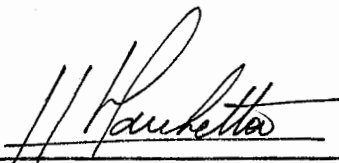


"SISTEMA DE BACK UP"

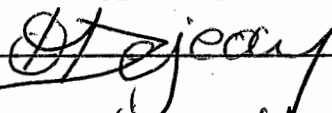
RONALDO PEIXOTO THOMPSON

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA (M.Sc.)

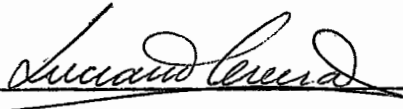
Aprovada por:



Presidente



Amor Vianna e Silva Filho



RIO DE JANEIRO
ESTADO DA GUANABARA - BRASIL
DEZEMBRO DE 1972

Agradeço,

ao programa de Engenharia de Sistemas e Computação e a COPPE pela o
portunidade;

à Cooperação técnica Francesa representada por: Henri Marchetta pelo
assunto;

Pierre Jean Lavelle pela introdução ao conhecimento do Mitra 15;

Daniel Dejean pela colaboração no estudo de interfaces;

à C.I.I. do Brasil que, por intermédio de Jacques Victoire, foi de
grande ajuda no conhecimento do Mitra 15;

ao RDC da PUC, e ao NCE da UFRJ pela utilização dos computadores
IBM /370, IBM /360, respectivamente;

à Eduardo Peixoto Thompson que soube converter meus hieroglifos num
texto datilografado;

ã.E. Sá Rego, M.L. Caldas e Moura, e M.E.B. de Almeida pela leitura
e correção do texto.

à Marta Maria

SINOPSE

Procura-se neste trabalho caracterizar um sistema de Back Up através de exemplos, definições e conceitos. Em seguida é proposto um sistema que será parcialmente implementado.

Foi usado um minicomputador Mitra 15 com 8k palavras de 16 bits.

Este sistema poderá ser usado em qualquer outro computador, desde que sejam feitas as adaptações necessárias.

ABSTRACTS

The aim of this work is to characterize a Back Up system through examples, definitions and concepts.

We propose a Back Up system that will be partially implemented.

We use a minicomputer Mitra 15 with 8k words of 16 bits.

This system may be used in any other computer with the necessary adaptations.

Capítulo I	Introdução	1
I-1	Generalidades	1
I-2	Motivos que levaram à pesquisa (Histórico)	1
I-3	Importância	2
I-4	Definição	2
I-5	Limitações	2
Capítulo II	Sistemaas de Back Up	3
II-1	Generalidades	3
II-2	Requisitos Básicos de um Sistema de Back Up	3
II-3	Dispositivos de Supervisão	4
II-3-1	Dispositivos de Supervisão Internos (DSI)	4
II-3-2	Dispositivos de Supervisão Externos (DSE)	4
II-3-2-1	Caixa Lógica de Decisão	5
II-3-2-2	Minuteria Cão de Guarda (Watch-Dog Timer)	5
II-4	Programas de Detecção de Erro	6
II-5	Funções Computador	6
II-6	Tipos de Sistemas de Back Up	7
II-6-1	Back Up Manual	7
II-6-2	Back Up Analógico	8
II-6-3	Back Up de Dois Computadores em Paralelo	8
II-7	Conclusão	12

Capítulo III Sistema Proposto	13
III-1 Generalidades	13
III-2 O Sistema	13
III-2-1 Função do dispositivo Supervisor 1	14
III-3 Conclusão	14
Capítulo IV Características do Mitra 15	15
IV-1 Generalidades	15
IV-2 Características Particulares	15
IV-2-1 Registros e Indicadores	18
IV-2-2 Instruções	20
IV-2-3 Interrupções	20
IV-3 Sistema Monitor "MOB"	23
IV-4 Micro-instruções e Executives	25
IV-4-1 Micro-Máquina	25
IV-4-1 Unidade Central	25
IV-4-1-1 Operador	25
IV-4-1-1-2 Registros	26
IV-4-1-2 Memória	26
IV-4-1-3 Memória de Comando e seus Circuitos	27
IV-4-1-3-1 Registro T	27
IV-4-1-3-2 Memória de Comando	27
IV-4-1-3-3 Memória de Executives	29
IV-5 Suspensão	29
IV-6 Conclusão	

Capítulo V Teste	31
V-1 Generalidades	31
V-2 Filosofia do Programa de Teste	31
V-3 Algoritmo de Escolha	32
V-4 Implementação do algoritmo	33
V-4-1 Leitura de Dados	33
V-5 Conclusão	35
Capítulo VI Programa de Teste	36
VI-1 Generalidades	36
VI-2-1 Carregamento e Lançamento de Programas	36
VI-2-2 Formação das Tabelas DVT e CTX (11), (10)	38
VI-3 Programa de Teste	38
VI-3-1 Procedimento de Teste	39
VI-4 Programa de Teste com Simulador de Erros	39
VI-5 Conclusão	46
Capítulo VII Minuteria Cão de Guarda	47
VII-1 Generalidades	47
VII-2 Função da Minuteria	47
VII-4 Estados da Minuteria	47
VII-4 Requisitos ao projeto da Minuteria Impostos pelo Computador	47

VII-4-1 Interface	48
VII-4-2 Sinais Recebidos e Transmítidos pela Interface	48
VII-5 Circuito Memorizador de nível de Interrupção	50
VII-6 Relógio da Minuteria	50
VII-6-1 Oscilador	50
VII-6-2 Contador	50
VII-6-3 Alarma	50
VII-7 Esquema Global	52
VII-9 Consumo	52
VII-10 Conclusão	54
Capítulo VIII Conclusão	55
Bibliografia	58
Apêndice A	59
Apêndice B	61

I - INTRODUÇÃO

I-1 Generalidades.

Este trabalho visa a implementação de uma técnica de segurança no funcionamento de computadores digitais em tarefas de controle, ou qualquer outra que necessite elevada confiabilidade do sistema.

A confiabilidade, a segurança e a disponibilidade são noções que de vem ser bem distintas, em particular nas operações de Tempo Real, onde as situações não são recuperáveis.

No projeto de um sistema de computação deste tipo, o funcionamento muitas vezes deverá ser mantido, mesmo sob a ocorrência de falhas no computador.

Para tanto o sistema poderá contar com dispositivos externos que o supervisionem, além daqueles internos à sua própria arquitetura.

I-2 Motivos que levaram a pesquisa (Histórico).

No final de 1971 foram propostos cinco projetos para serem desenvolvidos no sistema IBM /7, pelo programa de engenharia de sistemas e computação, sob a orientação do prof. Henry Marchetta.

Manifestado interesse pelo assunto, foi escolhido o item SISTEMA DE BACK UP, como tema para esta tese.

Posteriormente, tendo sido recebido o minicomputador Mitra 15, todo o trabalho foi nele desenvolvido.

I-3 Importância

Com a introdução das técnicas de controle (DDC e outras) em que o computador trabalha, em regime de Tempo Real, torna-se cada vez mais necessário o uso de técnicas do tipo Back Up. Como decorrência desta implementação, vê-se neste trabalho o desenvolvimento do diálogo "software-hardware", cujo profundo conhecimento se faz necessário nas aplicações Tempo Real do Mitra 15, aqui na COPPE.

I-4 Definição

Entende-se como Back Up a técnica de substituição de um equipamento por outro, em caso de falha no equipamento de trabalho.

I-5 Limitações

Será dado no decorrer da tese um breve relato bibliográfico. Nos subseqüentes capítulos tratar-se-á da discussão das pesquisas decorrentes da implementação do sistema proposto, dos seus resultados e suas conseqüências.

Como não dispunhamos meios para conectarmos o Mitra 15 a um outro sistema que fizesse seu Back Up, o trabalho de implementação se resume na detecção da falha e do alarma.

II - SISTEMAS DE BACK UP

II-1 Generalidades

Neste capítulo procura-se mostrar três tipos básicos de sistemas de Back Up:

- o Back Up manual
- o Analógico
- e o Back Up por outro computador.

No final deste capítulo propõe-se um sistema que será implementado, o que é assunto dos próximos capítulos (III-V-VI e VII).

II-2 Requisitos básicos de um sistema de Back Up.

A detecção da falha do computador de trabalho é o primeiro requisito de qualquer sistema de Back Up (4). Estas falhas são detetadas, com dispositivos externos à arquitetura do computador, e/ou dispositivos internos, ambos com ajuda ou programas especiais ou não. O segundo requisito é o isolamento do evento falha. Este isolamento pode ser, por uma simples substituição do computador, por outro sistema ou por técnicas de reconfiguração.

Na reconfiguração, todo o sistema é dividido em subsistemas. Em caso de falha num subsistema, os outros subsistemas isolam o subsistema falho e as funções de cada um são redistribuídas, não havendo prejuízo global no funcionamento do todo. O sistema continua funcionando mas em regime degradado (2). Num sistema de Back Up como o aqui proposto, detectada a falha, todo o computador será substituído.

II-3 Dispositivos de supervisão

Os dispositivos, cuja função seja supervisionar o funcionamento do computador, estão divididos em dois grupos:

Dispositivos de Supervisão Externos (DSE)

Dispositivos de Supervisão Internos (DSI).

II-3-1 Dispositivos de supervisão internos (DSI)

Os dispositivos de supervisão internos estão relacionados com a própria arquitetura do computador, sendo muitas vezes fornecidos como elemento opcional pelo fabricante.

Entre outras funções podemos citar: proteção de área de trabalho na memória a qual se faz com a verificação de determinado bit-ou um conjunto deles no endereço. Existem, frequentemente, dois tipos de instruções, quando se tem proteção de área - instruções privilegiadas ou não. Serão privilegiadas as instruções que tenham acesso, inclusive, as áreas protegidas. Se uma instrução não privilegiada tenta ter acesso a uma área protegida é emitido um sinal de erro.

- O teste de paridade é feito por hard-ware próprio que verifica a paridade de um determinado conjunto de bits, que formam a unidade de informação.

Na maioria dos computadores a unidade de informação é composta de oito bits ou de dois conjuntos desses, dezesseis bits, além do bit de paridade.

O bit de paridade é 1 ou 0 caso necessário para que se complete um número par de bits da unidade de informação.

II-3-2 Dispositivos de supervisão externos (DSE)

Qualquer sistema, que conectado ao computador funciona como supervisor do seu funcionamento, é um dispositivo externo de segurança (DES).

A única ligação a ser prevista entre o (DES) e o computador deve ser referente à troca de informações necessárias a sua supervisão. A caixa lógica de decisão e a minuteria cão de guarda são dois tipos de (DES).

II-3-2-1 Caixa lógica de decisão

É um sistema lógico proposto para receber informações em paralelo de dois computadores e tomar decisão quanto à concordância ou não entre estas informações (1).

No projeto de tais dispositivos deve-se prever que as informações não estarão em sincronismo. Então é previsto um intervalo de espera, Δt , para a chegada das informações nos registros da caixa lógica de decisão.

Esgotado o tempo Δt , ou em caso de não se encontrar concordância entre as informações recebidas, é emitida uma chamada a um programa de teste.

Este programa objetiva apontar ao sistema, em qual dos computadores foi encontrada a falha.

II-3-2-2 Minuteria cão de guarda (watch-dog timer)

A minuteria cão de guarda é empregada frequentemente na supervisão de um computador. Esta supervisão é feita com a inicialização periódica de um programa de teste. Findo o programa de teste, a minuteria recebe um sinal de bom ou mau funcionamento do computador supervisionado.

Ao receber sinal de bom funcionamento, é reinicializado e o ciclo recomeça. No caso da detecção de erro é enviado um sinal de alarma. Este sinal ativará o dispositivo de Back Up.

II-4 Programas de detecção de erro

É comum ser confundido programa de detecção de erro com programa de diagnose de erro (2).

Espera-se que um programa de diagnose de erro aponte o elemento do sistema que causou o erro, para posterior reparo. O termo elemento varia com a resolução do programa de teste.

Enquanto que um programa detector de erro verifica o bom ou mau funcionamento de um sistema:

Sistema OK, ou sistema não OK são suas possíveis saídas.

Com o advento da micro-programação, estes programas, em sistemas sofisticados são "escritos" ao nível de micro-programas.

Na série /370 os programas de diagnose estão numa pequena unidade de discos-discos Read Only-de acesso para a manutenção.

II-5 Funções computador

Duas funções caracterizam o emprego de um computador:

- a - Função passiva
- b - Função de Comando.

A função passiva é caracterizada por tarefas de manipulação de informação ou mesmo de computação de dados e posterior apresentação a um determinado operador, via qualquer sistema de comunicação homem-computador. Como exemplo de sistemas que trabalhem sob estas funções temos:

Centros de cálculo, banco de dados, e outros. Como se vê são tarefas importantes, mas não críticas!

Na função de comando o computador comanda algum sistema físico (ou o monitora).

Embora as duas funções necessitem muitas vezes do pleno funcionamento do computador, na última, a interrupção deste pode ser catastrófica.

II-6 Tipos de sistemas de Back Up

"O Back Up pode ser manual, analógico convencional ou por um outro sistema digital...dependendo das características do "plant" e suas justificativas econômicas..."

"... Quando um computador digital é usado no modo supervisor, ajustando o controlador de "Set Points", a falha no computador deixa o "Plant" sob o controle convencional. Mas no modo de controle digital direto (DDC), uma falha no computador poderá resultar numa completa perda de controle do sistema" (4).

Deste modo, a seguir descreveremos três tipos de sistemas de Back Up.

II-6-1 Back Up manual

Após emitido o sinal de alarma o operador assume o controle do sistema. Para tanto é necessário que as válvulas e demais elementos que atuam no sistema físico tenham possibilidade de serem posicionadas manualmente. A figura mostra um sistema que tem esta possibilidade. Pelas chaves o motor é controlado. A indicação da posição dos elementos é importante para informação do operador.

II-6-2 Back Up analógico

Num sistema que seja empregado este tipo de Back Up, o computador é substituído por elementos analógicos, nos "loops" críticos."... investigações mostram que talvez cinco a dez por cento de todos os "loops", num processo típico, tendo Back Up analógico, seria possível manter todo o processo em funcionamento com os "loops" não críticos sob Back Up manual...até se ter um reinício do controle automático D.D.C, ou até se atingir um ponto de parada..."(6).

Para que a comutação do computador digital com o sistema analógico seja feita com um mínimo de perturbação as seguintes condições devem ser satisfeitas (4).

1. Os "set points" dos controladores analógicos devem ser periodicamente atualizadas;
2. O termo integral deve ter sua condição inicial continuamente ajustada à posição do comando naquele instante.

Sob controle analógico a válvula da figura abaixo tem seu valor ajustado continuamente. Vê-se, na figura, a chave B que é comandada pelo D.E.S. para dar entrada ao sistema analógico de Backup.

II-6-3 Back Up de dois computadores trabalhando em paralelo (1).

Geralmente o Back Up de um computador por outro, compõe-se de sistema de dois computadores, trabalhando em paralelo e em pseudo-sincronismo, cujas saídas são submetidas a uma caixa lógica de decisão (vide II-3-2).

Caso esta unidade detecte uma não concordância entre as informações

BACK UP ANALÓGICO

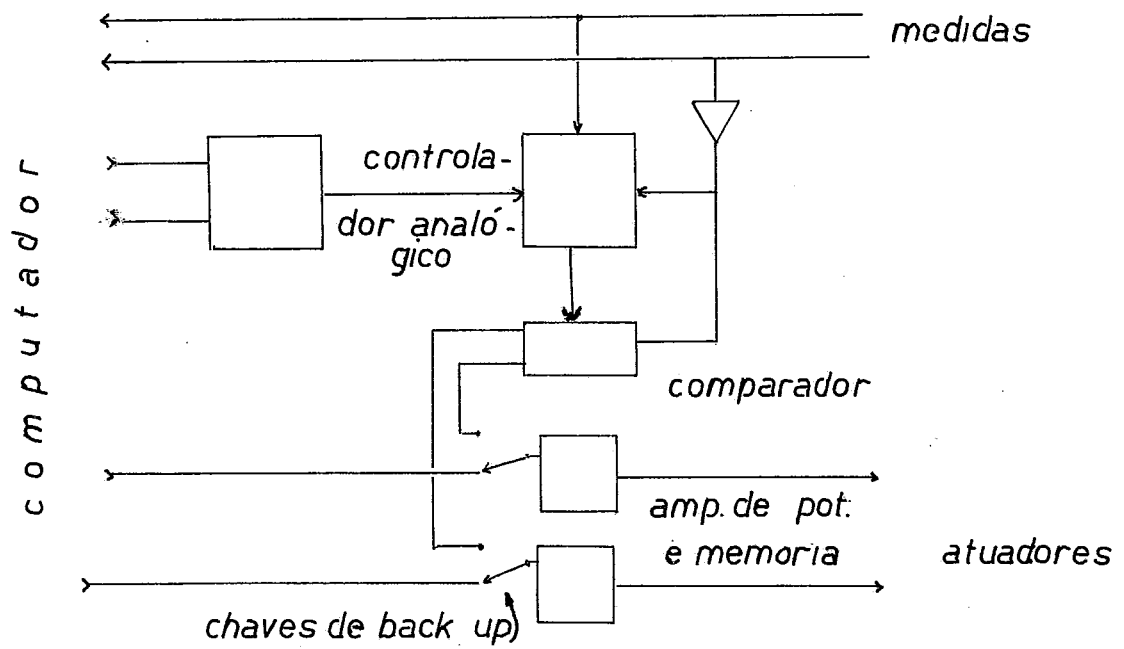


FIG. II-6-2

BACK UP MANUAL

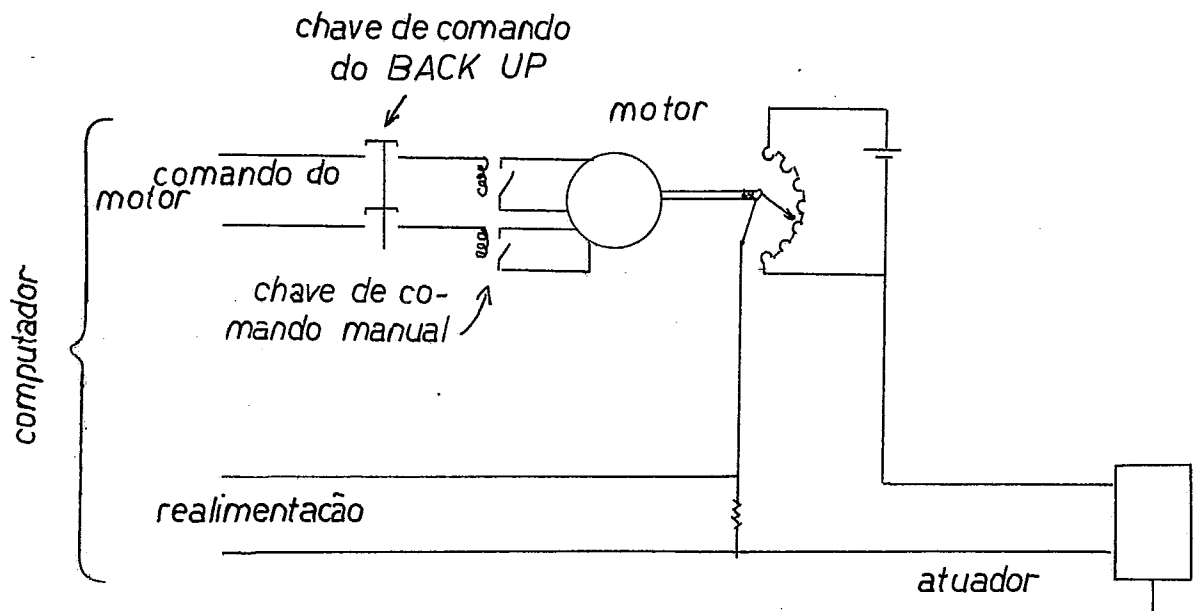


FIG. II-6-1

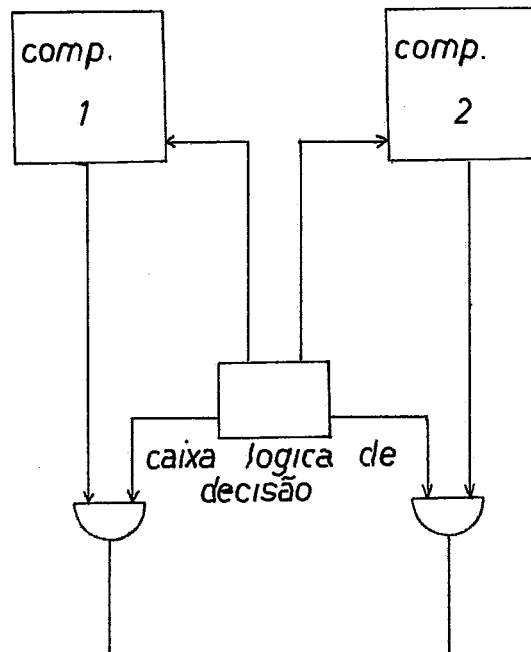
BACK UP COM DOIS COMPUTADORES EM PARALELO

FIG. II-6-3

emitidas pelos dois computadores, as saídas são bloqueadas por um sistema de portas lógicas "gates" e um sinal de prioridade mais elevada faz com que seja inicializado um programa de teste. Este programa informa ao sistema lógico, quais dos computadores está falho.

O computador falho é desconectado do sistema o qual passa a operar sob o controle do computador de Back Up.

O computador de Back Up, muitas vezes, assume a tarefa de aplicar "estimulos" ao computador falho e emitir um diagnóstico da falha. Deste modo a reparação da falha é mais rápida.

A reinicialização do sistema é feita pelo técnico de manutenção, seguindo uma rotina de desbloqueio das saídas e outras vias de comunicação. É necessário, entretanto, que o disco do computador que esteve parado seja regravado com o conteúdo do disco do computador de Back Up, esta tarefa é feita pelo computador de Back Up antes da inicialização.

II-7 Conclusão

No projeto de qualquer sistema de BackUp é preciso levar-se em conta custo e confiabilidade.

Sistemas com redundancia total, como aquele visto no parágrafo anterior, tem elevado custo e confiabilidade.

A diminuição do custo e conseqüentemente da confiabilidade poderá tomar um sistema de Back Up bastante acessível.

O sistema, que neste fim de capítulo propõe-se e descreve-se nos capítulos subsequentes, usará dois computadores mas não trabalhando em paralelo.

O computador de Back Up só entra no sistema quando deteta-se falha no computador de trabalho.

III - SISTEMA PROPOSTO

III-1 Generalidades

Depois de analisados vários sistemas, objetivando simplicidade e economia, foi elaborado aquele que passamos a descrever.

No apêndice C, será feita uma breve discussão das grandezas que medem confiabilidade, disponibilidade e segurança, tais com T.M.E.F. (Tempo Médio Entre falhas) e T.M.R. (Tempo Médio de Reparo).

Algum conhecimento matemático para os estudos de confiabilidade, como também dos parâmetros dos equipamentos que compõem um particular sistema, permitem o cálculo dessas grandezas.

III-2 O sistema

Não será necessário o trabalho contínuo de um segundo computador, computador de Back Up. Mas este computador deverá ter as características de Tempo Real, necessárias quando fôr chamado a fazer o Back Up.

Um sistema de memória auxiliar de rápido acesso - e.q.: disco - terá de ter seus dados continuamente renovados pelo computador que, no momento, está com o controle. Da mesma forma deverá ser disponível, e de localização definida, um conjunto de tabelas índices, onde conte os parâmetros necessários ao computador que assumir o controle, além dos dados ^{próprios} às equações do controle, parâmetros que definam o trabalho em curso-rotina, periféricos, dados lidos antes da transição etc.

III-2-1 Função do dispositivo supervisor

O dispositivo supervisor será uma minuteria Cão de Guarda. Periodicamente, esta minuteria ativa um determinado nível de interrupção de alta prioridade. Este nível dá entrada ao programa de teste, e a minuteria entra no estado de espera. Esgotado o Tempo de Espera, é emitido um sinal de alarma que, por um sistema de "gates", bloqueia todas as saídas do computador de trabalho, ativa um nível de interrupção que prepara este computador para as tarefas de Back Up e libera suas saídas.

III-3 Conclusão

Do sistema proposto, será implementado no MITRA 15, a minuteria, o acoplador de comunicação Minuteria-Mitra e o programa de teste.

Quando fôr o caso, a minuteria emitirá um alarma. Não existe diferença entre este sinal e o sinal de alarma do sistema proposto. Se fôr usado Back Up analógico ou manual este sinal comandará as chaves indicadas nos parágrafos (II-6-1 e II-6-2).

