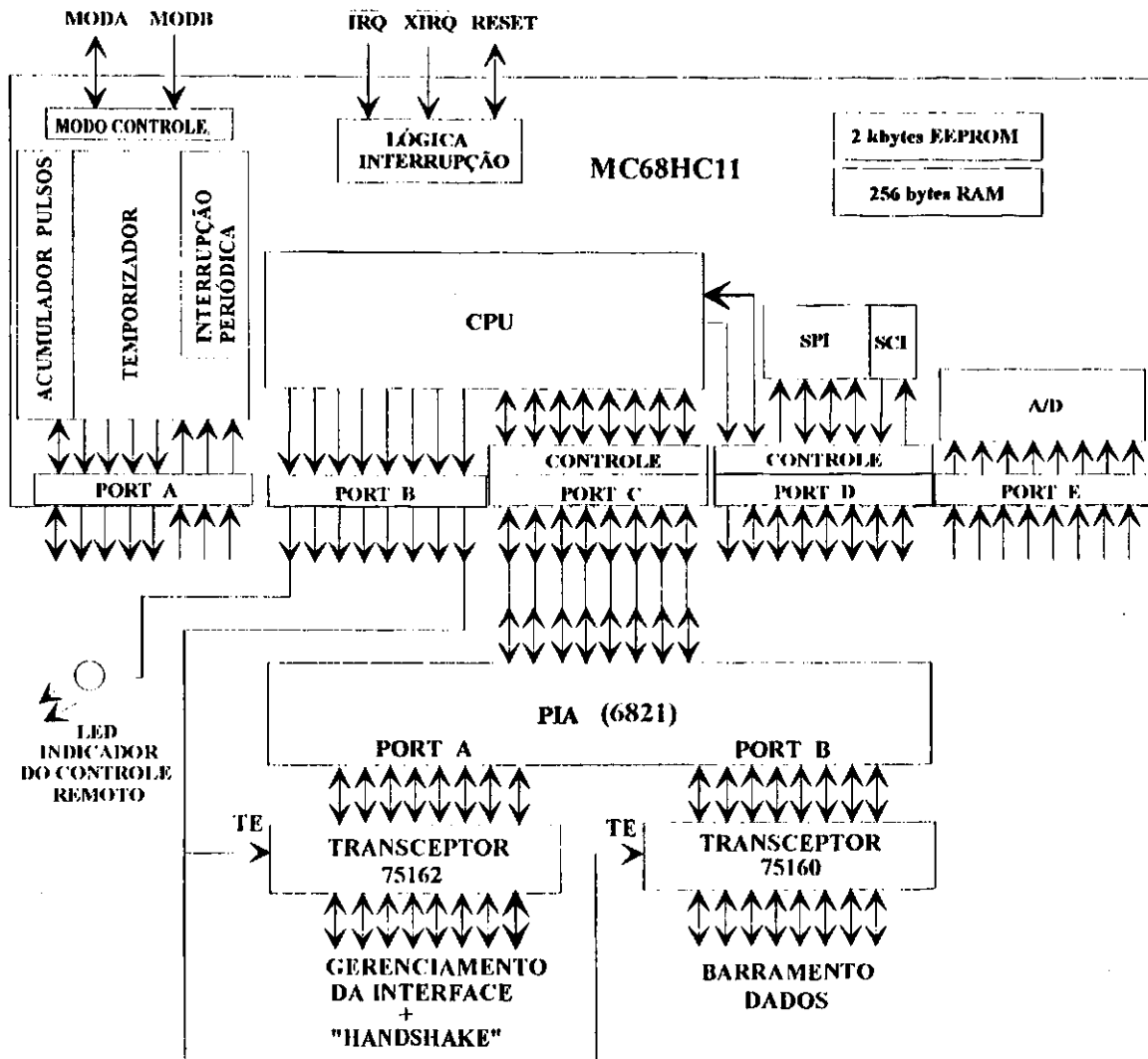


Figura 2.1: Diagrama do módulo para interface IEEE-488 baseado no 68HC11.

Desta forma, os pinos de entrada/saída da PIA são programados pelo microcontrolador e cada uma das 16 vias, conectadas ao barramento, pode ser programada para atuar como entrada ou saída. Os endereços \$6008 e \$6009 foram

usados, respectivamente, para dados do "port" A e para seu registrador de direção e controle. Os endereços \$600A e \$600B, de forma similar, para o "port" B.

O barramento IEEE-488, em operação normal, usa "drivers" de 48 mA, do tipo coletor aberto ou de 3 estados. Entretanto, a PIA não apresenta características elétricas compatíveis com a norma IEEE-488, sendo necessário, portanto, o uso de transceptores tipo SN75160 e SN75162 para manter estas características (figura 2.1) [11],[18].

O SN75160 é usado para implementar as 8 linhas do barramento de dados. A direção do fluxo de dados no barramento é controlada pela entrada TE - "talk enable" (figura 2.1). Desta forma, os oito canais estão simultaneamente no modo de recepção, quando os dados são transferidos do barramento para a PIA (TE em nível baixo); ou no modo de transmissão, quando os dados são transferidos da PIA para o barramento (TE em nível alto) [11],[18].

O SN75162 é um dispositivo usado como "buffer" das vias de gerenciamento e quando combinado com o SN75160 fornece uma configuração completa de 16 vias para o barramento IEEE-488. O SN75162 apresenta configuração interna para aplicações que incluem controlador, locutor e ouvinte. A direção das vias do SN75162 tanto para transmissão como recepção é determinada pelos pinos de habilitação DC - "Direction Control", TE - "talk enable" e SC - "System Controle". O pino TE do SN75162, que controla as linhas de handshake NRFD, NDAC e DAV, foi conectado ao pino TE do SN75160 e levado ao pino 0 do "port" B do microcontrolador. Quando TE está em nível baixo, NDAC e NRFD estão configuradas para transmissão e DAV para recepção. Quando TE está em nível alto, NDAC e NRFD estão configurados para recepção e DAV para transmissão [11],[18].

Na figura 2.1, um LED é ligado ao pino 7 do "port" B para indicar o estado remoto/local do instrumento ligado com este módulo de interface IEEE-488.

A PIA, os transceptores, o LED e o conector com padrão IEEE-488 foram montados em "wire-wrap", e ligados ao módulo de interface.

No capítulo III, apresenta-se o "software" desenvolvido para o módulo com interface IEEE-488.

CAPÍTULO III

EXECUTIVO EM TEMPO REAL DE GERENCIAMENTO

Neste capítulo, apresenta-se o "software" desenvolvido para gerenciar o barramento IEEE-488, adaptado de um executivo em tempo real.

III.1. EXECUTIVO EM TEMPO REAL

Os executivos em tempo real ou sistemas operacionais em tempo real, são sistemas supervisórios desenvolvidos para serem aplicados em computadores, microcontroladores, com o objetivo de maximizar o fator de utilização desses equipamentos, e garantir a execução de várias tarefas [21].

O principal componente de um executivo em tempo real é o escalonador das tarefas que, geralmente, são escalonadas e acionadas de acordo com uma lista de prioridades, predefinidas por um conjunto de condições temporais para cada tarefa. O escalonador necessita de um relógio em tempo real para cadenciar o escalonamento das tarefas periódicas ou críticas no tempo. O executivo em tempo real deve:

1- controlar interrupções internas ou externas a unidade de processamento central (CPU);

2- possuir estrutura modular para aceitar a inclusão ou exclusão de tarefas;

3- dispor de uma interface apropriada com o processo externo e com o operador, e;

4- introduzir sobrecarga mínima pela execução do escalonador e maximizar a utilização do microprocessador [21].

O executivo utiliza interrupções para cadenciar a ativação e a desativação das várias tarefas [21]. Para o chaveamento entre as tarefas, o escalonador deve armazenar todos os conteúdos dos registradores da CPU da tarefa que está deixando o executivo e carregar todos os registradores da tarefa que está entrando no executivo [22].

Uma tarefa desativável pode passar por três estados durante sua execução em tempo real (figura 3.1): estado executando, pronta para executar ou bloqueada. A mudança de estado da tarefa depende de eventos internos ou externos a CPU. Uma tarefa no estado bloqueada passará ao estado pronta (linha 4) por requisição de interrupção ou por solicitação de outra tarefa. Uma tarefa desativável no estado executando passará ao estado pronta (linha 2) quando houver a necessidade do acionamento de uma tarefa de maior prioridade. Uma tarefa no estado pronta passará a executando (linha 1) quando houver disponibilidade de processador. Uma tarefa passará do estado executando para bloqueada (linha 3) no final de sua execução [21].

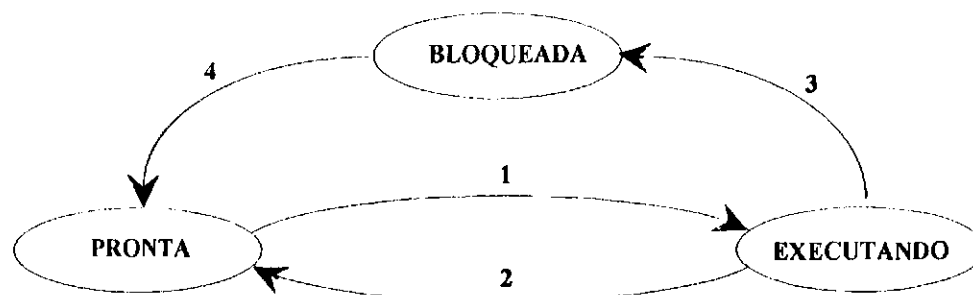


Figura 3.1: Estados das tarefas.

III.1.1. Executivo em Tempo Real do Módulo de Interface

Um executivo em tempo real [21],[23],[24] foi adaptado para gerenciar as tarefas da interface do módulo IEEE-488 apresentado neste trabalho. Devido ao

compartilhamento dos dados pelas tarefas, utilizou-se semáforos binários para gerenciar o acesso às variáveis críticas do executivo [25],[26], que foi escrito em linguagem de programação "assembly", do microcontrolador MC68HC11 [27].

Nas figuras 3.2 e 3.3 estão representadas as estruturas desse executivo. No início da execução do programa, os registradores das tarefas do relógio, os semáforos binários e a interrupção periódica RTI - "Real Time Interrupt" são inicializados [27].

A cada tarefa do executivo é associado um descritor. Os descritores das tarefas são encadeados numa lista de prioridades estática. Cada descritor possui 5 campos como é mostrado na Tabela III.1 [21],[23].

Tabela III.1: Campos dos descritores das tarefas.

CAMPO/VARIÁVEL	DESCRIÇÃO
STATUS	estados "pronta" "executando" e "bloqueado" das tarefas
FREQUÊNCIA	frequência entre duas ativações consecutivas da tarefa
CONTADOR	contador das interrupções do relógio em tempo real
LIGA	apontador para o próximo descritor na lista de prioridades
ENDEREÇO	identificador da tarefa

Uma interrupção em tempo real (RTI) é disponível no microcontrolador MC68HC11 com o objetivo de gerar interrupções por "hardware" em uma taxa fixa e periódica. Quatro taxas diferentes estão disponíveis para a RTI, sendo função da frequência do relógio do microcontrolador: 4,10; 8,19; 16,38 e 32,77 milisegundos [27].

Uma interrupção em tempo real (RTI), foi usada para acionar o escalonador de tarefas do executivo. Após a inicialização, o sistema permanece dentro de um "loop" esperando uma interrupção RTI ("Real Time Interrupt", interrupção em tempo real), que acontece a intervalos de aproximadamente 32,77 milisegundos. Quando ocorre uma interrupção RTI, o programa é desviado para o endereço apontado pelo vetor dessa interrupção, onde se encontra o escalonador das tarefas. No escalonador, os campos dos descritores das tarefas são verificados, iniciando pelo descritor 1 que corresponde a tarefa 1, como mostra a figura 3.4. Se o conteúdo do campo contador do descritor de tarefa 1 é diferente de zero, o campo contador é decrementado e, depois deste decremento, seu conteúdo é novamente verificado, podendo ocorrer dois casos:

1- o contador não é nulo (estado bloqueado), então o escalonador testa através do descritor da tarefa 1 se é a última tarefa indicada pelo campo liga; caso afirmativo o programa é desviado para o "loop" esperando outra interrupção RTI; caso contrário, é carregado a próxima tarefa apontada pelo campo liga e o processo se repete para todos os descritores e suas tarefas correspondentes, e;

