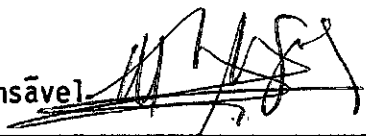


I

1. Classificação <i>INPE-COM. 2/NTI</i> <i>C.D.U.: 621.322:621.38</i>		2. Período	4. Critério de Distribuição: interna <input checked="" type="checkbox"/> externa <input type="checkbox"/>
3. Palavras Chaves (selecionadas pelo autor) <i>SISTEMA MONITOR</i> <i>LINGUAGEM DE MONITORAMENTO</i> <i>MICROCOMPUTADOR</i> <i>MINICOMPUTADOR</i>			7. Revisado por <i>EW Bergamini</i> <i>Eduardo W. Bergamini.</i>
5. Relatório nº <i>INPE-1333-NTI/111</i>	6. Data <i>Agosto, 1978</i>	9. Autorizado por <i>Parada</i> <i>Nelson de Jesus Parada.</i> <i>Diretor</i>	
8. Título e Sub-Título <i>MANUAL DO SISTEMA MONITOR HP-2116B</i>		10. Setor <i>DEE/GSD</i>	11. Nº de cópias <i>04</i>
12. Autoria <i>Wilson Yamaguti</i> <i>Fábio Vitaliano Filho</i>		14. Nº de páginas <i>74</i>	15. Preço
13. Assinatura Responsável 			
16. Sumário/Notas <i>Este manual surgiu da necessidade de padronização adotada na documentação do "software" e do "hardware" do sistema monitor, implementado no HP-2116 B/DOS do INPE. Ao mesmo tempo, a intenção é tornar o sistema monitor operacional, com vistas a sua utilização nos projetos do Programa de Sistemas Digitais e Analógicos em andamento. Na elaboração deste manual, baseou-se no trabalho preliminar de Dissertação de Mestrado do Eng. Fábio Vitaliano Filho, que iniciou o projeto deste sistema, implementando o "software" no HP-2116 B. Este manual também baseou-se no trabalho do Eng. Mário Mammoli, responsável pela implementação da primeira versão da interface ADAPTADOR-2116 B. Ficando ao nosso encargo, os testes finais, bem como a conclusão dos trabalhos de "software" e a subsequente operacionalidade do sistema completo.</i>			
17. Observações			

ÍNDICE

	Pág.
ABSTRACT	<i>vi</i>
LISTA DE FIGURAS	<i>viii</i>
 <u>CAPÍTULO I</u>	
INTRODUÇÃO	1
1.1 - Etapas na Programação de um Microcomputador	1
1.1.1 - Desenvolvimento de Programas	1
1.1.2 - Carga de Programas na Memória	2
1.1.3 - Testes de Programas	2
1.2 - Ferramentas Auxiliares	2
1.2.1 - Montadores Cruzados ("Cross Assembler")	3
1.2.2 - Compiladores Cruzados	3
1.2.3 - Montadores	3
1.2.4 - Carregadores ("Loaders")	3
1.2.5 - Simuladores	4
1.2.6 - Sistema Monitor	4
 <u>CAPÍTULO II</u>	
DESCRIÇÃO DO SISTEMA MONITOR	7
2.1 - Configuração do Sistema Monitor	7
2.1.1 - HP-2116B	7
2.1.2 - Comunicação com o Usuário	9
2.1.3 - Comunicação com o Microcomputador	9
2.1.4 - Sistema em Monitoramento	10
2.2 - Funções do Sistema Monitor	13
2.2.1 - Utilização de Equipamento Remoto	13
2.2.2 - Operações de Edição, Modificação e Transferência de Progra- mas ou dados	14

2.2.3 - Operação de Teste de Programas	15
2.2.4 - Operações de Controle de Microcomputador	16
2.3 - A Programação de Controle e Monitoramento	17
2.4 - Uma Sessão de Monitoramento	17

CAPÍTULO III

CONTROLES DO MICROCOMPUTADOR E ESTADOS DE PARADA	21
3.1 - Controle do Modo de Execução	21
3.2 - Controle do Sistema de Interrupção	22
3.3 - Controle do Estado de Parada	22
3.3.1 - Estados de Parada do Microcomputador	23
3.3.2 - Entradas e Saídas no Estado de Fim de Monitoramento	25
3.3.3 - Entrada e Saída no Estado de Espera de Interrupções	27
3.3.4 - Entradas e Saídas no Estado de Fim de Instrução ou Pausa ..	28
3.4 - Controle da Retomada de Programas	29
3.5 - Controle de Inicialização de um Programa	29
3.6 - Controle dos Canais de Interrupção	29

CAPÍTULO IV

LINGUAGEM DE MONITORAMENTO	31
4.1 - Componentes da Linguagem	31
4.2 - Símbolos Básicos e Palavras Reservadas	31
4.3 - Constantes	32
4.4 - Identificadores	33
4.5 - Estrutura do Programa	33
4.6 - Comandos da Secção de Configuração (Sec 1)	34
4.6.1 - Definição de Memória Local	35
4.6.2 - Definição de Memória Remota	35
4.6.3 - Definição do Endereço Inferior dos Periféricos Remotos	35
4.6.4 - Associação de Periféricos Remotos	36
4.6.5 - Finalização da Secção de Configuração (Sec 1)	36

4.7 - Comandos da Secção de Edição, Modificação e Transferência de Programas (Sec 2).....	37
4.7.1 - Comando de Armazenamento em um Endereço de Memória	38
4.7.2 - Comando de Transferência de um Programa	39
4.7.3 - Comando de Especificação de um Programa	42
4.7.4 - Comando de Apagamento de um Programa	43
4.7.5 - Comando de Preservação da Memória ou do Arquivo	43
4.7.6 - Comando de Recuperação da Memória ou do Arquivo	45
4.7.7 - Comando de Mapeamento do Diretório	45
4.7.8 - Finalização da Secção de Edição, Modificação e Transferência de Programas	46
4.8 - Comandos da Secção de Controle e Teste de Programas	46
4.8.1 - Comando Incondicional de Controle	47
4.8.1.1 - Controle do Modo de Execução	47
4.8.1.2 - Controle do Modo de Interrupção	48
4.8.1.3 - Comando de Controle do Canal de Interrupção	48
4.8.1.4 - Comando de Controle de Inicialização de um Programa	49
4.8.1.5 - Comando de Controle da Retomada de um Programa	49
4.8.2 - Comando Incondicional de Teste de Programas	50
4.8.3 - Comandos Condicionais	53
4.8.3.1 - Condição de Leitura da Memória	54
4.8.3.2 - Condição de Escrita na Memória	54
4.8.3.3 - Condição de Tempo	54
4.8.3.4 - Condição de Número de Instruções	55
4.8.3.5 - Condição de Status	55
4.8.3.6 - Comandos Associados	56
4.8.4 - Comando de Cancelamento	56
4.8.5 - Comando de Finalização da Secção de Controle e Teste de Programas	57

CAPÍTULO V

EXEMPLO DE APLICAÇÃO	59
----------------------------	----

Pág.

BIBLIOGRAFIA	63
APÉNDICE A	A.1
APÉNDICE B	B.1
APÉNDICE C	C.1

ABSTRACT

This manual was elaborated as the result of the standardization adopted for the software and hardware of the monitor system implemented in the HP 2116-B/DOS of INPE. Simultaneously, the purpose of this manual is to render operational the monitor system, aiming its utilization in the projects of the Digital and Analog Systems Program, presently under way. When this manual was elaborated, it was based on the MSc dissertation work developed by Eng. Fabio Vitaliano Filho, who initiated the project of the monitor system by means of its software implementation in the HP-2116 B/DOS System. This manual was equally based on the hardware implementation done by Eng. Mario Mammoli, responsible for the first version of the 2116 B - ADAPTER interface. It was our duty, the final tests, the completion of the software work and the subsequent operationality of the whole system.

LISTA DE FIGURAS

	Pág.
II.1 - Configuração do Sistema Monitor	8
II.2 - Diagrama de Blocos do Adaptador	11
II.3 - Diagrama de Transição das Secções de Monitoramento	19
III.1 - Estados de Parada do Microcomputador, sob a Supervisão do Sistema Monitor	26

CAPÍTULO I

INTRODUÇÃO

Atualmente, o uso de microcomputadores e, consequentemente, o uso de programação em projetos, não quais antes somente circuitos eram utilizados, vem se intensificando. Este fato deve-se principalmente ao grande aumento na velocidade dos microprocessadores, à sua flexibilidade, e ao seu custo, que se torna cada dia mais baixo. Associadas porém ao uso de programação, encontram-se algumas dificuldades que são inerentes e exigem para sua transposição eficiente, a utilização de ferramentas de suporte especializadas.

Neste capítulo, procuraremos apresentar uma visão global dos problemas que envolvem o desenvolvimento e implantação da programação de um microcomputador, e as ferramentas de suporte que podem auxiliar em cada etapa.

Por microcomputador, entenderemos um sistema digital, construído para fins gerais e que inclua um microprocessador.

1.1 - ETAPAS NA PROGRAMAÇÃO DE UM MICROCOMPUTADOR

O trabalho da programação, num projeto que utiliza um microcomputador, normalmente segue as seguintes etapas:

1.1.1 - DESENVOLVIMENTO DE PROGRAMAS

Durante esta fase, são desenvolvidos os programas que conduzirão o microcomputador à execução das tarefas específicas desejadas. O principal problema, encontrado nesta fase, reside na dificuldade de se programar em linguagem de máquina. Para minorá-la o usuário poderá utilizar-se de compiladores ou montadores, que permitem a programação em uma linguagem de nível mais alto que a de máquina.

1.1.2 - CARGA DE PROGRAMAS NA MEMÓRIA

Nesta etapa, os programas desenvolvidos devem ser carregados na memória do microcomputador, para posterior execução. Se nenhuma ferramenta auxiliar for utilizada, a carga dos programas deverá ser efetuada através de chaves, no painel do microcomputador, o que torna a operação bastante tediosa, mesmo no caso de programas de pequena extensão. Para auxiliar esta operação, o usuário deverá utilizar-se de um programa carregador, ou de um sistema monitor, que permitem que a carga de um programa se efetue a partir de sua cópia em um meio de armazenamento qualquer, como fita de papel, fita magnética ou cartões perfurados.

1.1.3 - TESTES DE PROGRAMAS

Os testes dos programas, desenvolvidos para o microcomputador, principalmente durante a etapa final, quando este pode estar conectado a dispositivos a serem por ele controlados, podem apresentar um alto grau de complexidade.

As dificuldades encontradas aqui devem-se, principalmente, à grande integração dos circuitos existentes em um microcomputador, e ao fato de alguns elementos desses circuitos só serem acessíveis por programação (é o caso dos registros internos do microprocessador, por exemplo). Também deve ser encarada, como fonte de dificuldade, a pouca familiaridade da maioria dos projetistas digitais com programação, o que a torna a principal suspeita, na ocorrência de falhas. As ferramentas mais utilizadas, nesta etapa, são os simuladores e sistemas monitor, que permitem ao usuário testar acuradamente a programação desenvolvida.

1.2 - FERRAMENTAS AUXILIARES

Nesta seção mostraremos uma visão global das principais ferramentas de suporte a projetos que utilizem microcomputadores.

1.2.1 - MONTADORES CRUZADOS ("CROSS ASSEMBLER")

Esta ferramenta auxiliar, durante a etapa de desenvolvimento de programas, consiste em um programa, desenvolvido para um computador já existente, (provavelmente escrito em uma linguagem de alto nível aceita por vários computadores correntes), que aceita como entrada um programa escrito em uma linguagem simbólica, desenvolvido para um microcomputador, e produz como saída o código objeto equivalente, que é aceito pelo microcomputador.

1.2.2 - COMPILADORES CRUZADOS

Também construídos para auxiliarem durante a etapa de desenvolvimento de programas, os compiladores cruzados, através da utilização de um computador já existente, permitem ao usuário que escreva os programas para o microcomputador em uma linguagem de alto nível. Como exemplos de compiladores cruzados já existentes, citamos o PL/M (Intel Corporation, PL/M programming manual) e o ALGOL/M (veja Correa, 1976).