SISTEMA PROPOSTO

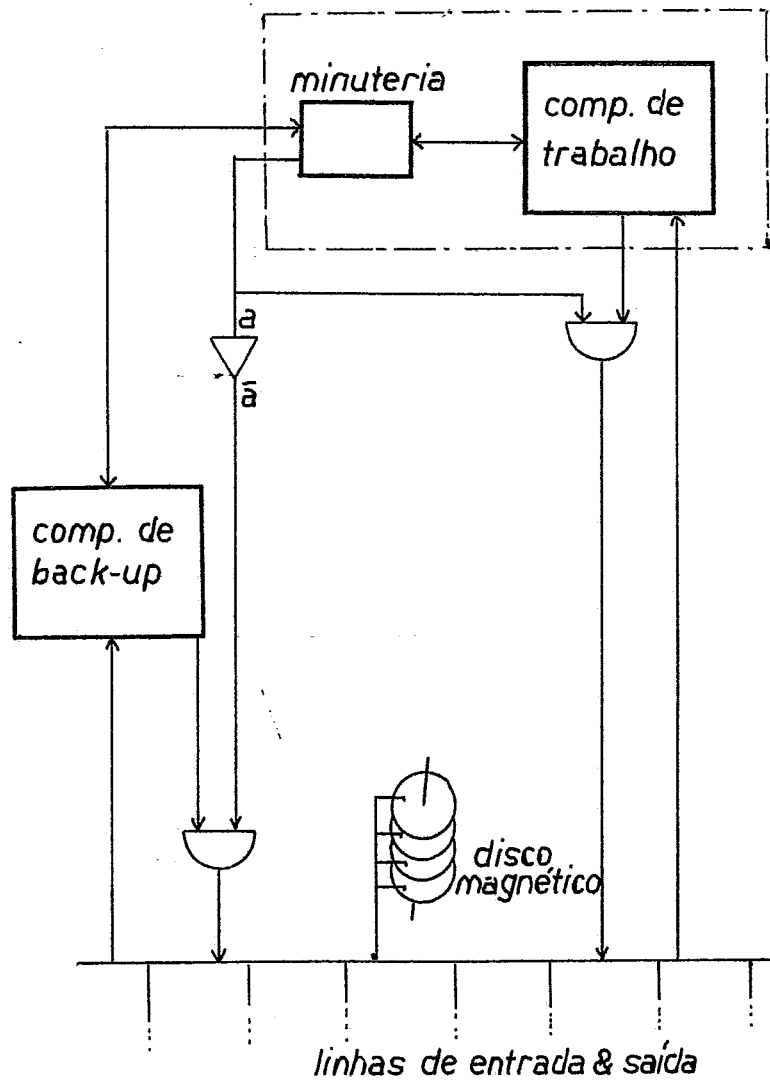


FIG. III-3

Este sistema compõe-se de:

Minuteria Cão de Guarda

Programa de Teste

Computador de trabalho

Computador de Back Up.

O computador de Back Up poderá ser um computador de função passiva do centro de processamento de empresa adaptado a Tempo Real.

IV - CARACTERISTICAS DO MITRA 15*

IV-1 Generalidades

O Mitra 15 é um computador para trabalhos em Tempo Real, de arquitetura simples e moderna baseada nas técnicas de micro-programação.

Sua unidade de tratamento é composta de memórias mortas (Read Only Memory) onde estão implantados os micro-programas. Estas ROMs distinguem o Processador Central (CPU) da Unidade de Intercâmbio, pelos micro-programas a elas associados.

O Mitra 15 é apresentado em dois modelos, cuja diferença está na potência do processador, e, conseqüentemente, na micro-máquina.

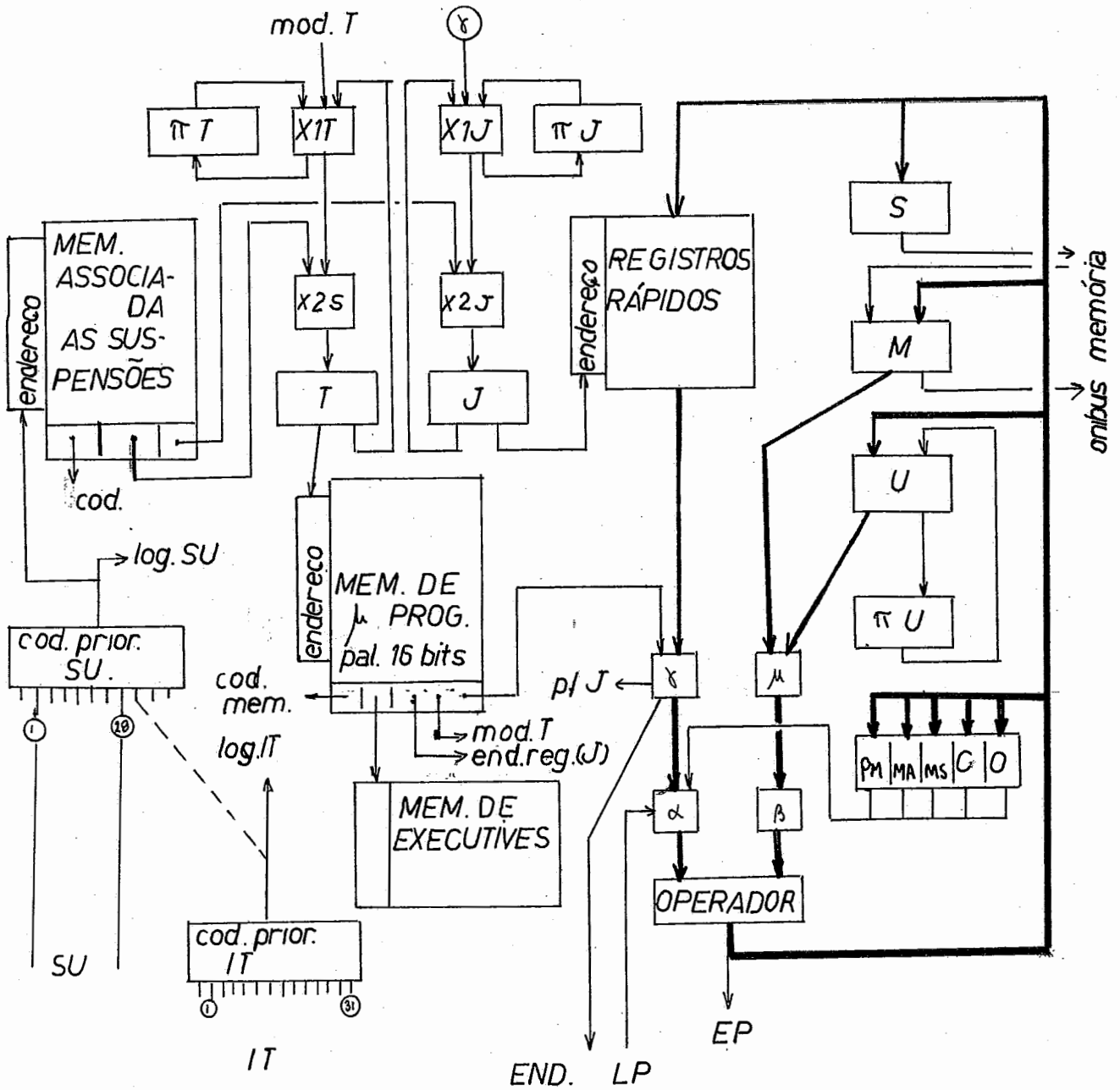
IV-2 Características particulares

Salvo restrições em contrário, toda descrição desse computador, apresentada neste capítulo, será referente ao modelo Mitra 15 onde todo o trabalho foi executado. A observância da figura IV-2 é necessária para o entendimento deste capítulo.

O modelo Mitra 16, do Laboratório de Simulação de Sistemas de COPPE, tem sua micro-máquina com 512 palavras de 16 bits, e tempo de acesso de 60 ns. A memória principal tem 16k bytes ou 8k palavra, com um ciclo de leitura de 800ns.

Um teletipo, com uma leitora e perfuradora de fita de papel, e a unidade de tratamento Mitra 15 compõem o sistema Mitra do Laboratório de Simulação e Sistemas.

* (10), (11), (12), (13).



RPTompson

FIG. IV-2

IV-2-1 Registros e indicadores

.Registros rápidos

A unidade de tratamento tem 8 blocos de 8 registros de 16 bits, totalizando 64 registros.

O bloco 0 afeta a execução dos programas. É composto dos 6 registros que indicam, ao sistema, a localização dos diversos segmentos, índice, acumulador e sua extensão e mais dois registros utilizados pelos micro-programas:

- P - Apontador da próxima instrução a ser executada;
- L - Apontador da base local;
- G - Apontador da base Geral;
- A - Acumulador;
- E - Extensão de A;
- X - Índice;
- V - Utilizado pela micro-máquina
- W - Utilizado pela micro-máquina.

O registro rápido 8 contém o dobro do valor do nível de interrupção do programa em curso.

Os outros são usados, normalmente, por interrupções rápida ou suspensões, com periféricos.

. Registros de Ligação com a Memória.

Fazem a ligação com a memória dois registros, S e M.

O registro S tem 16 bits (15 + 1 de proteção). É usado para endereça

mento na memória.

O registro M tem 11 bits (10 + 1 de proteção). É usado para acesso a memória.

. Outros Registros

O registro U tem 16 bits, sendo usado como acumulador da micro-máquina.

O registro J tem 8 bits sendo usado como endereçamento dos registros rápidos R.

. Indicadores

A Unidade Central possui 9 indicadores. Desses indicadores, quatro são acessíveis a programa:

C - Carry

O - Overflow

MS - Modo de Execução de Programa.

MS - 0 Modo Normal

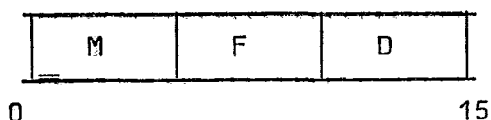
MS - 1 Modo Escravo

MA - Mascaramento de Interrupções

PR - Indica proteção de Memória é usada.

IV-2-2 Instruções

Este modelo tem um jôgo de 62 instruções. Estas instruções tem 16 bits.e o seguinte formato:



onde:

M - Modo de Endereçamento

F - Função

D - Deslocamento.

Cada instrução dá origem a um micro-programa que a executa. A primeira fase da execução deste micro-programa faz a decodificação do tipo de endereçamento usado e a família da instrução.

Existem três famílias e um total de 10 modos de endereçamento.

IV-2-3 Interrupções

O sistema possui 28 níveis externos de interrupções, hierarquizadas. A cada nível está associado o endereço do contexto do programa daquele nível.

Um nível de interrupção é ativado por um sinal externo diretamente sobre o minibus, com nível lógico TTL, 0.

O minibus é uma linha de comunicação do computador com o exterior, onde estão os periféricos. Os sinais são acessíveis aos acopladores por dois

conectores de 2 x 43 pontos de acesso. Cada um desses pontos tem função própria;

IT n : Ativamento de Interrupção Nível n.

DIT : Desativamento de Interrupção

PIT : Posicionamento de Interrupção

LEP : Leitura de Dados do Exterior

NEP : Envio de Dados ao Exterior

e sinais de sincronismo, alimentação e etc. (vide apêndice F)

O sinal de ativamento de interrupção deve ser mantido durante todo o trabalho por circuitos memorizadores tipo Flip-Flop. Se o nível de interrupção ativado for de prioridade superior ao nível de interrupção em curso, não é desativado, ficando na fila de espera, com seu nível lógico ativo.

Todo ativamento de interrupção dá entrada a um micro-programa com a seguinte função:

1. Copiar o contexto do programa em curso na memória, no endereço definido pela tabela de contexto e pelo conteúdo do registro 8.
2. Copiar no registro 8 o dobro do valor do nível de interrupção ativado.
3. Chamar o novo contexto dado pelo conteúdo do registro 8 e pela tabela de contexto.
4. Lançar o programa do novo contexto.

Todo o programa de interrupção deve terminar por uma macro-instrução do monitor M:EXIT. Uma das instruções desta macro é a instrução de desativamento de interrupção DIT. A instrução DIT emite, pelo ponto de acesso correspondente, um sinal que "resseta" o Flip-Flop de Interrupção. No caso de estarem ativados vários níveis de interrupção, o sinal DIT emitido no final do primeiro programa-o mais prioritário-desativaria todos os níveis ativados e na fila de espera. Para que este inconveniente seja evitado o sinal de desativamento DIT é usado com o sinal NEP (emitido pelo conteúdo da tabela DVT correspondente ao nível desativado).

Funções que sucedem o desativamento.

1. Atendimento do nível mais prioritário na fila das interrupções.
2. Modificação do conteúdo do registro 8.
3. Chamada do novo contexto.
4. Lançamento do programa apontado pelo contexto.

Quando um programa de interrupção se comunica com outro programa, em nível inferior, o nível de interrupção em curso, deve ser desativado.

O reativamento da interrupção, ou seu reposicionamento, é feito pelo sinal PIT e o conteúdo da tabela DVT: É o que ocorre quando um programa se comunica com o MOB. Se o programa estiver no nível 4, por exemplo, e chamar o MOB tem nível de interrupção 5 e executa suas tarefas no nível 0. O MOB desativará o nível 4, executará a tarefa e voltará a posicionar o nível 4 ao final do trabalho.

IV-3 Sistema monitor "MOB"

O sistema monitor do Mitra 15, MOB, é responsável por todo o controle "software" do computador. Entre outras funções cita-se: carregamento de programas para serem executados, tratamento de erro ao nível de execução, controle de interrupções, controle de E/S com o Teletipo (o controle do teletipo é feito por um micro-programa).

Os programas, para serem executados, são divididos em segmentos.

1. Segmento de dados comuns, CDS, apontado por "G";
2. Segmento de dados locais, LDS, apontado por "L";
3. Segmento de programa, LPS, apontado por "P";

Um segmento de programa pode constar de vários pequenos programas (sub-programas). Cada sub-programa pode ter sua, LDS, ou compartilhar da LDS de outros sub-programas. E ter acesso a zona comum de memória, CDS.

Quando um sub-programa faz a chamada de outro sub-programa, o retorno, ao sub-programa que fez a chamada, é assegurado pelo conteúdo das duas primeiras palavras de CDS associada ao sub-programa chamado. Nestas palavras estarão o endereço da LDS e da LPS do sub-programa que fez a chamada.

Em determinada zona do monitor localizam-se duas tabelas:

- DVT - Tabela de desativamento de interrupção;
- CTX - Tabela de contexto.

Cada uma dessas tabelas tem 32 palavras, o que corresponde a uma palavra para cada nível de interrupção (4 níveis de interrupção internos e 28 ex-

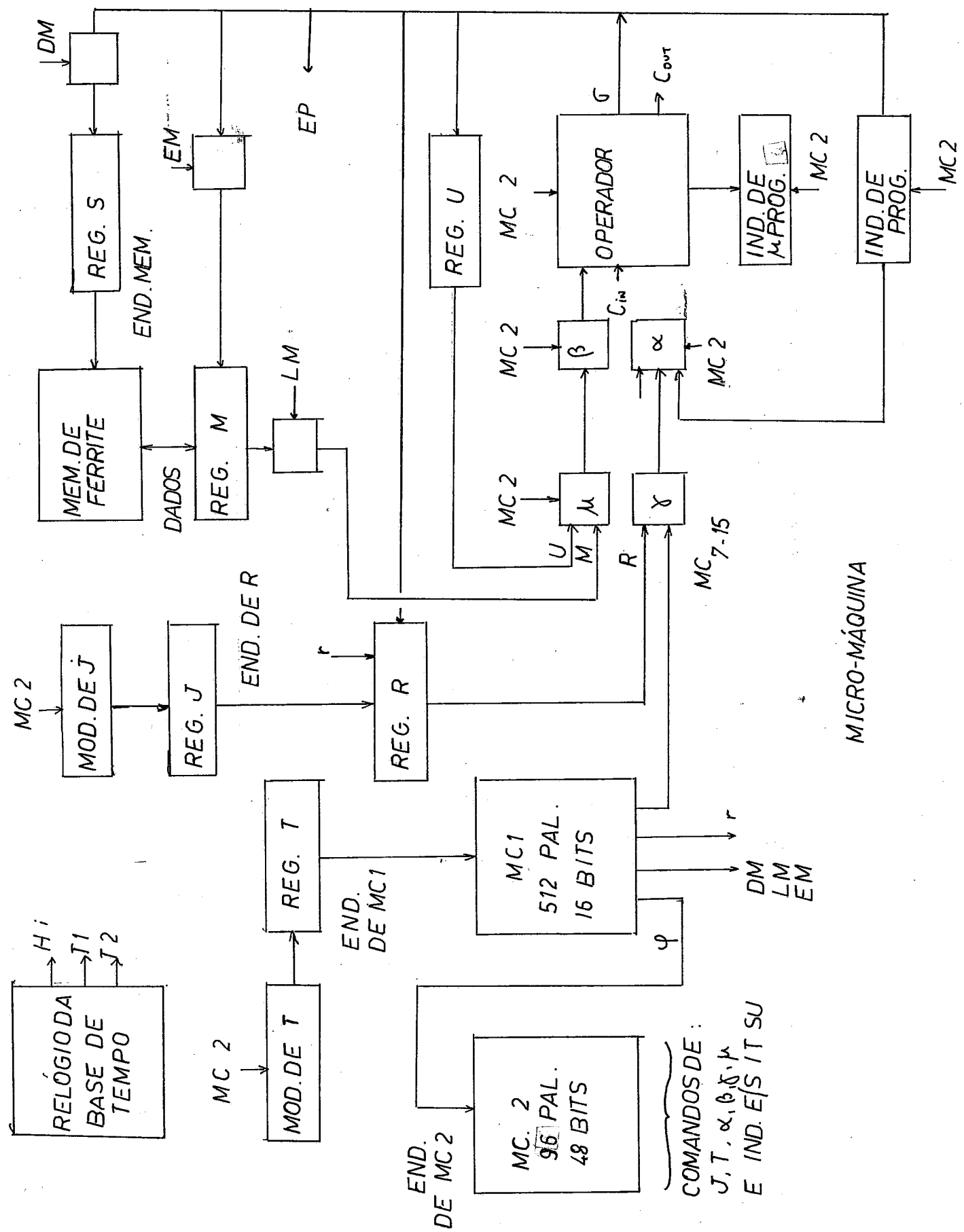


FIG. IV - 4

ternos).

Quando é ativado ou desativado determinado nível de interrupção, as tabelas CTX e DVT são consultadas (vide IV-2).

IV-4 Micro-instruções e executives

Todo o trabalho das micro-instruções dá-se na micro-máquina. A micro-máquina, a memória de executives e a memória de comando compõe a macro-máquina.

IV-4-1 Micro-máquina

Compõe a micro-máquina:

Unidade Central;
Memória de Ferrite;
Memória de Comando;
Interrupção e Suspensão.

IV-4-1-1 Unidade Central

Compõe-se de:

Operador;
Registros.

IV-4-1-1-1 Operador

Fornece uma saída, σ de 16 bits que comandam diversas partes da micro-máquina; memória, indicadores, linhas de saída etc.

As operações são sobre as entradas X e B cada uma com 16 bits. E mais uma entrada C_{in} e uma saída C_{out} que são usadas nas operações aritméticas.

IV-4-1-1-2 Registros

Vide IV-2-1

IV-4-1-2 Memória

A capacidade da memória é de 8k palavra (sua expansão é modular até 32 palavra em módulos de 4k palavras).

O acesso à memória é comandado por 3 sinais da memória de comando:

DM: Demanda de Memória

Carrega o registro S com o conteúdo de σ , zera M, e envia um pedido de acesso a memória.

LM: Se o registro M é diferente de zero faz o ciclo de reescritura e transfere o conteúdo do registro M para o σ .

EM: Transfere o conteúdo de σ para M e inicia o ciclo de escrever na memória.

IV-4-1-3 Memória de comando e seus circuitos

É formado por:

- Registro T (10 bits);
- Memória de Comando;
- Memória de Executives.

IV-4-1-3-1 Registro T

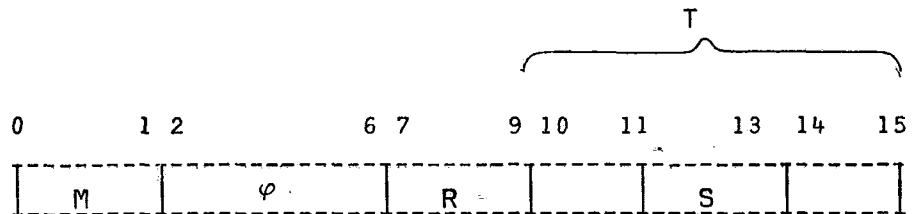
Tem 10 bits que fornecem o endereço da micro-instrução em curso.

IV-4-1-3-2 Memória de comando

Tem 512 palavras de 16 bits. Cada palavra corresponde a uma micro-instrução e fornece as seguintes informações:

- Código da Micro-instrução;
- Comando DM, LM e EM da Memória;
- Modificação do Conteúdo do Registro T.
- Parâmetros de MC_{7-15} para α .

Uma micro-instrução tem o seguinte formato:



Zona M: Comando da Memória:

- M= 11 → (nenhuma ação sobre a memória)
- M= 01 → DM
- M= 10 → LM
- M= 00 → EM

Zona φ : Endereço da Memória de Executiva.

Zona R : Endereço de um dos Registros Rápidos.

Zona T : Modificação do Registro T.

Zona S : Extensão da Zona φ .

IV-4-1-3-3 Memória de Executives: MC 2

É uma memória morta de 96 palavras de 48 bits. Existem 88 executivos possíveis de comando e 8 reservados à manutenção do sistema cujo acesso é pela chave MAINTENANCE do painel.

Esses executivos são responsáveis pelo comando de toda a micro-máquina, seus 48 bits estão assim distribuídos:

Um no Comando de μ ;

Um no Comando de σ ;

Dois no Comando de α ;

Nove no Comando de β ;

Onze no Comando do operador e entrada de sinais exteriores;

Três no Comando do relógio de transferência U e R;

Seis no Comando de T;

Dois no Comando de J;

Cinco no Comando dos indicadores de programa;

Quatro no Comando dos indicadores de micro-programa;

Dois no Comando dos periféricos;

Dois no Comando do sistema de interrupção e suspensão.

IV-5 Suspensão*

Dois tipos de sinais são traduzidos por uma parada no trabalho em curso: Interrupções e Suspensões: O primeiro já foi discutido.

*Será mencionado, embora não tenha sido usado diretamente.

Existem 27 níveis de suspensão externos e 5 internos.

Enquanto que uma interrupção esta associada a uma parada no programa e a uma mudança de seu contexto, a suspensão esta associada a uma parada no trabalho da micro-máquina diretamente.

Uma memória Associada às suspensões, guarda os valores dos registros U,J,T e os indicadores de micro-programas (não descritos aqui). Após a memorização desses parâmetros o micro-programa associado a esta suspensão começa a ser "processado".

IV-6 Conclusão

Uma instrução de acesso ao programador gera micro-códigos (executives e micro-instruções). Por meio dessas instruções, o programador terá acesso a toda unidade central.

V - TESTE

V-1 Generalidades

Neste capítulo será descrita a filosofia do programa de teste a ser elaborado, o algoritmo de EXCOLHA das instruções a serem testadas e a implementação deste algoritmo.

V-2 Filosofia do programa de teste

Na elaboração de um programa detector de erros, deseja-se que este, teste o maior número de partes do computador possível. Um meio de atingir este objetivo poderia ser abrir sobre uma mesa o esquema do Mitra 15 e, pacientemente, seguir todo o fluxo de informações das instruções. Entretanto, o capítulo anterior mostra (IV-6) que as micro-instruções e seus executives, são responsáveis por todo o funcionamento do sistema Mitra.

Como não se tem acesso as micro-instruções, a não ser indiretamente via as instruções*, essas serão usadas. O uso de todas as macro-códigos certamente a tivaría todas as micro-instruções e executives. Contudo, tendo por base que nos diversos micro-programas, associados aos macro-códigos, existem micro-instruções comuns a utilização de todos os macro-códigos, poderia conter redundâncias dispensáveis. O procedimento que será seguido é de escolher o menor número de macro-códigos que contenham, em seus micro-programas, todas as micro-instruções. O algoritmo que faz esta escolha será ALGORÍTMO DE ESCOLHA.

* O termo macro-código, será usado no lugar de instrução.

V-3 Algoritmo de escolha (Z)

Tendo-se o conjunto de macro-códigos e micro-instruções associadas nu ma forma matricial, onde cada coluna representa uma micro-instrução e cada linha um macro-código, e a intersecção, linha x coluna, é 1 ou 0 conforme a utilização da micro-instrução pelo macro-código ou não, inicia-se o algoritmo:

. Pesquisa dos macro-códigos essenciais

1. Percorre-se as colunas da matriz verificando-se o núme ro de macro-códigos que representam a micro-instrução correspondente a coluna per corrida. Um contador faz a contagem do número de 1s que aparecem em cada coluna. Se só um macro-código representa determinada micro-instrução, este é um macro-códi gido essencial.

2. Elimina-se a influência dos macro-códigos escolhidos, zerando-se todas colunas em que estes se fazem presentes.

. Pesquisa da existência de micro-instruções ainda não representadas.

3. Percorre-se as colunas verificando-se em alguma delas se esta presente 1.

. Escolha dos macro-códigos não essenciais que devem ser representados.

4. Percorre-se as linhas, contando-se o número de micro-instruções ^{co} presentadas por cada macro-código. O macro-código mais representati-vo é escolhido.