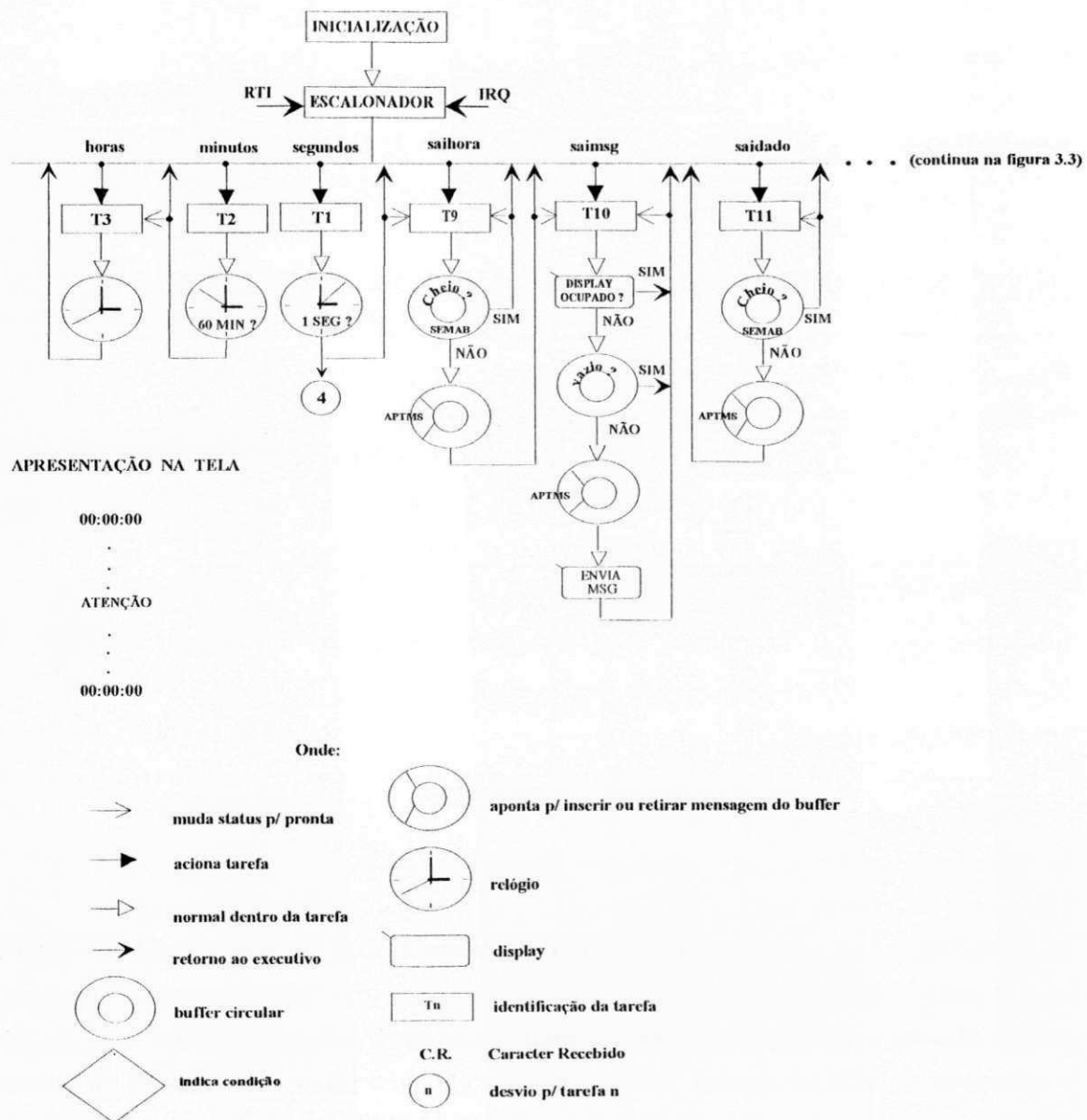


Figura 3.2: Fluxograma do executivo em tempo real.

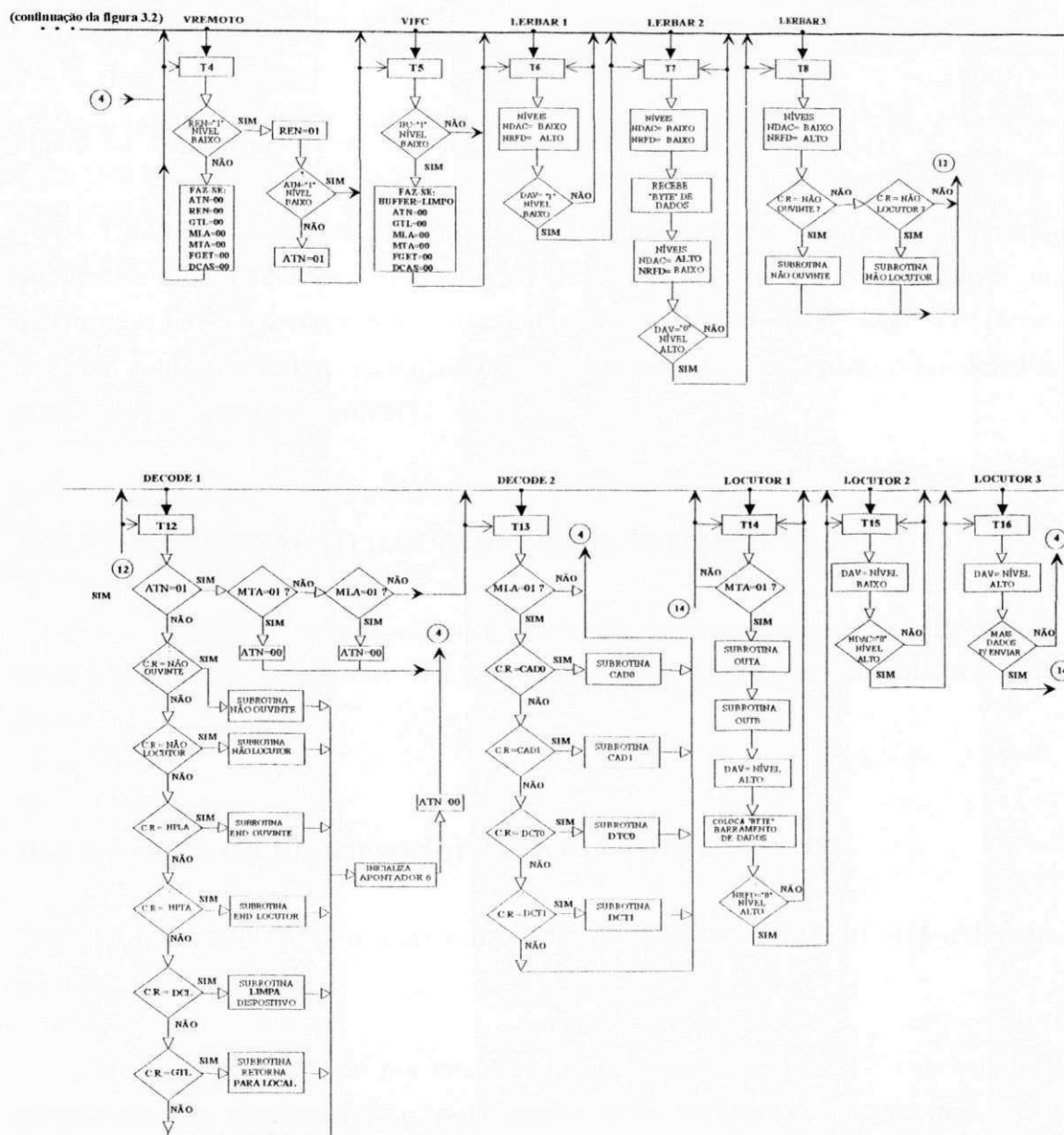


Figura 3.3: Continuação do fluxograma do executivo em tempo real.

2- o contador é nulo, então o conteúdo do campo frequência é movido para o campo do contador e o status é modificado para "ready" (status = 1). Neste momento, a tarefa que teve seu status modificado para "ready" é ativada. Quando termina a execução desta tarefa o programa retorna ao executivo e o escalonador testa através do descritor da tarefa se foi a última que foi ativada através do campo liga; caso afirmativo, o programa

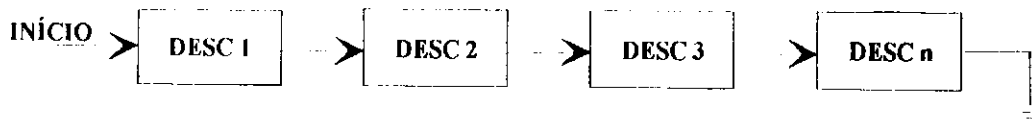


Figura 3.4: Lista de descritores das tarefas desativáveis ou não desativáveis.

Uma outra possibilidade de acionar uma determinada tarefa é fazer o campo status do descritor da tarefa em questão igual a 1, estado "ready". Para desativar uma determinada tarefa é necessário fazer o campo status igual a 0, estado bloqueado. Deve-se levar em consideração que a sequência de acionamento das tarefas não é pré-definida e pode ocorrer em qualquer ordem.

III.2. IMPLEMENTAÇÃO DAS TAREFAS DA INTERFACE IEEE-488

Um executivo em tempo real foi utilizado para implementar os diagramas de estados do módulo de interface IEEE-488, que foram definidas a partir do seu padrão (anexo G).

III.2.1. Notação dos Diagramas de Estados da Interface IEEE-488

A notação utilizada nos diagramas de estados da interface IEEE-488 é descrita a seguir:

1- o estado assumido por uma das funções citadas no item 1.2.4 da interface é identificado por um mnemônico, com quatro letras maiúsculas, circunscritas em um círculo, terminado com um S, (ex.: LOCS e REMS, figura 3.5);

2- todas as transições entre os estados de uma função da interface são representadas por linhas interligando os círculos e são descritas por uma ou mais expressões lógicas, que podem ser verdadeiras ou falsas (figura 3.5);

3- uma função deve permanecer em seu estado corrente se todas as expressões lógicas das transições entre os estados são falsas e deve entrar em um estado determinado se, e somente se, uma das expressões for verdadeira;

4- uma linha que não está conectada ao estado que lhe deu origem indica a origem de todos os estados da função (ex.: pon, figura 3.5);

5- uma expressão lógica consiste de uma ou mais mensagens locais ou remotas e estados de outras funções usadas em conjunto com operações "AND" (\wedge), "OR" (\vee) e "NOT" (\neg), e;

6- um estado de outra função é representado por um mnemônico com quatro letras maiúsculas, terminadas com S, circunscritas, (ex.: $\textcircled{\text{ACDS}}$, etc) [1],[3],[7],[11].

7- uma mensagem local é representada por um mnemônico com três letras minúsculas (ex.: pon, nba, etc) e uma mensagem remota (relacionadas com as vias da interface, por exemplo) são representadas por um mnemônico com três letras maiúsculas (ex.: IFC, ATN, etc).

III.2.1.1. Diagramas de estados da função RL ("remote/local"- remoto/local)

A função RL da interface IEEE-488, fornece aos instrumentos a capacidade de seleccionar dois modos de controle do funcionamento. No modo local, as instruções de controle são provenientes do painel frontal do equipamento e no modo remoto do barramento de interface.

O diagrama de estados da função RL do módulo IEEE-488, é mostrado na figura 3.5. No estado LOCS ("local state" - estado local), todos os controles funcionais do módulo desenvolvido estão no modo local, e ele não responde as mensagens que dependem do barramento da interface. Quando o módulo é ligado, a função RL está em LOCS.

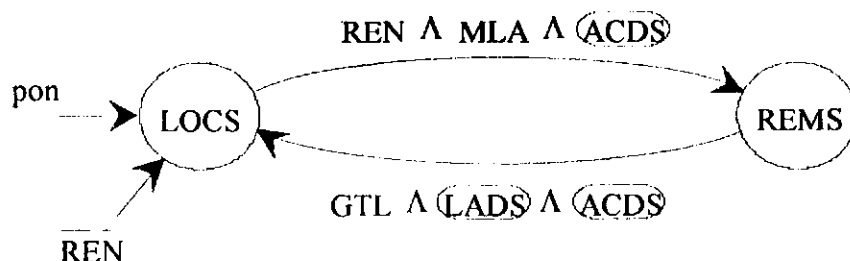


Figura 3.5: Diagrama de estados da função RL do módulo desenvolvido.

No estado LOCS, todos os controles locais do módulo desenvolvido estão operando normalmente, não respondendo, nem enviando mensagens pela interface. A função RL deve sair do estado LOCS para o estado REMS ("remote state" - estado remoto), se a via REN, a mensagem MLA ("my listen adress" - meu endereço de ouvinte) e o estado ACDS ("accept data state" - estado de aceitação de dados) estão ativos.

No estado REMS o equipamento está em remoto. Os controles locais associados com o instrumento estão inoperantes e ele não pode enviar mensagens pela interface.

A mudança do estado REMS para o estado LOCS ocorre se:

1- a mensagem GTL ("go to local" - vá para local) é verdadeira e ACDS e LDAS ("listener addressed state" - estado de ouvinte endereçado) estão ativas, ou;

2- a via REN está desativada.

III.2.1.2. Diagramas de estados da função AH ("acceptor handshake" - receptor de "handshake")

A função AH da interface IEEE-488 fornece a capacidade de receber mensagens dos instrumentos ligados ao barramento de interface (comandos, dados, endereços, etc). A transferência de cada "byte" é assíncrona entre um instrumento com função SH ("source handshake" - fonte de "handshake") e outros com função AH.

O diagrama de estados da função AH é mostrado na figura 3.6. Quando o módulo é ligado ele está no estado AIDS ("acceptor idle state" - estado de recepção desativado), com a função de recepção de "handshake" inativa. A mudança do estado AIDS para o ANRS ("acceptor not ready state" - estado de receptor não pronto) ocorre se:

1- a via ATN é verdadeira, ou;

2- LACS ("listener active state" - estado de ouvinte ativo) está ativo, ou;

3- LADS ("listener addressed state" - estado de ouvinte endereçado) está ativo.

No estado ANRS é indicado ao barramento de interface que o módulo não está pronto para continuar com o ciclo de "handshake". As vias NRFD e NDAC estão no estado verdadeiro.

A mudança do estado ANRS para o ACRS ("acceptor ready state" - estado de receptor pronto) ocorre se LADS ou LACS estiver ativo.