1.2.3 - MONTADORES

Analogamente às ferramentas descritas acima, os montadores se destinam a facilitar o desenvolvimento de programas para microcomputadores. Diferem, porém, dos montadores cruzados, pelo fato de não se utilizarem de um computador auxiliar, sendo, a transformação de um programa simbólico em código objeto, efetuada pelo próprio microcomputador, através de um programa, para ele desenvolvido.

1.2.4 - CARREGADORES ("LOADERS")

Desenvolvidos para auxiliarem durante a fase de carga na memória do microcomputador, os carregadores são programas, construídos para o microcomputador, que executam a transferência de programas ou dados para a memória dele, a partir de um meio de armazenamento em uma unidade periférica qualquer.

1.2.5 - SIMULADORES

Utilizados durante os testes da programação desenvolvida para o microcomputador, os simuladores são programas, escritos em uma linguagem qualquer de programação ou de simulação que, quando executados em um computador qualquer, simulam o funcionamento de um microcomputador na execução de um programa (SIM 8080/B-6700). Dessa forma, o usuário pode verificar passo a passo o conteúdo de qualquer dos registros internos, de portas de entradas e saídas, ou de posições de memória, testando assim a programação desenvolvida.

O maior problema, associado ao uso de simuladores, reside na dificuldade encontrada na sua construção, quando o projeto em desenvolvimento, a ser simulado, envolver, além do microcomputador, um circuito associado relativamente complexo, principalmente no caso de projetos que controlam múltiplos processos, trabalhando em tempo real e dependendo de entradas não determinísticas.

1.2.6 - SISTEMA MONITOR

Entenderemos por monitor ou sistema monitor a ferramenta, constituída por uma programação de controle e monitoramento, residentes em um computador de suporte (geralmente um mini ou mesmo microcomputador), e de um adaptador que permite a comunicação entre o computador de suporte e o microcomputador em desenvolvimento.

O sistema monitor permite que o usuário utilize as facilidades (periféricos, memória, suporte de programação) do computador de suporte de forma a executar vários tipos de operações de controle e monitoramento do microcomputador em desenvolvimento.

Esta ferramenta é bastante útil na etapa de carga de programas na memória do microcomputador e na de testes da programação desenvolvida, podendo-se dispensar, com a sua utilização, o uso de carregadores e simuladores, evitando-se assim o uso adicional de memória do microcomputador, para conter o programa carregador, e a programação e repro

gramação do simulador durante o desenvolvimento do projeto.

O sistema monitor, ora implantado, utiliza, como computador de suporte, o minicomputador HP-2116B e seus periféricos. O monitoramento do microcomputador é efetuado através de comandos fornecidos pelo usuário, ao sistema monitor, via terminal de vídeo ("SCOPUS"). A comunicação entre o mini e o microcomputador se processa através de um adaptador (interface) especialmente desenvolvido para esta finalidade.

CAPÍTULO II

DESCRIÇÃO DO SISTEMA MONITOR

O sistema monitor é constituído basicamente por:

"HARDWARE"

- HP2116B e seus periféricos;
- Terminal de Video ("Scopus");
- Adaptador (Interface);
- Microcomputador em teste;

"SOFTWARE"

- Programa de controle de monitoramento.

Suas operações podem ser classificadas em quatro grupos gerais:

- Utilização de equipamento remoto.
- Operações de edição, modificação e transferência de programas e dados.
- Operações de teste de programas.
- Operações de controle do microcomputador.

2.1 - CONFIGURAÇÃO DO SISTEMA MONITOR

Na figura II.1, é apresentada a configuração do sistema implantado.

2.1.1 - HP-2116B

Trata-se de um minicomputador com as seguintes características:

- CPU - palavra de 16 bits
- ciclo de memória de 1,6 μ s.

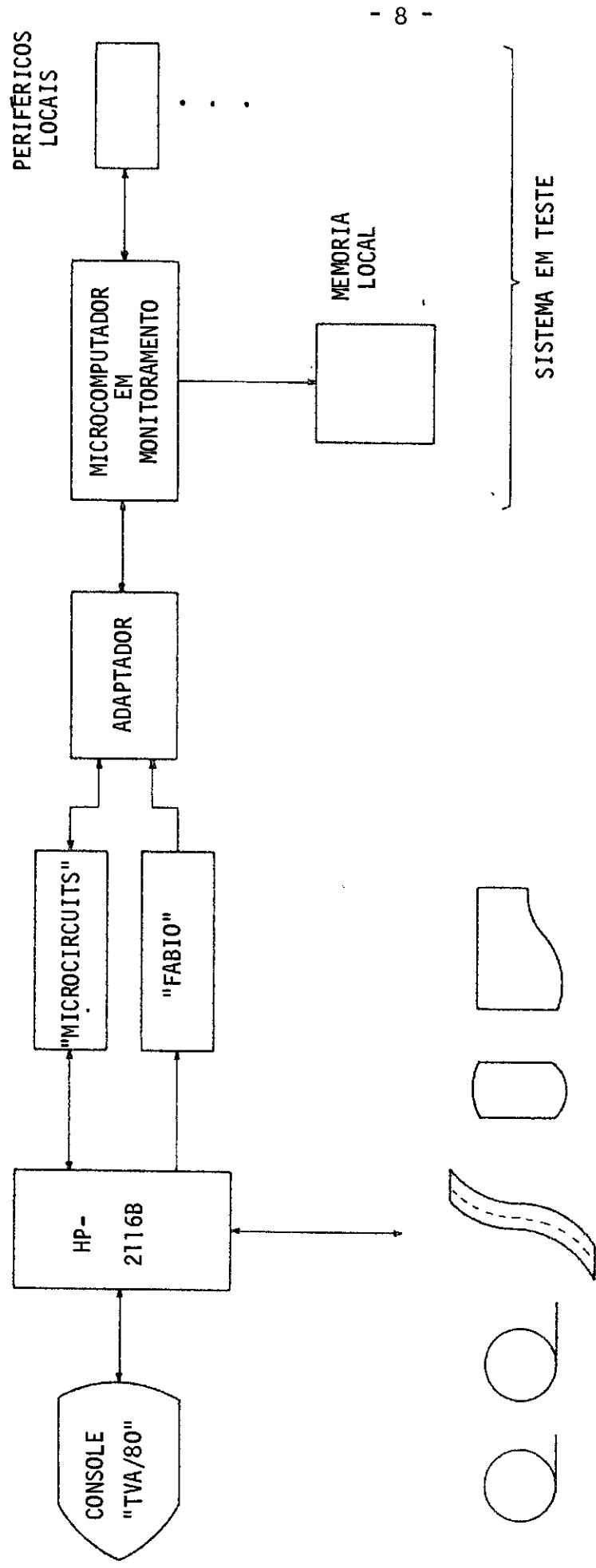


Fig. II.1 - Configuração do Sistema Monitor.

- 70 instruções básicas.
- 1024 palavras por página.

MEMÓRIA - 16 K palavras (configuração atual)

- PERIFÉRICOS - duas unidades de fita magnética
- uma leitora ótica de fita de papel
 - uma perfuradora de fita de papel
 - disco de 4,7 Megabytes
 - um terminal de video ("Scopus")
 - uma impressora

SISTEMA OPERACIONAL - DOS-III com ASSEMBLER, BASIC, HP-FORTRAN IV e HP-ALGOL.

2.1.2 - COMUNICAÇÃO COM O USUÁRIO

O sistema monitor utiliza, para a comunicação com o usuário, o terminal de video ("Scopus" TVA 80/20), por meio dele aceitando os comandos da linguagem de controle e monitoramento, e fornecendo, através desse terminal, mensagens que informam ao usuário seu estado de operação.

Como veremos mais a frente, devido à facilidade de implementação, as operações que causam a parada imediata do microcomputador serão comandadas não através de comandos fornecidos no terminal de video mas mediante o pressionamento de botões dispostos no painel do Adaptador.

2.1.3 - COMUNICAÇÃO COM O MICROCOMPUTADOR

A comunicação entre o mini e o microcomputador é efetuada através da interface ADAPTADDR, que permite, ao sistema monitor, o acesso às vias de dados, de endereços e de status do microcomputador em teste e circuitos de contagem do tempo e do número de instruções gastas, durante a execução de um programa.

Basicamente, o ADAPTADOR é um conjunto de registradores

e circuitos combinacionais de controle para o monitoramento do microcomputador.

O ADAPTADOR está conectado ao HP-2116 através da "MICRO CIRCUITS" - Porta de E/S nº 13B e da Interface FABIO - Porta de Controle nº 26B localizadas no "rack" de E/S do HP.

Na construção do ADAPTADOR, procurou-se torná-lo o menos dependente possível do tipo do microcomputador a ser monitorado, necessitando-se apenas que o microcomputador possua certas características básicas, para que possa ser conectado a esse Adaptador com o mínimo possível de alterações nos circuitos.

A Figura II.2 mostra o diagrama de blocos do ADAPTADOR com as interfaces "MICROCIRCUITS" e "FABIO".

2.1.4 - SISTEMA EM MONITORAMENTO

Devido à falta de padronização na arquitetura dos diversos microprocessadores existentes no mercado, um sistema monitor que pudesse monitorar microcomputadores baseados em qualquer microprocessador seria muito complexo de construir. Assim sendo, um sistema monitor foi desenvolvido, de modo que atendesse a uma parcela significativa desse conjunto sem contudo, aumentar em demasia sua complexidade.

As características impostas são as seguintes:

1) Acesso Assíncrono à Memória e aos periféricos

O acesso assíncrono, à Memória e aos periféricos, se refere ao fato de um microprocessador entrar num estado de espera, durante um acesso à memória ou periféricos. A saída desse estado de espera é dado por um sinal, gerado pelo próprio dispositivo, indicando fim de acesso. Não há limite máximo para o tempo de permanência nesse estado de parada. Com esta característica, o microprocessador poderá ter aces