5. Se não existe nenhuma micro-instrução a representar vá para 8.

6. Se houve empate na escolha do macro-código mais representativo siga, caso contrário volte para 2.

7. Verifique o macro-código de menor tempo de execução.

(volte para 2)

8. Os macro-códigos escolhidos em um, quatro e sete são aqueles que devem fazer parte do programa de teste."

V-4 Implementação do algoritmo (7)

Foi feito um programa em FORTRAN que implementou o algoritmo anterior. Como o número de dados manipulados foi grande, despendeu-se 300k de memória, com um tempo de CPU de 9.62 segundos.

O Programa de Escolha esta dividido em programa principal e três subrotinas. São as seguintes funções atribuídas a estas partes:

- . Programa Principal: Leitura dos dados e formação da matriz binária.
- . Subrotina Escolha: Faz a escolha do conjunto de macro-códigos.
- . Subrotina Ordem e Rendu: Operam juntas na compactação de vetores.

V-4-1 Leitura de dados

A decodificação de um macro-código num micro-programa é feita em duas partes:

1. Decodificação do modo de endereçamento.

2. Decodificação do comando do macro-código.

Deste modo não é necessário fazer parte do processo de escolha os macro-códigos com todas suas opções de endereçamento, pois qualquer macro-código gera o mesmo micro-programa de endereçamento. As micro-instruções que fizeram parte do processo foram:

Tipo
Código
e Família.

Exemplo 7: Macro-código XEX

Micro-instruções:

Tipo: 3
F11D
EDA1
F158
ED16

Código:1

Família: PH1

B3AA

DB67 5A67

D19E

C2A7 D18F

Além da leitura dos macro-códigos e suas micro-instruções são lidos os tempos de execução de cada macro-código.

V#5 Conclusão

Fizeram parte do processo de escolha 66 macro-códigos dos quais foram escolhidos 65! Isto se deve principalmente ao número de macro-códigos essenciais. Foram encontrados 64 macro-códigos essenciais e um não essencial.

O resultado do programa escolha é apresentado de acordo com a codificação do anexo B.

A soma do tempo de execução** foi de: 179.40

* Programa Escolha será o nome do programa que implementa o algoritmo ESCOLHA:

** Tempo calculado pela soma dos tempos individuais. Este tempo é aproximado pois não leva em conta a variação com o tipo de endereçamento.

VI - PROGRAMA DE TESTE

VI-1 Generalidades

Neste capítulo será discutido o programa de teste proposto e o programa de teste com o simulador de falhas, como também as facilidades de lançamento de um programa num determinado nível de interrupção (operação).

VI-2 Operação *

Como foi visto no capítulo quatro é necessária a formação das tabelas DVT e CTX para o lançamento de um programa em determinado nível de interrupção.

VI-2-1 Carregamento e lançamento de programas (11)

A instrução de monitor:

%L

Carrega um programa Link-Editado a partir do primeiro endereço livre da memória. Terminado o carregamento na memória de um programa o MOB envia pelo teletipo as seguintes informações:

A → XXXX → Endereço do início de implantação do programa;

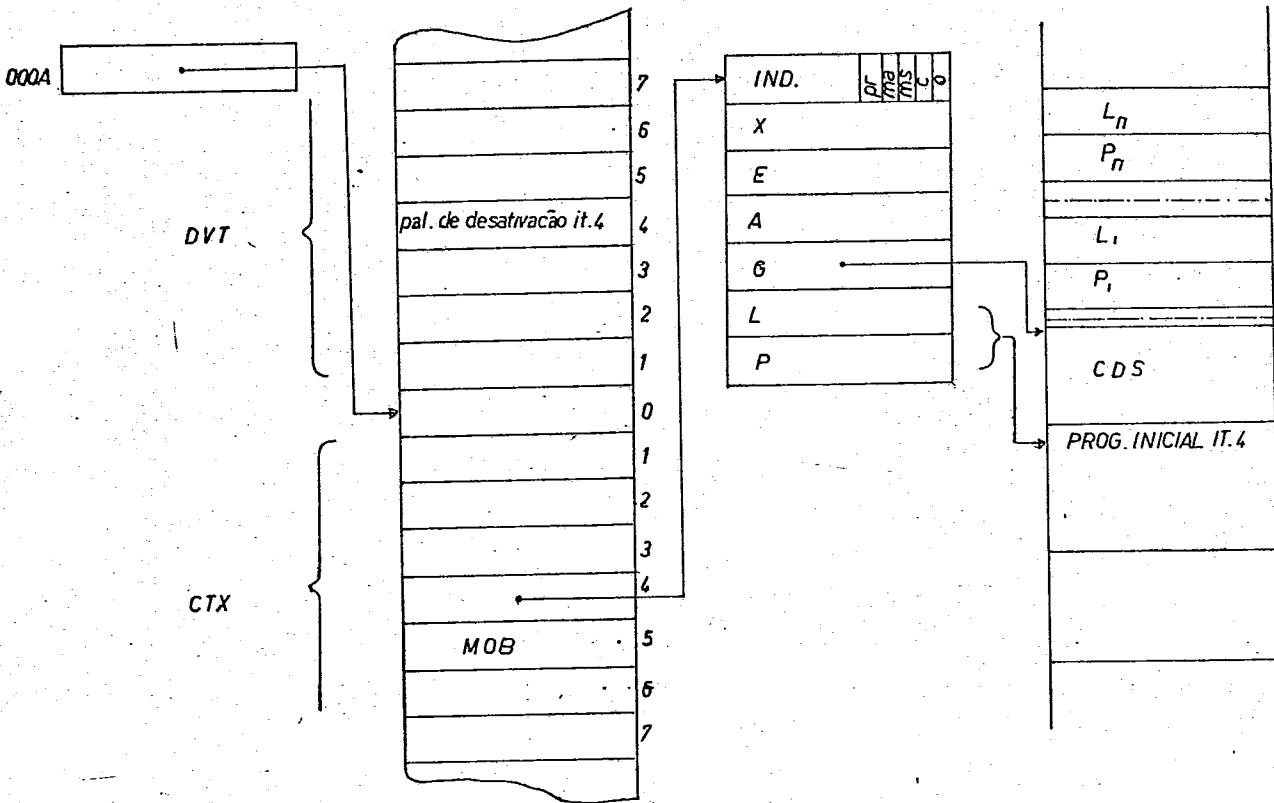
B → YYYY → Valor da base G;

C → WWW → Primeiro endereço livre após o programa.

Estas informações devem ser observadas no carregamento de outros programas, para que não sejam sobrepostos programas na memória. O carregamento de outros programas deverá ser após o endereço dado por C.

* Os comandos aqui descritos dão entrada pelo teletipo. Após cada comando deve-se bater retorno de carro, uma das teclas.

** Nível 0 não é necessário especificar, basta: %R.



LIGAÇÃO ENTRE UM PROGRAMA DE NIVEL DE IT.4 E A MICROMÁQUINA

FIG. VI-2

%L/A,@& (endereço \geq C)

Após carregado o programa na memória a instrução:

%R/I&(nível de IT em hexadecimal) **

faz seu lançamento no nível especificado. Lançado um programa, este só é ativado quando presente o sinal de ativamento de interrupção.

VI-2-2 Formação das tabelas DVT e CTX (11),(10)

A tabela CTX tem localização dada pelo conteúdo do endereço de memória 000A. Sua localização dentro do MOB é prevista pelo sistema numazo na reservada. O endereço de contexto de determinado nível de IT é modificado pelo sistema após recebido o comando %R/I& (nível de IT em hexadecimal).

A tabela DVT se localiza nas trinta e uma posições de memória anteriores a tabela CTX. Como seu conteúdo é codificado de acordo com o código de desativamento para cada acoplador, esta deve ser formada pelo operador. Usa-se a instrução:

%MO/A,@&(endereço a modificar): &(valor em hexadecimal) *

O endereço da palavra da DVT correspondente a determinado nível de interrupção é dado por:

ENDEREÇO DA DVT correspondente ao nível $n=2x \ n+0116$ **

VI-3 Programa de teste

No programa de teste, os macro-códigos escolhidos serão testados, comparando os resultados de suas operações, com resultados pré-estabelecidos.

As comparações serão feitas, normalmente, após as manipulações dos dados por vários macro-códigos. Assim diminui-se o tempo de execução dos testes sem sacrifício de sua performance.

Nos testes das instruções de "SHIFT" será testado o resultado de ca-

DA "SHIFT", sendo feitos o "SHIFT" do número de bits do operando, 16 ou 32.

Em caso de se encontrar erro o programa entra num "loop" infinito.

Devido ao tamanho do programa este foi dividido em quatro partes:

Um programa principal e três sub-programas.

A instrução do monitor M:KEY foi usado para teste da macro-código

RD e WD.

VI-3-1 Procedimento de teste

Serão dados os macro-códigos e a descrição de seus testes. O diagrama de bloco do programa de teste e a sua listagem estão nos anexos.

Caso os valores esperados nas comparações forem diferentes, o programa entra numa instrução de "loop" infinito. ***

As instruções CMP (compare), BCT e BCF (desvia se a comparação for verdadeira e desvia se a comparação for falsa), são usadas em quase todos os testes.

Teste 1 - LDA, STA e EOR.

A é carregado (LDA) com &AAAA. O conteúdo de A é armazenado na memória (STA).

Faz-se um ou exclusivo entre o conteúdo de A e o conteúdo do endereço de memória onde este conteúdo foi armazenado (EOR). O resultado esperado em A é:

&0000

* Esta instrução permite a modificação do conteúdo de qualquer endereço de memória.

** Tanto a adição quanto, n em hexadecimal.

*** BRU\$0

Teste 2 - IOR e AND.

Faz-se um ou lógico (IOR) entre A (que vem do teste anterior com conteúdo = &0000) e &AAAA. Em seguida é feito um E lógico entre A e &5555.

O resultado esperado é &0000.

Teste 3 - LBR e LBL.

Os dois bytes da esquerda e da direita de A são carregados com &55.

O resultado esperado é &5555.

Teste 4 - DST e DLD.

E é carregado com &AAAA e A com &5555 (DLD). Estes conteúdos são armazenados na memória (DST). O conteúdo desses endereços de memória são compara dos com &AAAA e &5555.

Teste 5 - SBR, SBL e ADM.

Os bytes da direita e da esquerda de determinado endereço da memória, são carregados com &AA, (SBR e SBL). O conteúdo desse endereço é somado a &AAAA (ADM). O resultado esperado, como conteúdo desse endereço, é: &5554.

Teste 6 - LDX e STX.

X é carregado com &0002 (LDX). O conteúdo de X é armazenado (STX). O valor armazenado é comparado com &0002.

Teste 7 - STE.

Note que desde o teste 4 o conteúdo de E é &AAAA. O conteúdo de E é armazenado na memória (STE). Este conteúdo da memória é comparado com &AAAA.

Teste 8 XAE, MUL, SUB e ADD.

A e E são carregados, respectivamente, com &0002 e &0000. O conteúdo de A é trocado com o conteúdo de E, e vice-versa (XAE).

O conteúdo de A, (&0002), é multiplicado por &5555 (MUL), e armazenado na memória. O conteúdo de A é subtraído de &5555, e somado com &5555 (SUB,ADD). O resultado dessa operações é comparado com o resultado da multiplicação.

Nota:

- a. Os testes de 9 a 22 estão em subrotinas.
- b. O registro X, nos testes 9 a 16, é usado para controlar o número de deslocamentos*.(DCX)
- c. Como não existe no repertório de instrução do Mitra 15, comparação entre X e memória; essas comparações serão feitas no registro A (XAX,BAZ).

Teste 9 - SLLS.

A é carregado com &5555, e em binário é representado por:

0101010101010101

X é carregado com 16

Com o conteúdo de A são feitos 16 deslocamentos de 1 bit. Após cada deslocamento X é decrementado (DCX).

Entre cada deslocamento o bit mais a esquerda será 0 ou 1 se a ordem do deslocamento for, respectivamente, $2k + 1$ ou $2k$.

Após completados 16 deslocamentos, o conteúdo de A, deverá ser zero. O macro-código SLLS, é supervisionado ao nível do bit deslocado e ao nível do conteúdo final esperado. A necessidade desta redundância no teste é discutível.

Teste 10 - SRLS.

A é carregado com &AAAA, que em binário é:

[1010][1010][1010][1010]

O procedimento restante é semelhante ao teste 9.

Teste 11 - SRCS.

A é carregado com &AAAA. O procedimento restante é semelhante ao teste 9, só que após completados 16 deslocamentos o conteúdo de A é comparado com &AAAA.

Teste 12-SLCS.

A é carregado com &AAAA.

O procedimento é semelhante ao teste 9, só que os deslocamentos de ordem $2k + 1$ e $2k$, devem ter o bit mais a esquerda 1 ou 0 respectivamente.

Teste 13 - SRLD e CCA.

E é carregado com &AAAA.

A é carregado com &5555.

O conteúdo de A é complementado.

X é carregado com 32.

O restante do teste é semelhante ao teste 10, com exceção que a comparação esta implícita no próximo teste.

Teste 14 - SLCD.

Após o teste 13, espera-se que o conteúdo de A e E seja &AAAA.

X é carregado com 32 o restante do teste é semelhante ao teste 12, exceto a comparação que não é feita.

Teste 15 - SAD.

Após o teste 14 o conteúdo de A e E deve ser &AAAA.

X é carregado com 31. O procedimento restante é semelhante ao teste 10, exceto o conteúdo final de A, deve ser &FFFF.

Teste 16 - SAS

A é carregado com &AAAA e X com 16. O procedimento restante é semelhante ao teste 11.

Teste 17 - XEX, CNA e CNX.

A e E carregados com &AAAA. Os conteúdos de A e E são trocados entre si (XEX). O conteúdo de A é complementado algebricamente (CNA). Os conteúdos de E e X são trocados entre si.

O conteúdo de X é complementado algebricamente (CNX). Espera-se que os conteúdos de X e A sejam iguais.

Teste 18 - XAA.

A é carregado com &0000. O byte da esquerda de A é carregado com &55. Os conteúdos dos bytes da esquerda e da direita são trocados entre si.

O conteúdo de A é comparado com &0055.

Teste 19 - CCE,AEE e KDE,"

O conteúdo de E que desde o teste 17 deve ser &0000 é complementado.
Os conteúdos de A e E são trocados entre si.

O registro E é carregado com &FFFF,
Faz-se um ou exclusivo entre A e E (AEE).
Espera-se que o conteúdo de A seja &0000.

Teste 10 - AIE.

Com o conteúdo de A(&0000) e de E (&FFFF) é feito um ou lógico(AIE).
O resultado esperado é &FFFF.

Teste 21

Nota:

Se o resultado do teste anterior fôr OK o indicador C é igual a 1
(como característica da operação de comparação CMP).

E é carregado com -1 (LNE).
O conteúdo de E é adicionado a C, (ACE).
O resultado esperado é: E=&0000.

Teste 22 - CHX.

X é carregado com &AAAA.
X é dividido por dois (CHX).
O resultado esperado é &D555.

Teste 23 - CLS e RTS.

Chamada de subrotina e retorno ao programa principal. Estas duas ins
truções foram usadas 3 vêzes desde o teste 8.

Teste 24 - WD,RD,RSV,CSV.

A instrução de monitor M:KEY contém estas quatro instruções.

Portanto o teste é feito como uso de M:KEY.

Teste 25 -ICL, desvios (branches) e DCL.

A base L é incrementada (ICL) de dois. Com o modo de endereçamento in direto geral é feito um desvio para uma posição de memória cujo conteúdo é outra instrução de desvio, com endereçamento indireto local. Esse desvio é para endereço na memória, cujo conteúdo é a instrução de decremento da base L à posição orig original(DCL).

Ente cada instrução de desvio tem duas instruções de loops infinitos (BRU\$00).

Se estas três instruções não forem executadas corretamente, não se atinge a instrução de monitor MEXIT.

Teste 26 - DIT.

O teste desta instrução é feito pela instrução de monitor M:EXIT. Se não for executada, não há desativamento do nível de interrupção.

Após a instrução M:EXIT (que é uma volta ao programa principal) é necessário um desvio para o começo do programa de teste.

Isto porque após o desativamento de interrupção, a tabela de contexto do programa de teste deverá ter o endereço do início deste programa.

VI-4 Programa de teste com simulador de erros

O programa de teste é ativado por uma minuteria cão de guarda. Após ativado o lançamento do programa, a minuteria entra no estado de espera, durante um período pré-determinado. Esgotado este período, caso não seja enviado um sinal de desativamento de interrupção, é emitido o alarma. A simulação do erro será feita, fazendo-se o programa ser executado várias vezes antes de emitir o sinal de desativamento de IT.

Foi anexado um M:KEY que lê o conteúdo das chaves do painel do MITRA, e executa o programa o número de vezes dado por este conteúdo. A listagem do programa com este módulo simulador mostra que não existem diferenças entre os dois programas.

VI-5 Conclusão

No programa de teste foram gastas 244 palavras de memória e um tempo de execução de 2,5 milissegundos. (Com o simulador estes resultados foram 258 palavras de memória e aproximadamente 2,5 milissegundos de tempo de execução).

VII - MINUTERIA CÃO DE GUARDA

VII-1 Generalidades

Neste capítulo será descrita a função da minuteria no sistema de BACK UP, seu projeto, custo e desempenho.

VII-2 Função da Minuteria

A minuteria enviará periodicamente um sinal de ativamento de um nível de interrupção de alta prioridade. Neste nível de interrupção está um programa de teste que é inicializado.

O programa de testes tem duas opções de saída:

1. Emitir um sinal de desativamento de interrupção (sistema OK).
2. Entrar em "loop" ou nem ser inicializado.

A minuteria cão de guarda, que após emitir o sinal de desativamento entra num estado de espera, caso receba o sinal 1, volta ao estado inicial e reinicia o ciclo, caso receba o sinal 2, (i.e. não receba sinal), esgota-se o tempo de espera e emite um sinal de alarma.

O sinal de alarma na minuteria aqui projetada e construída, é dado por um Diodo Emissor de luz (Light Emitter Diode), que é comandado por um Flip-Flop memorizador do alarma.

O reinício do ciclo, caso receba o sinal 1, é devido ao "reset" do contador de tempo pelo sinal de desativamento.

VII-3 Estados da minuteria

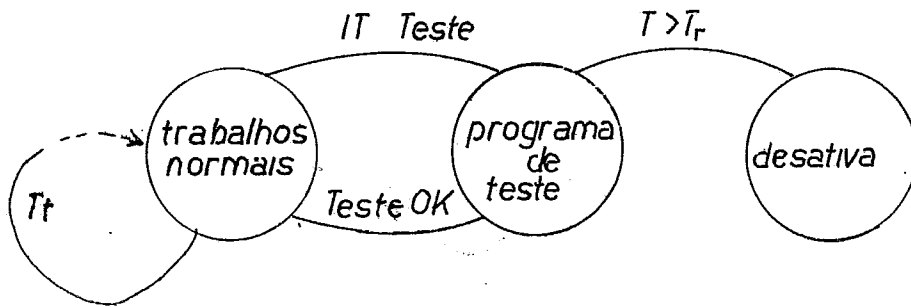
Os estados da minuteria e do sistema podem ser resumido nos diagramas da Fig. VII-3.

VII-4 Requisitos ao projeto da minuteria impostos pelo computador

A minuteria cão de guarda é encarada pelo computador como um periférico, pois está conectada diretamente no Minibus do Mitra 15. Para tanto é neces

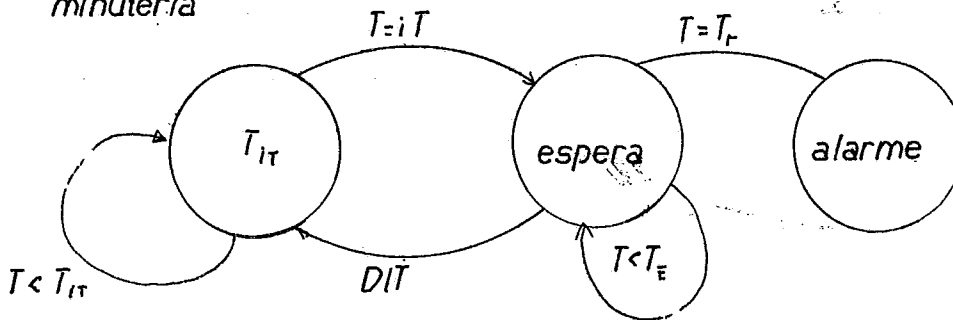
ESTADOS

sistema



T_r tempo de espera
 T_t Tempo de background

minuteria



T_{IT} ativamente

FIG. VII - 3

sário um circuito que faça a interface minibus-minuteria cão de guarda.

VII-4-1 Interface

Os circuitos de interface agem como isoladores, entre os circuitos externos e circuitos do minibus, e como fornecedores de nível lógico compatível com o minibus. O nível lógico do minibus Mitra 15 é TTL. Então, todos os que de rem entrada ou saída no minibus, serão deste nível.

Como circuitos de interface foram usados SN7404, SN7440, vide figura VII-4-1. Montados na extremidade de uma placa extensora de 2 x 43 pontos.

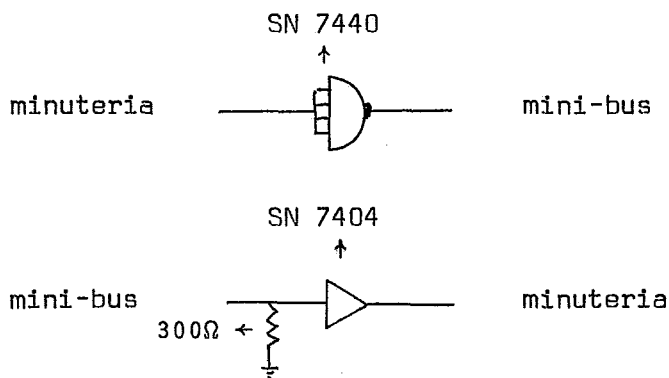


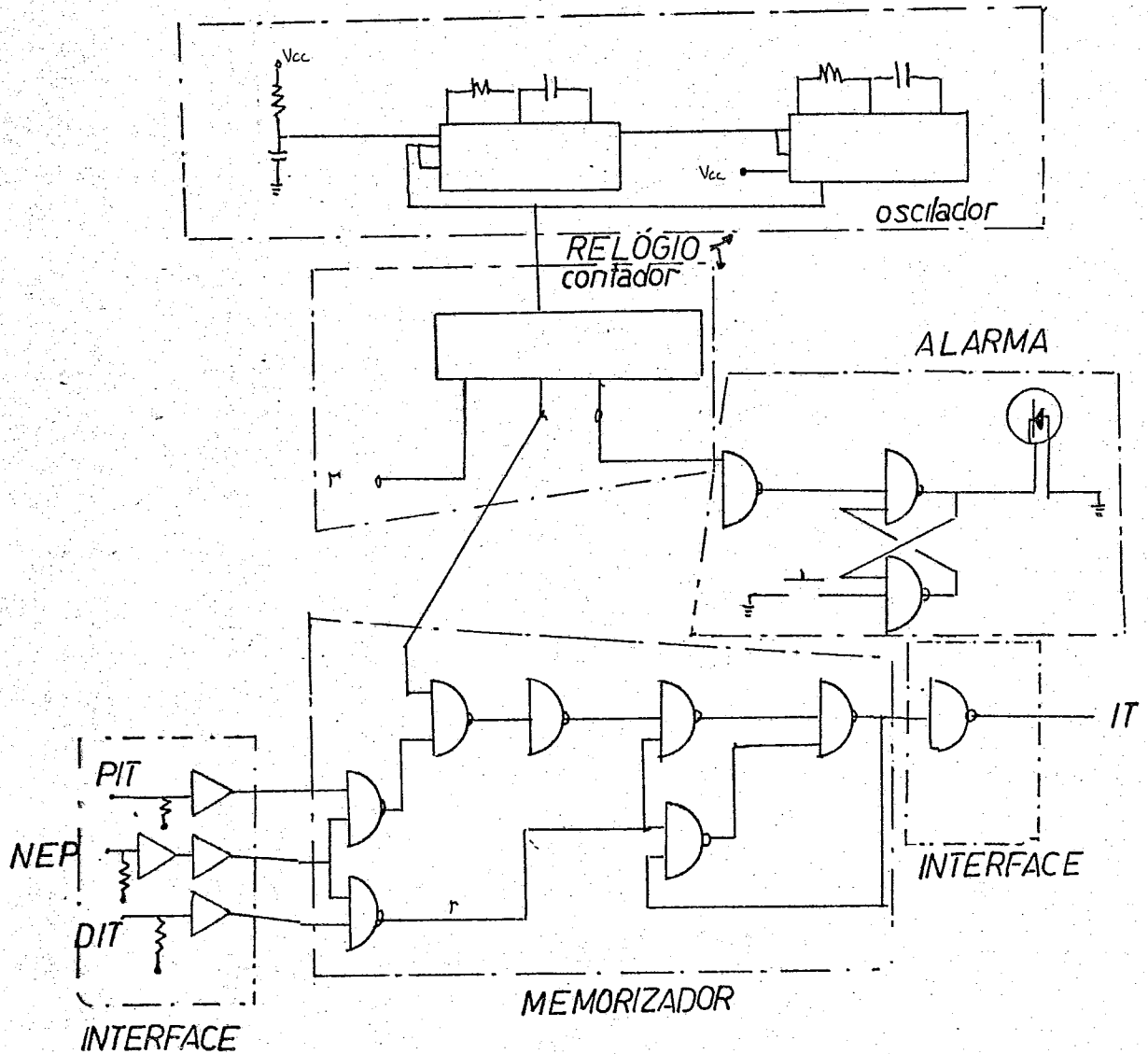
FIG. VII - 4-1

VII-4-2 Sinais recebidos e trasmitidos pela interface

Foram usados os seguintes sinais.