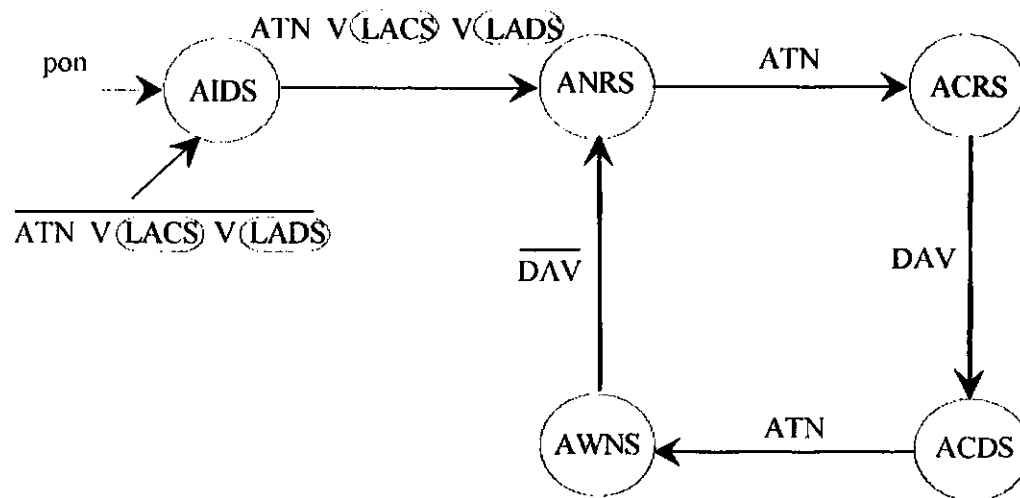


Figura 3.6: Diagrama de estados da função AH do módulo de interface.

No estado ACRS o módulo indica para interface que está preparado para receber uma mensagem. São impostos os estados verdadeiro e falso, respectivamente, para as vias NDAC e NRFD.

A mudança do estado ACRS para o ACDS ("accept data state" - estado de aceitação de dados) ocorre se a via DAV for verdadeira.

No estado ACDS, o módulo espera que o locutor (SH) mantenha válido o "byte" de mensagem. Este é o único em que uma mensagem no barramento de dados é válida. Neste, a mensagem está ou no modo de comandos (ATN= verdadeira) ou no modo de dados (ATN= falsa). As vias NDAC e NRFD estão em estado verdadeiro.

A mudança do estado ACDS deve ocorrer para:

1- o estado AWNS ("acceptor wait for new ciclo state" - estado de espera por um novo ciclo) se a via ATN for verdadeira;

2- o estado AIDS se a via ATN for falsa e tanto LADS quanto LACS são falsos.

No estado AWNS, a função AH indica que o módulo recebeu o "byte" sobre o barramento de dados. Nesse estado, a via NRFD é feita verdadeira e NDAC, falsa.

A mudança do estado AWNS deve ocorrer para:

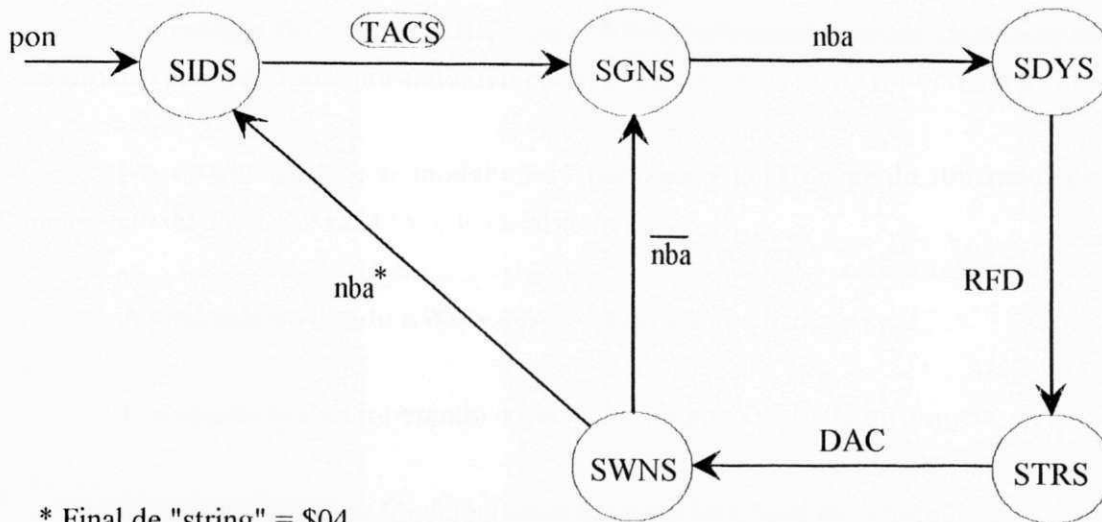
1- o estado ANRS se a via DAV for falsa, ou;

2- o estado AIDS se a via ATN for falsa e tanto o estado LADS quanto o LACS estão ativos.

III.2.1.3. Diagramas de estados da função SH ("source handshake"- fonte de "handshake")

A função SH da interface IEEE-488 fornece a capacidade de enviar mensagens assíncronas a um instrumento ligado ao barramento de interface (comandos, dados, endereços, etc). A transferência de cada "byte" é garantida entre um instrumento com função SH ("source handshake" - fonte de handshake) e outros com função AH. Esta função utiliza as vias RFD, DAC e DAV para efetuar a transferência de cada "byte".

O diagrama de estados da função SH do módulo IEEE-488 é mostrado na figura 3.7. No estado SIDS ("source idle state" - fonte em estado desativado), o módulo não está preparado para o ciclo de "handshake". Quando o módulo é ligado, a função SH está em SIDS.



* Final de "string" = \$04

Figura 3.7: Diagrama de estados da função SH do módulo de interface.

A mudança do estado SIDS para o SGNS ("source generate state" - estado de fonte geral) ocorre se o estado TACS ("talker active state" - locutor em estado ativo) estiver ativo.

No estado SGNS, o módulo de interface gera internamente um "byte" e espera que ele esteja disponível. Nesse estado a via DAV é falsa.

A mudança do estado SGNS para o SDYS ("source delay state" - fonte em estado de atraso) ocorre se existir um "byte" disponível.

No estado SDYS, o módulo espera um "byte" disponível sobre o barramento de dados depois da mudança do estado SGNS. Na função SH, o módulo também espera em SDYS que os ouvintes com a função AH indiquem que estão prontos para aceitar o "byte". A via DAV permanece ainda em estado falso.

A mudança do estado SDYS para o STRS ("source transfer state" - estado de transferência da fonte) ocorre se a via RFD for verdadeira.

No estado STRS, o módulo indica aos ouvintes que o "byte" sobre o barramento de dados é válido, a via DAV é verdadeira.

A mudança do estado STRS para o SWNS ("source wait for new ciclo state" - estado de espera da fonte por um novo ciclo) ocorre se a via DAC for verdadeira.

No estado SWNS, o módulo está esperando pelo início de um novo ciclo de mensagem de dado. A via DAV é feita falsa.

A mudança do estado SWNS deve ocorrer para:

- 1- o estado SGNS esperando o início de um novo ciclo de mensagem, ou;
- 2- o estado SIDS se foi detectado o caracter \$04, final de "string".

III.2.1.4. Diagramas de estados da função L ("listen" - ouvinte)

A função L da interface IEEE-488, fornece a um instrumento ligado ao barramento, a capacidade de receber comandos específicos do equipamento. Isto é possível, quando o módulo está endereçado como locutor.

O diagrama de estados da função L do módulo IEEE-488 é mostrado na figura 3.8. No estado LIDS ("listener idle state" - estado de ouvinte desativado), a função L no modulo não está pronta para receber mensagens que dependam do módulo. Quando o módulo é ligado, a função L está em LIDS.

A mudança do estado LIDS para o LADS ocorre se a via IFC for falsa, a mensagem MLA for verdadeira e ACDS estiver ativo.

No estado LADS, o módulo recebeu seu endereço de ouvinte e está preparado para receber mensagens específicas do instrumento.

A mudança do estado LADS deve ocorrer para:

- 1- o estado LACS se a via ATN for falso, ou;
- 2- o estado LIDS, se:

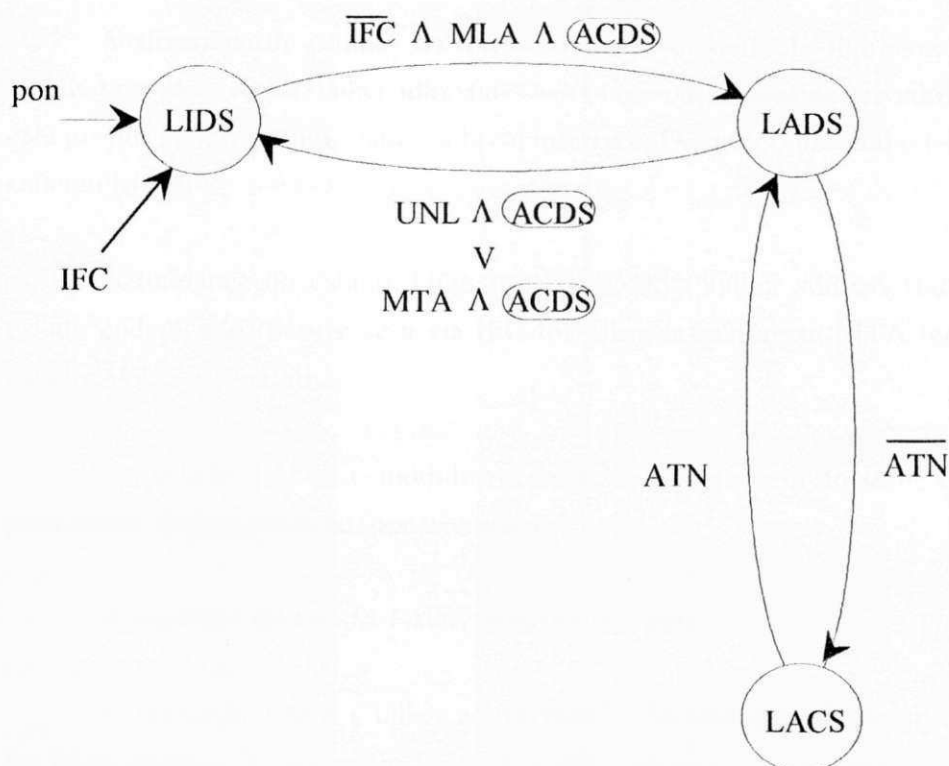


Figura 3.8: Diagrama de estados da função L do módulo de interface.

- 2.1- a mensagem UNL for verdadeira e o estado ACDS estiver ativo, ou;
- 2.2- a mensagem MTA ("my talk adress") for verdadeira e o estado ACDS estiver ativo, ou;
- 2.3- a via IFC for verdadeira.

No estado LACS, o módulo de interface está pronto para receber mensagens do instrumento locutor (DAB, EOS, STB ou END). O módulo IEEE-488 utiliza a função AH para controlar a recepção de mensagens da interface.

III.2.1.5. Diagramas de estados da função T ("talker"- locutor)

A função T da interface IEEE-488 fornece a capacidade de transmitir dados e mensagens para os ouvintes ligados ao barramento. Isto é possível, quando o módulo receber seu endereço de locutor.

O diagrama de estados da função T do módulo IEEE-488 é mostrado na figura 3.9. No estado TIDS ("talker idle state" - locutor em estado desativado), o módulo não está pronto para transmitir dados sobre a interface. Quando o módulo é ligado, a função L está em LIDS.

A mudança do estado TIDS para o TADS ("talker address state" - locutor em estado endereçado) ocorre se a via IFC for falsa, a mensagem MTA for verdadeira e o estado ACDS estiver ativo.

No estado TADS, o módulo recebeu seu endereço de locutor, e está preparado para enviar dados (medição, por exemplo).

A mudança do estado TADS deve ocorrer para:

1- o estado TACS ("talker active state" - locutor em estado ativo) se a via ATN for falsa, ou;

2- o estado TIDS, se:

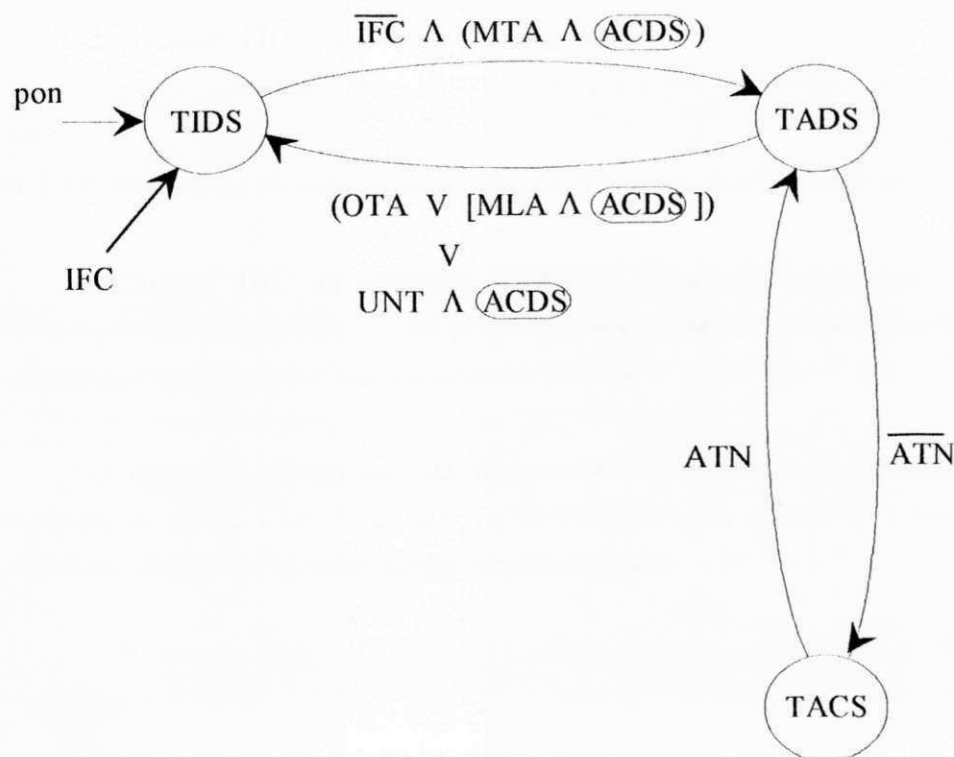


Figura 3.9: Diagrama de estados da função T do módulo de interface.