DIAGRAMA DE BLOCOS - INTERFACE HP-2116B - MICROCOMPUTADOR

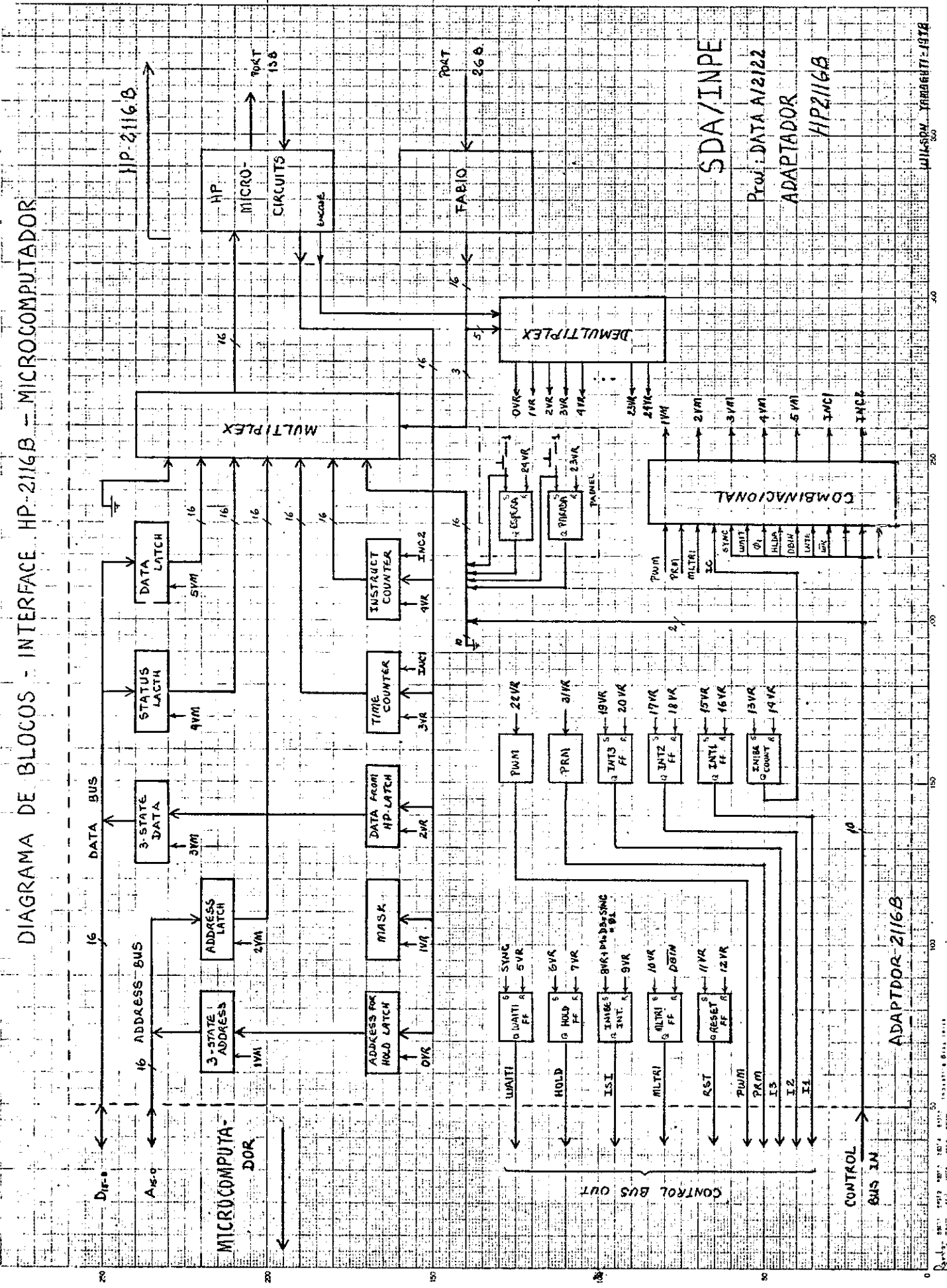


Fig. II.2 - Diagrama de Blocos do Adaptador.

so à memórias de qualquer velocidade e, no caso de acesso a periféricos, não há necessidade de sincronização por programação ou através de interrupções sucessivas.

No sistema monitor desenvolvido, o acesso assíncrono à memória foi utilizado nos seguintes fins:

- a) acesso à memória remota
- b) teste de condições referentes ao acesso à memória
- c) criação do estado fim de instrução
- d) Implementação do desvio do microcomputador para um programa que faz as listagens dos registros internos do microprocessador.

E o acesso assíncrono a periféricos foi utilizado somente na criação de periféricos remotos.

2) Acesso direto à memória

Quanto ao acesso direto à memória, para leitura e para escrita, não deve haver limite máximo para o tempo de duração desse acesso. Durante um acesso à memória, o microprocessador deve parar e suas entradas e saídas de dados e endereços, devem entrar num estado de alta impedância, permitindo que o sistema monitor faça uso dessas vias para acesso à memória local. Quando requisitado, um acesso direto à memória deve ser atendido nas seguintes condições:

- a) Quando um pedido de acesso direto à memória for solicitado pelo sistema monitor, com o microcomputador parado no estado de fim de monitoramento, esta solicitação deverá ser atendida imediatamente e, após terminado o acesso, o microcomputador deverá voltar ao estado de fim de monitoramento.
- b) Quando o sistema monitor requisitar um acesso direto à memória, com o microcomputador parado durante um acesso à dita memória, esta requisição deverá ser atendida imediatamente, após o fim

do acesso em curso ser indicado pelo sinal de sincronização pertinente. Quando o sinal, indicando fim do acesso direto à memória, for fornecido ao microcomputador pelo sistema monitor, este deverá continuar a executar a instrução durante a qual o acesso direto a memória foi requisitado.

Esta característica de acesso direto à memória, será utilizada pelo sistema monitor na execução de instruções de monitoramento que executem carga de programas ou dados na memória local ou executem listagens de blocos desta memória.

3) Reinicialização

Deve ser possível ao sistema monitor, sempre que solicitado, reinicializar o microcomputador, isto é, causar desvio para uma posição especificada de memória, que contenha o endereço de um programa a ser executado ou uma instrução de desvio para este endereço.

2.2 - FUNÇÕES DO SISTEMA MONITOR

Podemos classificar as operações do sistema monitor em quatro grupos gerais:

- a) Utilização de equipamento remoto;
- b) Operações de edição, modificação e transferência de programas e dados;
- c) Operações de teste de programas;
- d) Operações de controle do microcomputador.

2.2.1 - UTILIZAÇÃO DE EQUIPAMENTO REMOTO

Chamaremos de utilização de equipamento remoto à facilidade introduzida pelo sistema monitor, que permite que o microcomputador, em desenvolvimento, se utilize de endereços de memória e de periféricos que não façam parte física de seus circuitos, mas sejam uma extensão vir

tual permitida pelo sistema monitor. Desta maneira, a capacidade de memória e periféricos do microcomputador pode ser ampliada sem a necessidade da conexão de circuitos com estes elementos.

Por memória remota e periféricos remotos, entenderemos a extensão virtual da memória e dos periféricos do microcomputador, respectivamente. Por analogia, equipamentos locais serão os equipamentos (memória e periféricos) para os quais existam circuitos físicos no projeto (i.é. podem ser utilizados mesmo sem o monitoramento do sistema monitor).

No início de uma sessão de monitoramento, o usuário deverá definir os equipamentos locais e remotos que serão utilizados.

Abaixo, exemplificaremos, em linguagem informal, alguns comandos que poderiam definir os equipamentos locais e remotos utilizados:

- a) A memória local está compreendida entre os endereços 0 e 2047;
- b) A memória remota está compreendida entre os endereços 2048 e 4097;
- c) Não existem periféricos remotos.

2.2.2 - OPERAÇÕES DE EDIÇÃO, MODIFICAÇÃO E TRANSFERÊNCIA DE PROGRAMAS OU DADOS

Estas operações devem permitir, ao usuário, a manipulação de programas e de dados entre as unidades periféricas do minicomputador e a memória (local ou remota) do microcomputador.

Através das operações de edição, modificação e transferência de programas, o usuário poderá carregar programas ou dados na memória do microcomputador, a partir de um periférico qualquer do mini, modificar ou editar um programa já carregado, manter um arquivo em disco, contendo programas para posterior execução, ou obter listagens dos pro

gramas desenvolvidos.

Abaixo, em linguagem informal, exemplificamos algumas instruções que poderiam comandar a execução dessas operações:

- a) Arquive em disco o programa XT20 que se encontra na memória do microcomputador.
- b) Carregue no endereço 1500 de memória o código "10001010".
- c) Carregue o programa XT50 começando no endereço 1500 de memória a partir de sua cópia armazenada em fita magnética, disponível na unidade 8.
- d) Obtenha na impressora uma listagem do programa S0X1 que se encontra no arquivo em disco do sistema.

2.2.3 - OPERAÇÃO DE TESTE DE PROGRAMAS

Estas operações, quando comandadas, permitirão ao usuário testar os programas desenvolvidos para o microcomputador, através da obtenção de informações sobre o conteúdo de endereços de memória, registros, canais de entrada e saída do microcomputador ou, ainda, informações relativas ao tempo gasto ou ao número de instruções executadas durante o desenvolvimento de um programa. Deverão possibilitar, ainda, que o usuário associe a obtenção de informações sobre o microcomputador, vistas acima, com condições sobre o respectivo estado geral, de forma que as informações somente serão fornecidas ao usuário quando a condição associada tiver sido satisfeita.

Abaixo, forneceremos uma lista das condições possíveis de serem associadas:

- a) Quando o microcomputador tiver tido acesso a um determinado endereço de memória para leitura.
- b) Quando o microcomputador tiver tido acesso a um determinado endereço de memória para escrita.
- c) Após uma certa quantidade de tempo ter decorrido na execução de um programa.

- d) Após um determinado número de instruções ter sido executado no desenvolver de um programa.
- e) Quando o microcomputador tiver atingido um determinado "STATUS".

Normalmente, um microcomputador possui algumas saídas que informam ao circuito externo a natureza das tarefas sendo executadas. Por status entenderemos uma configuração válida dessas saídas.

Abaixo exemplificamos algumas instruções, que comandariam operações de teste de programas, em linguagem informal, explícita.

- a) Forneça uma listagem do conteúdo do registro acumulador;
- b) Indique quanto tempo decorreu desde a inicialização do programa que está sendo executado;
- c) A cada 25 ciclos liste o endereço da memória 1895;
- d) Quando o endereço 1791 da memória for lido, indique o tempo decorrido desde a inicialização do programa que está sendo executado.

2.2.4 - OPERAÇÕES DE CONTROLE DE MICROCOMPUTADOR

Através do comando destas operações o usuário deverá poder inicializar, parar ou continuar a execução de um programa, selecionar o modo de operação do microcomputador, de forma que os programas sejam executados passo a passo (i.é uma instrução por vez) ou automaticamente, ativar um canal de interrupção do microcomputador, de forma a simular a ocorrência de uma interrupção ou, ainda, controlar o sistema de interrupção de forma que o microcomputador aceite ou não interrupções durante o desenvolvimento de um programa.

Analogamente, às operações de testes de programas, estas operações devem poder ser associadas a condições, de forma a somente serem executadas quando a condição associada for satisfeita, sendo as condições, possíveis de serem associadas, as mesmas vistas na secção 2.2.3.

Alguns exemplos de comandos de controle são:

- a) Inicialize a execução do programa XYK.
- b) Permita que as interrupções sejam aceitas.
- c) Execute o programa XYK passo a passo.
- d) Quando ler o endereço 1235 de memória, pare.
- e) A cada 250 ciclos ative o canal 4 de interrupções.

2.3 - A PROGRAMAÇÃO DE CONTROLE E MONITORAMENTO

A programação do sistema monitor encontra-se dividida em três secções:

- a) Secção de configuração do microcomputador;
- b) Secção de edição, modificação e transferência de programas;
- c) Secção de teste e controle de programas.

Cada secção é encarregada do reconhecimento das instruções e das rotinas, que executarão as operações de configuração do microcomputador, de edição, modificação e transferência de programas ou de teste de programas e controle do microcomputador, respectivamente.

A seguir, antes de examinarmos detalhadamente alguns aspectos relevantes de construção, e as instruções aceitas pelo sistema monitor em cada uma destas secções, resumiremos de maneira breve, como se deverá processar normalmente, uma sessão de monitoramento completa.

2.4 - UMA SESSÃO DE MONITORAMENTO

Os programas encarregados do controle e monitoramento, acham-se armazenados no disco do minicomputador, em forma objeto, e assim a inicialização de uma sessão de monitoramento se processará através

dos procedimentos que inicializarão a execução da primeira secção (de configuração do microcomputador) dessa programação.

Durante a primeira secção, o usuário deverá informar ao sistema monitor, qual a configuração planejada para o microcomputador, (tamanho da memória local, da memória remota e quais os periféricos locais e remotos utilizados e, finalmente, fornecerá ao sistema monitor uma instrução que encerrará a execução desta secção, e inicializará a secção de edição, modificação e transferência de programas.

No decorrer da mencionada secção, o sistema monitor aceitará instruções que permitirão ao usuário comandar a execução das operações de manipulação de programas, vistas no item 2.2.2. A seguir, após a execução das operações desejadas, o usuário poderá comandar a inicialização da secção de configuração, caso deseje reconfigurar o microcomputador, ou inicializar a terceira secção.

Durante o processamento da secção de controle e teste de programas, as operações, descritas nos itens 2.2.3 e 2.2.4, podem ser executadas sob comando do usuário.

Após encerrar o processamento dos testes e controles desejados no microcomputador, o usuário poderá reinicializar qualquer das outras duas secções, conforme as operações que deseje executar, ou poderá encerrar a sessão de monitoramento, através do comando adequado.

Na Figura II.3, estão esquematizadas as relações entre as secções e, através dos comandos de finalização, a mudança de secção.