- DIT - Desativamento de Interrupção ponto de acesso número 34 A.
- NEP07 - Posicionamento de Dados (DVT) ponto de acesso número 39 A.
- IT04 - Ativamento de Interrupção - Ponto de acesso número 21 B.
- PIT - Posicionamento de Interrupção - Ponto de acesso número 7 B.

Memorizador, Relógio e Alarma.



MINUTERIA CÃO DE GUARDA E INTERFACE

FIG. VII-6

VII-5 Circuito memorizador de nível de interrupção

Tem como função memorizar o nível de interrupção enquanto este não fôr desativado. O desativamento e o posicionamento da interrupção é feito pelos sinais:

DIT "E" NEP07 desativamento;
 PIT "E" NEP07 posicionamento.

E o ativamento feito pelo sinal C, do relógio da minuteria, quando este atingir o tempo de ativamento, (vide o circuito na figura VII-5).

VII-6 Relógio da minuteria (8)

O relógio da minuteria fornece o sinal de ativamento de interrupções e o sinal de alarma. O período das interrupções é variado com a decodificação dos números binários do contador.

VII-6-1 Oscilador

O oscilador fornece uma sequência de 1s ao contador.

No projeto do oscilador foram usados dois monoestáveis SN74121 que sob o esquema da figura VII-6 funciona como um estável.

VII-6-2 Contador

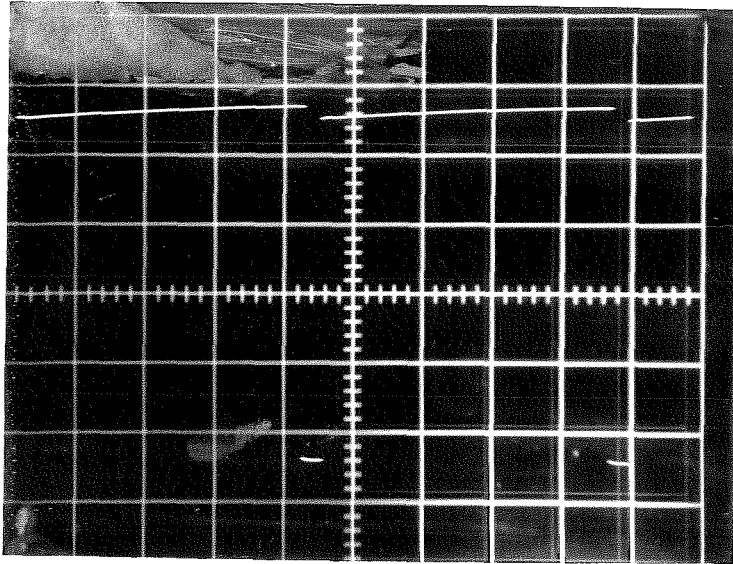
O contador fornece o sinal de ativamento de interrupções ao sistema memorizador, após a contagem de certo número de pulsos do oscilador. E o sinal de alarma.

O contador é formado pelo circuito integrado SN7493 (contador binário).

O sinal de desativamento é recebido pelo sistema de reset do contador.

VII-6-3 Alarma

O alarma é dado pelo excitamento de um LED por um FLIP_FLOP que me-



A fotografia mostra a forma de onda gerada pelo os
cilador (escala: Vert. 0,5V/div, Hor. 1 ms/div.).

moriza esta condição. O "reset" do alarma, é feito por uma chave (vide figura VII-6).

VII-7 Esquema global

Vide figura VII-6.

VII-8 Custo da minuteria

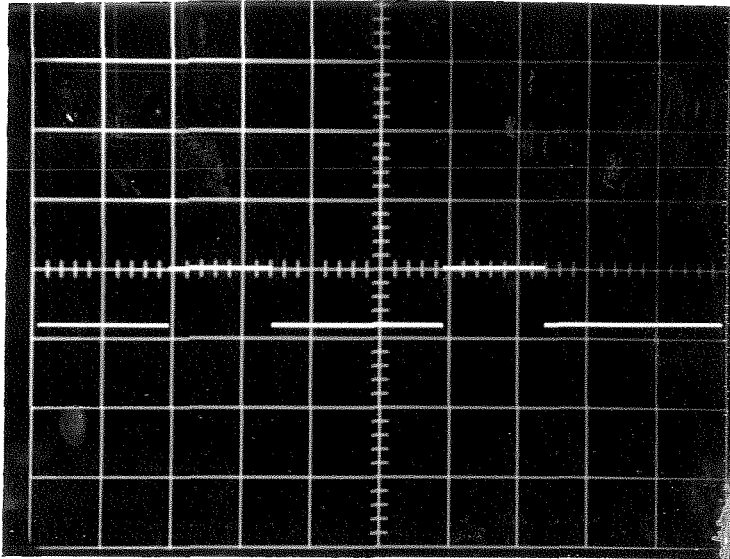
Foram gastos na minuteria:

3 - C.I. SN 7400 (Quad. 2 entradas, Nand)	Cr\$ 45.
1 - C.I. SN 7440 (Dual 4 entradas Buffer Nand)	Cr\$ 15.
1 - C.I. SN 7404 (Hex Inversor)	Cr\$ 18..
1 - C.I. SN 7493 (Contador Binário 4 bits)	Cr\$ 35.
2 - C.I. SN 74121 (Multivibrador Monoestável)	Cr\$ 80.
6 - (Resistores)	Cr\$ 2.
3 - (Capacitores)	Cr\$ 2.
+ - TOTAL	Cr\$ 197.

VII-9 Consumo

O acoplador tem alimentação fornecida pelo próprio computador. A minuteria teve sua alimentação independente, como recomendação do parágrafo II-3-2.

O consumo de corrente foi de 30mA, sob tenção de 5V, fornecidos por uma fonte Hewlett Pachard.



A fotografia mostra o lançamento periódico do programa de teste no nível de interrupção número 4 feito pela minuteria Cão de Guarda. (escala: Vert. 5V/div e Hor. 2 ms/div.)

VII-10 Conclusão.

Não foi necessário a utilização de linhas de endereçamento, devido aos sinais escolhidos.

O circuito memorizador deve ser montado junto com o acoplador. Nesta implementação, o acoplador ficou junto ao computador que fornecia sua alimentação, e o circuito memorizador de interrupção, junto a minuteria.

O custo da minuteria/acoplador foi satisfatório.

A estabilidade do oscilador foi satisfatória.

No caso de montagem do acoplador junto ao circuito memorizador, este deverá ser síncrono com o pulso de sincronismo externo do computador.

VIII - CONCLUSÃO

A soma dos tempos de execução das instruções escolhidos para o teste, e o tempo de execução do programa de teste, é de aproximadamente 17 vezes.

Para o teste de determinadas instruções, ocorre um tempo morto, devido ao uso de instruções testadas anteriormente. Este tempo morto poderá ser diminuído com a otimização do programa de teste ou com o teste micro-programado. Sem entrar no mérito da viabilidade econômica da última opção, esta parece ser razoável, sobretudo nos computadores com tarefas "stand alone". O programa de teste aqui elaborado executa periódicas comparações e, em caso de erro, para em "loops" infinitos. A vantagem deste tipo de teste, decorre das informações que possa oferecer quanto a natureza da falha - muito embora não se tenha preocupação com a diagnose propriamente num conjunto restrito de instruções. A diminuição do número destas comparações, que poderá ser reduzida a um no final da manipulação de determinado dado pelo conjunto de instruções, certamente simplificará o programa como diminuirá ser tempo de execução e memória gasta.

O algoritmo de escolha tem semelhança àqueles usados em minimização de funções booleanas.

O sistema implementado foi posto a funcionar no Mitra 15 do Laboratório de Sistemas e Simulação em duas opções: com a formação da tabela CTX e DVT e programas em linguagem de máquina, sem o auxílio do sistema monitor, dando-se entradas as instruções e dados pelo teclado; e com o MOB.

Sem o monitor não é necessário o uso de sinal PIT e, programas e tabelas devem dar entradas via chaves. Com o MOB a facilidade é maior, sobretudo na

na modificação de programas e uso de módulos do monitor - M:KEY, M:WAIT. O desempenho da minuteria cãõ de guarda foi o mesmo nos dois casos.

A falta de informação do sistema MITRA 15, como seu "software" - montador Mitra 1 e 2 -, diluiu grande parte do tepo gasto neste trabalho, o que marcou o todo.

Um dos programas com que, experimentalmente, o sistema foi testado emite uma cadeia de caracteres(BACKGROUND) pelo teletipo. O programa de teste é executado (no nível de TT4) entre a emissão de cda caracter desta cadeia. Não interrompendo, aparentemente, o trabalho de E/S do programa(BACKGROUND).

Os sistemas de Back Up manual e analógico (II-6) necessitarãõ de um sinal que comande as "chaves de Back Up". O sinal de alarma do sistema implementado poderã ser usado para o comando dessas chaves.

A necessidade de trabalho contínuo de dois computadores no sistema descrito no parágrafo (II-6-3), torna-o, em muitas aplicações, demais oneroso. A aplicação do computador de Back Up em outras atividades, fora do "plant" de controle, dá ao sistema proposto as vantagens dessa disponibilidade o que se traduz no maior uso de um computador e, conseqüentemente, menor custo operacional.

A implementação completa do sistema proposto, necessitarã de unidades de disco magnético na qual se permita o acesso de dois computadores independentemente.

O desenvolvimento de um trabalho neste item trará, sem dúvida, sensível contribuição a área de sistemas digitais entre nós.

A diagnose de falhas, (Fault Diagnose), é um dos assuntos que atualmente tem grande desenvolvimento, no mundo da ciência da computação. O emprêgo dessas técnicas usando micro-programas poderá dar boas colheitas à quem neste campo semear.

BIBLIOGRAFIA

1. Internal Report, Take Over System, Cerci, Paris, April, 71.
1. Chang, Fault Diagnosis of Digital Systems, Wiley Interscience, New York, 70.
3. E.M.Prell, Automatic Trouble Isolation in Duplex Central Control Employing Matching, Spring Joint Conference, 76.
4. C.M.Cundall, Back-Up-Methods for DDC, N.Y., McGraw-Hill, Control Engineering, Sept., 71.
5. R.E.Forbes, A Self Diagnosable Computer, Proceeding Fall Joint Computer Conference, 65.
6. G.C.Hendrie, Reliability Still Means Back Up, McGraw-Hill, Control Engineering, Sept., 66.
7. Knuth D.E, Art of Computer Programming, Addison Wesley, 68.
8. Electronique & Microelectronique Industrielle, Paris, Fev.
9. Myers, Reliability Engineering for Electronic Systems, John Wiley & Sons, Inc., 64.
10. Manual de Presentation Mitra 15, CII, Paris, 72
11. Reference Manual Mitra 15, CII, Paris, 72.
12. CPU Manual Mitra 15 vol. 1 e 2, CII, Paris, 72.
13. Temps Reel Mitra 15, CII, paris, 72.

APÊNDICE A

A-I Introdução.

Para medir a confiabilidade de um sistema são frequentemente usadas as grandezas, Tempo Médio Entre Falhas e Tempo Médio de Reparo, respectivamente TMEF e TMR.

A-II TMEF e TMR

O TMEF é determinado fazendo-se vários testes no equipamento, verificando-se a incidência de falhas durante estes testes e tornando-se o Tempo Médio por Número de Falhas. Se ocorrer cinco falhas num período de teste de 1000 Hs o Tempo Médio entre falhas é de:

$$\text{TMEF} = \frac{1000}{5} = 200 \text{ Horas.}$$

O TMEF aproxima-se do verdadeiro com o aumento do número de amostras.

Outro importante conceito é o da capacidade de manutenção do responsável pela manutenção do equipamento. É representada pela grandeza Tempo Médio de Reparo, TMR: O cálculo do TMR é obtido, tomando-se o número total de horas em que o equipamento está parado, dividindo-se pelo número total de falhas.

A disponibilidade pode ser calculada através do TMEF e TMR:

$$D = \frac{\text{TMEF}}{\text{TMEF} + \text{TMR}}$$

A-2

Se definirmos a disponibilidade como sendo dada pelo número de horas (horas em funcionamento = horas em reparo).

O seguinte exemplo mostra que devemos ter um pleno conhecimento das grandezas para evitarmos enganos.

Dois computadores, A e B, são garantidos como tendo disponibilidade de 99,5% num período de 6 meses.

O computador A tem TMEF de 1000 Hs. e sua equipe de manutenção gasta em média 5 hs. para chegar ao local da instalação e sanar o defeito, depois de chamada. O computador B tem TMEF de 66 hs., mas sua equipe gasta 20 minutos para sanar o defeito, depois de chamada. Aqueles que comprarem A terão o conceito de confiabilidade, com relação a disponibilidade, diferente dos que comprarem B.

O TMEF poderá ser aumentado acrescentando-se partes redundantes ao sistema. Isto é, o sistema pode ser obtido através da duplicação dos componentes críticos. Esta substituição deve ser automática e imediata após o defeito.

O TMEF será dado pela formula empírica:

$$\text{TMEF} = \frac{M (3R + M)}{2R}$$

onde: M é o TMEF de cada parte redundante.

R é o Tempo de Reparo da Parte Falha.

Assim conhecido M e R de um sistema Back Up seu TMEF pode ser calculado.

APÊNDICE B

Neste apêndice são apresentadas:

- . As listagens do programa ESCOLHE.
- . Uma lista das micro-instruções, em hexadecimal, no se-

guinte formato:

XX Y WWW

MICROS QUE COMPÕE O MICRO- PROGRAMA ANTERIOR

HHHH HHHH

onde:

- XX = ao código do macro-código (vide apêndice C)
- Y = ao número de micro-instruções do seu micro-programa
- WWW = ao tempo de execução do macro-código
- HHHH = a micro-instruções.

- . Resultado do programa ESCOLHE.

B-1

FORTRAN IV G LEVEL 20

MAIN

DATE = 73031

20/08/04

PAGE

```

C
C   PROGRAMA PRINCIPAL
C   FUNCAO :COLOCAR UM CONJUNTO DE ELEMENTOS
C           NA FORMA DE MATRIZ BINARIA,QUE E' A
C           FORMA NECESSARIA PARA O TRABALHO DA SUBROUTINA ESCOLHE.
C
C   M=MATRIZ MACRO X MICRO LIDA
C   NM=MATRIZ MACRO X MICRO NA FORMA BINARIA
C   R=VETOR QUE FAZ A REALOCAO DOS ELEMENTOS DA MATRIZ MACRO/MICRO
C   TPO=GUARDA O TEMPO DE CADA MACRO
C   L=VETOR QUE APONTA AS LINHAS DA MATRIZ MACRO X MICRO
C

```

```

0001   INTEGER TP
0002   INTEGER U,C1,A,CC,Q,GG,G2,C
0003   INTEGER M(100,500)
0004   INTEGER R(1000)
0005   INTEGER*2 TPO(100),L(100),NNMI(100)
0006   LOGICAL*1 NM(100,500)
0007   COMMON NM/AREAL/R,NNMI,M
0008   DATA Q,U,C,CC/0,0,0,0/
0009   DATA GG,C1,K,LL/0,0,0,0/
0010   M(1,1)=0

```

```

C
C   LEITURA DO NUM DE MICROS EM CADA MICROPROGRAMA
C

```

```

0011   2   READ(5,1,END=901)MA,NMI,TP
0012   WRITE(6,1991)MA,NMI,TP
0013   1   FORMAT(2I3,I4)
0014   1991  FORMAT(2I6,I8)
0015   LL=LL+1
0016   TPO(LL)=TP
0017   L(LL)=MA
0018   NNMI(LL)=NMI
0019   IF(GG.GE.NMI)GOTO 3331
0020   701  GG=NMI

```

```

C
C   LEITURA DAS MICROS SENDO UMA EM CADA CARTAO
C

```

```

0021   3331 READ(5,4)((IMU,M(LL,C1)),C1=1,NMI)
0022   4   FORMAT(7(I6,Z4))
0023   PRINT 31
0024   31  FORMAT('   MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR')
0025   WRITE(6,63)(M(LL,C1),C1=1,NMI)
0026   63  FORMAT(20(2X,Z4),/)

```

```

C
C   VETOR DE REALOCAO N(K)
C

```

```

0027   3  DO 9 IP=1,NMI
0028   K=K+1
0029   R(K)=M(LL,IP)
0030   9  CONTINUE
0031   CALL ORDEN(R,K)
0032   CALL REDUN(R,K)
0033   GOTO 2
0034   901 NMI=GG
0035   MA=LL
0036   PRINT 32
0037   32  FORMAT('   VETOR REALOCAO')

```

B-2

FORTRAN IV G LEVEL 20

MAIN

DATE = 73031

20/08/04

PAG

```

0038      WRITE(6,63)(R(J),J=1,K)
0039      DO 19 LI=1,LL
0040      DO 19 LJ=1,K
0041      NM(LI,LJ)=.FALSE.
0042 19     CONTINUE
0043      DO 20 J=1,LL
0044      IF(NNMI(J).GE.NMI)GOTO 20
0045 702    NN=NNMI(J)+1
0046      DO 21 JJ=NN,NMI
0047      M(J,JJ)=0
0048 21     CONTINUE
0049 20     CONTINUE
0050      PRINT 33
0051 33     FORMAT('          FORMACAO DA MATRIZ MACRO X MICROINSTRUcoes')
0052      WRITE(6,63)((M(L1,C1),L1=1,LL),C1=1,NMI)

C
C     PREPARA O CONJUNTO PARA O FORMATO
C     QUE DARA ENTRADA NA SUBROTINA ESCOLHA
C

0053 681  DO 11 I=1,MA
0054      DO 11 J=1,NMI
0055      DO 12 K1=1,K
0056      IF(M(I,J).EQ.R(K1))GOTO 13
0057 706    GOTO 12
0058      13 NM(I,K1)=.TRUE.
0059      GOTO 11
0060      12 CONTINUE
0061      11 CONTINUE
0062      WRITE(6,14)
0063 14     FORMAT(///,9X,'MATRIZ MACRO X MICRO NO FORMATO BINARIO',//)
0064      WRITE(6,60)((NM(I,K1),I=1,MA),K1=1,K),R(K1)
0065 60     FORMAT(2X,82I1,5X,Z4)
0066      CALL SCGLHE(MA,TPD,L,K)
0067      CALL EXIT
0068      END

```


FORTRAN IV G LEVEL 20

SCOLHE

DATE = 73031

20/08/04

PAC

```

C      TESTA SE A MACRO E ESSENCIAL E ELIMINA SEUS PONTOS DE INFLUENCIA
C
0036  331  DO 26 K=1,MA
0037      IF(EU(K))26,26,666
0038  666  DO 6 CL=1,MI
0039      IF(.NOT.N(EU(K),CL))GOTO 6
0040      DO 25 J=1,MA
0041      N(J,CL)=.FALSE.
0042  25   CONTINUE
0043  6   CONTINUE .
0044  26  CONTINUE
0045      END=END+1
C
C      ORDENA AS MELAORES
C
0046      PRINT 34
0047  34  FORMAT ('          VETOR QUE GUARDA O NUM DE MICROS',/, '          AINDA N
      *AO REPRESENTADAS')
0048      DO 9 J=1,MA
0049      C(J)=0
0050      DO 8 Y=1,MI
0051      IF(.NOT.N(J,Y))GOTO 8
0052  111  C(J)=C(J)+1
0053  8   CONTINUE
0054  70  FORMAT(4X,'C(',I3,')=',I3,2X,'C(',I2,')=',I3,2X,'C(',I2,')=',I3)
0055  9   CONTINUE
0056      WRITE(6,70)(J,C(J),J=1,MA)
0057      MX=MA-1
0058      DO 21 J=1,MX
0059      YJ=J+1
0060      DO 10 Y=YJ,MA
0061      IF(C(Y).GE.C(J))GOTO 10
0062      A=C(J)
0063      C(J)=C(Y)
0064      C(Y)=A
0065      A=L(Y)
0066      L(Y)=L(J)
0067      L(J)=A
0068      A=TPO(Y)
0069      TPO(Y)=TPO(J)
0070      TPO(J)=A
0071      DO 22 IA=1,MI
0072      Z=N(J,IA)
0073      N(J,IA)=N(Y,IA)
0074      N(Y,IA)=Z
0075  22  CONTINUE
0076  10  CONTINUE
0077  21  CONTINUE
0078      IF(C(MA).EQ.0)GOTO 333
0079      B=0
0080      DO 12 J=1,MA
0081      IF(C(MA).NE.C(MA-J))GOTO 19
0082      B=J
0083  12  CONTINUE
0084  19  IF(B.EQ.0)GOTO 108
0085  1080 MAB=MA-B
0086      MA1=MA-1
0087      DO 23 O=MAB,MA1

```

FORTRAN IV G LEVEL 20

SCOLHE

DATE = 73031

20/08/04

PA(

```

0088      OI=O+1
0089      DO 13 K=O1,MA
0090      IF(TPO(K).LE.TPO(O))GOTO 13
0091      A=TPO(O)
0092      TPO(O)=TPO(K)
0093      TPC(K)=A
0094      A=C(O)
0095      C(O)=C(K)
0096      C(K)=A
0097      A=L(O)
0098      L(O)=L(K)
0099      L(K)=A
0100      DO 24 IA=1,MI
0101      Z=N(O,IA)
0102      N(O,IA)=N(K,IA)
0103      N(K,IA)=Z
0104      24 CONTINUE
0105      13 CONTINUE
0106      23 CONTINUE
0107      108 I=I+1
0108      DO 71 J=1,MA
0109      EU(J)=0
0110      71 CONTINUE
0111      TPI(I)=TPO(MA)
0112      G(I)=L(MA)
0113      EU(I)=MA
0114      IF(END.LT.MA)GOTO 331
0115      333 WRITE(6,200)END
0116      200 FORMAT(4X,'END=',I3,'MARCA O NUM DE PESQ.FEITAS')
0117      WRITE(6,15)
0118      15 FORMAT(4X,'LISTA DAS MACROS ESCOLHIDAS')
0119      WRITE(6,20)(Q,G(Q),Q,TPI(Q),Q=1,I)
0120      20 FORMAT(10X,'G(',I3,')=',I3,2X,'TPO(',I3,')=',I9,5X,'G(',I3,')=',I3
*,2X,'TPO(',I3,')=',I9)
0121      ITPI=0
0122      DO 39 JK=1,I
0123      ITPI=ITPI+TPI(JK)
0124      39 CONTINUE
0125      WRITE(6,89)ITPI
0126      89 FORMAT(//,2X,'+ A SOMA DOS TEMPOS DAS MACROS ESCOLHIDAS =',I8)
0127      RETURN
0128      END

```

FORTRAN IV G LEVEL 20

ORDEN

DATE = 73031

20/08/04

PAGE

```

0001      SUBRGUTINE ORDEN(R,K)
          C
          C      SUBRGUTINA ORDEN;
          C      FUNCAO:  ORDENAR UM CONJUNTO DE VALGRES
          C
0002      INTEGER R(1000)
0003      INTEGER A
0004      INTEGER GG,G2
0005      GG=K-1
0006      DO 9 I=1,GG
0007      G2=I+1
0008      DO 8 J=G2,K
0009      IF(R(I).LE.R(J))GOTO 8
0010      703  A=R(I)
0011      R(I)=R(J)
0012      R(J)=A
0013      8  CONTINUE
0014      9  CONTINUE
0015      RETURN
0016      END

```

FORTRAN IV G LEVEL 20

REDUN

DATE = 73031

20/08/04

PA

```
0001      SUBROUTINE REDUN(R,K)
          C
          C      SUBROUTINA REDUN;
          C      FUNCAO:TIRAR OS TERMOS REDUNDANTES DE UM CONJUNTO
          C      DE VALORES ORDENADOS.
          C
0002      INTEGER R(1000)
          C
          C      TIRA OS TERMOS REDUNDANTES
          C
0003      1      KG=K-1
0004      DO 5 I=1,KG
0005      J=I+1
0006      IF(R(I).NE.R(J))GOTO 5
0007      DO 21 KI=J,K
0008      R(KI-1)=R(KI)
0009      21      CONTINUE
0010      R(K)=0
0011      K=K-1
0012      GOTO 1
0013      5      CONTINUE
0014      RETURN
0015      END
```