2.1- a mensagem UNT for verdadeira e o estado ACDS estiver ativo, ou;

2.2- a mensagem OTA ("other talk adress") for verdadeira e o estado ACDS estiver ativo, ou;

2.3- a mensagem MLA for verdadeira e o estado ACDS estiver ativo, ou;

2.4- a via IFC for verdadeira.

No estado TACS, o módulo de interface está pronto para enviar mensagens para os instrumentos endereçados como ouvintes. O módulo IEEE-488 utiliza a função SH para controlar a transmissão de mensagens sobre a interface e a ativação da linha EOI, indicando final de transmissão.

A mudança do estado TACS deve ocorrer para:

1- o estado TADS se a via ATN for verdadeira, ou;

2- o estado TIDS, se a via IFC for verdadeira.

III.2.1.6. Diagramas de estados da função DC ("device clear" - limpa instrumento)

A função DC, da interface IEEE-488 fornece a capacidade de limpar um instrumento individualmente ou um grupo de instrumentos. Este pode ser um subconjunto ou todos os equipamentos endereçados no sistema.

O diagrama de estados da função DC do módulo de interface IEEE-488 é mostrado na figura 3.10. No estado DCIS ("device clear idle state" - instrumento limpo em estado desativado), a função DC está desativada.

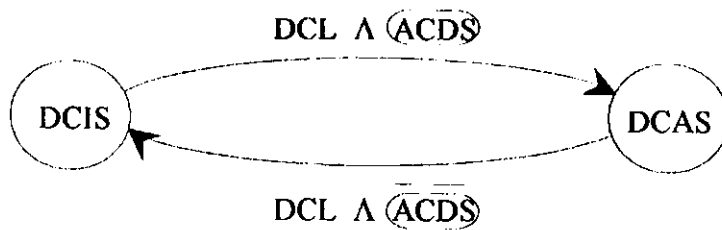


Figura 3.10: Diagrama de estados da função DC do módulo de interface.

A mudança do estado DCIS para o DCAS ("device clear active state" - instrumento limpo em estado ativo) ocorre se ACDS estiver ativo e a mensagem DCL ("device clear" - instrumento limpo) for verdadeira.

No estado DCAS, a função limpa o módulo de interface. Nesse, ela deve retornar ao estado DCIS, se ACDS estiver desativado e a mensagem DCL não for mais ativa.

III.2.2. Registradores de Supervisão do Barramento

Com o objetivo de supervisionar as vias de gerenciamento geral e o barramento de dados no módulo da interface, foram designados alguns registradores, que indicam o tipo do "byte" recebido no barramento de interface (modo de dados ou modo de comandos, através de ATN), o estado em que se encontra o módulo (local ou remoto, através de REM) e os comandos recebidos (ex.: GTL e DCL). Esses registradores têm as seguintes funções:

-ATN, indica o tipo do "byte" recebido pelo módulo e se o mesmo pertence ao modo de dados ou ao modo de comandos;

-REM, indica se o módulo está em estado remoto ou local;

-MTA, indica se o endereço de locutor no módulo foi recebido;

-MLA, indica se o endereço de ouvinte no módulo foi recebido;

-FGET, indica se o comando "Group Execute Trigger" foi recebido;

-DCAS, indica se o comando "Device Clear" foi recebido, e;

-GTLS, indica se o comando "Go To Local" foi recebido.

Na inicialização do executivo em tempo real, a PIA e os transceptores são programados na subrotina IMPAB, as vias REN, ATN, IFC, EOI, DAV e o barramento de dados são colocados para recepção e as vias DAC, RFD e SRQ para transmissão. Os registradores ATN, GTLS, DCAS, REM, MTA, MLA e FGET são zerados.

III.2.3. Tarefas da Interface IEEE-488

O executivo em tempo real cujo fluxograma está representado nas figuras 3.2 e 3.3 tem 16 tarefas, mas outras podem ser acrescentadas. Periodicamente, as vias do barramento IEEE-488 são testadas e os sinais de controle são gerados, a partir dessas tarefas.

III.2.3.1. Tarefas 1, 2 e 3

As tarefas 1, 2 e 3 são responsáveis respectivamente pelos minutos, segundos e horas do relógio;

III.2.3.2. Tarefas 4

A tarefa 4 (VREMOTO) é ativada pela tarefa 1 dos segundos, e foi desenvolvida a partir do diagrama de estados da função RL, tendo as seguintes etapas:

1- o estado da via REN é testada, se ela está no estado desativado (nível alto):

1.1- todos os registradores, incluindo o "buffer" do módulo desenvolvido são inicializados;

1.2- o LED de remoto é desativado;

1.3- o módulo permanece no estado local;

2- a via REN está ativa (nível baixo):

2.1- o registrador REM é feito igual a \$01;

2.2- a via ATN é testada se está em nível alto ou baixo. Pode acontecer dois casos:

2.2.1- ATN em nível baixo, os bytes recebidos pertencem ao modo de comando, são: comandos de endereçamento, comandos universais, etc, e;

2.2.2- ATN em nível alto, os bytes recebidos pertencem ao modo de dados, são: códigos de estado, dados de medição ou do terminal de vídeo, etc.

2.3- a tarefa 5 é ativadas, e;

2.4- a tarefa 4 retorna ao escalonador.

III.2.3.3. Tarefas 5

A tarefa 5, é mostrada na estrutura da figura 3.3. Esta tarefa, é relacionada com a via IFC, de gerenciamento geral da interface IEEE-488. Embora o padrão IEEE-488 não apresente um diagrama de estado relacionado com IFC, esta via faz parte de outras funções da interface (anexo G). Esta tarefa é desenvolvida conforme regras do padrão IEEE-488.

No módulo desenvolvido a tarefa 5 (VIFC) é ativada pela tarefa 4, e apresenta as seguintes etapas:

1- o estado da via IFC é testada, se está no estado desativado (nível alto):

1.1- as tarefas 6 (LERBAR 1), 7 (LERBAR 2) e 8 (LERBAR 3), respectivamente, são ativadas para recepção de dados e comandos, e correspondem ao diagrama de estado da tarefa AH ("Acceptor Handshake"- receptor de "handshake"), anexo G;

2- a via IFC está ativa (nível baixo):

2.1- o "buffer" e os registradores MLA, MTA, são inicializados;

2.2- as tarefas 6 (LERBAR 1), 7 (LERBAR 2) e 8 (LERBAR 3), respectivamente, são ativadas para recepção de dados e comandos, e correspondem ao diagrama de estado da tarefa AH ("Acceptor Handshake"- receptor de "handshake"), anexo G, e;

2.3- a tarefa 5 retorna ao escalonador.

III.2.3.4. Tarefas 6, 7 e 8

As tarefas 6, 7 e 8, foram definidas a partir do diagrama de estados da função AH, e são responsáveis pela recepção de mensagens remotas sobre o barramento da interface IEEE-488, e utilizam para este fim as vias de "handshake": NRFD, NDAC e DAV.

Na tarefa 6 (LERBAR 1), é iniciado a recepção dos "bytes", onde:

1- a via NRFD é colocada para nível alto e a via NDAC para nível baixo;

2- é testado o estado da via DAV:

2.1- se a via DAV está em nível baixo, a tarefa 7 (LERBAR 2) é ativada, e;

2.2- se a via DAV está em nível alto, a tarefa 6 ativa-se novamente.

Na tarefa 7 (LERBAR 2), têm-se:

1- a via NRFD é colocada para nível baixo e a via NDAC para nível baixo;

2- o "byte" é recebido pelo barramento de dados e comandos;

3- a via NRFD é colocada para nível baixo e a via NDAC para nível alto;

4- é testado o estado da via DAV:

4.1- se a via DAV está em nível alto, a tarefa 8 (LERBAR 3) é ativada, e;

4.2- se a via DAV está em nível baixo, a tarefa 7 ativa-se novamente.

Na tarefa 8 (LERBAR 3), têm-se:

1- a via NRFD é colocada para nível alto e a via NDAC para nível baixo;

2- o "byte" recebido é comparado com o caracter de não ouvinte, têm-se:

2.1- se igual, a subrotina não ouvinte é acionada e depois ativa a tarefa 12;

2.2- se diferente, o "byte" recebido é comparado com o caracter de não locutor, têm-se:

2.3- se igual, chama a subrotina não locutor e depois ativa a tarefa 12;

2.4- se diferente, a tarefa 12 é ativada,

III.2.3.5. Tarefas 9, 10 e 11

A tarefa 9 tem como objetivo inserir as horas, minutos e segundos no "buffer" circular.

A tarefa 10 tem como objetivo retirar as horas, minutos e segundos ou a mensagem (ATENÇÃO) do "buffer" circular e apresentá-las no terminal de vídeo.

A tarefa 11 tem como objetivo inserir a mensagem ATENÇÃO no "buffer" circular.

Embora o "buffer" circular seja usado apenas para inserir as horas e mensagens, nada impede que o mesmo seja usado para outras finalidades.

III.2.3.6. Tarefa 12

A tarefa 12 (DECODE1) tem como função decodificar o "byte" recebido pelo barramento de dados. Inicialmente, o registrador ATN é testado com dois valores possíveis: \$01 (modo de dados) ou \$00 (modo de comandos).

No caso de ele ser \$00, esta tarefa testa se o "byte" recebido é um comando universal, comando de endereçamento, endereço ou comando de desativação e chama sua subrotina correspondente.

No caso do conteúdo do registrador ATN ser \$01, esta tarefa testa se o módulo está endereçado como ouvinte (registrador MLA):

1- caso afirmativo, ativa-se a tarefa 13 (DECODE2), para decodificar os comandos do instrumento (do conversor A/D, gerador de pulsos, etc), e;

2- caso negativo, ativa a tarefa 4 e retorna ao escalonador.

III.2.3.7. Tarefa 13

A tarefa 13 (DECODE2) tem como função decodificar os "bytes" dos comandos recebidos pelo barramento de dados. Inicialmente, é testado o registrador MLA (se o módulo está endereçado como ouvinte) e:

1- caso afirmativo, os comandos do conversor A/D (CAD0 ou CAD1) ou do gerador de pulsos (FRQ1 ou FRQ2) são decodificados, e sua subrotina correspondente ativada;

2- caso negativo, é ativada a tarefa 4 e a tarefa 13 retorna ao escalonador.

Se o módulo foi endereçado como ouvinte e o comando enviado pelo controlador não corresponde aos encontrados neste decodificador de comandos, a tarefa 4 é ativada e a tarefa 13 retorna ao escalonador;

III.2.3.8. Tarefas 14, 15 e 16

As tarefas 14, 15 e 16, foram definidas a partir do diagrama de estados da função SH, e são ativadas pela tarefa 12 (DECODE1), quando o endereço de locutor (MTA) é recebido no barramento de dados. Elas são responsáveis pela transmissão de mensagens remotas sobre o barramento da interface IEEE-488, utilizando para este fim as vias de "handshake": NRFD, NDAC e DAV.

Na tarefa 14 (TALK1), inicia-se a transmissão de dados de medição para os ouvintes (computador, por exemplo). O módulo testa o registrador MTA e:

1- se igual a \$00, a tarefa 14 retorna ao escalonador;

2- se igual a \$01 (equipamento endereçado como locutor), as subrotinas OUTA e OUTB são executadas e a PIA e os transceptores programados: as vias REN, ATN, IFC, NDAC e NRFD são colocadas para recepção, as vias EOI, DAV, SRQ e o barramento de dados para transmissão, além disso:

2.1- a via DAV é feita alto;

2.2- o primeiro "byte" para transmissão é carregado e comparado com final de "string", caracter \$04, têm-se: a) se igual, a tarefa 4 e a subrotina IMPAB (restaura a configuração inicial da PIA e dos transceptores) são ativadas, os apontadores de transmissão e recepção são inicializados e a tarefa 14 retorna ao escalonador, ou; b) se diferente, o caracter é complementado e enviado sobre o barramento de dados.

2.3- é testada a via NRFD, estado falso ou verdadeiro, se o ouvinte está pronto para receber o "byte" presente no barramento de dados, têm-se: a) no estado falso, a tarefa 14 ativa-se novamente, ou; b) no estado verdadeiro, a tarefa 15 é ativada.

Na tarefa 15 (TALK2), têm-se:

1- a via DAV é feita em nível baixo, e;

2- testa-se a via NDAC, estado falso ou verdadeiro, indicando se o ouvinte accitou o "byte" enviado no barramento de dados, têm-se: a) no estado falso, a tarefa 15 ativa-se novamente, ou; b) no estado verdadeiro, a tarefa 16 é ativada.

Na tarefa 16 (TALK3), têm-se:

- 1- o próximo caracter de medição é carregado;
- 2- a via DAV é feita em nível baixo, e;
- 3- a tarefa 14 é ativada, para um novo ciclo de transmissão.

III.2.4 DIMENSIONAMENTO DE MEMÓRIA PARA ESTE MÓDULO

A tabela III.2 motra a quantidade de memória e o número de instruções por tarefa do módulo de interface IEEE-488 e do relógio do executivo em tempo real.

Um ciclo de relógio no MC68HC11 é de 0,5 μ segundos para uma frequência de 2 MHz.