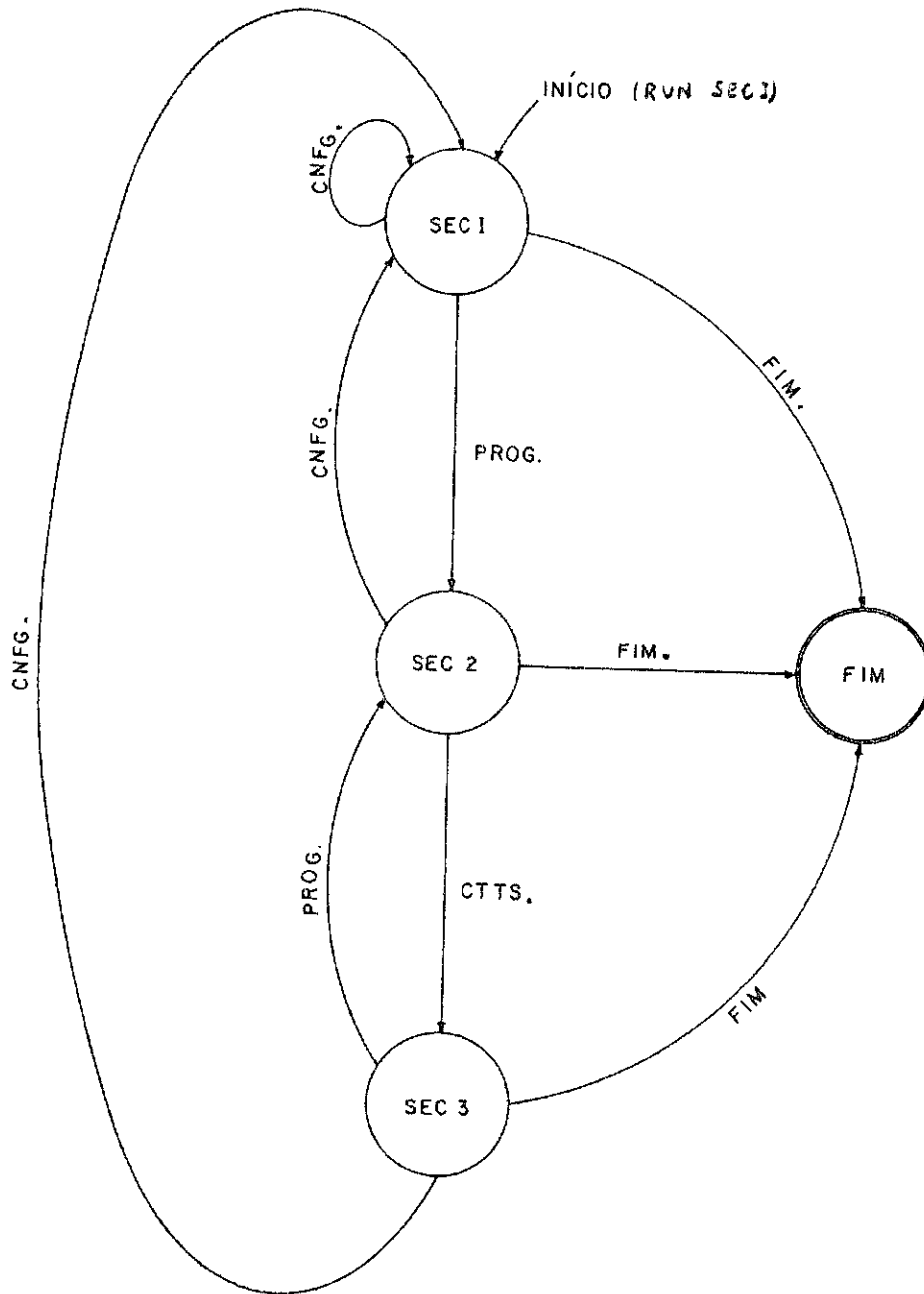


Fig. II.3 - Diagrama de Transição das secções de Monitoramento.

CAPÍTULO III

CONTROLES DO MICROCOMPUTADOR E ESTADOS DE PARADA

As operações de controle do microcomputador, são executadas pela manipulação dos comandos e botões de controle, disponíveis no sistema monitor. Os possíveis controles disponíveis no sistema monitor, bem como os estados de parada do microcomputador são descritos neste capítulo.

Os comandos de controle e testes podem ser classificados em 2 tipos:

- Imediatos.
- Condicionais.

Os imediatos permitem ao usuário obter informações e/ou execução de operações de controle, logo após o seu fornecimento ao sistema monitor, enquanto que os condicionais são aqueles associados a uma condição, de modo que a operação de controle é executada quando esta condição for satisfeita.

3.1 - CONTROLE DO MODO DE EXECUÇÃO

Através deste controle, o usuário deverá poder selecionar o modo pelo qual o microcomputador, executará um programa, havendo duas escolhas possíveis.

- a) Modo passo a passo
- b) Modo automático

Quando um modo passo a passo for selecionado, as instruções de um programa do microcomputador serão executadas passo a passo

(i.é, após a execução de cada instrução o microcomputador para), sendo o passo seguinte comandado pelo usuário através da operação de controle adequada.

A seleção do modo de execução automático, acarretará a execução automática de um programa pelo microcomputador - uma instrução após a outra - sem a necessidade da intervenção do usuário para comandar cada passo.

3.2 - CONTROLE DO SISTEMA DE INTERRUPÇÃO

O sistema monitor, deverá controlar os pedidos de interrupções que ocorram no microcomputador, permitindo ou não que estas sejam aceitas, havendo assim duas opções de escolha possíveis para o usuário:

- a) Interrompível
- b) Não interrompível

Quando foi selecionado o controle "Interrompível" o sistema monitor não interferirá em nenhum dos canais de interrupção do microcomputador, deixando que as interrupções se processem normalmente. Se, no entanto, o controle "Não interrompível" for selecionado, o sistema monitor inibirá externamente o sistema de interrupções do microcomputador, não permitindo assim o atendimento de quaisquer interrupções.

Deve-se, contudo, notar que os microcomputadores, de modo geral, controlam internamente o seu sistema de interrupções e, dessa forma, mesmo se o controle externo "Interrompível" for selecionado, uma interrupção só será aceita se o sistema interno o permitir.

3.3 - CONTROLE DO ESTADO DE PARADA

Como vimos, quando a seção de teste e controle do sistema monitor estiver sendo processada, o microcomputador poderá traba

lhar na execução de um programa do usuário, sendo o funcionamento do microcomputador monitorado pelo sistema.

A execução de operações de teste e controle de programas serão, contudo, requisitadas através de instruções que somente poderão ser fornecidas pelo usuário com o microcomputador parado, constituindo exceções as operações que levam o microcomputador a parar imediatamente que, por serem aceitas com o microcomputador funcionando, serão, para facilidade de implementação, comandadas por pressionamento de botões. Além do mais, o conjunto de instruções de teste e controle aceito, depende do estado particular em que se encontra o microcomputador quando parado. Desse maneira, um estudo detalhado dos possíveis estados do microcomputador, quando parado, torna-se necessário e será feito a seguir.

3.3.1 - ESTADOS DE PARADA DO MICROCOMPUTADOR

São três os estados de parada do microcomputador, aparentes ao usuário, quando sob supervisão do sistema monitor:

a) Estado de fim de monitoramento

Chamaremos de estado do fim de monitoramento ao estado em que se encontra o microcomputador, sempre após a execução de uma instrução local de parada, com o controle do sistema de interrupções do sistema monitor desativado, inibindo assim todas as interrupções que possam ocorrer.

Para evitar confusão, chamaremos de instruções locais, às instruções executáveis, do repertório de instruções do microcomputador, em teste pelo sistema monitor. Por instrução local de parada, entenderemos toda a instrução que, quando executada pelo microcomputador, causa a suspensão das suas atividades de busca e execução de instruções, até que uma interrupção ocorra ou um sinal de reinicialização seja causado externamente.

Quando o microcomputador estiver neste estado de parada, o usuário poderá, através do comando de teste e controle adequado, solicitar a execução imediata ou condicional de qualquer operação de teste e controle, com exceção das operações que exigem entrada imediata no estado de fim de instrução, vide c - estado de fim de instrução.

b) Estado de espera de interrupções

Por estado de espera de interrupções, entenderemos o estado em que se encontra o microcomputador, após a execução de uma instrução local de parada, com o controle de interrupções ativado, permitindo a passagem livre das interrupções que ocorrerem.

Deve-se notar que, mesmo neste estado, certas interrupções podem não ser atendidas, devido às máscaras controladas internamente no microcomputador.

Sempre que, no fim da execução de um programa, desejarmos atender a interrupções, este estado deverá ser utilizado. Quando o microcomputador estiver neste estado, somente poderá ser executada a operação de controle responsável pela entrada imediata do microcomputador no estado de fim de monitoramento, descrito acima.

c) Estado de fim de instrução ou pausa

Ao contrário dos dois estados de parada, descritos acima, que envolvem a execução, pelo microcomputador, de uma instrução local de parada, a entrada no estado de pausa se dará quando o microcomputador estiver executando uma instrução local, que não seja de parada, e um sinal for lançado pelo sistema monitor, na entrada adequada do microcomputador, causando a suspensão das atividades deste, após o término da execução da instrução local presente.

Quando a retomada do programa for solicitada, através da instrução de controle adequada, o sistema monitor lançará um sinal na entrada para isto adequado ao microcomputador, e este retomará a execução no programa, a partir do ponto em que a entrada no estado de pausa foi efetuada.

Este estado será utilizado no fim de cada instrução, no modo passo a passo, ou sempre que o usuário desejar uma pausa, com possibilidade de continuação (para o fornecimento de comandos de teste e controle, por exemplo).

Com exceção da operação de inicialização de um programa, que é imediata, e das que exigem encerramento desta seção, todas as demais operações, imediatas ou condicionais podem ser efetuadas neste estado.

O diagrama da Figura III.1, mostra o relacionamento dos diversos estados de parada sob a supervisão do Sistema Monitor.

Após termos visto os estados de parada do microcomputador, aparentes ao usuário, quando sob supervisão do sistema monitor, descreveremos nas próximas seções, como se processam a entrada e a saída do microcomputador em cada um desses estados.

3.3.2 - ENTRADAS E SAIDAS NO ESTADO DE FIM DE MONITORAMENTO

A entrada neste estado, se dará nas seguintes condições

- a) O usuário pressiona o botão que comanda a operação de entrada imediata neste estado.
- b) O microcomputador executa uma instrução local de parada, e o controle do sistema de interrupções, do sistema monitor, fica desativado (no estado não interrompível).