704C C908 AC18 5649 AC49 9AAB EC24 5649 9AAA DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 120 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 FOC8 EDA3 ECD5 F109 FLA4 ED15 DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 127 16 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 FOD0 EDA3 FL56 ECE1 F197 ED57 DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 134 14 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 FL1D ECA1 FL58 ED16 DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 139 11 60
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 FECE DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 141 11 60
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 F616 DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 143 21 400
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 CB19 C191 CC04 5AAE 84DB 7081 9AAB EC2A 56A0 9A8F EC4F DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 C18F
 155 14 60
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C9A0 C560 C59F DD1F DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 160 11 60
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 FAD6 DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 162 11 90
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 F124 DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 164 12 60
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 E34B FB57 DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 168 12 90
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 F124 F124 DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 171 11 90
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 FB3A DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 173 13 90
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 FAD5 C58C DCCC DB67 C2A7 5A67 D18F B3AA D19E DB67 C2A7 5A67 D18F
 180 5 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 9045 DB67 C2A7 5A67 D18F
 182 5 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 8957 DB67 C2A7 5A67 D18F
 192 45 520
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C121 DB08 4C0A ADC0 C708 C404 5FEO 80F1 C708 C402 58D5 ADE5 42A1 05C0 CC05 ED89 59D0 FF91 041B E388
 C868 4C0A ADD3 C708 C080 C618 ADDC 4282 CC05 ED8F 86E1 59CA FF93 860C E38D CA47 7022 DB67 C2A7 5A67 D18F
 D18F E348 C911 C191 D183
 230 10 350
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 F901 ECC5 DB67 C2A7 5A67 D18F B3AA C911 C191 D183
 233 1C 350
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR

FODE F906 DB67 C2A7 5A67 D16F B3A8 C911 C191 D183
 236 5 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 B84F DB67 C2A7 5A67 D18F
 238 5 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 B851 DB67 C2A7 5A67 D18F
 240 25 870
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 8264 752C ED43 709E EDE7 1E18 59E2 F081 1E59 57C6 05C4 CC14 C461 4C0C ADC4 F194 AC55 79F2 57CC AC0D
 4282 CE67 C2A7 5A67 D18F
 262 24 820
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 825A C24A DE9C C088 9AAD 8DC9 C904 300F 57E8 305D F1E9 EC55 57A2 9A97 6C0E F090 1E11 57EC F0A8 1E5D
 DB67 C2A7 5A67 D18F
 233 10 360
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 ADEA FFD1 C414 C708 FF31 EED5 DB67 C2A7 5A67 D18F
 290 12 420
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 B140 8191 ADD0 F0E0 FFD1 C61C C708 FF11 DB67 C2A7 5A67 D18F
 315 5 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 AECO C387 C3A4 D19F D19F
 317 5 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 AF02 C387 C3A4 D19F D19F
 319 5 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 AF44 C387 C3A4 D19F D19F
 321 5 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 8AC6 C387 C3A4 D19F D19F
 323 7 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 ECD3 B08B F8D3 C387 C3A4 D19F D19F
 327 5 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 9CCA C387 C3A4 D19F D19F
 329 5 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 88D4 C387 C3A4 D19F D19F
 331 5 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 A6CE C387 C3A4 D19F D19F
 337 5 210
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 A4D2 C387 C3A4 D19F D19F
 341 16 110
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 ADA0 FE0F C808 C84D F194 DEDA CC02 C84E C58C C54C C3CC F0DA C387 C3A4 D19F D19F
 354 32 900
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C82F AC48 F51A CC10 EDE9 F0DC FB91 E8D8 E3CD C89E C9CB F193 CA44 CA59 DF0D FD04 C50F F0EA E8CF F10D
 FD08 F1C2 C8DA F007 DFOE F008 D888 E918 C387 C3A4 D19F D19F
 383 8 260
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR

C8C1 B4C5 B2C5 C505 C387 C3A4 D19F D19F.
 388 8 260
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C8E6 BEE6 BEC9 C510 C387 C3A4 D19F D19F
 393 8 260
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C8E6 BF66 BF48 C51F C387 C3A4 D19F D19F
 398 8 340
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 EDEA AF28 57C3 ACC3 C387 C3A4 D19F D19F
 403 5 220
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 30C2 C387 C3A4 D19F D19F
 405 5 220
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 3104 C387 C3A4 D19F D19F
 407 5 220
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 3146 C387 C3A4 D19F D19F
 409 7 250
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C8CF 3CD1 3CD7 C387 C3A4 D19F D19F
 413 7 250
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C8CD 3CD8 3CDD C387 C3A4 D19F D19F
 417 7 350
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 EDE7 3102 57E5 D19F C387 C3A4 D19F
 421 6 240
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C6A7 ICCE C387 C3A4 D19F D19F
 424 10 280
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C2A9 CC04 C823 IA12 DE94 IE14 C387 C3A4 D19F D19F
 431 10 340
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 FB09 E099 F0C1 FB0F 2783 8287 C387 C3A4 D19F D19F

B-12

LISTA DAS MACROS ESCOLHIDAS

G(1)= 25	TPC(1)=	290	G(2)= 33	TPO(2)=	1900
G(3)= 36	TPO(3)=	220	G(4)= 40	TPC(4)=	220
G(5)= 44	TPO(5)=	0	G(6)= 48	TPO(6)=	210
G(7)= 50	TPC(7)=	210	G(8)= 53	TPO(8)=	210
G(9)= 59	TPO(9)=	210	G(10)= 76	TPO(10)=	210
G(11)= 78	TPC(11)=	210	G(12)= 81	TPO(12)=	210
G(13)= 86	TPC(13)=	210	G(14)= 91	TPC(14)=	210
G(15)= 94	TPO(15)=	210	G(16)= 97	TPC(16)=	210
G(17)=100	TPC(17)=	210	G(18)=110	TPO(18)=	270
G(19)=120	TPO(19)=	210	G(20)=127	TPO(20)=	210
G(21)=134	TPC(21)=	210	G(22)=139	TPO(22)=	60
G(23)=141	TPC(23)=	60	G(24)=143	TPO(24)=	400
G(25)=155	TPO(25)=	60	G(26)=160	TPC(26)=	60
G(27)=164	TPC(27)=	60	G(28)=171	TPO(28)=	90
G(29)=173	TPO(29)=	90	G(30)=180	TPO(30)=	210

G(31)=182	TPO(31)=	210	G(32)=192	TPO(32)=	520
G(33)=230	TPC(33)=	350	G(34)=233	TPO(34)=	350
G(35)=236	TPO(35)=	210	G(36)=238	TPC(36)=	210
G(37)=240	TPC(37)=	870	G(38)=262	TPO(38)=	820
G(39)=283	TPC(39)=	360	G(40)=290	TPC(40)=	420
G(41)=315	TPO(41)=	210	G(42)=317	TPC(42)=	210
G(43)=319	TPC(43)=	210	G(44)=321	TPO(44)=	210
G(45)=323	TPO(45)=	210	G(46)=327	TPC(46)=	210
G(47)=329	TPO(47)=	210	G(48)=331	TPO(48)=	210
G(49)=337	TPC(49)=	210	G(50)=341	TPO(50)=	110
G(51)=354	TPO(51)=	900	G(52)=383	TPC(52)=	260
G(53)=388	TPC(53)=	260	G(54)=393	TPO(54)=	260
G(55)=398	TPC(55)=	340	G(56)=403	TPC(56)=	220
G(57)=405	TPO(57)=	220	G(58)=407	TPC(58)=	220
G(59)=409	TPC(59)=	250	G(60)=413	TPO(60)=	250
G(61)=417	TPO(61)=	350	G(62)=421	TPC(62)=	240
G(63)=424	TPO(63)=	280	G(64)=431	TPO(64)=	340
G(65)=162	TPC(65)=	90	G(

+ A SOMA DOS TEMPOS DAS MACROS ESCOLHIDAS = 17940 μ s.

APÊNDICE D

Constam deste apêndice as listagem do programa de teste e seu diagrama de blocos.

PROGRAMA DE TESTE

%ASS2/SI,LO

```

1          CDTEST  CDS
2 0000          RES  16  *   AREA DE TRABALHO MOB
3 0020 0000 A  P12      DATA P1
4 0022 AAAA    DAD02    DATA &AAAA
5 0024 5555    DAD03    DATA &5555
6 0026 FFFF    DAD04    DATA &FFFF
7 0028 0000 A  P1      DATA P2
8 002A 0002    DAD01C   DATA &0002
9 002C 0000    DAD01    DATA &0000
10 002E 55     DAD03C   DATA,1 &55
11          BND
12          FIN
13          LDS10  LDS
14 0000          RES  2
15          FIN
16          FI10  LPS  LDS10
17 0000 4024    LDA  #DAD03 *   A=&5555
18 0002 2210    LDX  =16  *   X=16
19 0004 F301    DCX  =1   *   X=X-1
20 0006 F001    SLLS =1   *   SHIFTLOGICO EQUERDA DE
21 0008 C000    BCT  $-C  *   SE C=1 PARE!
22 000A F301    DCX  =1   *   X=X-1
23 000C F001    SLLS =1   *   SHIFT LOGICO A ESQUERDA
24 000E F104    XAX          *   X<->A
25 0010 C504    BAZ  $+4  *   SE A =C V.P. 4 POS. A E
26 0012 F104    XAX
27 0014 C808    BCT  $-8  *   SE C=1 VOLTE 8 POS.
28 0016 C700    BRU  $-C  *   SE NAO PARE
29 0018 F104    XAX          *   X<->A
30 001A C502    BAZ  $+2  *   SE A=C AVANCE 2 POS.
31 001C C700    BRU  $-C  *   SE NAO PARE
32 001E 4022    LDA  #DAD02 *   A=&AAAA
33 0020 2210    LDX  =16  *X=16
34 0022 F301    DCX  =1   *   X=X-1
35 0024 F001    SRLS =1   *   SHIFT LOGICO SIMPLES DEI
36 0026 C000    BCT  $-C  *   SE C=1 PARE
37 0028 F301    DCX  =1   *   X=X-1
38 002A F001    SRLS =1   *   SHIFT LOGICO SIMPLES P
39 002C F104    XAX          *   X<->A

```

D-3

40	002E	C504	BAZ	\$+4	*	SE A =C AVANCE 4 POS.
41	0030	F104	XAX		*	X<->A
42	0032	C808	BCT	\$-8	*	SE C=1 VOLTE 8 POS.
43	0034	C700	BRU	\$-0	*	SE NAO PARE
44	0036	F104	XAX		*	X<->A
45	0038	4B2C	CMP	#DAD01	*	A=&CCCC
46	003A	C300	BCF	\$+0	*	SE NAO PARE
47	003C	4C22	LDA	#DAD02	*	A=&AAAA
48	003E	2210	LDX	=16	*	X=16
49	0040	F301	DCX	=1	*	X=X-1
50	0042	F021	SRCS	=1	*	SHIFT CIRCULAR P/ DIR. ▢
51	0044	C000	BCT	\$+0	*	SE C=1 PARE
52	0046	F021	SRCS	=1	*	SHIFT CIRCULAR P/ DIR. ▢
53	0048	F301	DCX	=1	*	X=X-1
54	004A	F104	XAX		*	X<->A
55	004C	C504	BAZ	\$+4	*	SE A=C AVANCE 4 POS.
56	004E	F104	XAX			
57	0050	C808	BCT	\$-8	*	SE C=1 VOLTE 8 POS.
58	0052	C700	BRU	\$+0	*	SE NAO PARE
59	0054	F100	RTS			VOLTA AO PROG.PRINC.
60			FIN			
61		LDS11	LDS			
62	0000		RES	2		
63			FIN			
64			*****			
65		FI11	LPS	LDS11		
66	0000	F104	XAX		*	X<->A
67	0002	4B22	CMP	#DAD02	*	A=&AAAA ?
68	0004	C300	BCF	\$+0	*	SE NAO PARE
69	0006	2210	LDX	=16	*	X=16
70	0008	F301	DCX	=1	*	X=X-1
71	000A	F081	SLCS	=1	*	SHIFT CIRCULAR P/ DIR. ▢
72	000C	C300	BCF	\$+0	*	SE C=0 PARE
73	000E	F301	DCX	=1	*	X=X-1
74	0010	F081	SLCS	=1	*	SHIFT CIRCULAR P/ DIR. ▢
75	0012	F104	XAX		*	X<->A
76	0014	C504	BAZ	\$+4	*	SE A=C AVANCE 4 POS.
77	0016	F104	XAX			
78	0018	C808	BCF	\$-8	*	SE C=0 VOLTE 8 POS.
79	001A	C700	BRU	\$+0	*	SE NAO PARE
80	001C	F104	XAX		*	X<->A

D-4

```

81 001E 4B22          CMP #DAD02          * A=&AAAA?
82 0020 C300          BCF $+0           * SE NAO PARE
83 0022 5022          DLD #DAD02        * E=&AAAA,A=&5555
84 0024 F110          CCA                * A=A'=&AAAA
85 0026 2220          LDX =32           * X=32
86 0028 F301          DCX =1            * X=X-1
87 002A F0E1          SRGD =1          * SHIFT CIRCULAR P/DIR.
88 002C 0000          BCT $+0          * SE C=1 PARE!
89 002E F301          DCX =1            * X=X-1
90 0030 F0E1          SRGD =1          * SHIFT CIRCULAR P/DIR.
91 0032 F104          XAX                * X<->A
92 0034 C504          BAZ $+04         * SE A=C AVANCE 4 POS.
93 0036 F104          XAX
94 0038 C808          BCT $-8           * SE C=1 VOLTE 8 POS.
95 003A C700          BRU $+0           * SE NAO PARE!
96 003C F104          XAX                * X<->A
97 003E 2220          LDX =32           * X=32
98 0040 F301          DCX =1            * X=X-1
99 0042 F061          SLCD =1           * SHIFT CIRCULAR P/ESQ.
100 0044 C300          BCF $+0           * SE C=0 PARE!
101 0046 F301          DCX =1            * X=X-1
102 0048 F061          SLCD =1           * SHIFT CIRCULAR P/ESQ.
103 004A F104          XAX                * X<->A
104 004C C504          BAZ $+4           * SE A=C AVANCE 4 POS.
105 004E F104          XAX
106 0050 C808          BCF $-8           * SE C=0 VOLTE 8 POS.
107 0052 C700          BRU $+0
108 0054 F100          RTS
109                      FIN
110                      LDS12  LDS
111 0000                      RES 2
112 0004 0000 A          P3      DATA  P5
113 0006 AAAA            P5      DATA  &AAAA
114 0008 D555            LUGA31  DATA  &D555
115 000A 0000            LUGAR3  DATA  &0000
116 000C 0055            DAD05   DATA  &0055
117                      FIN
118                      *****
119                      FI12  LPS  LDS12
120 0000 F104          XAX                * A<->A

```


D-5

121	0002	221F	LDX	=31	*	X=31
122	0004	F301	DCX	=1	*	X=X-1
123	0006	F041	SAD	=1	*	SHIFT ARITMETICO P/DIR
124	0008	F104	XAX			
125	000A	C507	BAZ	\$+7		
126	000C	F104	XAX			
127	000E	C000	BCT	\$+0	*	SE C=1 PARE!
128	0010	F301	DCX	=1	*	X=X-1
129	0012	F041	SAD	=1	*	SHIFT ARITMETICO P/DIR
130	0014	C808	BCT	\$-8	*	SE C=1 VOLTE 8 POS.
131	0016	C700	BRU	\$+0	*	SE NAO PARE
132	0018	F104	XAX		*	X<->A
133	001A	4B26	CMP	#DAD04	*	A=&FFFF ?
134	001C	C300	BCF	\$+0	*	SE NAO PARE!
135	001E	6004	LDA	@P3	*	A=&FFFF
136	0020	2210	LDX	=16	*	X=16
137	0022	F301	DCX	=1	*	X=X-1
138	0024	F0A1	SAS	=1	*	SHIFT SIMPLES P/DIR.
139	0026	C000	BCT	\$+0	*	SE C=1 PARE!
140	0028	F301	DCX	=1	*	X=X-1
141	002A	F0A1	SAS	=1	*	SHIFT SIMPLES P/DIR.
142	002C	F104	XAX		*	X<->A
143	002E	C504	BAZ	\$+4		
144	0030	F104	XAX			
145	0032	C808	BCT	\$-8	*	SE C=1 VOLTE 8 POS.
146	0034	C700	BRU	\$+0	*	SE NAO PARE!
147	0036	F104	XAX			
148	0038	4022	LDA	#DAD02	*	A=&AAAA
149	003A	4122	LDE	#DAD02	*	E=&AAAA
150	003C	F102	XAE		*	A<-EEE
151	003E	F11C	CNA		*	A=-A
152	0040	F106	XEX		*	E<->X
153	0042	F114	CNX		*	X=-X
154	0044	130A	STX	LUGAR3	*	LUGAR3=X
155	0046	0BCA	CMP	LUGAR3	*	A=LUGAR3 ?
156	0048	C300	BCF	\$+0	*	SE NAO PARE!
157	004A	402C	LDA	#DAD01	*	A=&0000
158	004C	4D2E	LBL	#DAD030	*	A=&5500
159	004E	F108	XAA		*	A=&0055
160	0050	0BCC	CMP	DAD05	*	A=&0055?

D-6

```

161 0052 F10A      CCE                      * E=&0000' =&FFFF
162 0054 F102      XAE                      * A=&FFFF, E=&0055
163 0056 4126      LDE #DAD04             * E=&FFFF
164 0058 F112      AEE                      * A=&0000
165 005A 4B2C      CMP #DAD01             * A=&0000?
166 005C C300      BCF $+C                * SE NAO PARE!
167 005E F116      AIE                      * A=&0000 OU &FFFF=&FFF
168 0060 4B26      CMP #DAD04             * A=&FFFF
169 0062 C300      BCF $+C                * SE NAO PARE!
170 0064 F11A      LNE                      * E=-1
171 0066 F10E      ACE                      * E=E+C
172 0068 F102      XAE                      * A<->E
173 006A 4B2C      CMP #DAD01             * A=&0000?
174 006C C300      BCF $+C                * SE NAO PARE!
175 006E 4222      LDX #DAD02             * X=&AAAA
176 0070 F11E      CHX                      * X=X/2
177 0072 130A      STX LUGAR3             * LUGAR3=&D555
178 0074 0008      LDA LUGA31             * A=&D555
179 0076 0BCA      CMP LUGAR3             * A=&D555?
180 0078 C300      BCF $+C                * SE NAO PARE!
181 007A F100      RTS
182                      FIN
183                      *****
184                      LDSFIC  LDS
185 0000                      RES 2
186 0004 0000      LUGAR1  DATA &0000
187 0006 0000      LUGAR2  DATA &0000
188 0008 5554      LUGP    DATA &5554
189 000A 0000  A    P4    DATA P11
190 000C 0000  A    P6    DATA P7
191                      FIN
192                      FIG    LPS  LDSFIC
193 0000 4022      P11    LDA #DAD02     * A=&AAAA
194 0002 1106      STA LUGAR2          * LUGAR2=&AAAA
195 0004 0306      EOR LUGAR2          * A=A EOR LUGAR2
196 0006 4B2C      CMP #DAD01          * A=&0000 ?
197 0008 C300      BCF $-C             * SE NAO PARE.
198 000A 4722      IOR #DAD02          * A= A OU DAD02=&AAAA
199 000C 4924      AND #DAD03          * A= A E DAD03=&0000
200 000E 4B2C      CMP #DAD01          * A=&0000 ?

```

201	0010	C300	BCF	\$-0	*	SE NAO PARE.
202	0012	4E2E	LBR	#DAD030	*	A=&0055
203	0014	4D2E	LBL	#DAD030	*	A=&5555
204	0016	4B24	CMP	#DAD03	*	A=&5555 ?
205	0018	C300	BCF	\$+0	*	SE NAO PARE.
206	001A	5022	DLA	#DAD02	*	E=&AAAA, A=5555
207	001C	1604	DST	LUGAR1	*	LUGAR1=&AAAA, LUGAR2=&55
208	001E	0B06	CMP	LUGAR2	*	A=LUGAR2 ?
209	0020	C300	BCF	\$+0	*	SE NAO PARE
210	0022	4022	LDA	#DAD02		*A=&AAAA
211	0024	0B04	CMP	LUGAR1	*	A=LUGAR1 ?
212	0026	C300	BCF	\$+0	*	SE NAO PARE
213	0028	1506	SBR	LUGAR2	*	LUGAR2=&=55AA
214	002A	1406	SBL	LUGAR2	*	LUGAR2=&AAAA
215	002C	1706	ADM	LUGAR2		*A=LUGAR2=&5554
216	002E	0B08	CMP	LUGP		*A=&5554 ?
217	0030	422A	LDX	#DAD010		*X=0002
218	0032	1304	STX	LUGAR1	*	LUGAR1=&0002
219	0034	8020	LDA	@#P12, X	*	A=&0002
220	0036	0B04	CMP	LUGAR1	*	A=LUGAR1?
221	0038	C300	BCF	\$-0	*	SE NAO PARE.
222	003A	1206	STE	LUGAR2	*	LUGAR2=&AAAA
223	003C	4022	LDA	#DAD02		*A=&AAAA
224	003E	0B06	CMP	LUGAR2	*	A=LUGAR2?
225	0040	C300	BCF	\$+0	*	SE NAO PARE.
226	0042	502A	DLA	#DAD010	*	E=&0002, A=&0000
227	0044	F102	XAE		*	E <-> A
228	0046	4C24	MUL	#DAD03	*	A=A X &5555 =&AAAA
229	0048	1104	STA	LUGAR1	*	LUGAR1=&AAAA
230	004A	4624	SUB	#DAD03	*	A=&AAAA-&5555=&5555
231	004C	4524	ADD	#DAD03	*	A=&5555+&5555=&AAAA
232	004E	0B04	CMP	LUGAR1	*	A=LUGAR1?
233	0050	C300	BCF	\$-0	*	SE NAO PARE
234	0052	0000	CLS	F110		
235	0054	0000	CLS	F111		
236	0056	0000	CLS	F112		
237	0058	2200	LDX	=0		
238	005A	0000	CSV	M:KEY		
239	005C	2201	LDX	=1		
240	005E	0000	CSV	M:KEY		

D-8

241	0060	F502		ICL =2		
242	0062	DF28		BRU @#P1	*	VA PARA P2,USANDO ENDE
243			*			IND. GERAL.
244	0064	C700		BRU \$+C	*	SE O BALTO NAO FOR EXE
245			*			DO PARE!
246	0066	C700		BRU \$+C		
247	0068	D70A	P2	BRU @P4	*	COM A BASE L INC.DE 2,0E
248			*			TO VAI ATE P7.
249	006A	C700		BRU \$+C	*	SE O SALTO NAO FOR EXE
250			*			DO PARE!
251	006C	C700		BRU \$+C	*	SE O SALTO NAO FOR EXE
252			*			PARE!
253	006E	F602	P7	DCL =2	*	VOLTA BASE L A POSICAO
254			*			GINAL.
255	0070	0000		CSV M:EXIT		
256	0072	CF39		BRU P11		
257				FIN P11		
258				END FIC		

%ASS2/

D-9

PROGRAMA DE TESTE
(com simulador)

%ASS2/SI,LO

```

1          DTEST  CDS
2 0000          RES  16  *   AREA DE TRABALHO MOB
3 0020 0000 A  P12      DATA P1
4 0022 AAAA    DAD02    DATA &AAAA
5 0024 5555    DAD03    DATA &5555
6 0026 FFFF    DAD04    DATA &FFFF
7 0028 0000 A  P1       DATA P2
8 002A 0002    DAD01C   DATA &0002
9 002C 0000    DAD01    DATA &0000
10 002E 55     DAD03C   DATA,1  &55
11          BND
12          FIN
13          LDS10  LDS
14 0000          RES  2
15          FIN
16          FI10   LPS  LDS10
17 0000 4024    LDA     #DAD03 *   A=&5555
18 0002 2210    LDX   =16  *   X=16
19 0004 F301    DCX   =1   *   X=X-1
20 0006 F001    SRLS  =1   *   SHIFT LOGICO EQUERDA DE
21 0008 C000    BCT   $-0  *   SE C=1 PARE!
22 000A F301    DCX   =1   *   X=X-1
23 000C F001    SRLS  =1   *   SHIFT LOGICO A ESQUERDA
24 000E F104    XAX          *   X<->A
25 0010 C504    BAZ   $+4  *   SE A =C V.P. 4 POS. A E
26 0012 F104    XAX
27 0014 C808    BCT   $-8  *   SE C=1 VOLTE 8 POS.
28 0016 C700    BRU   $-0  *   SE NAO PARE
29 0018 F104    XAX          *   X<->A
30 001A C502    BAZ   $+2  *   SE A=C AVANCE 2 POS.
31 001C C700    BRU   $-0  *   SE NAO PARE
32 001E 4022    LDA   #DAD02 *   A=&AAAA
33 0020 2210    LDX   =16  *X=16
34 0022 F301    DCX   =1   *   X=X-1
35 0024 F001    SRLS  =1   *   SHIFT LOGICO SIMPLES DEI
36 0026 C000    BCT   $-0  *   SE C=1 PARE
37 0028 F301    DCX   =1   *   X=X-1
38 002A F001    SRLS  =1   *   SHIFT LOGICO SIMPLES PE
39 002C F104    XAX          *   X<->A
40 002E C504    BAZ   $+4  *   SE A =C AVANCE 4 POS.
41 0030 F104    XAX          *   X<->A
42 0032 C808    BCT   $-8  *   SE C=1 VOLTE 8 POS.