O "software" apresentado neste trabalho ocupou um total de 1,4 "Kbytes" de memória e foi gravado em memória EEPROM interna ao microcontrolador, entre as locações de memória \$B800 e \$BFFF.

O escalonador ocupou 53 "bytes" de memória, com 25 instruções e seu tempo de execução foi de 52,5 μ segundos.

Apresenta-se no próximo capítulo as aplicações relacionadas ao módulo de interface IEEE-488 e um pequeno manual para inserção de novas tarefas e comandos ao microcontrolador MC68HC11.

Tabela III.2: Quantidade de memória e número de instruções por tarefa.

TAREFAS	MEMÓRIA ("bytes")	Nº. DE INSTRUÇÕES	TEMPO DE EXECUÇÃO (μ seg.)
1	47	32	53
2	38	28	44,5
3	33	26	41,5
4	79	36	71
5	25	15	41
6	34	19	33,5
7	53	27	47
8	46	24	45
9	95	44	84
10	71	31	60,5
11	42	43	84
12	333	283	384
13	322	231	397
14	95	45	128
15	34	19	34,5
16	30	17	33
TOTAL	1377	920	1604



APLICAÇÕES

Neste capítulo, apresentam-se aplicações do módulo de interface IEEE-488 como digitalizador e como gerador de pulsos. Mostra-se ainda como retirar, modificar ou introduzir novas tarefas no executivo. A introdução de novas tarefas é particularmente importante quando se deseja usar os sub-sistemas internos ao microcontrolador MC68HC11.

IV.1. APLICAÇÕES EM INSTRUMENTOS DE MEDIDA

Para avaliar e verificar o funcionamento do módulo de interface IEEE-488, utilizou-se alguns sub-sistemas internos ao microcontrolador MC68HC11 na implementação simples de instrumentos dotados dessa interface. Um gerador de pulsos e um digitalizador foram implementados usando, respectivamente, o temporizador e o conversor A/D internos do 68HC11.

IV.1.1. Digitalizador com o Módulo IEEE-488

Um digitalizador foi implementado com o módulo de interface desenvolvido com o 68HC11 (no endereço \$C857). Para este fim, foi usado o conversor A/D interno ao microcontrolador. Este conversor tem 8 entradas multiplexadas, é de 8 bits e usa a técnica de aproximação sucessiva. O registrador ADCTL (\$1030) programa o conversor A/D,

que guarda os resultados das conversões nos registradores ADR1, ADR2, ADR3 e ADR4 (\$1031, \$1032, \$1033 e \$1034, respectivamente). Cada conversão se completa em 32 ciclos do relógio (16 μ s, para um relógio de 2 MHz).

O processo de digitalização de uma tensão contínua se inicia pela solicitação de conversão feita pelo controlador do barramento enviando uma mensagem CAD0 ou CAD1, que corresponde, respectivamente, aos canais 0 e 1 do conversor A/D do microcontrolador. Após a solicitação, o digitalizador faz uma conversão e guarda o resultado no "buffer" de transmissão, entre as locações \$C690 e \$C69F. O controlador deve então se endereçar como ouvinte e endereçar o digitalizador como locutor para poder receber o resultado da conversão A/D.

Para o teste do digitalizador foi usada a estrutura representada na figura 4.1, onde foram colocados no barramento: um cartão controlador inserido em um "slot" de um computador (controlador com função de locutor), uma fonte de tensão cc programável e um multímetro digital, além do digitalizador.

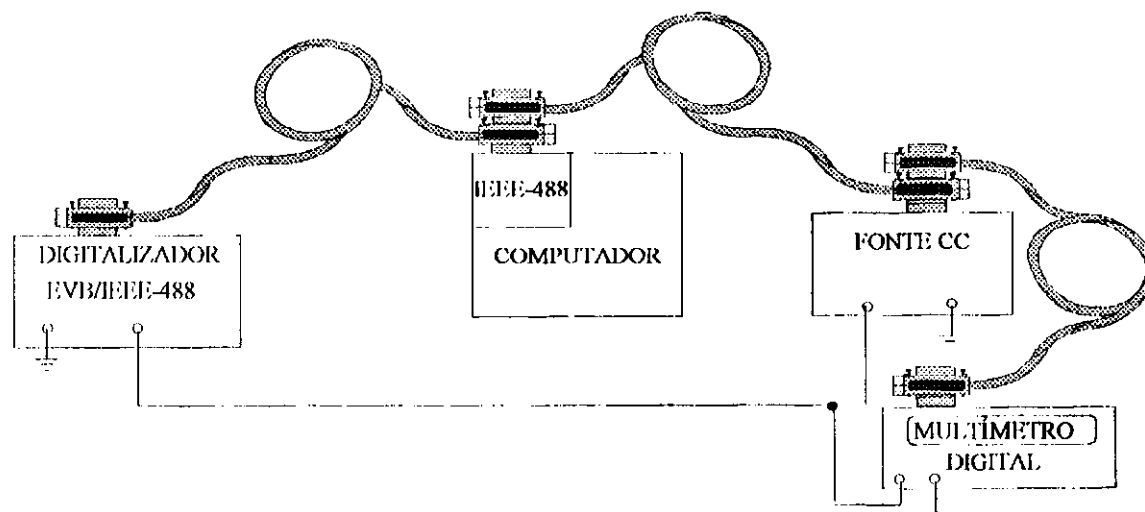


Figura 4.1: Disposição dos equipamentos para testes no módulo de interface IEEE-488.

O procedimento de teste foi o seguinte:

1- o controlador programa todos os instrumentos ligados ao barramento como ouvinte (inclusive o digitalizador);

2- a fonte gera uma tensão cc em sua saída, com amplitude, programada pelo controlador, entre 0 e 5 volts;

3- o multímetro digital e o digitalizador fazem leituras do sinal de saída da fonte;

4- o controlador programa o multímetro digital como locutor, que envia o dado adquiridos para o computador, e;

5- o passo d é repetido para o digitalizador.

Os resultados das leituras feitas pelo digitalizador e pelo multímetro digital são apresentados na tabela IV.1. Os valores fornecidos pelo digitalizador são em hexadecimal e foram convertidos no computador para tensão entre 0 e 5 V.

Tabela IV.1: Resultados das leituras do módulo de interface e do multímetro digital.

LEITURAS nº	FONTE CC	MULTÍMETRO DIGITAL	DIGITALIZADOR	
	Tensões (Volts)	Tensões (Volts)	Valor Hexadecimal	Val. Tensões (Volts)
1	0,5	0,5012	1A	0,5098
2	1,0	0,9985	34	1,0190
3	2,0	1,9858	67	2,0190
4	3,0	2,9669	9A	3,0190
5	4,0	3,959	CD	4,0196

O digitalizador só faz uma aquisição a cada solicitação do controlador, entretanto, pode-se facilmente modificar o programa para que o controlador possa programar também o número de conversões desejadas. Além disso, pode-se acrescentar tarefas para que este digitalizador funcione como voltímetro, fornecendo valores eficaz, de pico, médio, etc.

IV.1.2. Gerador de Pulsos

Um gerador de pulsos foi implementado com o módulo de interface desenvolvido com o 68HC11. Para este fim, foi usado o temporizador interno ao microcontrolador.

Este temporizador usa um contador de 16 bits e tem 3 entradas de captura, 4 saídas de comparação e uma entrada/saída associada a um contador de pulsos de 8 bits. Na implementação do gerador de pulsos usou-se uma das saídas de comparação.

Para o teste do gerador de pulsos foi usada a estrutura representada na figura 4.2, onde foram colocados no barramento: um cartão controlador inserido em um "slot" de um computador (controlador com função de locutor) e um osciloscópio, além do gerador de pulsos.

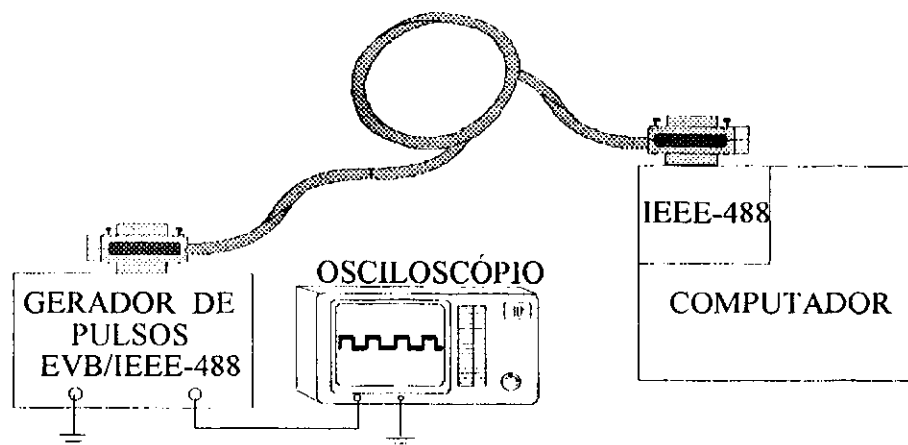


Figura 4.2: Disposição dos equipamentos para testes no módulo de interface IEEE-488.

A programação do gerador de pulsos é feita pelo controlador, enviando os comandos DCT0 e DCT1 para o gerador de pulsos. Estes comandos atuam modificando o ciclo de trabalho do sinal gerado para 20% ou 50%. A princípio fez-se a frequência constante, mas ela pode ser alterada, mudando apenas o "software" da tarefa do decodificador 2.

Observou-se com um osciloscópio digital que o sinal obtido na saída do gerador de pulsos estava de acordo com o esperado.

IV.2. MODIFICAÇÕES NO EXECUTIVO

A apresentação a seguir, tem como finalidade ajudar o usuário na utilização do módulo de interface IEEE-488 para modificação e introdução de novas tarefas, comandos e uso dos sub-sistemas internos ao microcontrolador MC68HC11, além da possibilidade

de mudança do endereço de locutor e/ou ouvinte do módulo e o uso dos "ports" de propósito geral do microcontrolador.

IV.2.1. Introdução de Novas Tarefas no Executivo em Tempo Real

Novas tarefas são necessárias sempre que se deseja utilizar um dos sub-sistemas do microcontrolador para um fim específico. Para introduzir uma nova tarefa no executivo em tempo real, deve-se definir seu descritor, como apresentado no item III.1.1 (página 26). O descritor deve ter campos STATUS, FREQUENCIA, CONTADOR, LIGA e ENDEREÇO, além disso ele deve ter seu nome. Como exemplo, apresenta-se a seguir o descritor da tarefa 1:

	ORG	\$C010	
TASK1	FCB	\$0	* STATUS
	FDB	\$001F	* FREQUENCIA
	FDB	\$001F	* CONTADOR
	FDB	TASK2	* LIGA
	FDB	VSEGU	* END

1- \$C010 é o endereço do descritor da tarefa 1;

2- TASK1 é a representação da tarefa 1 no descritor das tarefas e deve ser introduzida de acordo com a sequência de eventos das outras tarefas que se deseja monitorar ou controlar;

3- a frequência de ativação desta tarefa está programada nos endereços \$C011 e \$C012 com o valor #\$001F e o contador das interrupções do relógio em tempo real está no endereço \$C013 e \$C014;

4- TASK2, indica que no final da tarefa 1 o executivo irá escalonar o descritor da tarefa 2 através do campo liga;

5- o campo END indica o endereço da tarefa 1, onde o executivo é desviado para executá-la.

Uma tarefa deve apresentar a seguinte estrutura:

Nome	PSHX	* GUARDA O VALOR DE X.
	PSHY	* GUARDA O VALOR DE Y.
	PSHA	* GUARDA O VALOR DE A.
	PSHB	* GUARDA O VALOR DE B.
	.	
	.	
	.	
	PULB	* RESTAURA O VALOR DE B.
	PULA	* RESTAURA O VALOR DE A.
	PULY	* RESTAURA O VALOR DE Y.
	PULX	* RESTAURA O VALOR DE X.
	JMP	NAMUDA1
		* RETORNA AO ESCALONADOR.

A tarefa deve armazenar os acumuladores A, B, X e Y na pilha, quando ela está sendo executada, e restaurá-los quando deixa o executivo. No final de uma tarefa, o executivo deve retornar ao escalonador.

IV.2.2. Troca dos Endereços de Locutor e Ouvinte do Módulo de Interface IEEE-488

No executivo em tempo real, o endereço de locutor e ouvinte do módulo de interface IEEE-488 são:

- 1- HPLA, endereço de ouvinte, igual a \$23, e;
- 2- HPTA, endereço de locutor, igual a \$43.

Estes endereços podem ser trocados, tomando-se como base os endereços de locutor e ouvinte da Tabela A.1 do anexo A. Deve-se levar em consideração as regras da interface IEEE-488, no capítulo 1, onde um barramento composto por vários instrumentos deve apresentar endereços distintos para cada equipamento com funções de locutor e ouvinte.

IV.2.3. Novos Comandos dos Sub-Sistemas do Módulo de Interface IEEE-488

A tarefa 13 é responsável pela decodificação dos comandos recebidos pelo barramento, através do "buffer" de recepção de comandos, localizados entre os registradores \$C640 e \$C68F. Ela é responsável também pela ativação de um sub-sistema interno ao microcontrolador, contido no módulo de interface IEEE-488

A introdução de novos comandos dos sub-sistemas do módulo de interface IEEE-488 é feita definindo-se um código de 4 caracteres para este comando. No microcontrolador este código vai ser visto como 4 "bytes". Como exemplo pode-se citar os comandos CAD0 e CAD1 dos canais 0 e 1 do conversor A/D.