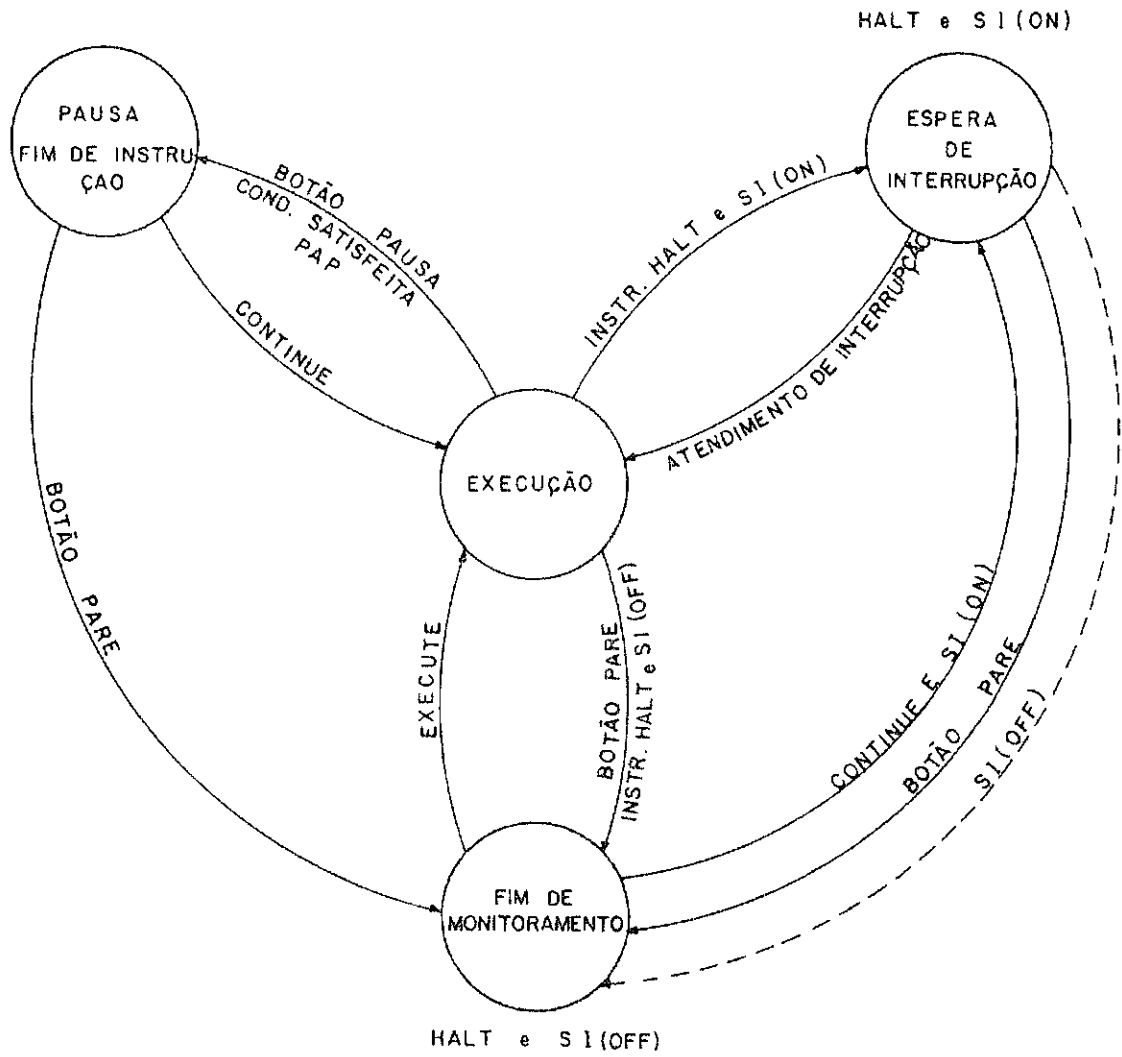


Fig. III.1 - Estados de Parada do microcomputador, sob a supervisão do Sistema Monitor.

A saída do estado de fim de monitoramento se dará nas seguintes condições:

- a) O usuário fornece ao sistema monitor a instrução imediata, que comanda a operação de inicialização de execução de um programa.
- b) O usuário ativa o controle de interrupções, através do fornecimento da instrução de controle adequada e, a seguir, fornece ao sistema monitor uma instrução de controle imediata, que comanda a retomada do programa, a partir do ponto em que foi encontrada a instrução local de parada, que causou a entrada no estado de fim de monitoramento. Neste caso o microcomputador entrará no estado de espera de interrupções.

3.3.3 - ENTRADA E SAIDA NO ESTADO DE ESPERA DE INTERRUPTÕES

A entrada do microcomputador neste estado se processará nas seguintes condições:

- a) O microcomputador executa uma instrução local de parada e, após a execução desta instrução, o controle de interrupções, pelo sistema monitor se encontra ativado (interrompível).
- b) O microcomputador se encontra no estado de fim de monitoramento e o usuário, através da instrução de controle adequado, ativa o controle de interrupções exigindo a seguir a retomada do programa, a partir do ponto em que a instrução local de parada, que causou a entrada no estado de fim de monitoramento foi encontrada.

A saída deste estado se dará quando:

- a) Uma interrupção não mascarada internamente ocorrer. Neste caso o microcomputador deixará este estado de parada e passará a execução do programa, associado com a interrupção que ocorreu.

- b) O usuário pressiona o botão que comanda a operação de entrada imediata no estado de fim de monitoramento.

3.3.4 - ENTRADAS E SAIDAS NO ESTADO DE FIM DE INSTRUÇÃO OU PAUSA

Quando alguma das situações abaixo ocorrer, o microcomputador entrará no estado de pausa:

- a) Após a execução pelo microcomputador de uma instrução local que não seja de parada, com o controle de execução selecionando um modo passo a passo.
- b) Quando o usuário pressionar o botão que comanda a execução da operação de entrada imediata neste estado, com o microcomputador executando uma instrução local que não seja de parada. Neste caso, após o término da execução da instrução presente, o microcomputador entrará no estado de pausa.
- c) Quando a condição de um comando de teste e controle condicional, anteriormente fornecida, for satisfeita, e um dos comandos associados a esta condição exigir entrada no estado de pausa. Isto, desde que a condição satisfeita não envolva a execução, pelo microcomputador, de uma instrução local de parada pois, neste caso, ele entraria no estado de fim de monitoramento ou no de espera de interrupção, conforme o controle do sistema de interrupções estivesse ativado ou não.

A saída do estado de pausa se processará mediante a ocorrência de alguma das seguintes situações:

- a) O usuário pressiona o botão que comanda a operação de entrada imediata no estado de fim de monitoramento.
- b) O usuário fornece ao sistema monitor um comando de controle imediato, que comanda a operação de retomada do programa que estava sendo executado antes da entrada no estado de pausa.

3.4 - CONTROLE DA RETOMADA DE PROGRAMAS

A operação de retomada de um programa será comandada por um comando imediato, que pode ser fornecida ao sistema monitor no estado de fim de monitoramento ou no de pausa, com efeitos diferentes, conforme seja o estado no qual é fornecida.

- a) Se fornecida no estado de fim de monitoramento, então, se o controle de interrupção foi previamente ativado, o microcomputador passará ao estado de espera de interrupções, considerando como sendo a última instrução local executada, a instrução de parada que causou a entrada no estado de fim de monitoramento e encontrando-se, na sua via de endereços, o endereço dessa instrução. Se, contudo, o controle de interrupções se encontrar desativado, o microcomputador permanecerá no estado de fim de monitoramento, sendo esta instrução sem qualquer efeito.
- b) Se fornecido no estado de pausa, este comando, causará a retomada do programa pelo microcomputador, a partir do ponto em que se deu a entrada no estado de pausa.

3.5 - CONTROLE DE INICIALIZAÇÃO DE UM PROGRAMA

A operação de inicialização de um programa, será acionada através de comando de controle imediato, que somente será aceito pelo sistema monitor quando o microcomputador se encontrar no estado de fim de monitoramento, devendo especificar o programa a ser executado.

As especificações desse programa devem estar relacionados no diretório da memória, estando portanto o programa carregado.

3.6 - CONTROLE DOS CANAIS DE INTERRUPTÃO

O sistema monitor permite ao usuário que forneça pedidos de interrupção ao microcomputador, em até três canais de interrupção diferentes.

Esta operação é efetuada por comandos de controle imediatos ou condicionais, que permitem ativar ou desativar cada um dos três canais de interrupção disponíveis no sistema monitor.

Deve-se alertar que todos os três canais de interrupções, controlados pelo sistema monitor, estarão, como os demais, sujeitos ao controle de interrupções pelo sistema monitor, e ao controle interno do microcomputador.

< letra > :: = A | B | C | D | E | F | G | H | I | J | K | L | M |
N | O | P | Q | R | S | T | U | V | W | X | Y | Z

< dígito > :: = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

< caracteres especiais > :: = . | , | ; | * | + | = | < espaço >

< espaço > :: = { uma ou mais posições em branco }

< palavras reservadas > :: = < palavra chave > |
< conectivo >

< palavra chave > :: = FIM | CNFG | PROG | CTTS | NAO |
PERIFERICO | CARREGUE | TRANSFIRA |
ESPECIFIQUE | APAGUE | PRESERVE | RECUPERE |
MAPEIE | ATIVE | DESATIVE | EXECUTE |
CONTINUE | TEMPO | INSTRUCAO | DADOS |
ENDERECO | REGISTRO | MEMORIA | LISTE |
PAUSA | LE | ESCREVE | STATUS |
CANCELE

< conectivo > :: = ATE | ROTINA | ENDERECO | CONTEUDO |
PAP | SI | 1INT | 2INT | 3INT | BINARIO |
OCTAL | DECIMAL | HEXADECIMAL

4.3 - CONSTANTES

Sintaxe

< número > :: = B < número binário > |
O < número octal > |
D < número decimal > |
H < número hexadecimal >

< número binário > ::= < dígito binário > |
< número binário > < dígito binário >

< dígito binário > ::= 0 | 1

< número octal > ::= < dígito octal > |
< número octal > < dígito octal >

< dígito octal > ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7

< número decimal > ::= < dígito > |
< número decimal > < dígito >

< número hexadecimal > ::= < dígito hexadecimal > |
< número hexadecimal > < dígito hexadecimal >

< dígito hexadecimal > ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A |
B | C | D | E | F

4.4 - IDENTIFICADORES

Sintaxe:

< identificador > ::= < letra ou dígito > |
< identificador > < letra ou dígito >

< letra ou dígito > ::= < letra > |
< dígito >

4.5 - ESTRUTURA DO PROGRAMA

Sintaxe:

< estrutura do programa > ::= < SEC 1 >

```
< SEC 1 > :: = < definições de memória e periféricos >  
              < finalização SEC 1 >  
  
< finalização SEC 1 > :: = FIM . | CNFG . < SEC 1 > | PROG . < SEC 2 >  
  
< SEC 2 > :: = < lista comando SEC 2 > < finalização SEC 2 > |  
              < finalização SEC 2 >  
  
< finalização SEC 2 > :: = FIM . | CNFG . < SEC 1 > | CTTS . < SEC 3 >  
  
< SEC 3 > :: = < lista comando SEC 3 > < finalização SEC 3 > |  
              < finalização SEC 3 >  
  
< finalização SEC 3 > :: = FIM . | CNFG . < SEC 1 > | PROG . < SEC 2 >
```

Basicamente, uma sessão de monitoramento é composta de três secções (SEC 1, SEC 2 e SEC 3) dadas na sintaxe de estrutura do programa. SEC 1, SEC 2 e SEC 3 correspondem, respectivamente, à secção de configuração do microcomputador, à secção de edição, modificação e transferência de programas e à secção de teste e controle de programas, descritas no Capítulo II. Note-se que a transição de uma secção a outra é efetuada através dos comandos de finalização (vide fig II.3). Nos itens subsequentes é dada a sintaxe de cada uma dessas secções, juntamente com suas definições e comandos.

4.6 - COMANDOS DA SECÇÃO DE CONFIGURAÇÃO (SEC 1)

Sintaxe:

```
< SEC 1 > :: = < definições de memória e periféricos >  
              < finalização SEC 1 >
```

< definições de memória e periféricos > ::= < def. memória local > |
< def. memória remota > . < def. end. inferior dos perif. remotos > . |
< def. memória local > . < def. memória remota > . < def. end. infe-
rior dos perif. remotos > . < assoc. periféricos remotos > .

4.6.1 - DEFINIÇÃO DE MEMÓRIA LOCAL

Sintaxe:

< def. de memória local > ::= < número > ATE < número > |
NAO

Os endereços de memória local especificados devem cor-
responder sempre a endereços físicos de unidades de memória existente.
NAO para o caso de não haver memória local.

4.6.2 - DEFINIÇÃO DE MEMÓRIA REMOTA

Sintaxe:

< def. de memória remota > ::= < número > ATE < número > |
NAO

Quando um bloco de endereços especificados como memória
remota, corresponder a endereços de unidades de memória existente, estas
unidades não serão utilizadas pelo microcomputador, prevalecendo assim,
a especificação remota.

NAO utilizada no caso de não haver memória remota.

4.6.3 - DEFINIÇÃO DO ENDEREÇO INFERIOR DOS PERIFÉRICOS REMOTOS

Sintaxe:

< def. do endereço inferior dos
periféricos remotos > ::= < número > |
NAO

O < número > fornecido, correspondente ao endereço inferior dos periféricos remotos, deve estar compreendida entre 0 e 255 e de maneira coerente à associação de periféricos remotos.