```

```

43 0034 C700      BRU $-0      *      SE NAO PARE
44 0036 F104      XAX          *      X<->A
45 0038 4B2C      CMP #DAD01  *      A=&0000
46 003A C300      BCF $+0     *      SE NAO PARE
47 003C 4022      LDA #DAD02  *      A=&AAAA
48 003E 2210      LDX =16     *      X=16
49 0040 F301      DCX =1      *      X=X-1
50 0042 F021      SRCS =1     *      SHIFT CIRCULAR P/ DIR.
51 0044 C000      BCT $+0     *      SE C=1 PARE
52 0046 F021      SRCS =1     *      SHIFT CIRCULAR P/ DIR.
53 0048 F301      DCX =1      *      X=X-1
54 004A F104      XAX          *      X<->A
55 004C C504      BAZ $+4     *      SE A=C AVANCE 4 POS.
56 004E F104      XAX
57 0050 C808      BCT $-8     *      SE C=1 VOLTE 8 POS.
58 0052 C700      BRU $+0     *      SE NAO PARE
59 0054 F100      RTS
60              FIN
61              LDS11  LDS
62 0000              RES 2
63              FIN
64
65              *****
66              F111  LPS  LDS11
67 0000 F104      XAX          *      X<->A
68 0002 4B22      CMP #DAD02  *      A=&AAAA ?
69 0004 C300      BCF $+0     *      SE NAO PARE
70 0006 2210      LDX =16     *      X=16
71 0008 F301      DCX =1      *      X=X-1
72 000A F081      SLCS =1     *      SHIFT CIRCULAR P/ DIR.
73 000C C300      BCF $+0     *      SE C=C PARE
74 000E F301      DCX =1     *      X=X-1
75 0010 F081      SLCS =1     *      SHIFT CIRCULAR P/ DIR.
76 0012 F104      XAX          *      X<->A
77 0014 C504      BAZ $+4     *      SE A=C AVANCE 4 POS.
78 0016 F104      XAX
79 0018 C808      BCF $-8     *      SE C=C VOLTE 8 POS.
80 001A C700      BRU $+0     *      SE NAO PARE
81 001C F104      XAX          *      X<->A
82 001E 4B22      CMP #DAD02  *      A=&AAAA?
83 0020 C300      BCF $+0     *      SE NAO PARE
84 0022 5022      DLD #DAD02  *      E=&AAAA,A=&5555

```

D-11

84	0024	F110	CCA	*	A=A'=&AAAA
85	0026	2220	LDX =32	*	X=32
86	0028	F301	DCX =1	*	X=X-1
87	002A	F0E1	SRCD =1	*	SHIFT CIRCULAR P/DIR. \boxtimes
88	002C	C000	BCT \$+0	*	SE C=1 PARE!
89	002E	F301	DCX =1	*	X=X-1
90	0030	F0E1	SRCD =1	*	SHIFT CIRCULAR P/DIR. \boxtimes
91	0032	F104	XAX	*	X<->A
92	0034	C504	BAZ \$+04	*	SE A=C AVANCE 4 POS.
93	0036	F104	XAX		
94	0038	C808	BCT \$-8	*	SE C=1 VOLTE 8 POS.
95	003A	C700	BRU \$+0	*	SE NAO PARE!
96	003C	F104	XAX	*	X<->A
97	003E	2220	LDX =32	*	X=32
98	0040	F301	DCX =1	*	X=X-1
99	0042	F061	SLCD =1	*	SHIFT CIRCULAR P/ESQ. \boxtimes
100	0044	C300	BCF \$+0	*	SE C=C PARE!
101	0046	F301	DCX =1	*	X=X-1
102	0048	F061	SLCD =1	*	SHIFT CIRCULAR P/ESQ. \boxtimes
103	004A	F104	XAX	*	X<->A
104	004C	C504	BAZ \$+4	*	SE A=C AVANCE 4 POS.
105	004E	F104	XAX		
106	0050	C808	BCF \$-8	*	SE C=C VOLTE 8 POS.
107	0052	C700	BRU \$+0		
108	0054	F100	RTS		
109			FIN		
110			LDS12		
111	0000		RES 2		
112	0004	0000 A	P3 DATA P5		
113	0006	AAAA	P5 DATA &AAAA		
114	0008	D555	LUGA31 DATA &D555		
115	000A	0000	LUGAR3 DATA &0000		
116	000C	0055	DAD05 DATA &0055		
117			FIN		
118			*****		
119			FI12 LPS LDS12		
120	0000	F104	XAX	*	A<->A
121	0002	221F	LDX =31	*	X=31
122	0004	F301	DCX =1	*	X=X-1
123	0006	F041	SAD =1	*	SHIFT ARITMETICO P/DI \boxtimes

124	0008	F104	XAX			
125	000A	C507	BAZ	\$+7		
126	000C	F104	XAX			
127	000E	C000	BCT	\$+0	*	SE C=1 PARE!
128	0010	F301	DCX	=1	*	X=X-1
129	0012	F041	SAD	=1	*	SHIFT ARITMETICO P/DIR.
130	0014	C808	BCT	\$-8	*	SE C=1 VOLTE 8 POS.
131	0016	C700	BRU	\$+0	*	SE NAO PARE
132	0018	F104	XAX		*	X<->A
133	001A	4B26	CMP	#DAD04	*	A=&FFFF ?
134	001C	C300	BCF	\$+0	*	SE NAO PARE!
135	001E	6004	LDA	@P3	*	A=&FFFF
136	0020	2210	LDX	=16	*	X=16
137	0022	F301	DCX	=1	*	X=X-1
138	0024	F0A1	SAS	=1	*	SHIFT SIMPLES P/DIR.
139	0026	C000	BCT	\$+0	*	SE C=1 PARE!
140	0028	F301	DCX	=1	*	X=X-1
141	002A	F0A1	SAS	=1	*	SHIFT SIMPLES P/DIR.
142	002C	F104	XAX		*	X<->A
143	002E	C504	BAZ	\$+4		
144	0030	F104	XAX			
145	0032	C808	BCT	\$-8	*	SE C=1 VOLTE 8 POS.
146	0034	C700	BRU	\$+0	*	SE NAO PARE!
147	0036	F104	XAX			
148	0038	4022	LDA	#DAD02	*	A=&AAAA
149	003A	4122	LDE	#DAD02	*	E=&AAAA
150	003C	F102	XAE		*	A<-EEE
151	003E	F11C	CNA		*	A=-A
152	0040	F106	XEX		*	E<->X
153	0042	F114	CNX		*	X=-X
154	0044	130A	STX	LUGAR3	*	LUGAR3=X
155	0046	0BCA	CMP	LUGAR3	*	A=LUGAR3 ?
156	0048	C300	BCF	\$+0	*	SE NAO PARE!
157	004A	402C	LDA	#DAD01	*	A=&0000
158	004C	4D2E	LBL	#DAD030	*	A=&5500
159	004E	F108	XAA		*	A=&0055
160	0050	0B0C	CMP	DAD05	*	A=&0055?
161	0052	F10A	CCE		*	E=&0000' =&FFFF
162	0054	F102	XAE		*	A=&FFFF, E=&0055
163	0056	4126	LDE	#DAD04	*	E=&FFFF


```

164 0058 F112      AEE          *      A=&0000
165 005A 4B2C      CMP #DAD01   *      A=&0000?
166 005C C300      BCF $+0     *      SE NAO PARE!
167 005E F116      AIE          *      A=&0000 OU  &FFFF=&FFF
168 0060 4B26      CMP #DAD04   *      A=&FFFF
169 0062 C300      BCF $+C     *      SE NAO PARE!
170 0064 F11A      LNE          *      E=- 1
171 0066 F10E      ACE          *      E=E+C
172 0068 F102      XAE          *      A<->E
173 006A 4B2C      CMP #DAD01   *      A=&0000?
174 006C C300      BCF $+0     *      SE NAO PARE!
175 006E 4222      LDX #DAD02   *      X=&AAAA
176 0070 F11E      CHX          *      X=X/2
177 0072 130A      STX LUGAR3   *      LUGAR3=&D555
178 0074 0008      LDA LUGA31   *      A=&D555
179 0076 0BCA      CMP LUGAR3   *      A=&D555?
180 0078 C300      BCF $+0     *      SE NAO PARE!
181 007A F100      RTS
182                FIN
183                *****
184                LDSFIC  LDS
185 0000                RES  2
186 0004 0000          LOOP  DATA &0000
187 0006 0000          LUGAR1 DATA &0000
188 0008 0000          LUGAR2 DATA &0000
189 000A 5554          LUGP  DATA &5554
190 000C 0000 A        P4    DATA P11
191 000E 0000 A        P6    DATA P7
192                FIN
193                FIG    LPS  LDSFIC
194 0000 2200          P11   LDX  =0
195 0002 0000          CSV  M:KEY
196 0004 2201          LDX  =1
197 0006 0000          CSV  M:KEY
198 0008 1104          STA  LOOP
199 000A 0004          RP   LDA  LOOP
200 000C F104          XAX
201 000E F301          DCX  =1
202 0010 F104          XAX
203 0012 1104          STA  LOOP
204 0014 4022          LDA  #DAD02 *      A=&AAAA

```

D-14

205	0016	1108	STA	LUGAR2	*	LUGAR2=&AAAA
206	0018	0308	EOR	LUGAR2	*	A=A EOR LUGAR2
207	001A	4B2C	CMP	#DAD01	*	A=&0000 ?
208	001C	C300	BCF	\$-C	*	SE NAO PARE.
209	001E	4722	IOR	#DAD02	*	A= A OU DAD02=&AAAA
210	0020	4924	AND	#DAD03	*	A= A E DAD03=&0000
211	0022	4B2C	CMP	#DAD01	*	A=&00000 ?
212	0024	C300	BCF	\$-C	*	SE NAO PARE.
213	0026	4E2E	LBR	#DAD030	*	A=&0055
214	0028	4D2E	LBL	#DAD030	*	A=&5555
215	002A	4B24	CMP	#DAD03	*	A=&5555 ?
216	002C	C300	BCF	\$+C	*	SE NAO PARE.
217	002E	5022	DLD	#DAD02	*	E=&AAAA, A=5555
218	0030	1606	DST	LUGAR1	*	LUGAR1=&AAAA, LUGAR2=&55
219	0032	0B08	CMP	LUGAR2	*	A=LUGAR2 ?
220	0034	C300	BCF	\$+C	*	SE NAO PARE.
221	0036	4022	LDA	#DAD02	*	*A=&AAAA
222	0038	0B06	CMP	LUGAR1	*	A=LUGAR1 ?
223	003A	C300	BCF	\$+C	*	SE NAO PARE
224	003C	1508	SBR	LUGAR2	*	LUGAR2=&=55AA
225	003E	1408	SBL	LUGAR2	*	LUGAR2=&AAAA
226	0040	1708	ADM	LUGAR2	*	*A=LUGAR2=&5554
227	0042	0BCA	CMP	LUGP	*	*A=&5554 ?
228	0044	422A	LDX	#DAD010	*	*X=0002
229	0046	1306	STX	LUGAR1	*	LUGAR1=&0002
230	0048	8020	LDA	@#P12,X	*	A=&0002
231	004A	0B06	CMP	LUGAR1	*	A=LUGAR1?
232	004C	C300	BCF	\$-C	*	SE NAO PARE.
233	004E	1208	STE	LUGAR2	*	LUGAR2=&AAAA
234	0050	4022	LDA	#DAD02	*	*A=&AAAA
235	0052	0B08	CMP	LUGAR2	*	A=LUGAR2?
236	0054	C300	BCF	\$+C	*	SE NAO PARE.
237	0056	502A	DLD	#DAD010	*	E=&0002, A=&0000
238	0058	F102	XAE		*	E <-> A
239	005A	4024	MUL	#DAD03	*	A=A X &5555 =&AAAA
240	005C	1106	STA	LUGAR1	*	LUGAR1=&AAAA
241	005E	4624	SUB	#DAD03	*	A=&AAAA-&5555=&5555
242	0060	4524	ADD	#DAD03	*	A=&5555+&5555=&AAAA
243	0062	0B06	CMP	LUGAR1	*	A=LUGAR1?
244	0064	C300	BCF	\$-C	*	SE NAO PARE

D-15

```

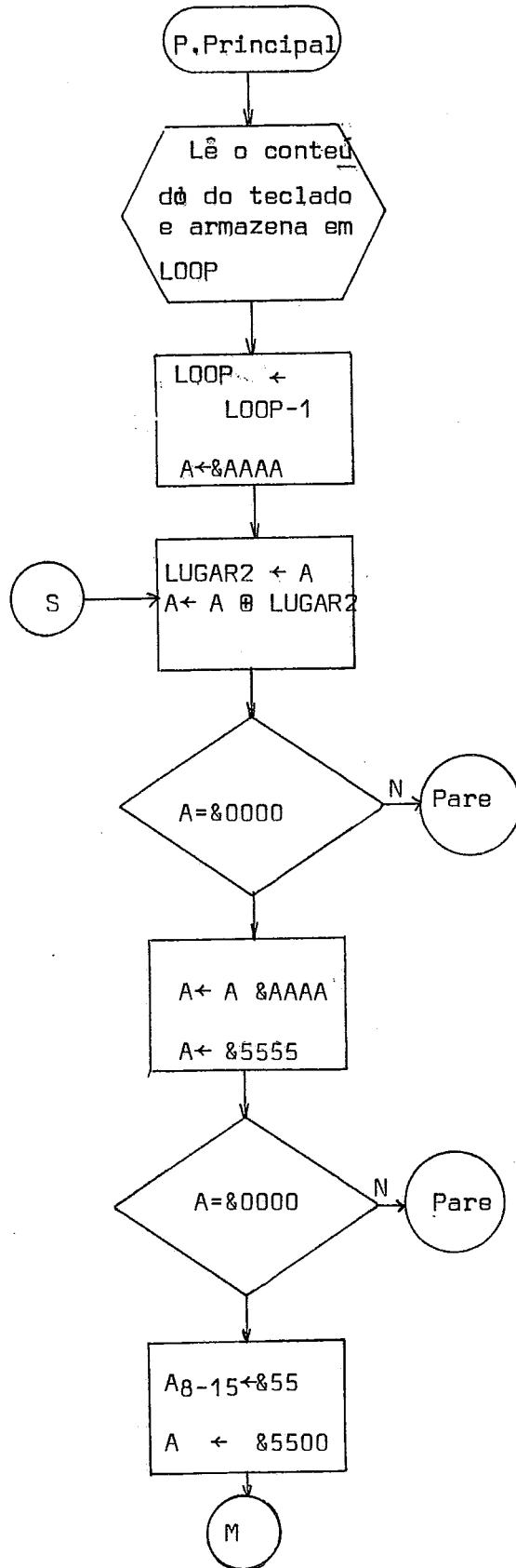
245 0066 0000          CLS  FI10
246 0068 0000          CLS  FI11
247 006A 0000          CLS  FI12
248 006C F502          ICL  =2
249 006E DF28          BRU  @#P1   *      VA PARA P2,USANDO ENDE
250                               *      IND. GERAL.
251 0070 C700          BRU  $+0   *      SE O BALTO NAO FOR EXE
252                               *      DO PARE!
253 0072 C700          BRU  $+0
254 0074 D700          P2    BRU  @P4   *      COM A BASE L INC.DE 2,0E
255                               *      TO VAI ATE P7.
256 0076 C700          BRU  $+0   *      SE O SALTO NAO FOR EXE
257                               *      DO PARE!
258 0078 C700          BRU  $+0   *      SE O SALTO NAO FOR EXE
259                               *      PARE!
260 007A F602          P7    DCL  =2           *      VOLTA BASE L A POSICAO
261                               *
262 007C 0004          LDA  LOOP
263 007E C502          BAZ  $+2
264 0080 CF3B          BRU  RP
265 0082 0000          CSV  M:EXIT
266 0084 CF42          BRU  P11
267                               FIN  P11
268                               END  FIC