CAD0 ⇒ C = \$43
 A = \$41
 D = \$44
 0 = \$30

CAD1 ⇒ C = \$43
 A = \$41
 D = \$44
 1 = \$31

CAPÍTULO V

CONCLUSÕES

Apresentou-se neste trabalho o desenvolvimento em "hardware" e "software" de um módulo de interface com barramento no padrão IEEE-488. O "hardware" foi baseado no microcontrolador MC68HC11, sendo usado também uma PIA e dois transceptores; o "software" usou um executivo em tempo real para gerenciar todas as tarefas da interface.

O executivo em tempo real foi subdividido em 16 tarefas, cada uma com descritor. Esse executivo foi desenvolvido de modo a permitir modificações futuras, como introdução de novas tarefas, com relativa facilidade. As tarefas 1, 2 e 3 estão associadas a um relógio em tempo real, com frequência de ativação de 1 KHz; as tarefas 8, 9 e 10 estão associadas com um "buffer" circular e, as demais se encarregam do gerenciamento da interface.

Algumas funções do barramento IEEE-488 não foram implementadas (C, DT, PP e SR), mas elas podem facilmente ser acrescentadas ao executivo em tempo real que gerencia a interface.

O "software" do executivo em tempo real usou 1,4 "kbyte" de memória EEPROM interna ao microcontrolador, desse modo ainda sobra 0,6 "kbyte" para implementação de rotinas de inicialização e de novas tarefas.

Na implementação de instrumentos simples, é necessário apenas acrescentar novas tarefas ao "software". Para instrumentos mais complexos pode-se acrescentar mais memória ao módulo e "hardware" específico para a aplicação desejada.

Duas aplicações foram feitas com o intuito de testar o módulo desenvolvido: um digitalizador (que pode ser adaptado para um voltímetro digital), e um gerador de pulsos, usando, respectivamente, o conversor A/D e o temporizador internos ao microcontrolador. Os testes realizados em laboratório se mostraram satisfatórios.

Como sugestões para trabalhos futuros pode-se citar:

- implementar as funções C ("controller" - controlador) no módulo de interface com o objetivo de gerenciar as ações do barramento de interface, DT ("device trigger" - gatilho do instrumento) no módulo com a finalidade de iniciar ações individuais ou coletivas nos instrumentos, PP ("parallel pool" - "pool" paralelo) para coletar informações dos instrumentos de maneira rápida e eficiente e SR ("service request" - pedido de serviço) capacitando o módulo de interface avisar ao controlador do barramento que os dados estão prontos para transmissão ou ocorreu uma condição de erro, e;

- desenvolver instrumentos para aplicações específicas utilizando os sub-sistemas internos ao microcontrolador (conversor A/D, temporizador, comunicação serial), como condutivímetro, medidor de PH, gerador de funções, voltímetro digital.

O PROCESSO DE "POLL"

Em equipamentos que agregam vários sub-sistemas internos e compartilham recursos gerenciados por um mesmo instrumento, é comum a conversação entre os diversos sub-sistemas e este equipamento gerenciador denominado em inglês de "poll".

Na interface com padrão IEEE-488, um instrumento pode enviar um pedido de serviço para o controlador, ex.: quando ele terminou uma medição, quando ele detectou uma condição crítica de erro ou por outro motivo. O modo no qual o serviço é pedido e a maneira na qual o controlador trata tal pedido pode ser dividido em "poll" serial e "poll" paralelo.

- "Poll" serial, é uma sequência na qual habilita o controlador para ler o "status" dos instrumentos. Neste modo, o controlador pode determinar se um ou vários equipamentos necessitam de serviço. Desta forma, até 14 instrumentos são capazes de responder a este serviço e o controlador pode iniciar o questionamento individual e sucessivo de todos os equipamentos para determinar quem solicitou serviço.

Um instrumento deve possuir a função de pedido de serviço para ativar a via SRQ do barramento. Ele deve ter a função de locutor, para ser hábil em uma sequência de "poll" serial, e decodificar os comandos SPE e SPD.

Os principais eventos durante uma sequência de "poll" serial são as seguintes:

1- um instrumento ativa a via SRQ, para indicar ao controlador que ele está pedindo serviço. O controlador observa o estado da via SRQ = nível baixo, o mesmo pode iniciar uma rotina de interrupção. O gerenciamento da interrupção depende da natureza do "software" do controlador;

2- se existir vários instrumentos no barramento que podem gerar um SRQ, o controlador investiga qual dos equipamentos fez o pedido. Para este propósito, o controlador coloca o sistema no modo de "poll" serial com a adição do comando SPE sobre o barramento;

3- o controlador então checa os instrumentos individualmente e sucessivamente. O controlador endereça um dado equipamento como locutor e então inspeciona o estado das via de dados ("byte" de estados) no modo de dados. Se um equipamento pediu serviço, ele responde pela colocação de DIO7 para verdadeiro (mensagem RQS). Outras vias podem também ser colocadas para verdadeiro para indicar a natureza do pedido de serviço (mensagem STB), e;

4- o controlador termina o modo de "poll" serial com o comando sobre o barramento SPD ("poll" serial desativado). O controlador gera ações para aquele equipamento que fez um pedido de serviço.

- "Poll" Paralelo, é executado por iniciativa do controlador para obter informações dos equipamentos. Quando o controlador inicia um "poll" paralelo, cada instrumento retorna um bit de "status" através de uma das vias de dados, DIO1 ... DIO8. A associação de cada equipamento a uma das vias DIO1 ... DIO8 é feita através de chaves ou estrapes ou ainda pelo controlador, através do comando PPC ("Parallel Poll Configure").

No "Poll" paralelo o controlador checa ao mesmo tempo o número máximo de 8 instrumentos para determinar seus estados.

MENSAGENS DE ENDEREÇOS

No padrão IEEE-488 cada equipamento ligado ao barramento pode atuar como locutor e/ou ouvinte. A sua identificação é feita por um "endereço", que é simplesmente um código selecionado por chaves no próprio equipamento.

Um endereço consiste de 7 bits e os códigos válidos estão listados na tabela B.1. O bit mais significativo (MSB) entre os 8 bits sobre as linhas de dados é o (B_8). Este não tem influência sobre os endereços e é usado em certos casos como paridade; sendo indicado por um "X" na tabela B.1. O bit menos significativo (LSB) é o (B_1) sobre as linhas de dados e corresponde a DIO1.

O sexto e o sétimo bits determinam portanto se o endereço em questão é um endereço de locutor ou de ouvinte. Os cinco bits principais determinam o endereço do dispositivo envolvido.

O endereço de um equipamento pode, geralmente, ser trocado mudando-se as posições das chaves de endereçamento no próprio equipamento, as quais selecionam os cinco bits apropriados de endereço.

Tabela B.1: códigos para os endereços de ouvinte e locutor.

Endereços de Ouvinte									Endereços de Locutor								
Bits (representação Binária)								ASCII	Bits (representação Binária)								ASCII
B ₈	B ₇	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁		B ₈	B ₇	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	
x	0	1	0	0	0	0	0	SP	x	1	0	0	0	0	0	0	@
x	0	1	0	0	0	0	1	!	x	1	0	0	0	0	0	1	A
x	0	1	0	0	0	1	0	"	x	1	0	0	0	0	1	0	B
x	0	1	0	0	0	1	1	#	x	1	0	0	0	0	1	1	C
x	0	1	0	0	1	0	0	\$	x	1	0	0	0	1	0	0	D
x	0	1	0	0	1	0	1	%	x	1	0	0	0	1	0	1	E
x	0	1	0	0	1	1	0	&	x	1	0	0	0	1	1	0	F
x	0	1	0	0	1	1	1	'	x	1	0	0	0	1	1	1	G
x	0	1	0	1	0	0	0	(x	1	0	0	1	0	0	0	H
x	0	1	0	1	0	0	1)	x	1	0	0	1	0	0	1	I
x	0	1	0	1	0	1	0	*	x	1	0	0	1	0	1	0	J
x	0	1	0	1	0	1	1	+	x	1	0	0	1	0	1	1	K
x	0	1	0	1	1	0	0	,	x	1	0	0	1	1	0	0	L
x	0	1	0	1	1	0	1	^	x	1	0	0	1	1	0	1	M
x	0	1	0	1	1	1	0	.	x	1	0	0	1	1	1	0	N
x	0	1	0	1	1	1	1	/	x	1	0	0	1	1	1	1	O
x	0	1	1	0	0	0	0	0	x	1	0	1	0	0	0	0	P
x	0	1	1	0	0	0	1	1	x	1	0	1	0	0	0	1	Q
x	0	1	1	0	0	1	0	2	x	1	0	1	0	0	1	0	R
x	0	1	1	0	0	1	1	3	x	1	0	1	0	0	1	1	S
x	0	1	1	0	1	0	0	4	x	1	0	1	0	1	0	0	T
x	0	1	1	0	1	0	1	5	x	1	0	1	0	1	0	1	U
x	0	1	1	0	1	1	0	6	x	1	0	1	0	1	1	0	V
x	0	1	1	0	1	1	1	7	x	1	0	1	0	1	1	1	W
x	0	1	1	1	0	0	0	8	x	1	0	1	1	0	0	0	X
x	0	1	1	1	0	0	1	9	x	1	0	1	1	0	0	1	Y
x	0	1	1	1	0	1	0	:	x	1	0	1	1	0	1	0	Z
x	0	1	1	1	0	1	1	;	x	1	0	1	1	0	1	1	[
x	0	1	1	1	1	0	0	<	x	1	0	1	1	1	0	0	\
x	0	1	1	1	1	0	1	=	x	1	0	1	1	1	0	1]
x	0	1	1	1	1	1	0	>	x	1	0	1	1	1	1	0	^

COMANDOS UNIVERSAIS

Todos os comandos universais do barramento (tabela C.1), atuam em todos os instrumentos, endereçados ou não, tão logo eles tenham decodificados os comandos em questão, que são:

Tabela C.1: comandos universais.

Comando	Mnemônico	ASCII	HEXA
"Device Clear"	DCL	DC4	14
"Local Lock Out"	LLO	DC1	11
"Serial Poll Enable"	SPE	CAN	18
"Serial Poll Disable"	SPP	EM	19
"Parallel Poll Unconfigure"	PPU	NAK	15

1- DCL "Device Clear" (limpar instrumento), todos os instrumentos capazes de responder retornam para um estado pré-determinado;

2- LLO "Local Lockout" (chave local), este comando desabilita a chave de seleção REMOTO/LOCAL de todos os instrumentos do barramento, sensíveis a este comando, independentemente de estarem eles habilitados como ouvintes ou não. As chaves de seleção somente são reabilitadas com a transição de verdadeiro para falso da via REN, o que os coloca, adicionalmente, no modo local.

3- SPE "Serial Poll Enable" (habilita "poll" serial), este comando inicia uma sequência de "poll" serial sobre o barramento, feito por um dos instrumentos no barramento, pela ativação da via SRQ (anexo A);

4- SPD "Serial Poll Disable" (desativa "poll" em série), este comando desativa o modo de "poll" em série sobre o barramento e pode ser usado para terminar uma sequência de "poll" em série, e;

5- PPU "Parallel Poll Unconfigure" (paralelo "poll" não configurado), este comando coloca todos os instrumentos sobre o barramento com a capacidade para "poll" paralelo para um estado pré-definido (anexo A).

COMANDOS DE ENDEREÇAMENTO

Todos os comandos de endereçamento (tabela D.1) afetam todos os instrumentos endereçados, quando podem decodificá-los, eles são:

Tabela D.1: comandos de endereçamento

Comando	Mnemônico	ASCII	HEXA
"Selective Device Clear"	SDC	EOT	04
"Go To Local"	GTL	SOH	01
"Group Execute Trigger"	GET	BS	08
"Take Control"	TCT	HT	09
"Parallel Poll Configure"	PPC	ENQ	05

1- SDC ("Select Device Clear" - selecione dispositivo para limpar), este comando inicializa instrumentos endereçados como ouvintes para um estado pré-determinado;

2- GTL ("Go To Local" - vá para local), os instrumentos endereçados como ouvintes retornam para controle local;

3- GET ("Group Execute Trigger" - grupo execute gatilho), o controlador pode através deste comando, iniciar uma ação pré-programada por um ou mais equipamentos endereçados como ouvinte;

4- TCT ("Take Control" - tome o controle), este comando enviado pelo controlador, passa o controle do barramento para o equipamento endereçado como locutor, e;

5- PPC ("Parallel Poll Configure" - configuração para "poll" paralelo), este comando permite a alguns instrumentos responder a um "poll" paralelo por meio de algumas vias DIO1 ... DIO8.

COMANDOS SECUNDÁRIOS

São usados para endereços de locutor e ouvinte estendido (caracteres ASCII de 96 a 127 em representação decimal) e comandos de "poll" paralelo secundários, PPE e PPC, anexo A.

A interface IEEE-488 fornece a possibilidade do formato de endereço estendido pela adição de um segundo carácter sobre o endereço principal.

O sexto e o sétimo bit deste segundo carácter (B_6 e B_7) são iguais a "1".

Um exemplo de um endereço estendido é: "Ad",

Onde: "A" = x 1000001 (endereço primário)

"d" = x 1100100 (endereço secundário)

O endereço estendido pode ser usado para seleccionar partes de um dado instrumento, ou para identificar um instrumento de um grupo de instrumentos idênticos.

COMANDOS DE DESATIVAÇÃO

Estes comandos são usados para desativar os instrumentos endereçados como ouvinte ou locutor (tabela F.1), são:

Tabela F.1: comandos de desativação.