Caso seja fornecido NAO, o endereço inferior é assumido como sendo 256.

4.6.4 - ASSOCIAÇÃO DE PERIFÉRICOS REMOTOS

Sintaxe:

```
< assoc. periféricos remotos > :: = < assoc. periférico remoto > . |
                                     < assoc. periféricos remotos >
                                     < assoc. periférico remoto > .

< assoc. periférico remoto > :: = PERIFÉRICO < número > ROTINA
                                     < número >
```

Através da associação de periféricos remotos, associamos o endereço do periférico remoto com um parâmetro da rotina PRNT do sistema Monitor. Esta rotina, por meio das facilidades do minicomputador e do ADAPTADOR, permite a comunicação entre os computadores, simulando desta maneira, o funcionamento de vários periféricos remotos ao microcomputador.

Os endereços especificados para os periféricos remotos não devem corresponder a endereços de periféricos locais existentes, pois no caso de acesso do microcomputador para leitura do periférico, poderia ocorrer um curto circuito, devido aos dados fornecidos pelo periférico existente e pelo sistema monitor (através do ADAPTADOR de comunicações) serem diferentes.

4.6.5 - FINALIZAÇÃO DA SECÇÃO DE CONFIGURAÇÃO (SEC 1)

Sintaxe:

< finalização SEC 1 > ::= FIM . | CNFG . < SEC 1 > | PROG .
< SEC 2 >

Através dos comandos de finalização se faz a transição de secções. No caso de FIM, termina-se a monitoração do sistema em teste.

4.7 - COMANDOS DA SECÇÃO DE EDIÇÃO, MODIFICAÇÃO E TRANSFERÊNCIA DE PROGRAMAS (SEC 2)

Sintaxe:

< SEC 2 > ::= < lista comando SEC 2 > < finalização SEC 2 > |
< finalização SEC 2 >

< lista comando SEC 2 > ::= < comando SEC 2 > |
< lista comando SEC 2 > < comando SEC 2 >

< comando SEC 2 > ::= < armazenamento em endereço de memória > . |
< transf. de um programa > . |
< especificação de um programa > . |
< apagamento de um programa > . |
< preservação da memória ou do arquivo > . |
< recuperação da memória ou do arquivo > . |
< mapeamento do diretório > . |

Algumas observações úteis no entendimento desta secção são feitas a seguir:

- a) O microcomputador deverá estar parado, durante esta fase de manipulação de programas, e somente na secção de teste e controle poderá executar um programa;
- b) O sistema monitor deverá manter um diretório para os programas carregados na memória ou no arquivo. Estes diretórios deverão

conter informações que especifiquem os programas, e que serão denominados especificações de programa (vide comando de especificação de um programa);

- c) A execução de um programa na memória somente será permitida se as especificações do programa se encontrarem no diretório da memória, sendo para lá transferida quando a carga do programa na memória for efetuada;
- d) Quando fita magnética ou de papel forem utilizadas no armazenamento de programas, cada registro físico destes meios de armazenamento será constituído por 128 palavras de 16 bits;
- e) Na inicialização desta secção, o sistema monitor cria uma imagem da memória local numa área, em disco, e as transferências de códigos, destinadas à memória local, são, na verdade, efetuadas para essa área. E somente na inicialização da terceira secção, o sistema atualizará realmente a memória local.

4.7.1 - COMANDO DE ARMAZENAMENTO EM UM ENDEREÇO DE MEMÓRIA

Sintaxe:

< armazenamento em endereço de memória > ::= CARREGUE ; < especificação de carga >

< especificação de carga > ::= ENDEREÇO < número > CONTEUDO
< número > |
CONTEUDO < número > |
< especificação de carga > ; ENDEREÇO
< número > CONTEUDO < número > |
< especificação de carga > ; CONTEUDO
< número >

Através da utilização deste comando, o usuário poderá armazenar em endereços especificados da memória (local ou remota) do microcomputador, dados ou instruções, cuja configuração é representada pelo valor numérico fornecido.

Esta operação é bastante útil na modificação de códigos ou dados de programas, ou mesmo quando auxiliada pela operação de especificação de programa, para a carga de pequenos programas na memória do microcomputador.

4.7.2 - COMANDO DE TRANSFERÊNCIA DE UM PROGRAMA

Sintaxe:

< transf. de um programa > ::= TRANSFIRA < número > , < número >
{ , < identificador > }

< código > ::= BINARIO | OCTAL | DECIMAL | HEXADECIMAL

Este comando, permitirá ao usuário a obtenção de listas ou a transferência de programas entre os periféricos do minicomputador, o arquivo de programas e a memória do microcomputador.

O primeiro campo numérico deste comando indicará o elemento onde o programa deve ser transferido (destino) e o segundo, o elemento onde o programa se encontra armazenado (fonte) sendo esses valores numéricos associados aos elementos de acordo com o Apêndice C..

As chaves { } foram adicionadas na sintaxe do comando visando simplificação e com o seguinte significado:

- Quando o destino especificado for a impressora ou o teletipo (ou terminal de vídeo), o CÓDIGO no qual a listagem deverá ser fornecida deve ser indicado;

- Quando a fonte de transferência for a memória ou o arquivo, o nome do programa a ser transferido, deve ser indicado por IDENTIFICADOR.

A seguir são apresentadas algumas observações quanto ao MODO de transferência:

a) DESTINO: MEMÓRIA ou ARQUIVO

Quando o destino do programa for a memória ou o arquivo, as especificações desse programa são transferidas para o diretório desse elemento.

b) DESTINO: FITA MAG. OU PAPEL

Quando o destino do programa for fita magnética ou de papel, as especificações são transferidas para o primeiro registro físico desse meio de armazenamento, no FORMATO abaixo:

- 19 pal. do reg.: contém o nome do programa (2 caracteres em ASCII)
- 29 pal. do reg.: contém o endereço inicial de carga em formato binário absoluto.
- 39 pal. do reg.: contém o endereço final de carga em formato binário absoluto.
- 49 pal. do reg.: contém o endereço inicial de execução em formato binário absoluto.

As demais palavras deste registro não são utilizadas.

A seguir, os códigos constituintes do programa são armazenados nos N registros físicos subsequentes com

N = menor inteiro maior ou igual a

$$\left(\begin{array}{l} \text{endereço final} \\ \text{de carga} \end{array} - \begin{array}{l} \text{endereço inicial} \\ \text{de carga} \end{array} + 1 \right) / 128$$

c) DESTINO: TELETIPO OU IMPRESSORA

Quando o destino da transferência for o teletipo (ou terminal de video) ou a impressora, o usuário obterá uma listagem das especificações do programa transferido no formato:

NOME	ENDER.INICIAL	ENDER.FINAL	ENDER.INIC.EXECUÇÃO
< identificador >	< número >	< número >	< número >

e a seguir, os códigos constituintes do programa, são colocados no formato:

ENDEREÇO	CONTEUDO
< número >	< número >
< número >	< número >
.	.
.	.
.	.

d) FONTE: MEMÓRIA OU ARQUIVO

Quando a fonte do programa a ser transferido, for a memória ou o arquivo, suas especificações devem se encontrar no diretório desse elemento.

e) FONTE: FITA MAG. OU PAPEL

Quando a fonte do programa for a unidade de fita magnética ou de papel, o programa deve-se encontrar armazenado nesse meio, no formato descrito na observação b) DESTINO: FITA MAG. OU PAPEL.

4.7.3 - COMANDO DE ESPECIFICAÇÃO DE UM PROGRAMA

Sintaxe:

< especificação de um programa > :: = ESPECIFIQUE < identificador > , < número > ,
< número > , < número >

O comando de especificação de um programa permite ao usuário relacionar, como programas, os conteúdos de blocos de memória, que não façam parte de nenhum programa já relacionado no diretório. Por exemplo, blocos cujas especificações foram retiradas do diretório pelo comando de apagamento de um programa.

Para que um bloco de memória possa ser aceito como um programa, pelo sistema monitor, o diretório da memória deve necessariamente conter as especificações do programa, carregado nos endereços deste bloco.

O formato e o significado de cada campo de um registro de especificações de programa é dado abaixo:

- 1º campo: Nome do programa (4 caracteres quaisquer);
- 2º campo: Endereço inicial da memória, a partir do qual os códigos do programa devem ser carregados (valor inteiro positivo menor que 64 K);
- 3º campo: Endereço final da memória, ocupado pelo programa (valor inteiro positivo menor que 64 K e maior ou igual ao anterior);
- 4º campo: Endereço da primeira instrução do programa. É o endereço para o qual o microcomputador deve ramificar quando a execução do programa for requisitada (valor inteiro positivo compreendido entre o endereço inicial de carga e o endereço final de carga).

4.7.4 - COMANDO DE APAGAMENTO DE UM PROGRAMA

Sintaxe:

< apagamento de um programa > ::= APAGUE < número > , < identificador >

Este comando, permitirá ao usuário que retire do diretório da memória (9) ou do arquivo (2), conforme especificado por <número>, o programa identificado no sistema pelo nome indicado em <identificador>.

Quando fornecido ao sistema monitor, este comando tem, como única função, a retirada das especificações do programa referenciado no diretório, permanecendo inalterada a área (do arquivo ou da memória) onde esse programa se encontra, podendo porém esta área vir a ser ocupada por outros programas, o que não é permitido, se a área pertencer a algum programa especificado. Dessa forma, no caso da retirada de um programa da MEMÓRIA, se o usuário souber aonde este se encontra armazenado, poderá restaurar este programa mediante o fornecimento de um comando de especificação de um programa, antes que os conteúdos dessa área sejam modificados por um comando de transferência qualquer.

No caso da retirada de um programa do ARQUIVO, a restauração só poderá ser efetuada, mediante o fornecimento de um comando de transferência desse programa para o arquivo, a partir de sua cópia na memória ou em outra unidade periférica qualquer. Isto se deve ao fato de o usuário não dispor de acesso às informações sobre a posição ocupada por um programa no arquivo, não sendo assim possível especificar-se uma área do arquivo, como é o caso com áreas de memória, através do comando de especificação de um programa.

4.7.5 - COMANDO DE PRESERVAÇÃO DA MEMÓRIA OU DO ARQUIVO

Sintaxe:

< preservação da memória ou do arquivo > ::= PRESERVE < número > , < número >

Quando termina uma sessão de monitoramento e o sistema monitor deixa de ser utilizado, os programas armazenados na memória remota do microcomputador, ou no arquivo de programas, poderão vir a ser destruídos por outros usuários do microcomputador. Dessa maneira, torna-se necessário um comando que, quando fornecido pelo usuário, preserve a configuração dos programas na memória ou no arquivo, em fita magnética ou de papel.

O comando de preservação de um programa deverá causar o armazenamento, em fita magnética ou de papel, das especificações e dos códigos formadores de todos os programas que constem do diretório da memória ou do arquivo, conforme especificado. A recuperação dos programas preservados deve ser feita mediante a utilização do comando de recuperação da memória ou do arquivo, conforme o caso.

O primeiro campo numérico, deste comando, indica o periférico no qual os programas a serem preservados deverão ser armazenados (fita magnética ou de papel) e, o segundo, indica se o comando solicita a preservação da memória ou do arquivo.

O arquivo em fita magnética ou de papel, gerado por esse comando, terá a seguinte estrutura:

(1º registro físico: 128 palavras de 16 bits)

1º pal. do reg.: contém o número de programas preservado, em formato binário absoluto.

2º pal. do reg.: contém o identificador do elemento preservado (memória ou arquivo).

As demais palavras deste registro físico não são utilizadas.

Os demais registros físicos são utilizados para o armazenamento das especificações e dos códigos de todos os programas preservados, sendo estes programas armazenados sequencialmente, um após o outro, cada um no formato descrito na observação b) DESTINO: FITA MAG. OU PAPEL, do comando de transferência de um programa.