```

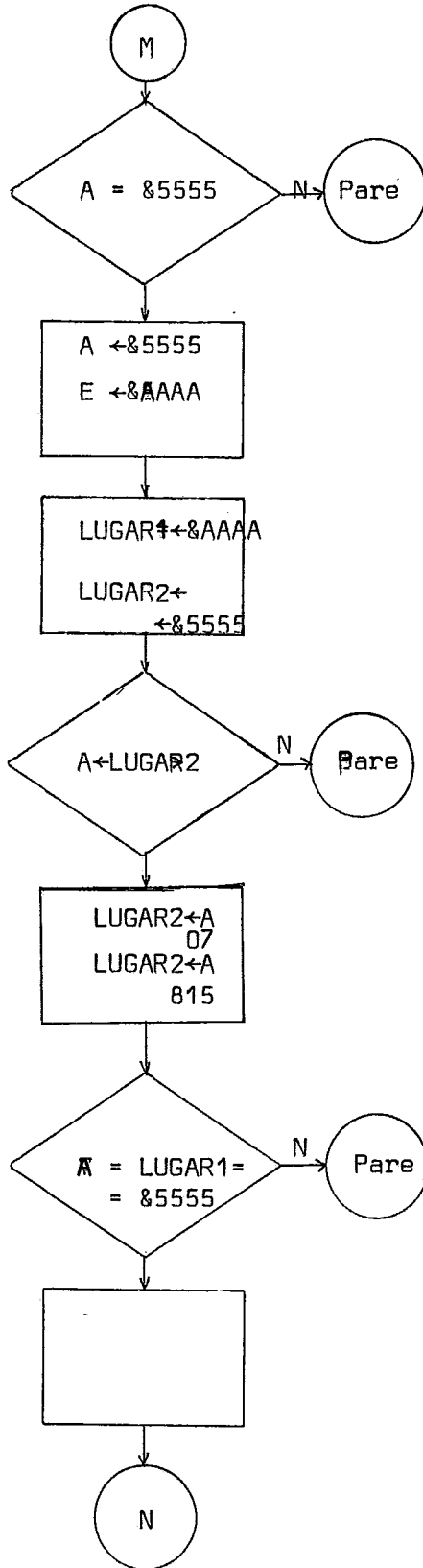
%ASS2/

Fluxograma do programa de teste

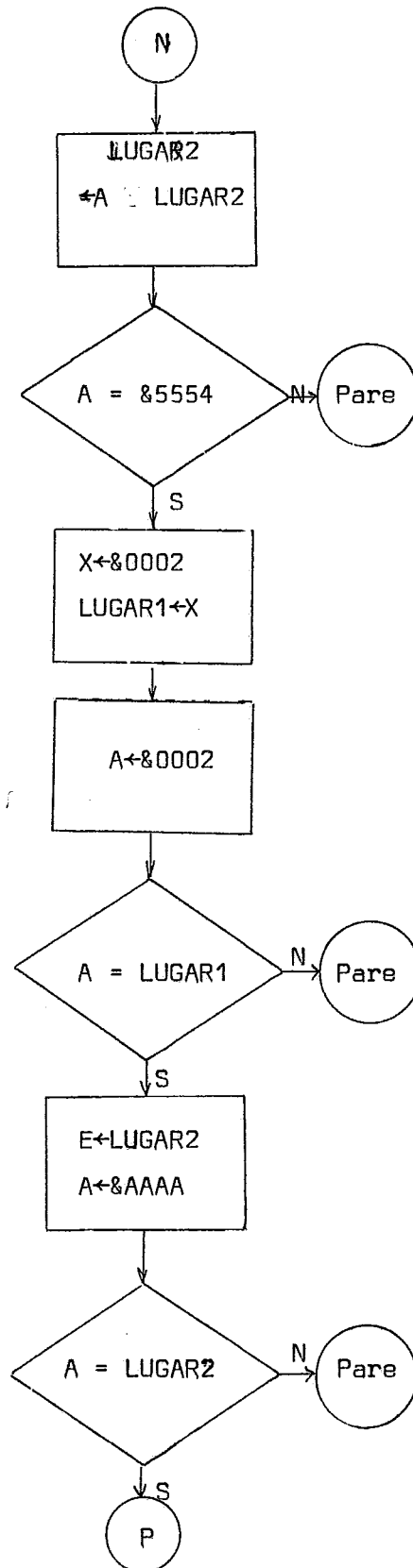
Programa Principal



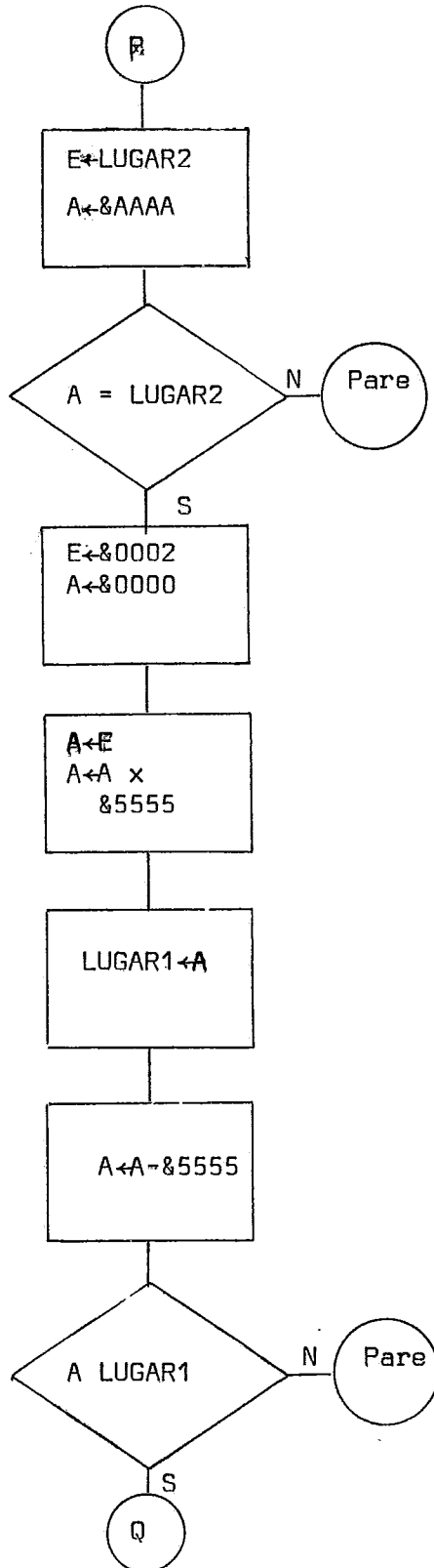
Simulador de Falhas

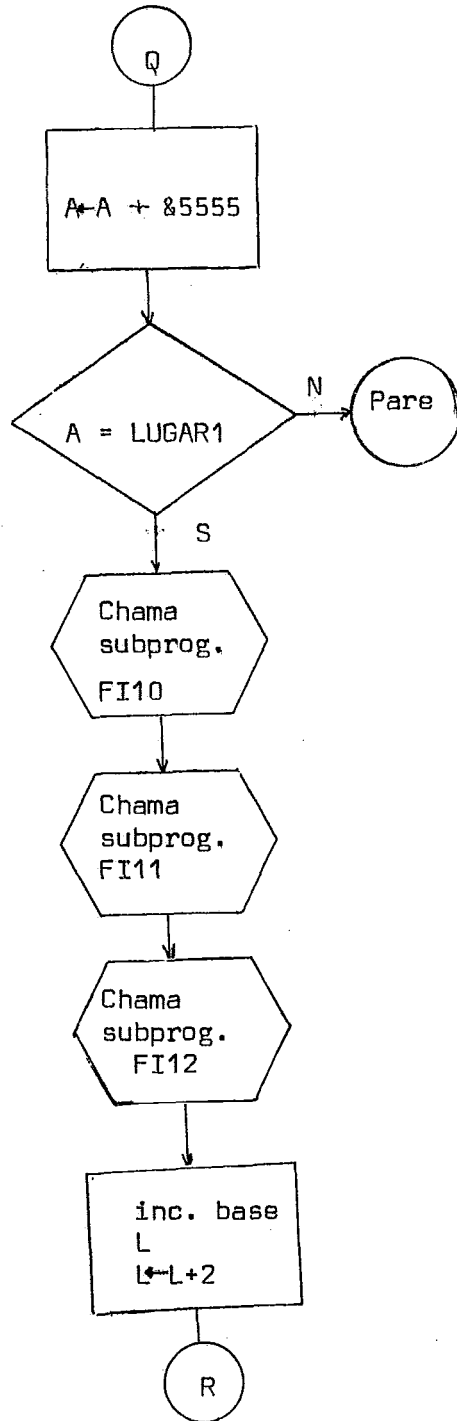


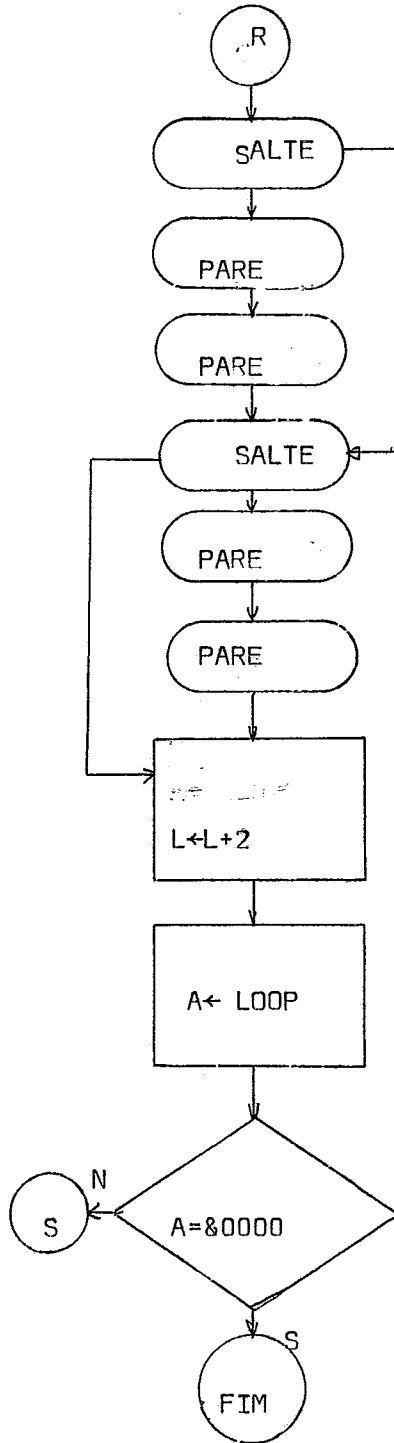
D-18



D-19



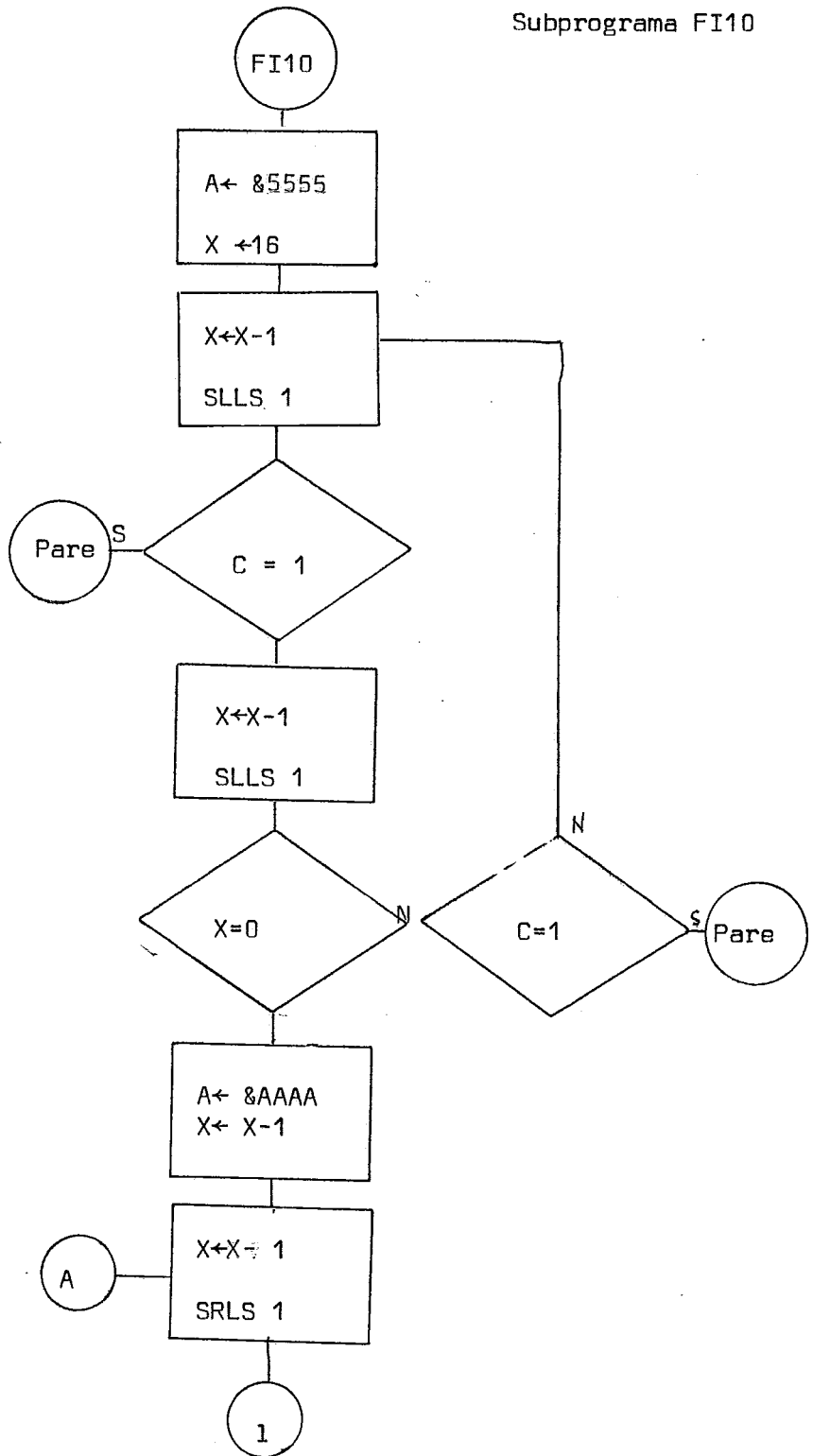




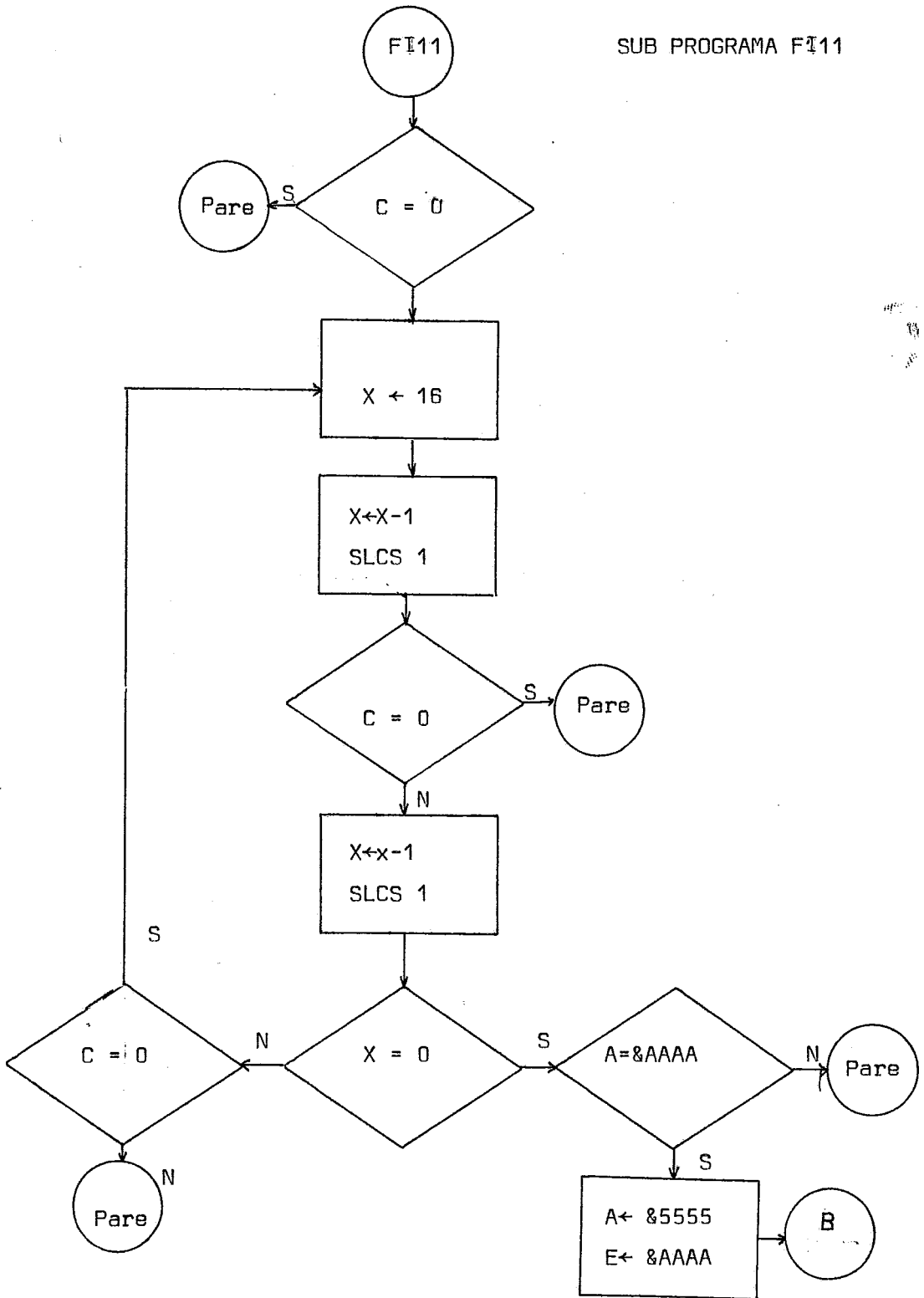
SIMULADOR DE

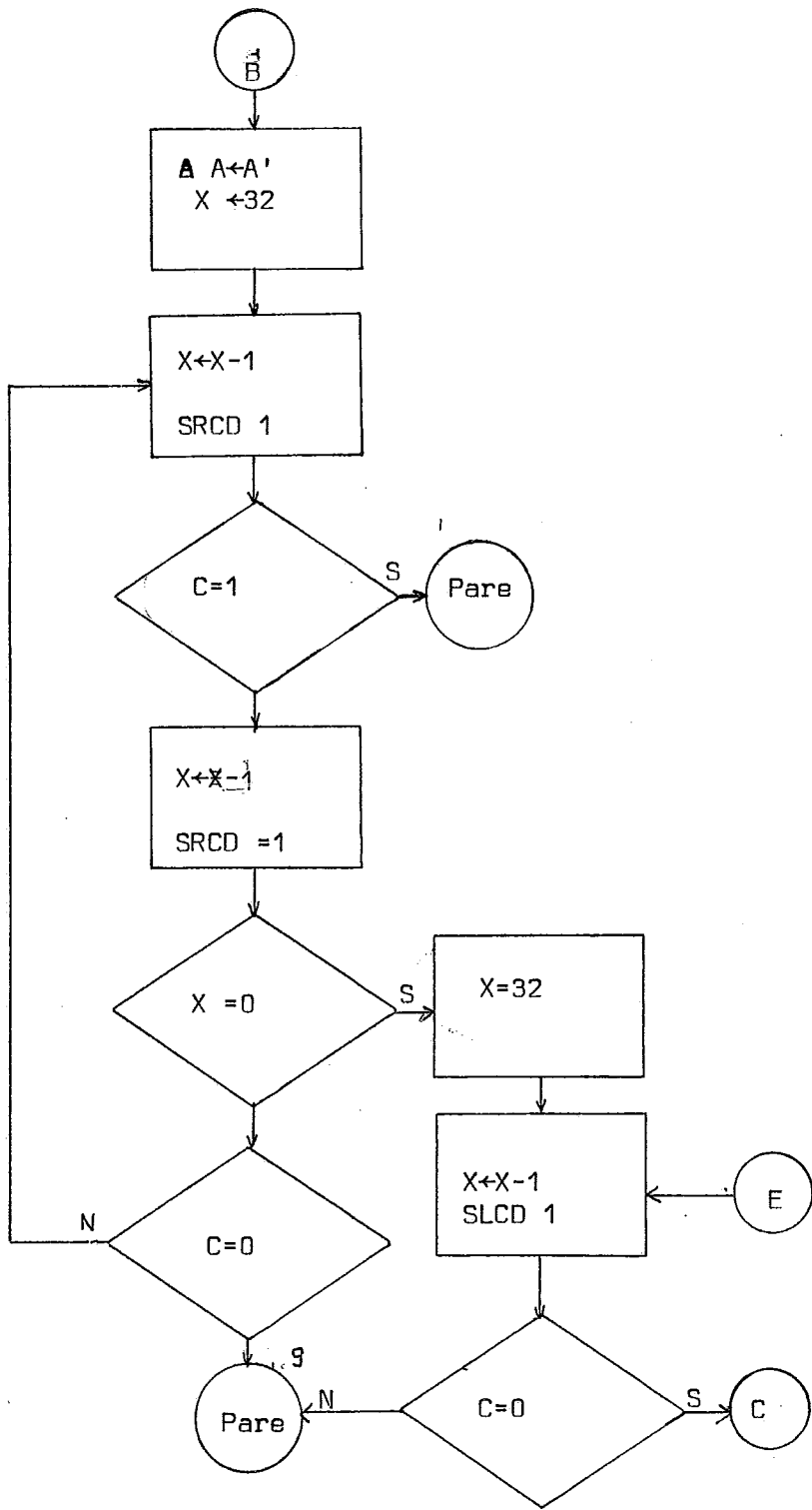
FALHAS

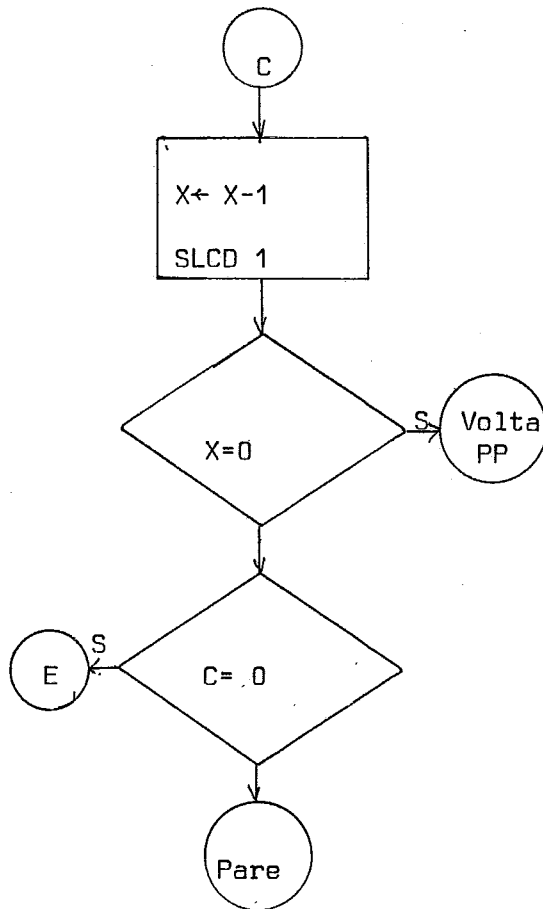
Subprograma FI10

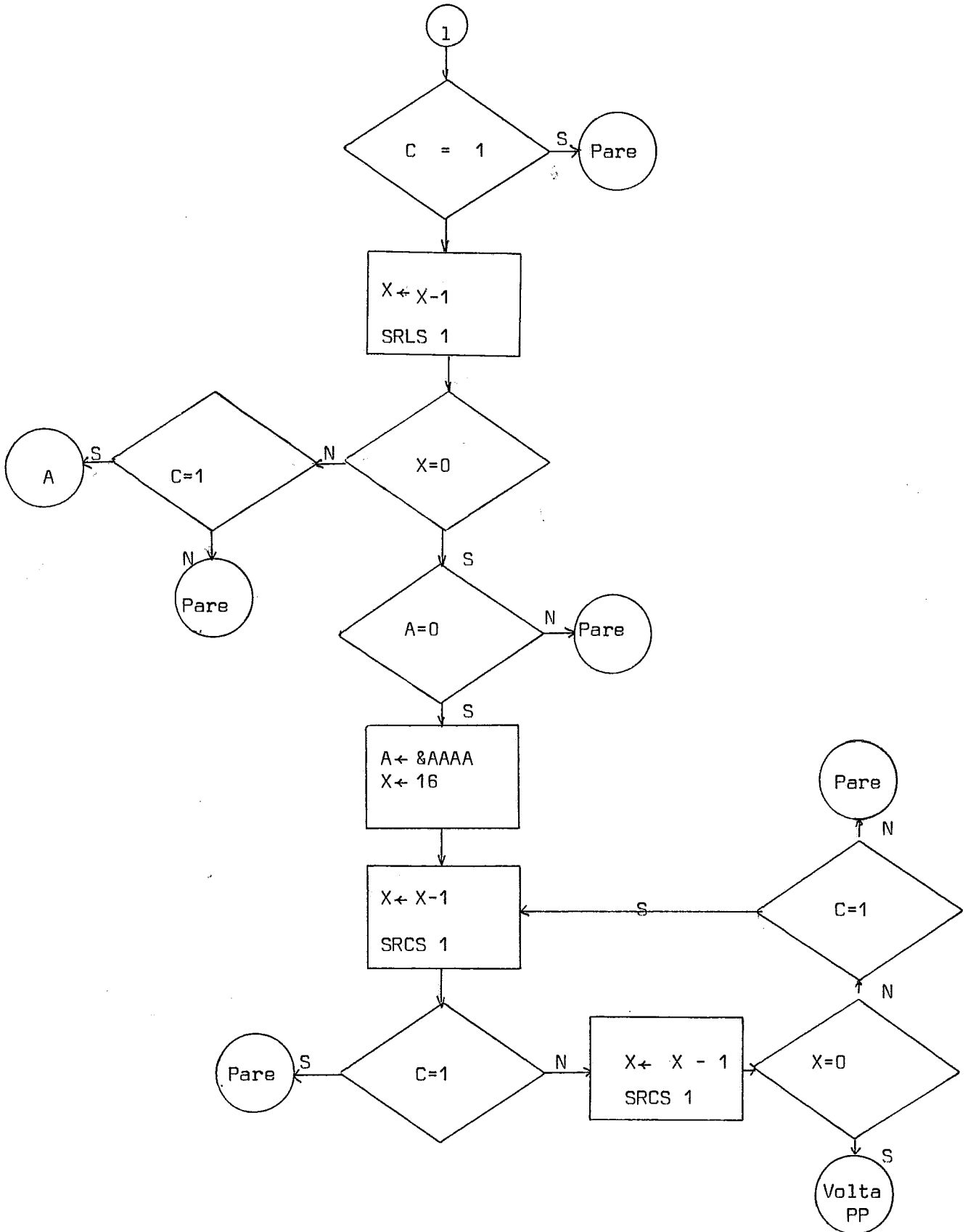


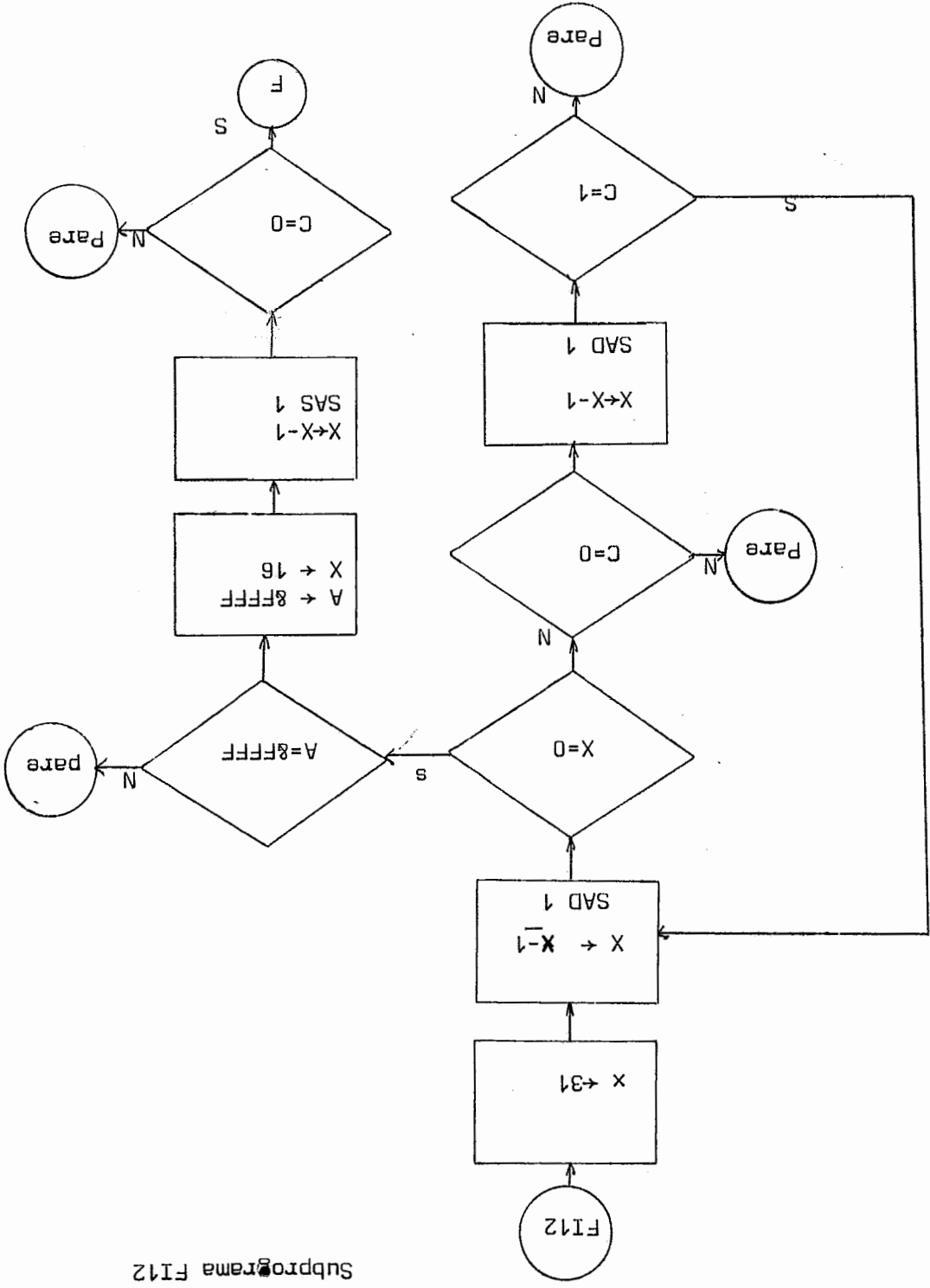
SUB PROGRAMA FI11



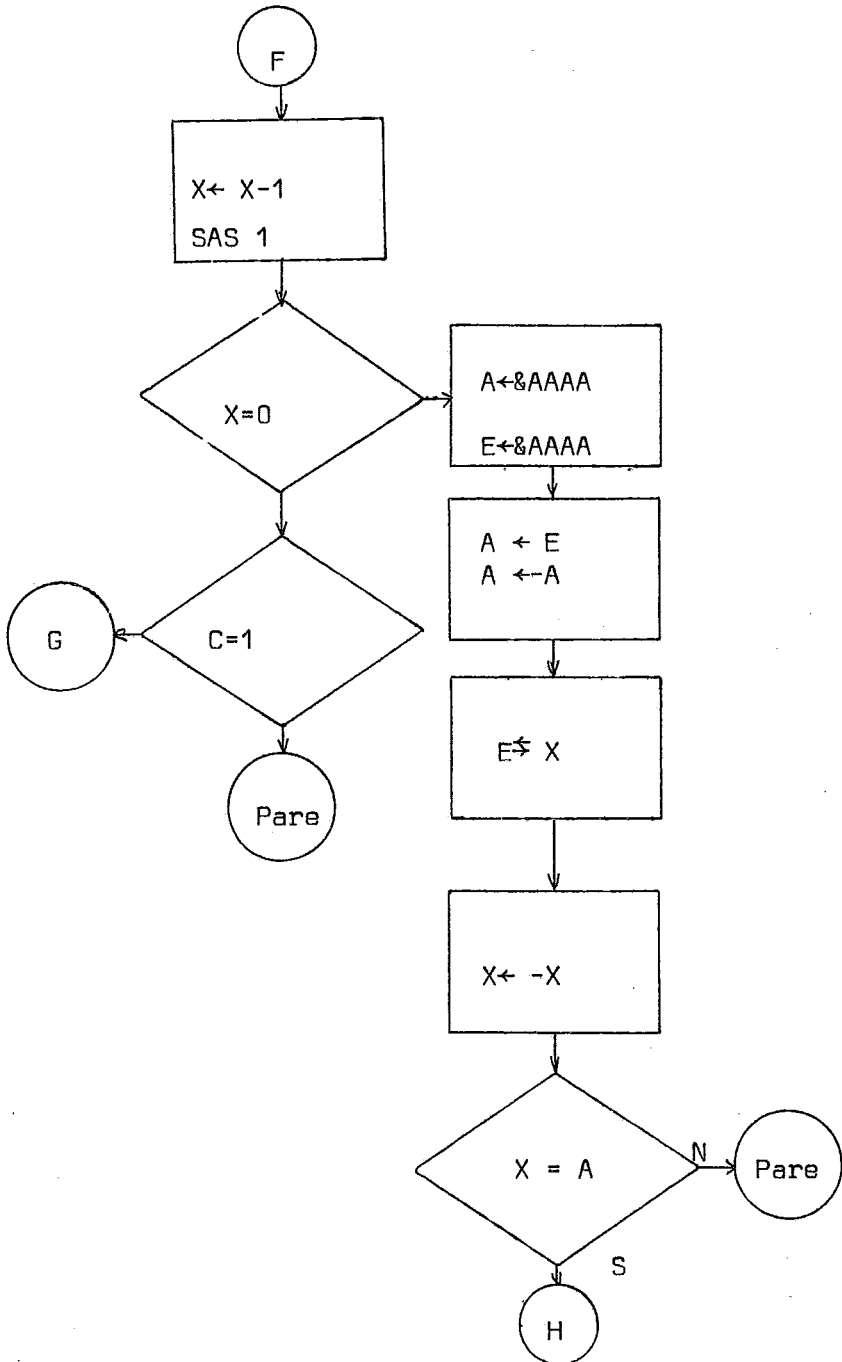


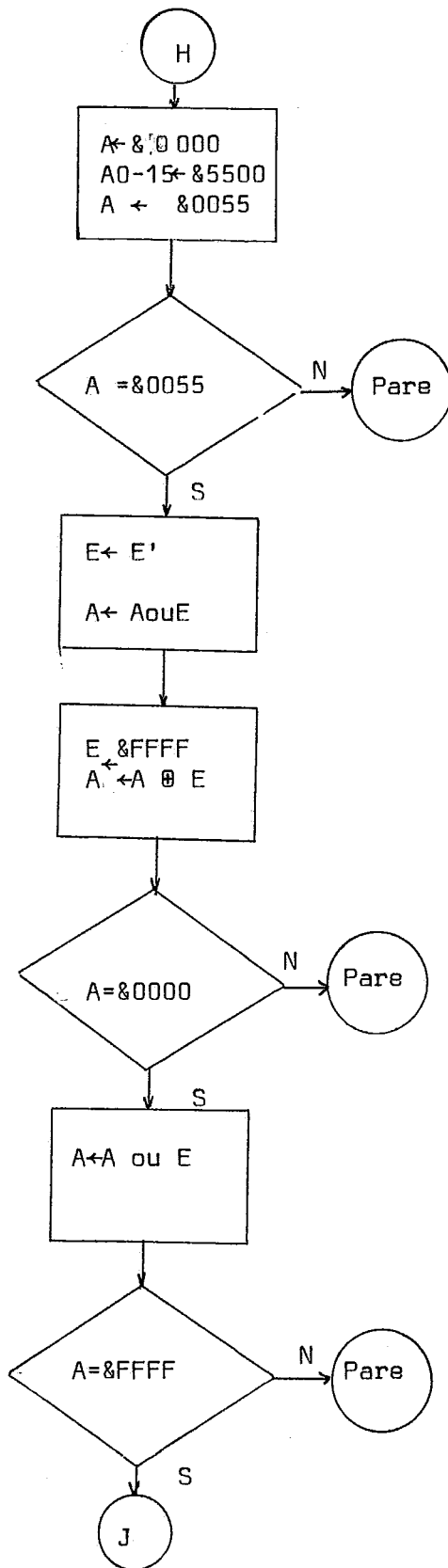




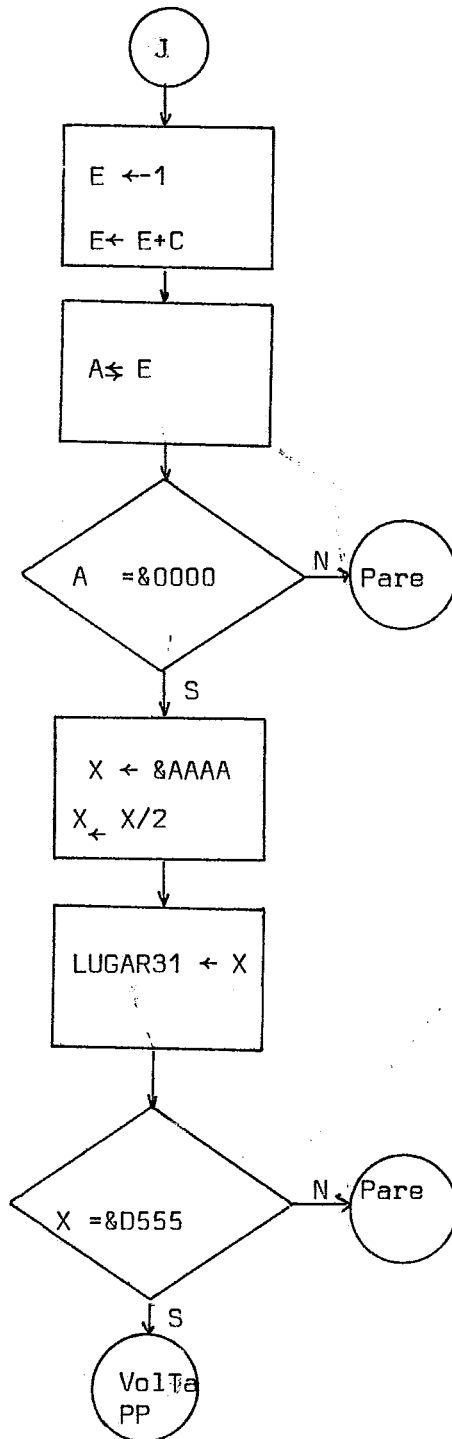


Subprograma FI12





D-30



APÊNDICE - C

REPERTÓRIO DE INSTRUÇÕES DO MITRA 15^I

Instrução	Código do P. Escolhe	Função	Tempos nos diversos modos de endereçamento em micro/seg.										
			DE	DG	P	IGX	PX	RP	RM	IL	IG	ILX	
LDA	315	A←Y	2,3	2,2	2,5	3,4	-	-	-	3,4	-	3,4	
LDE	317	E←Y	2,3	2,3	2,5	3,4	-	-	-	3,4	-	3,4	
LDX	319	X←Y	2,3	2,3	2,5	3,4	-	-	-	3,4	-	3,4	
LDR	283	R←Y	3,7	-	3,7	-	3,7	-	-	-	-	-	
LBL	383	A ₀₋₇ ←Y	2,6	2,6	3	3,6	-	-	-	3,6	-	3,6	
LBR	388	A ₈₋₁₅ ←Y	2,6	2,6	3	3,6	-	-	-	3,6	-	3,6	
LBX	393	X ₀₋₇ ←Y	2,6	2,6	3	3,6	-	-	-	3,6	-	3,6	
LEA	323	A←Y - (G)	2,7	2,7	3,1	3,7	3,7	-	-	3,7	-	3,7	
DLD	398	E←Y, A←Y+2	3,6	3,6	-	4,8	-	-	-	4,8	-	4,8	
SBL	409	Y←A ₀₋₇	2,5	2,5	-	3,5	-	-	-	3,5	-	3,5	
SBR	413	Y←A ₈₋₁₅	2,5	2,5	-	3,5	-	-	-	3,5	-	3,5	
STA	403	Y←A	2,2	2,2	-	3,2	-	-	-	3,2	-	3,2	
STE	405	Y←E	2,2	2,2	-	3,2	-	-	-	3,2	-	3,2	

SRCD VIII	\$600	Shift circ.p/dir. de A e E	DC	DG	P	IGX	PX	RP	RM	IL	IG	ILX
XAE	126	A ← E	-	-	4,3	-	-	-	-	-	-	-
XAX	127	A ← X	-	-	4,3	-	4,3	-	-	-	-	-
XAA	139	A ₀₋₇ ← A ₈₋₁₅	-	-	2,8	-	-	-	-	-	-	-
CCE	141	E ← E	-	-	2,8	-	-	-	-	-	-	-
ACE	155	E ← E + C	-	-	3,4	-	-	-	-	-	-	-
CCA	160	A ← A	-	-	2,8	-	-	-	-	-	-	-
AEE	171	A ← A ⊕ E	-	-	3,1	-	-	-	-	-	-	-
CNX		X ← -X	-	-	3,1	-	-	-	-	-	-	-
AIE	173	A ← A ∨ E	-	-	3,1	-	-	-	-	-	-	-
AAE	168	A ← A ∧ E	-	-	3,1	-	-	-	-	-	-	-
LNE	164	E ← -1	-	-	3,1	-	-	-	-	-	-	-
CNA	162	A ← -A	-	-	2,4	-	-	-	-	-	-	-
CHK	177	X ← X/2	-	-	3,1	-	-	-	-	-	-	-
BRU	44	Desvio incondicional	-	-	-	-	-	1,8	1,8	3	3	-
BRX	50	Desvio inc. Indx.	-	-	-	-	-	2,1	2,1	3,2	3,2	-
BCT IX	25	Desvia se C=1	-	-	-	-	-	2,1	2,1	3,3	3,3	-
BOT IX	53	Desvia se O=1	-	-	-	-	-	2,1	2,1	3,3	3,3	-
BCF IX	58	Desvia se C=0	-	-	-	-	-	2,1	2,1	3,3	3,3	-
BOF IX	33	Desvia se O=0	-	-	-	-	-	2,1	2,1	3,3	3,3	-
BAZ VIII	40	Desvia se A=0	-	-	-	-	-	2,4	2,4	3,5	3,5	-
BAN	36	Desvia se A 0	-	-	-	-	-	2,4	2,4	3,5	3,5	-
CLS VI	262	Chama subprograma	8,3	-	8,3	-	8,3	-	-	-	-	-
RTS VI	110	Retorna a programa	-	-	4,3	-	-	-	-	-	-	-
CSV	240	Chama supervisor	9,9	-	9,9	-	9,9	-	-	-	-	-

STX	407	$Y \leftarrow X$	2,2	2,2	-	3,2	-	-	-	3,2	-	3,2
STR	209	$Y \leftarrow \text{Reg.}$	4	-	4	-	4	-	-	3,2	-	-
SPA ^{II}	424	$Y \leftarrow P$	3,1	3,1	-	4,2	-	-	-	4,2	-	4,2
STS	431	$Y \leftarrow Y \wedge (\bar{E}) \vee (A) \wedge (E)$	3,4	3,4	-	4,4	-	-	-	4,4	-	4,4
DST	417	$Y \leftarrow E, Y+2, A$	3,6	3,6	-	4,7	-	-	-	4,7	-	4,7
ADD	327	$A \leftarrow (A) + Y$	2,3	2,3	2,5	3,4	-	-	-	3,4	-	3,4
ADM	421	$A e Y \leftarrow (A) + Y$	2,6	2,6	-	3,7	-	-	-	3,7	-	3,7
SUB	329	$A \leftarrow (A) - Y$	2,3	2,3	2,5	3,4	-	-	-	3,4	-	3,4
MUL ^{III}	354	$A \leftarrow (A) \times Y$	8,1	8,1	8,5	9,1	-	-	-	9,1	-	9,1
IDR	331	$A \leftarrow (A) \vee Y$	2,3	2,3	2,5	3,4	-	-	-	3,4	-	3,4
EOR	321	$A \leftarrow (A) \oplus Y$	2,3	2,3	2,5	3,4	-	-	-	3,4	-	3,4
AND	337	$A \leftarrow (A) \wedge Y$	2,3	2,3	2,5	3,4	-	-	-	3,4	-	3,4
CMP ^{IV}	341	compara (A) c/ (Y)	3,2	3,2	3,6	4,2	-	-	-	4,2	-	4,2
ICX	180	$X \leftarrow (X) + Y$	2,2	-	2,2	-	2,2	-	-	-	-	-
DCX	182	$X \leftarrow (X) - Y$	2,2	-	2,2	-	2,2	-	-	-	-	-
ICL	236	$L \leftarrow (L) + Y$	2,2	-	2,2	-	2,2	-	-	-	-	-
DCL	238	$L \leftarrow (L) - Y$	2,2	-	2,2	-	2,2	-	-	-	-	-
SILS ^V	76	Shift log.p/esq.simples	-	-	4,3	-	4,3	-	-	-	-	-
SRCS ^{VI}	78	Shift circ.P/dir.simples	-	-	4,3	-	4,3	-	-	-	-	-
SAD ^{VII}	81	Shift arit.p/dir.de e E	-	-	4,3	-	4,3	-	-	-	-	-
S.CD ^{VIII}	86	Shift circ.p/esq.de A eE	-	-	4,3	-	4,3	-	-	-	-	-
SILCS ^{VI}	91	Shift circ.p/esq.simples	-	-	4,3	-	4,3	-	-	-	-	-
SAS ^{VI}	94	Shift arit.p/dir.simples	-	-	4,3	-	4,3	-	-	-	-	-
SRLS ^{VI}	97	Shift log.dir.de A	-	-	4,3	-	4,3	-	-	-	-	-

RSV	143	Retorna ao supervisor	-	-	6,7	-	-	-	-	-	-	-	-
DIT	192	Desativa interrupção	-	-	32,5	-	-	-	-	-	-	-	-
DITR		Desativa IT rápida	-	-	6,1	-	6,1	-	-	-	-	-	-
RD	230	Leitura de acoplador	-	-	3,5	-	-	-	-	-	-	-	-
WD	333	Escreve no acoplador	-	-	3,5	-	-	-	-	-	-	-	-

I - micromáquina de 512 palavras.

II- os tempos indicados são para o modo Mestre (MS), para o modo escravo some 0,6 micro-segundos.

III- os tempos indicados são para multiplicação programada, para multiplicações executadas por micro-programas o tempo é 40ns.

IV- os tempos indicados é para comparação =, para comparação maior será:

4,1; 4,1;4,5; 5,1; -; -; 5,1 respectivamente, para comparação menor será 4,5;4,5;5,5,6;5,6;-.

V - acrescentar a este tempo+ 1,2n

VI- acrescentar a este tempo +1,5n

VII- acrescentar a este tempo +2,1n

VIII-acrescentar a este temp +2,7n

IX - este tempo é seu ocorrer desvio, caso contrário: 1,9; 1,9; 1,9; 1,9; 1,9.

X - este tempo é seu ocorrer desvio, caso contrário:2,2; 2,2; 2,2; 2,2; 2,2;

XI- este tempo é em modo escravo, MS será igual : 7,9; 7,9; 7,9;

XII- este tempo é em modo escravo, MS será igual: 4,7

C(31)= 0 C(32)= 0 C(33)= 0
 C(34)= 0 C(35)= 0 C(36)= 0
 C(37)= 0 C(38)= 0 C(39)= 0
 C(40)= 0 C(41)= 0 C(42)= 0
 C(43)= 0 C(44)= 0 C(45)= 0
 C(46)= 0 C(47)= 0 C(48)= 0
 C(49)= 0 C(50)= 0 C(51)= 0
 C(52)= 0 C(53)= 0 C(54)= 0
 C(55)= 0 C(56)= 0 C(57)= 0
 C(58)= 0 C(59)= 0 C(60)= 0
 C(61)= 0 C(62)= 0 C(63)= 0
 C(64)= 0 C(65)= 0 C(66)= 0

END= 2MARCA O NUM DE PESQ, FEITAS

LISTA DAS MACROS ESCOLHIDAS

G(1)= 25	TPC(1)=	290	G(2)= 33	TPO(2)=	1900
G(3)= 36	TPO(3)=	220	G(4)= 40	TPO(4)=	220
G(5)= 44	TPO(5)=	0	G(6)= 48	TPO(6)=	210
G(7)= 50	TPC(7)=	210	G(8)= 53	TPO(8)=	210
G(9)= 59	TPO(9)=	210	G(10)= 76	TPO(10)=	210
G(11)= 78	TPC(11)=	210	G(12)= 81	TPO(12)=	210
G(13)= 86	TPO(13)=	210	G(14)= 91	TPO(14)=	210