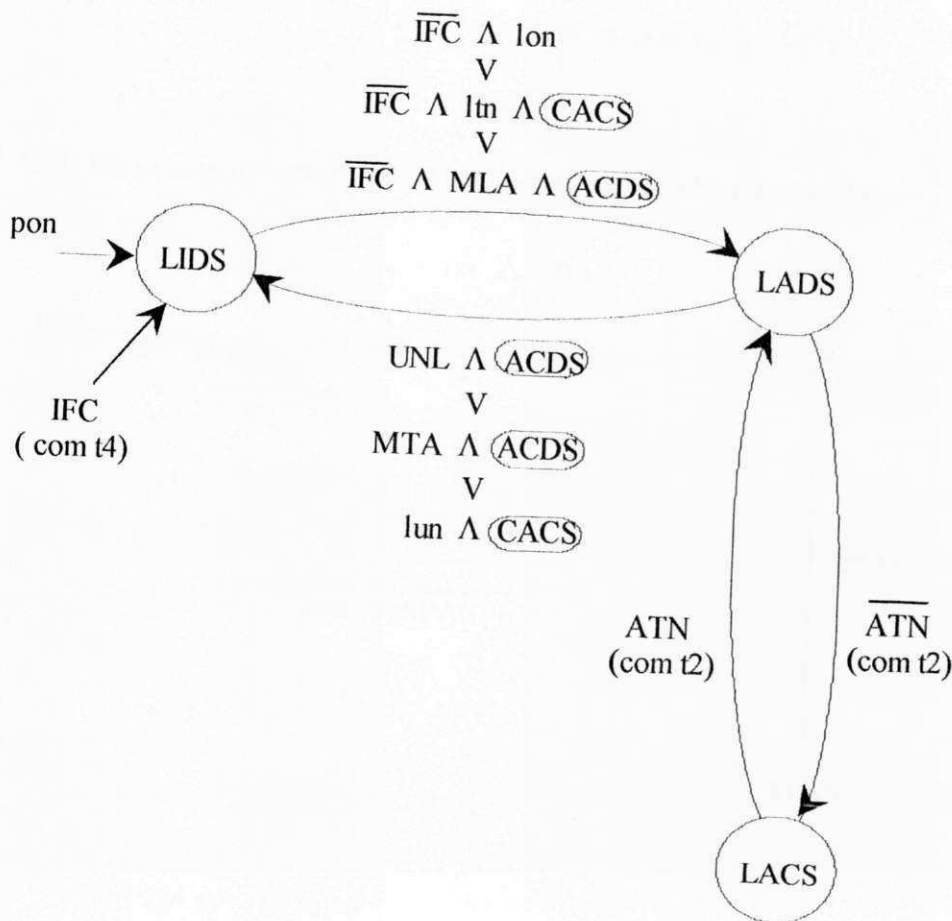
Comando	Mnemônico	ASCII	HEXA
"unlisten"	UNL	?	3F
"untalk"	UNT	-	5F

1- UNL ("UNListen" - não ouvinte), todos os ouvintes são inicializado para um estado não ativo de endereçamento. Este comando pode ser usado antes de um novo ouvinte ter sido endereçado, e;

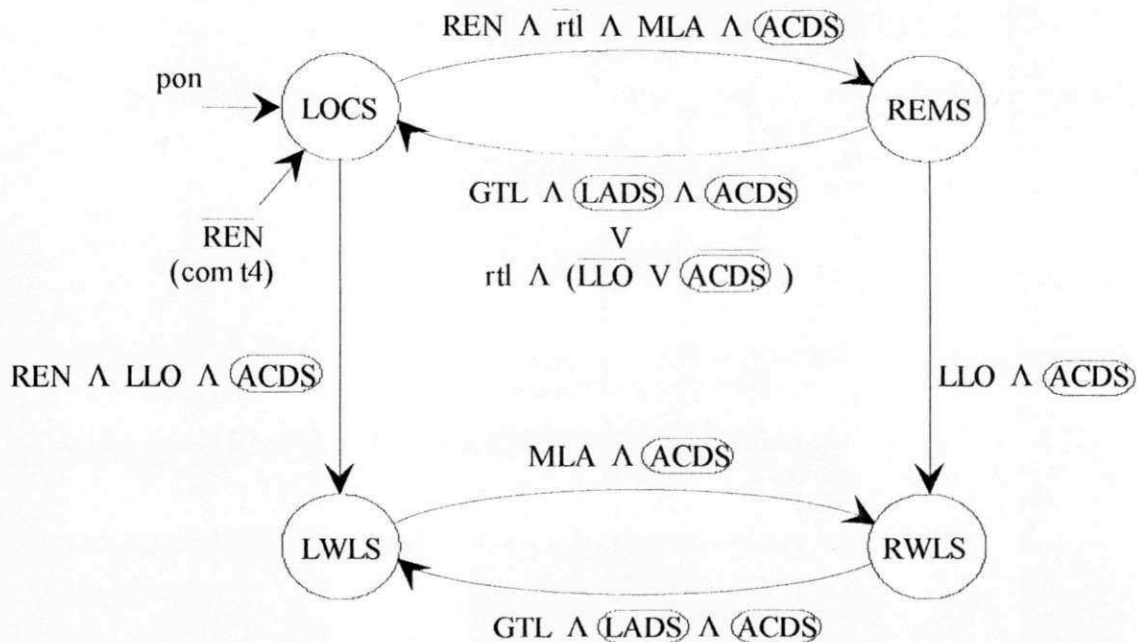
2- UNT ("UNTalk" - não locutor), o locutor principal será desativado de tal modo que nenhum locutor permanecerá sobre o barramento. Um locutor pode também ser inicializado pela transmissão de outro endereço de locutor sobre o barramento.

REPERTÓRIO DAS FUNÇÕES DA INTERFACE IEEE-488

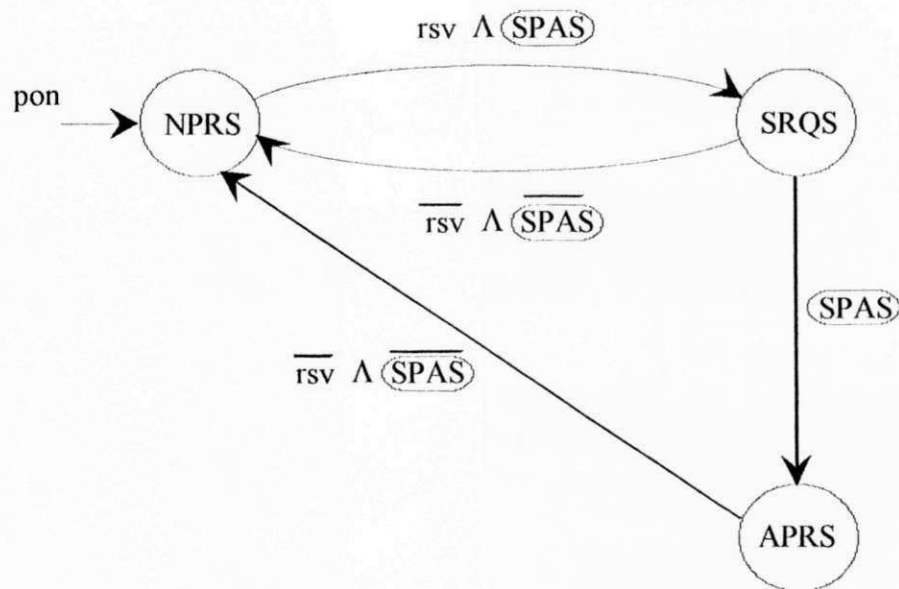
G.1. Diagrama de Estados da Função L da interface IEEE-488



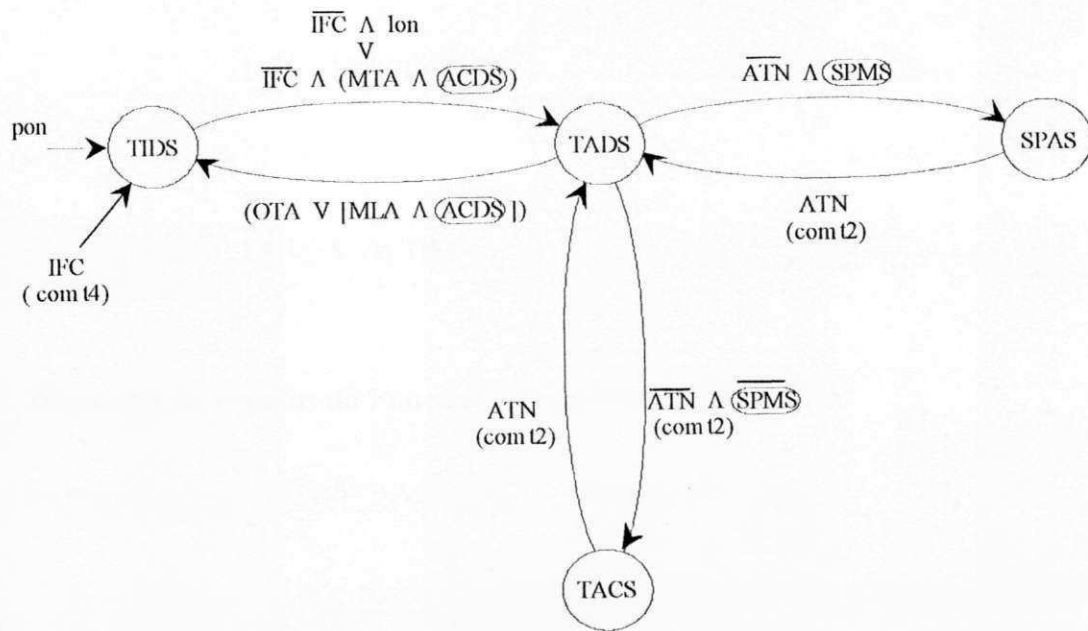
G.2. Diagrama de Estados da Função RL da interface IEEE-488



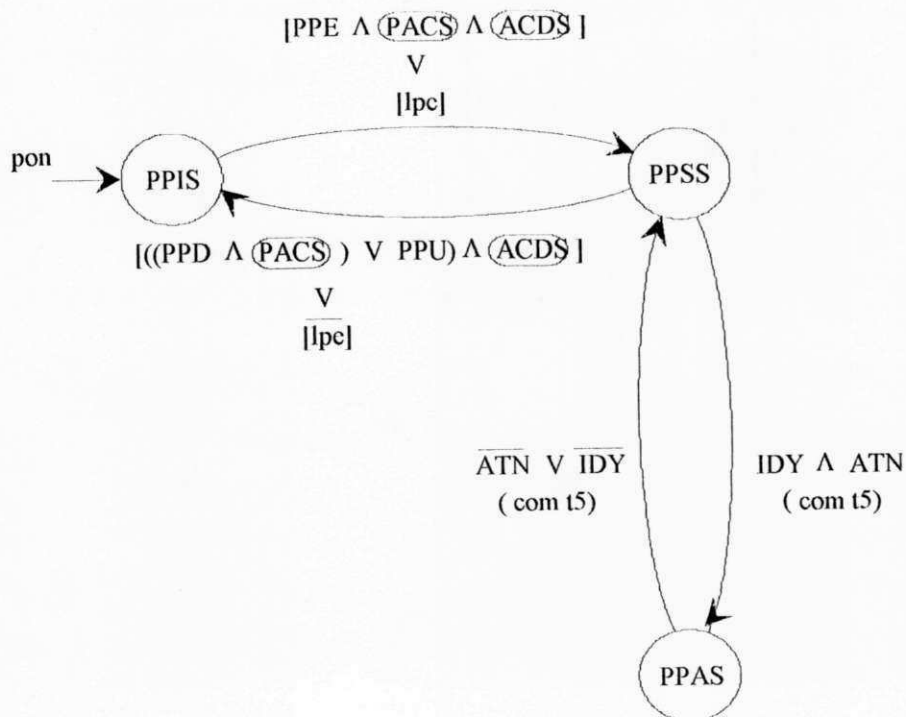
G.3. Diagrama de Estados da Função SR da interface IEEE-488



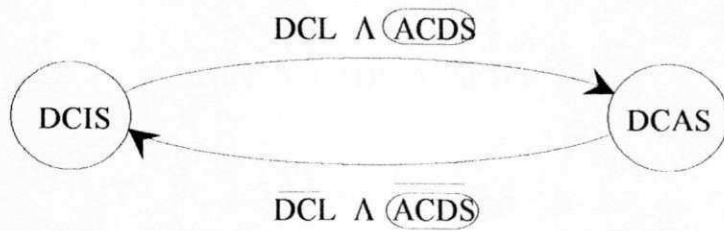
G.4. Diagrama de Estados da Função T da interface IEEE-488