4.7.6 - COMANDO DE RECUPERAÇÃO DA MEMÓRIA OU DO ARQUIVO

Sintaxe:

< recuperação da memória ou do arquivo > :: = RECUPERE < número > ,
< número >

Este comando é utilizado na recuperação dos programas da memória ou do arquivo de programas, a partir de fita magnética ou de papel, gerado pelo comando de preservação da memória ou do arquivo, descrito no item anterior.

O primeiro campo numérico indica se a recuperação a ser executada é da memória do microcomputador ou do arquivo, e o segundo indica a unidade periférica em que se encontra o arquivo, a partir do qual a recuperação deverá ser feita.

4.7.7 - COMANDO DE MAPEAMENTO DO DIRETÓRIO

Sintaxe:

< mapeamento do diretório > :: = < número > , < número > , < código >

Este comando, permite a obtenção de uma listagem no terminal teletipo (terminal de vídeo) ou na impressora, das especificações de todos os programas relacionados no diretório da memória ou do arquivo.

O primeiro valor numérico indica qual a unidade periférica em que a listagem será efetuada e, o segundo, indica qual o diretório a ser listado.

As listagens serão obtidas no código especificado e no seguinte formato:

NOME DO PROG	ENDER. INICIAL	ENDER. FINAL	ENDER. EXECUÇÃO
< identificador >	< número >	< número >	< número >

4.7.8 - FINALIZAÇÃO DA SECÇÃO DE EDIÇÃO, MODIFICAÇÃO E TRANSFERÊNCIA DE PROGRAMAS

Sintaxe:

< finalização SEC 2 > ::= FIM . | CNFG . < SEC 1 > | CTTS .
< SEC 3 >

Se FIM . , causará o encerramento desta secção e a parada do minicomputador, terminando, assim, o monitoramento do sistema em teste.

Caso seja fornecido CNFG . ou CTTS . , causará a inicialização das secções SEC 1 e SEC 3, respectivamente.

4.8 - COMANDOS DA SECÇÃO DE CONTROLE E TESTE DE PROGRAMAS

Sintaxe:

< SEC 3 > ::= < lista Comando SEC 3 > < finalização > |
< finalização SEC 3 >

```
< lista Comando SEC 3 > ::= < Comando SEC 3 > |  
                                < lista Comando SEC 3 > < Comando SEC 3 >
```

```
< Comando SEC 3 > ::= < Incondicional de Controle > . |  
                    < Incondicional de Teste > . |  
                    < Condicional > . |  
                    < Cancelamento > .
```

4.8.1 - COMANDO INCONDICIONAL DE CONTROLE

Sintaxe:

```
< Incondicional de Controle > ::= < Controle do modo de execução > |  
                                < Controle do sistema de inter-  
                                rupção > |  
                                < Controle do Canal de inter-  
                                rupção > |  
                                < Controle de inicialização de  
                                um programa > |  
                                < Controle da retomada de um  
                                programa >
```

A seguir, são descritas a sintaxe e a semântica de cada um dos comandos vistos acima. Antes, porém, deve-se notar que as operações de controle que causam a entrada imediata nos estados de fim de monitoramento e pausa, são comandadas por meio de dois botões existentes no painel do adaptador, não se fazendo portanto, referência a estas operações nesta secção.

4.8.1.1 - CONTROLE DO MODO DE EXECUÇÃO

Sintaxe:

```
< Controle do modo de execução > ::= ACTIVE PAP |  
                                       DESACTIVE PAP
```

Este comando permite ao usuário, selecionar o modo de execução do microcomputador. A forma DESATIVE PAP, indica que o modo selecionado é AUTOMÁTICO, enquanto que ATIVE PAP indica modo PASSO A PASSO.

Quando nenhuma instrução de controle do modo de execução for fornecida pelo usuário, o modo selecionado será o AUTOMÁTICO, que corresponde a desativação do controle.

4.8.1.2 - CONTROLE DO MODO DE INTERRUPÇÃO

Sintaxe:

```
< Controle do sistema de interrupção> ::= ATIVE SI |  
                                         DESATIVE SI
```

Este comando, é utilizado na manipulação do controle de interrupções do microcomputador.

A forma ATIVE SI ativarã o SI F/F no adaptador construído (controle de interrupções), de forma que o sistema monitor deixe que todas as interrupções que ocorram, passem livremente, permitindo, assim, o seu atendimento pelo microcomputador, desde que não estejam mascaradas internamente.

A forma DESATIVE SI, desativará o SI F/F (controle de interrupções), inibindo todas as interrupções que ocorram, não permitindo que o microcomputador as atenda.

Quando nenhuma instrução de controle do sistema de interrupções for fornecida, este controle permanecerã desativado.

4.8.1.3 - COMANDO DE CONTROLE DO CANAL DE INTERRUPÇÃO

Sintaxe:

```
< Controle do canal de interrupção > ::= = ATIVE < canal > |  
                                         DESATIVE < canal >
```

```
< canal > ::= = 1INT | 2INT | 3INT
```

Este comando, permite ao usuário causar interrupção no microcomputador, em qualquer dos três canais controlados pelo sistema monitor.

A forma ATIVE < canal >, levará ao nível lógico baixo o canal correspondente, indicando o término de um pedido de interrupção. Inicialmente, todos os canais de interrupção se encontram desativados.

4.8.1.4 - COMANDO DE CONTROLE DE INICIALIZAÇÃO DE UM PROGRAMA

Sintaxe:

```
< Controle de inicialização de um programa > ::= = EXECUTE <iden-  
                                         tificador >
```

Este comando, é responsável pelo controle de inicialização do programa, relacionado com o nome especificado por identificador no diretório da memória.

4.8.1.5 - COMANDO DE CONTROLE DA RETOMADA DE UM PROGRAMA

Sintaxe:

```
< Controle da retomada de um programa > ::= = CONTINUE
```

Este comando, executa a operação de retomada de um programa, a partir do ponto de parada.

É aceito nos estados de fim de monitoramento e pausa, com efeitos diferentes conforme o estado em que é fornecido.

ESTADO FIM DE MONITORAMENTO

Se o controle de interrupção foi previamente ativado, o microcomputador passará ao estado de espera de interrupções, considerando como sendo a última instrução local executada a instrução de parada que causou a entrada no estado de fim de monitoramento e encontrando-se na sua via de endereços o endereço dessa instrução.

Se contudo, o controle de interrupções se encontrar desativado, o microcomputador permanecerá no estado de fim de monitoramento, sendo este comando sem qualquer efeito.

ESTADO PAUSA

Se fornecido no estado de pausa, este comando causará a retomada do programa pelo microcomputador, a partir do ponto em que se deu a entrada no estado de pausa.

4.8.2 - COMANDO INCONDICIONAL DE TESTE DE PROGRAMAS

Sintaxe:

< incondicional de teste > ::= LISTE < periférico > , < código > ,
< elemento a ser listado >

< Código > ::= BINARIO | OCTAL | DECIMAL | HEXADECIMAL

< periférico > ::= < número >

< elemento a ser listado > ::= < bloco de memória > | < registros > |
TEMPO | INSTRUÇÃO | DADOS | ENDEREÇO

< bloco de memória > ::= = MEMORIA < número > ATE < número >

< registros > ::= = REGISTRO < lista de registros >

< lista de registros > ::= = < identificador > |
< lista de registros > , < identi-
ficador >

O comando de teste de programas, permite ao usuário a obtenção de listagens no teletipo (terminal de video) ou na impressora, conforme especificado em periférico, através do número associado ao periférico (teletipo = 1 e impressora = 6) e no código especificado, dos diversos elementos do microcomputador.

Em função do elemento a ser listado tem-se:

1. LISTAGEM DE POSIÇÕES DE MEMÓRIA

A listagem dos conteúdos das posições de memória com preendidas entre os endereços especificados, será fornecida no seguinte formato:

ENDEREÇO < valor numérico > CONTEUDO < valor numérico >

2. LISTAGEM DE REGISTROS

Os registros internos do microcomputador, serão reconhecidos pelo sistema monitor, através de identificadores, associados aos registros, mediante a carga desses identificadores no "array" registros do sistema monitor, durante o processo de preparação do dito sistema para um determinado microcomputador.

A listagem será fornecida no formato abaixo, no código especificado para a configuração dos registros.

REGISTRO < identificador > CONTEUDO < valor numérico >

3. LISTAGEM DO TEMPO DE EXECUÇÃO

O número de ciclos decorridos no processamento do programa, que está sendo executado desde sua inicialização, será fornecido no formato abaixo, e no código especificado.

TEMPO = < valor numérico >

4. LISTAGEM DO NÚMERO DE INSTRUÇÕES EXECUTADAS

Será fornecido o número de instruções do programa presente sendo processado, que foram executadas desde sua inicialização, no código especificado e no formato abaixo:

NUMERO DE INSTRUCOES EXECUTADAS = < valor numérico >

5. LISTAGEM DA VIA DE DADOS

Será fornecido ao usuário, o conteúdo da via de dados do microcomputador no momento do fornecimento deste comando, no formato abaixo e no código especificado para o valor numérico.

VIA DE DADOS : < valor numérico >

6. LISTAGEM DA VIA DE ENDEREÇOS

Será fornecido ao usuário, o conteúdo da via de endereços do microcomputador no momento do fornecimento deste comando, no formato abaixo e no código especificado para o valor numérico.

VIA DE ENDEREÇOS : < valor numérico >

4.8.3 - COMANDOS CONDICIONAIS

Sintaxe:

```
< comando condicional > ::= < cabeça > |  
                                < cabeça > ; < lista dos comandos  
                                associados >  
  
< cabeça > ::= < condição > |  
              < condição > , < número >  
  
< condição > ::= < condição de leitura na memória > |  
                < condição de escrita na memória > |  
                < condição de tempo > |  
                < condição do número de instruções > |  
                < condição de status >  
  
< lista dos comandos associados > ::= < comando associado > |  
                                       < lista dos comandos associados >;  
                                       < comando associado >  
  
< comando associado > ::= < controle do modo de execução > |  
                          < controle do sistema de interrupções > |  
                          < controle do canal de interrupções > |  
                          PAUSA |  
                          < comando incondicional de teste >
```

As instruções condicionais permitem que se associe uma condição a uma lista de comandos, de forma que os comandos serão executados, quando a condição associada a eles for satisfeita.

Na cabeça de um comando condicional pode haver um valor numérico, juntamente com a condição. Este valor numérico indicará o número de vezes que a condição deve ser satisfeita, antes de deixar de ser

4.8.3.4 - CONDIÇÃO DE NÚMERO DE INSTRUÇÕES

Sintaxe:

< condição de número de instruções > ::= INSTRUÇÃO = * +
< número >

Esta condição será satisfeita quando o número de instruções, especificado pelo comando, for executado. A contagem se inicia logo após o fornecimento do comando.

4.8.3.5 - CONDIÇÃO DE STATUS

Sintaxe:

< condição de status > ::= STATUS < identificador >

Analogamente aos registros internos do microcomputador, vide comando incondicional de teste - item 2 (4.7.2). Os estados (Status) possíveis do microcomputador, serão reconhecidos pelo sistema monitor, através de identificadores a eles associados, mediante a carga desses identificadores no "array" status do sistema monitor, durante o processo de preparação do sistema para o monitoramento de um microcomputador determinado.

Esta condição será satisfeita quando o microcomputador atingir o status caracterizado pelo identificador.

Quando a condição de um comando condicional é satisfeita, o sistema monitor imprime no terminal teletipo (ou terminal de vídeo), uma mensagem indicando o número da condição que acaba de ser satisfeita, no seguinte formato:

CONDICAO < número > SATISFEITA

Este número associado à condição é fornecido ao usuário logo após a aceitação do comando condicional. Isto permite, ao usuário, cancelar comandos condicionais, desde que ainda não estejam vencidos, através do Comando de Cancelamento.

4.8.3.6 - COMANDOS ASSOCIADOS

Com exceção do comando PAUSA, todos os comandos que podem ser associados a condições, podem também ser fornecidos incondicionalmente. Estes comandos encontram-se descritos nos itens 4.7.1 e 4.7.2.