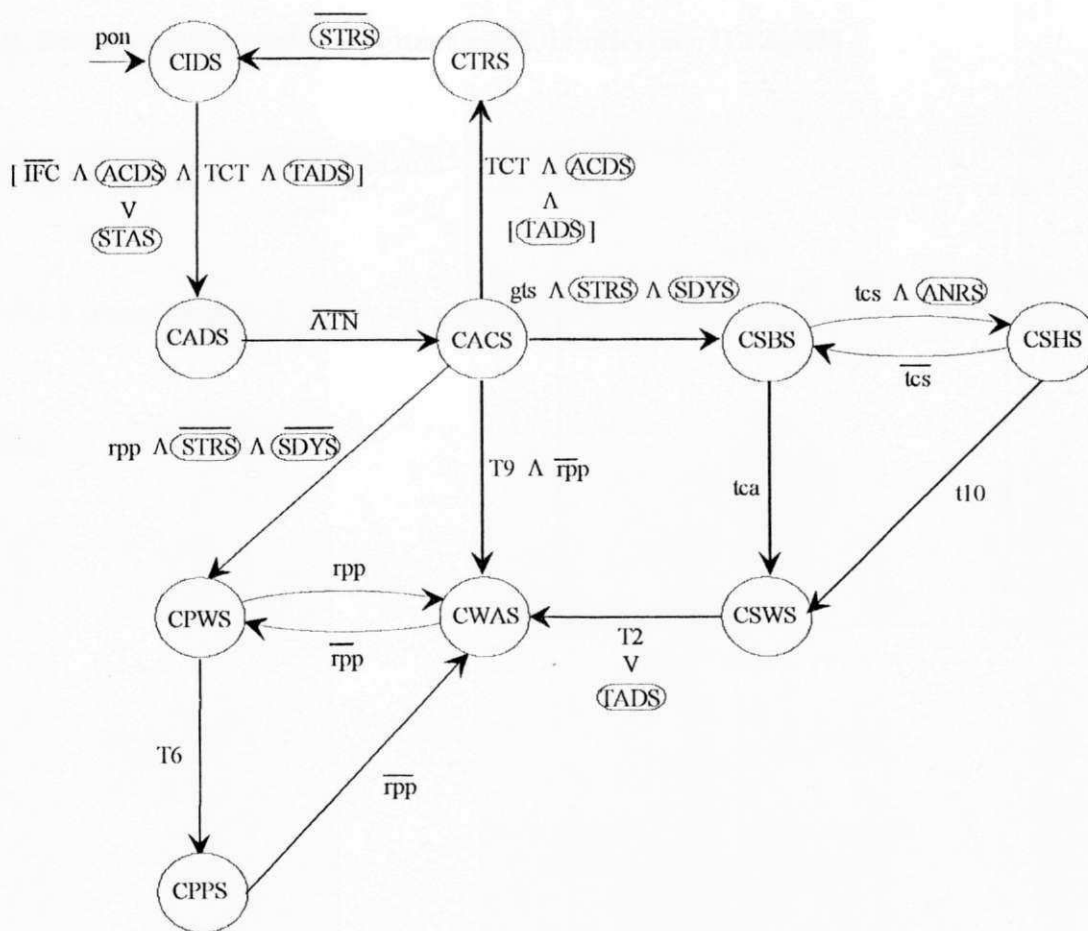
G.5. Diagrama de Estados da Função PP da interface IEEE-488



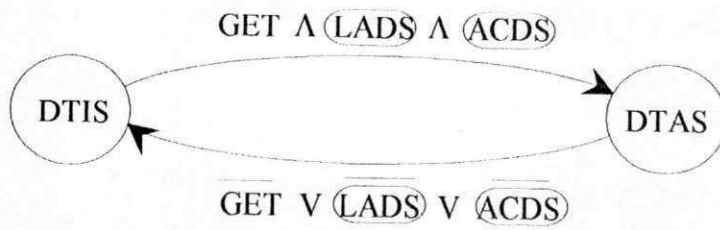
G.6. Diagrama de Estados da Função DC da interface IEEE-488



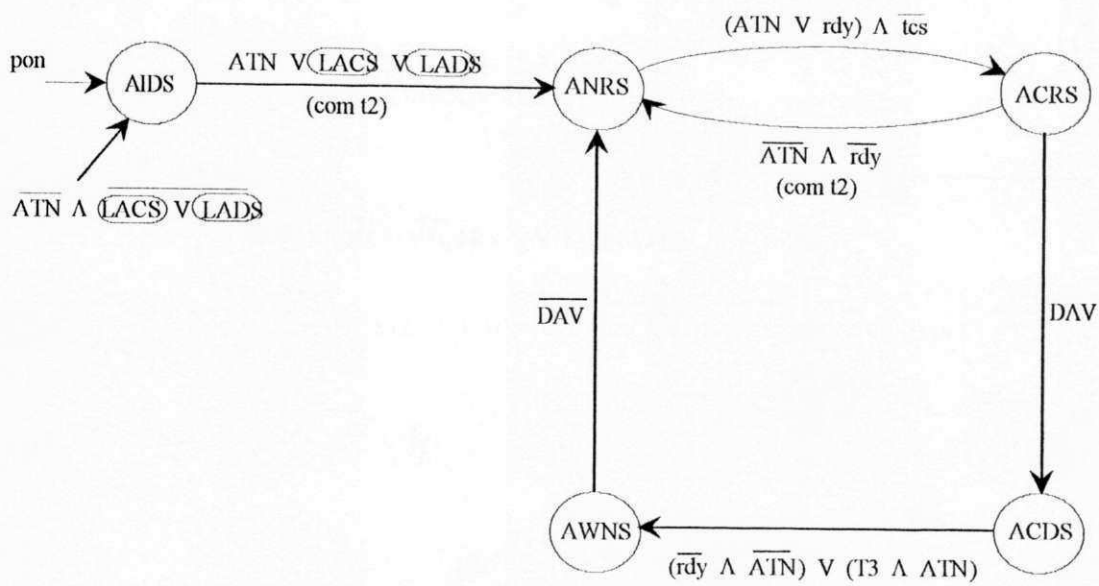
G.7. Diagrama de Estados da Função C da interface IEEE-488



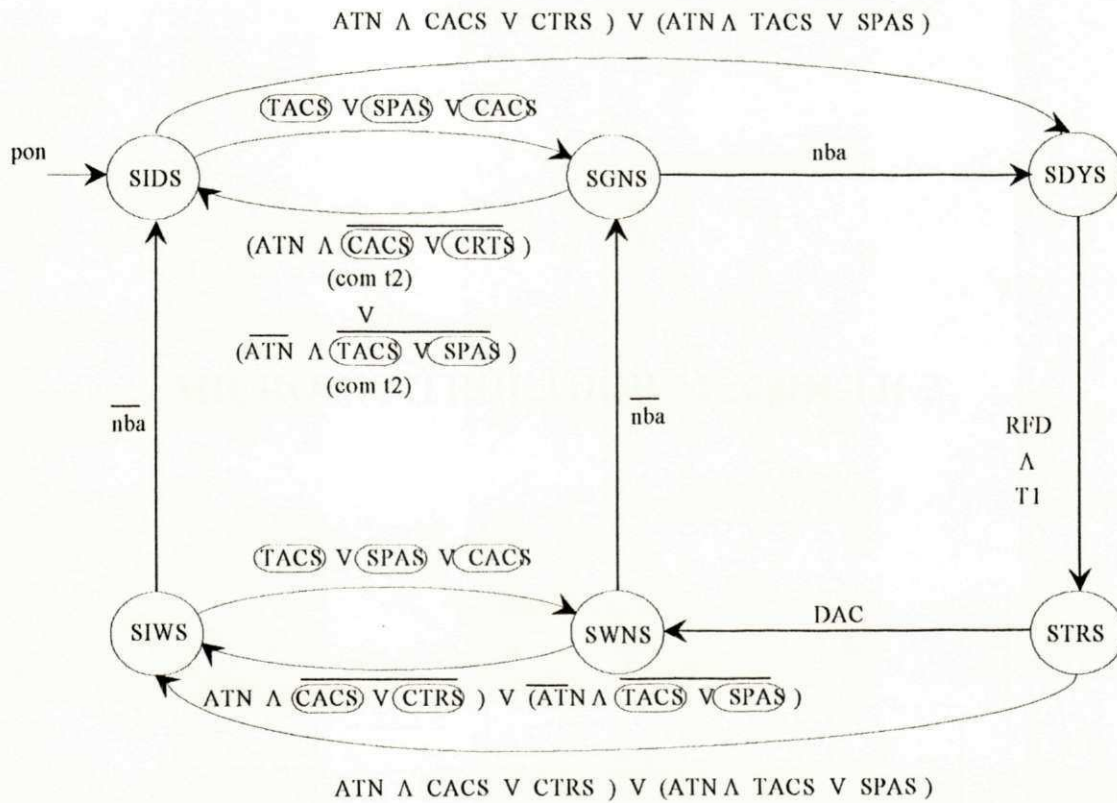
G.8. Diagrama de Estados da Função DT da interface IEEE-488



G.9. Diagrama de Estados da Função AH da interface IEEE-488

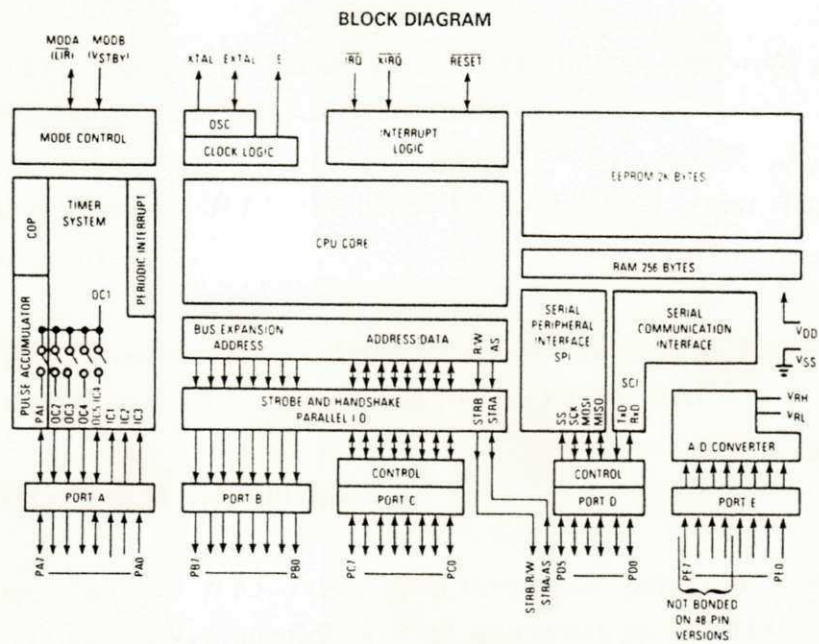


G.10. Diagrama de Estados da Função SH da interface IEEE-488



ANEXO H

MICROCONTROLADOR MC68HC11E2



REFERÊNCIAS BIBLIOGRÁFICAS

- [1] IEEE-488 Standard 488.1-1987, *Digital Interface for Programmable Instrumentation*, The IEEE, Inc., 345 East 47th St., New York, NY, Jun. 1987.
- [2] Hewlett Packard, *Tutorial Description of the Hewlett Packard Interface Bus*, 1980.
- [3] Grimberg, J.A.M., *Digital Instrument Course - IEC Bus Interface*, Philips, parte 4.
- [4] Bastide, G. & Vellas, J.R., *Mise en Oeuvre du Bus IEEE-488 - Utilisation et réalisation d'appareils*, Paris, ed. Editests, 1982.
- [5] Krutz, R.K., *Interfacing Techniques in Digital Design with Emphasis on Microprocessors*, New York, ed. Wiley, 1988.
- [6] National Instruments, *IEEE-488 and VXI Bus Control, Data Acquisition, and Analysis*, USA, 1992.
- [7] Zuffo, J.A., *Microprocessadores: Dutos do Sistema, Técnicas de Interface e Sistemas de Comunicação de Dados*, Ed. Edgard Blücher LTDA, 1981.
- [8] Onstee, H.G., T & M News, Philips.
- [9] *Speed-up, Simplify IEEE-488 System-Peripheral Interfaces with a Bipolar Talkers/Listener*, *Electronic Design*, v. 28, nº 4, 1980, pp. 109-114.
- [10] Lipovac, V. & S.J. Oh, *Design an IEEE-488 Bus into an FPLA*, *Electronic Design*, v. 25, nº 24, 1977, pp. 104-111.
- [11] Texas Instruments, *TMS 9914A General Purpose Interface Bus (GPIB) Controller, Data Manual*".

- [12] Chen, C.F. & Chen, C.S., *IBM PC-Based IEEE-488 Bus Controller*, IEEE, 1984.
- [14] Hootman, J., *See How the IEEE-488 Bus Works by Designing a Compatible A/D-D/A System*, *Electronic Design*, nº 20, 1978.
- [15] Intel, *MSC-86 Product Description*, 1979.
- [16] Motorola, *Microprocessor, Microcontroller and Peripheral Data*, v. II, pp. 1844-1863, USA, 1988.
- [17] Motorola, *8 bit Microprocessor & Peripheral Data*, pp. 131-148, USA, 1983.
- [18] Texas Instruments, *The Interface Circuits Data Book for Design Engineers*, segunda edição, pp. 171-183, USA, out. 1980.
- [19] Motorola, *Microprocessor, Microcontroller and Peripheral Data*, vol. 2, pp. 1610-1651, USA, 1988.
- [20] Motorola, *Microprocessor, Microcontroller and Peripheral Data*, vol. 2, pp. 1692-1702, 1988, USA.
- [21] Cavalcanti, J.H.F. & Deep, G.S. & Lima, A.M., *Executivo em Tempo Real para Controle de Motores Elétricos*, 9º CBA, pp. 1137-1139, 1992.
- [22] Shemitz, J., *Multitasking State Machine*, *The C Users Jornal*, pp. 23- 31, nov. 1992.
- [23] Cavalcanti, J.H.F. & Deep, G.S., *Um Kernel para Controle de Processos Industriais em Tempo Real Usando Microcomputadores*.
- [24] Cavalcanti, J.H.F. & Deep, G.S. & Lira, J.G.A., *Real Time Executive for Digital Signal Processor Employed in Process Control*, *IECON'88 Proceedings*, pp. 397-402, 1980.
- [25] Guimarães, C.C., *Princípios de Sistemas Operacionais*, Editora Campus, 5ª edição, 1986.

[26] Allworth, S.T., *Introduction to Real-Time Software Design*, Ed. Macmillan Press LTD, 1981.

[27] MOTOROLA - M68HC11, *HC11 - Reference Manual*, USA, 1990.