Toda a vez que uma condição possuir associado a ela, entre outros, o comando PAUSA, quando tal condição for satisfeita, após a execução das operações solicitadas pelos demais comandos associados, o microcomputador entrará no estado de PAUSA, desde que a última instrução local executada pelo microcomputador não tenha sido de parada.

Em uma instrução condicional, os comandos associados devem ser fornecidos à razão de um comando por linha do terminal de vídeo, indicando com o caracter e ponto e vírgula, o fim de cada comando.

4.8.4 - COMANDO DE CANCELAMENTO

Sintaxe:

< comando de cancelamento > ::= CANCELE < número >

Este comando permite ao usuário cancelar um comando condicional, caracterizado pelo número fornecido ao sistema monitor.

O comando condicional, a ser cancelado, não deve estar vencido pois, caso contrário, uma mensagem de erro será publicada.

4.8.5 - COMANDO DE FINALIZAÇÃO DA SECÇÃO DE CONTROLE E TESTE DE PROGRAMAS

Sintaxe:

< finalização SEC 3 > ::= FIM . | CNFG . < SEC 1 > | PROG .
< SEC 2 >

Se FIM . , causará o encerramento desta secção e a para da do microcomputador, encerrando assim uma sessão de monitoramento.

Se CNFG . , causará o encerramento desta secção e a ini cialização da secção de configuração.

Se PROG . , causará o encerramento desta secção e a ini cialização da secção de edição, modificação e transferência de programas.

CAPÍTULO V

EXEMPLO DE APLICAÇÃO

É apresentada a seguir, um exemplo de aplicação, mostrando a interação do usuário e do sistema Monitor. O microcomputador em teste utiliza a CPU 8080 da INTEL, cujos identificadores de registros internos e status são fornecidos no apêndice B.

Ainda no exemplo abaixo, utilizou-se no início da linha, o caractere "*" para indicar mensagem do sistema Monitor.

: RUN, SEC 1

* ESCREVA OS LIMITES DA MEMORIA LOCAL

D0 ATE D5

* ESCREVA OS LIMITES DA MEMORIA REMOTA

D6 ATE D10

* ESCREVA O ENDERECO INFERIOR DOS PERIFERICOS REMOTOS

NAO

* ASSOCIAR UM PERIFERICO REMOTO OU TERMINAR ESTA SECCAO

PROG .

* ACEITA INSTRUCAO

ESPECIFIQUE TESTE , D0 , D5 , D0 .

* ACEITA INSTRUCAO

TRANSFIRA D1 , D9 , BINARIO , TESTE .

*	NOME	COMEÇO	FIM	END-COD
*	TE	B00 0	B00 0	B00 0
*	END HO 000	CONT H00 01
*	END HO 001	CONT H00 02
	:	:	:	:
	:	:	:	:
*	END HO 05	CONT H00 FF6

* ACEITA INSTRUCAO
ESPECIFIQUE MAR , D0 , D5 , D0 .

* ERRO **** 22 (Ender. já especificado!)

* ACEITA INSTRUCAO
CARREGUE ;
ENDereco DO CONT HO ;
CONT H1 ;
CONT H5 .

* ACEITA INSTRUCAO
MAPEIE D1 , D9 , HEXA .

* DIRETORIO DA MEMORIA

*	NOME	COMEÇO	FIM	END-CAD
*	TE	H00 0	H00 ... 05	H00 0

* ACEITA INSTRUCAO
PRESERVE D4 , D9 .
(LISTAGEM EM FITA DE PAPEL)

* ACEITA INSTRUCAO
APAGUE D9 , TESTE

- * ACEITA INSTRUCAO
ESPECIFIQUE MAR , DO , D5 , DO .
- * ACEITA INSTRUCAO
CTTS .
- * SEC 3
- * ACEITA INSTRUCAO
ATIVE PAP .
- * PAP ATIVADO
- * ACEITA INSTRUCAO
ATIVE SI .
- * SI ATIVADO
- * ACEITA INSTRUCAO
LI D1 , OCT , REGISTRO A, B .
- * REG A CONT 000 .. 12
- * REG B CONT 000 .. 0
- * ACEITA INSTRUCAO
STATUS M1 ;
ATIVE 2I .
- * COMANDO CONDICIONAL 1
- * ACEITA INSTRUCAO
LE MEMORIA DO ATE D5 ;
PAUSA .

* COMANDO CONDICIONAL 3

* ACEITA INSTRUCAO
CANCELE INSTRUCAO D3 .

* ACEITA INSTRUCAO
FIM .

@ → FIM DE MONITORAMENTO

BIBLIOGRAFIA

- AMERICAN MICROSYSTEMS INCORPORATION - *6800 Microprocessores*. Santa Clara, Ca., Oct. 1975.
- GRIES, D. *Compiler Construction for Digital Computers*. New York, Wiley 1971.
- FILHO, F.V. *Um sistema Monitor de Microcomputadores, baseado no mini computador HP-2116B*, 1978. (Preliminar de Dissertação de Mestrado).
- HEWLETT PACKARD. *HP Algol*. Cupertino, Ca., Apr. 1974.
- HEWLETT PACKARD. *DOS-III - Disc Operating System*. Cupertino, Ca., Jun. 1973.
- HEWLETT PACKARD. *A pocket Guide to Hewlett-Packard Computers*. Palo Alto, Ca.
- HEWLETT PACKARD. *2116B - Interface Manual*. Palo Alto, Ca., Dec. 1968.
- INTEL CORPORATION. *8080 Microcomputer System Manual*. Santa Clara, Ca., Jan. 1974.
- NATIONAL SEMICONDUCTORS. *IMP-16C Application Manual*. Santa Clara, Ca., Jan. 1974.

APÊNDICE A

MENSAGENS DE ERRO EMITIDAS PELO SISTEMA MONITOR

<u>Nº DO ERRO</u>	<u>SITUAÇÕES POSSÍVEIS</u>
0	- Rotina GETAVAIL: Memory Array lotado.
1	- Estado FM.
2	- Identificador não declarado ou programa inexistente.
3	- Endereço fora do espaço de memória especificado.
4	- Número fornecido é negativo ou maior que 32k.
5	- Identificador de Status inválido.
6	- Número associado a condição $\notin [1, @ 077777]$.
7	- ROTINA GETNUM: Número Inválido e/ou posição inválida (Ponteiro do Buffer de Entrada é maior que 71).
8	- Comando Condicional Inválido.
9	- Identificador de Registro não definido.
10	- Elemento Inválido para ser listado.
11	- Elemento a ser Ativado/Desativado não é válido.
12	- Código Inexistente, permitido apenas BIN, DCT, DEC e HEX.
13	- Endereço fora do espaço de Memória Local/Remota declarado.
14	- Esperado ponto, ponto e vírgula ou CONTEUDO.
15	- Comando Inválido
16	- FIM, PROG ou CNFG fornecido em estado diferente de FM.

- 17 - Comando de Transferência : Fonte Inválida.
- 18 - Comando de Transferência : Destino Inválido.
- 19 -
- 20 - Programa inexistente e/ou diretório incorreto.
- 21 - Identificador já declarado.
- 22 - Comando de especificação incorreto e/ou endereço já especificado.
- 23 - Programa muito longo (> 928 blocos de 128 bytes).
- 24 - Limites de Memória Local incoerentes.
- 25 - Limites de Memória Remota incoerentes.
- 26 - Endereço inferior dos periféricos remotos está fora do intervalo [0,255]
- 27 - Endereço do periférico associado está fora do intervalo [end. inferior , 255]
- 28 - É permitido somente a associação de 3 periféricos.
- 29 - Não há espaço suficiente no disco.
- 30 - Comando CONTINUE dado em estado FM com HALT = 1 ou CTIN = 0.

APÊNDICE B

IDENTIFICADORES ASSOCIADOS AO MICROCOMPUTADOR

INTEL 8080

O sistema monitor foi preparado para monitorar o micro computador COBACS, baseado no 8080, com as seguintes mnemônicos.

B.1) Identificadores de Registros

< ident. de registro > ::= A | B | C | D | E | H | L

Foi mantida a mesma nomenclatura utilizada pelo fabricante.

B.2) Identificadores de Status

< ident. de status > ::= MI | MR | MW | SR | SW | PR |
PW | IN | IH | HL |

onde

<u>Ciclos de Máquina</u>	<u>D₇</u>	<u>D₆</u>	<u>D₅</u>	<u>D₄</u>	<u>D₃</u>	<u>D₂</u>	<u>D₁</u>	<u>D₀</u>
MI: "Fetch"	1	0	1	0	0	0	1	0
MR: "Memory Read"	1	0	0	0	0	0	1	0
MW: "Memory Write"	0	0	0	0	0	0	0	0
SR: "Stack Read"	1	0	0	0	0	1	1	0
SW: "Stack Write"	0	0	0	0	0	1	0	0
PR: "Input Read"	0	1	0	0	0	0	1	0
PW: "Output Write"	0	0	0	1	0	0	0	0
IN: "Interrupt Acknowledge"	0	0	1	0	0	0	1	1
IH: "Interrupt Ack. While Halt"	1	0	0	0	1	0	1	0
HL: "Halt Acknowledge"	0	0	1	0	1	0	1	1

APENDICE C

PERIFÉRICOS DISPONÍVEIS E SEUS NÚMEROS ASSOCIADOS

Os diversos periféricos do minicomputador, bem como a memória do microcomputador serão referenciados através de valores numéricos convencionados abaixo:

TIPO DE PERIFÉRICO	NÚMERO ASSOCIADO
Terminal de vídeo	1
Arquivo de programas	2
Perfuradora de fita de papel	4
Leitora Ótica de fita de papel	5
Impressora	6
Unidade de fita Magnética (Canal 7)	7
Unidade de fita Magnética (Canal 8)	8
Memória do Microcomputador	9

APÊNDICE B

IDENTIFICADORES ASSOCIADOS AO MICROCOMPUTADOR

INTEL 8080

O sistema monitor foi preparado para monitorar o micro computador COBACS, baseado no 8080, com as seguintes mnemônicos.

B.1) Identificadores de Registros

< ident. de registro > ::= A | B | C | D | E | H | L

Foi mantida a mesma nomenclatura utilizada pelo fabricante.

B.2) Identificadores de Status

< ident. de status > ::= MI | MR | MW | SR | SW | PR |
PW | IN | IH | HL |

onde

<u>Ciclos de Máquina</u>	<u>D₇</u>	<u>D₆</u>	<u>D₅</u>	<u>D₄</u>	<u>D₃</u>	<u>D₂</u>	<u>D₁</u>	<u>D₀</u>
MI: "Fetch"	1	0	1	0	0	0	1	0
MR: "Memory Read"	1	0	0	0	0	0	1	0
MW: "Memory Write"	0	0	0	0	0	0	0	0
SR: "Stack Read"	1	0	0	0	0	1	1	0
SW: "Stack Write"	0	0	0	0	0	1	0	0
PR: "Input Read"	0	1	0	0	0	0	1	0
PW: "Output Write"	0	0	0	1	0	0	0	0
IN: "Interrupt Acknowledge"	0	0	1	0	0	0	1	1
IH: "Interrupt Ack. While Halt"	1	0	0	0	1	0	1	0
HL: "Halt Acknowledge"	0	0	1	0	1	0	1	1

- B.2 -

Para maiores detalhes quanto ao diagrama de estados e sinais de controle da CPU, vide INTEL CORPORATION - 8080 microcomputer systems manual.

APÊNDICE C

PERIFÉRICOS DISPONÍVEIS E SEUS NÚMEROS ASSOCIADOS

Os diversos periféricos do minicomputador, bem como a memória do microcomputador serão referenciados através de valores numéricos convencionados abaixo:

TIPO DE PERIFÉRICO	NÚMERO ASSOCIADO
Terminal de vídeo	1
Arquivo de programas	2
Perfuradora de fita de papel	4
Leitora Ótica de fita de papel	5
Impressora	6
Unidade de fita Magnética (Canal 7)	7
Unidade de fita Magnética (Canal 8)	8
Memória do Microcomputador	9