G(15)= 94	TPO(15)=	210	G(16)= 97	TPO(16)=	210
G(17)=100	TPC(17)=	210	G(18)=110	TPO(18)=	270
G(19)=120	TPO(19)=	210	G(20)=127	TPO(20)=	210
G(21)=134	TPO(21)=	210	G(22)=139	TPO(22)=	60
G(23)=141	TPO(23)=	60	G(24)=143	TPO(24)=	400
G(25)=155	TPO(25)=	60	G(26)=160	TPC(26)=	60
G(27)=164	TPC(27)=	60	G(28)=171	TPO(28)=	90
G(29)=173	TPO(29)=	90	G(30)=180	TPO(30)=	210

G(31)=182	TPO(31)=	210	G(32)=192	TPO(32)=	520
G(33)=230	TPC(33)=	350	G(34)=233	TPO(34)=	350
G(35)=236	TPO(35)=	210	G(36)=238	TPO(36)=	210
G(37)=240	TPC(37)=	370	G(38)=262	TPO(38)=	820
G(39)=283	TPC(39)=	360	G(40)=290	TPO(40)=	420
G(41)=315	TPO(41)=	210	G(42)=317	TPC(42)=	210
G(43)=319	TPO(43)=	210	G(44)=321	TPO(44)=	210
G(45)=323	TPO(45)=	210	G(46)=327	TPC(46)=	210
G(47)=329	TPO(47)=	210	G(48)=331	TPO(48)=	210
G(49)=337	TPC(49)=	210	G(50)=341	TPO(50)=	110
G(51)=354	TPO(51)=	900	G(52)=383	TPO(52)=	260
G(53)=388	TPC(53)=	260	G(54)=393	TPO(54)=	260
G(55)=398	TPC(55)=	340	G(56)=403	TPC(56)=	220
G(57)=405	TPO(57)=	220	G(58)=407	TPC(58)=	220
G(59)=409	TPC(59)=	250	G(60)=413	TPO(60)=	250
G(61)=417	TPO(61)=	350	G(62)=421	TPO(62)=	240
G(63)=424	TPO(63)=	280	G(64)=431	TPO(64)=	340
G(65)=162	TPC(65)=	90	G(

* A SOMA DOS TEMPOS DAS MACROS ESCOLHIDAS = 17940


```

C
C   PROGRAMA PRINCIPAL
C   FUNCAO :COLOCAR UM CONJUNTO DE ELEMENTOS
C           NA FORMA DE MATRIZ BINARIA,QUE E' A
C           FORMA NECESSARIA PARA O TRABALHO DA SUBROUTINA ESCOLHE.
C
C   M=MATRIZ MACRO X MICRO LIDA
C   NM=MATRIZ MACRO X MICRO NA FORMA BINARIA
C   R=VETOR QUE FAZ A REALOCAAO DOS ELEMENTOS DA MATRIZ MACRO/MICRO
C   TPO=GUARDA O TEMPO DE CADA MACRO
C   L=VETOR QUE APONTA AS LINHAS DA MATRIZ MACRO X MICRO
C
0001   INTEGER TP
0002   INTEGER U,C1,A,CC,Q,GG,G2,C
0003   INTEGER M(100,500)
0004   INTEGER R(1000)
0005   INTEGER*2 TPO(100),L(100),NNMI(100)
0006   LOGICAL*1 NM(100,500)
0007   COMMON NM/AREAL/R,NNMI,M
0008   DATA Q,U,C,CC/0,0,0,0/
0009   DATA GG,C1,K,LL/0,0,0,0/
0010   M(1,1)=0
C
C   LEITURA DO NUM DE MICROS EM CADA MICROPROGRAMA
C
0011   2   READ(5,1,END=901)MA,NMI,TP
0012   WRITE(6,1991)MA,NMI,TP
0013   1   FORMAT(2I3,I4)
0014   1991  FORMAT(2I6,I8)
0015   LL=LL+1
0016   TPO(LL)=TP
0017   L(LL)=MA
0018   NNMI(LL)=NMI
0019   IF(GG.GE.NMI)GOTO 3331
0020   701  GG=NMI
C
C   LEITURA DAS MICROS SENDO UMA EM CADA CARTAO
C
0021   3331 READ(5,4)((IMU,M(LL,C1)),C1=1,NMI)
0022   4   FORMAT(7(I6,Z4))
0023   PRINT 31
0024   31  FORMAT(' MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR')
0025   WRITE(6,63)(M(LL,C1),C1=1,NMI)
0026   63  FORMAT(20(2X,Z4),/)
C
C   VETOR DE REALOCAAO N(K)
C
0027   3 DO 9 IP=1,NMI
0028   K=K+1
0029   R(K)=M(LL,IP)
0030   9 CONTINUE
0031   CALL ORDEN(R,K)
0032   CALL REDUN(R,K)
0033   GOTO 2
0034   901 NMI=GG
0035   MA=LL
0036   PRINT 32
0037   32  FORMAT(' VETOR REALOCAAO')

```

```
0038          WRITE(6,63)(R(J),J=1,K)
0039          DO 19 LI=1,LL
0040          DO 19 LJ=1,K
0041          NM(LI,LJ)=.FALSE.
0042          19  CONTINUE
0043          DO 20 J=1,LL
0044          IF(NNMI(J).GE.NMI)GOTO 20
0045          702 NN=NNMI(J)+1
0046          DO 21 JJ=NN,NMI
0047          M(J,JJ)=0
0048          21  CONTINUE
0049          20  CONTINUE
0050          PRINT 33
0051          33  FORMAT('          FORMACAO DA MATRIZ MACRO X MICROINSTRUCOES')
0052          WRITE(6,63)((M(L1,C1),L1=1,LL),C1=1,NMI)
          C
          C  PREPARA O CONJUNTO PARA O FORMATO
          C  QUE DARA ENTRADA NA SUBROTINA ESCOLHA
          C
0053          681 DO 11 I=1,MA
0054          DO 11 J=1,NMI
0055          DO 12 K1=1,K
0056          IF(M(I,J).EQ.R(K1))GOTO 13
0057          706 GOTO 12
0058          13  NM(I,K1)=.TRUE.
0059          GOTO 11
0060          12  CONTINUE
0061          11  CONTINUE
0062          WRITE(6,14)
0063          14  FORMAT(///,9X,'MATRIZ MACRO X MICRO NO FORMATO BINARIO',//)
0064          WRITE(6,60)((NM(I,K1),I=1,MA),K1=1,K),R(K1)
0065          60  FORMAT(2X,82I1,5X,Z4)
0066          CALL SCOLHE(MA,TPO,L,K)
0067          CALL EXIT
0068          END
```

```

0001      SUBROUTINE SCOLHE(MA,TPO,L,MI)
0002      INTEGER T,U,S,CL,Y,YJ,BJ,Q,O,B,01,END,A
0003      INTEGER EU(100)
0004      INTEGER*2 L(100),C(100),G(100),TPO(100),TPI(100)
0005      LOGICAL*1 N(100,500),Z
0006      COMMON N/AREA1/G,C,EU,TPI

C
C      SUBROUTINA ESCOLHE;
C      FUNCAO:ESCOLHER UM CONJUNTO MINIMO DE ELEMENTOS
C      CRITERIOS:
C          1. ELEMENTO ESCENCIAL
C          2. ELEMENTO DE MAIOR REPRESENTATIVIDADE
C          3. ELEMENTO MAIS RAPIDO
C
C          C=VETOR QUE PESQUISA O NUMERO DE MICROS REPRESENTADAS POR UMA MACRO
C          EU=VETOR QUE APONTA AS LINHAS ESCENCIAIS
C          G=GUARDA AS MACROS ESCOLHIDAS
C          TPI=GUARDA O TEMPO DAS MACROS ESCOLHIDAS
C          N=MATRIZ BINARIA MACRO VERSUS MICRO
C          END=PARAMETRO QUE MARCA O NUMERO DE PESQUISAS FEITAS NA MATRIZ BINARIA

0007      I=0
0008      END=0

C
C      PESQUISA DAS MACROS ESSENCIAIS
C
0009      DO 3 T=1,MI
0010      A=0
0011      DO 2 U=1,MA
0012      IF(.NOT.N(U,T))GOTO 2
0013      222  A=A+1
0014      A2=A-2
0015      IF(A2)444,4,4
0016      444  I=I+1

C
C      GUARDA A LINHA ESSENCIAL
C
0017      EU(I)=U
0018      GOTO 2
0019      4  EU(I)=0
0020      I=I-1
0021      GOTO 3
0022      2  CONTINUE
0023      3  CONTINUE
0024      IF(I.LE.1)GOTO 331
0025      WRITE(6,63)(II,EU(II),II=1,I)
0026      CALL ORDEN(EU,I)
0027      CALL REDUN(EU,I)
0028      DO 701 IK=1,I
0029      G(IK)=L(EU(IK))
0030      TPI(IK)=TPO(EU(IK))
0031      701  CONTINUE
0032      3331 WRITE(6,62)
0033      62  FORMAT(4X,'AS LINHAS ESSENCIAIS ENCONTRADAS FORAM')
0034      WRITE(6,63)(II,EU(II),II=1,I)
0035      63  FORMAT(10X,'EU(',I3,')=',I10,2X,'EU(',I3,')=',I10,2X,'EU(',I3,')=',
*,I10,2X,'EU(',I3,')=',I10,2X,'EU(',I3,')=',I10,2X,'EU(',I3,')=',I1
*0)

C

```

```

C      TESTA SE A MACRO E ESSENCIAL E ELIMINA SEUS PONTOS DE INFLUENCIA
C
0036  331  DO 26 K=1,MA
0037      IF(EU(K))26,26,666
0038  666  DO 6 CL=1,MI
0039      IF(.NOT.N(EU(K),CL))GOTO 6
0040      DO 25 J=1,MA
0041      N(J,CL)=.FALSE.
0042  25   CONTINUE
0043      6  CONTINUE
0044  26   CONTINUE
0045      END=END+1

C
C      ORDENA AS MELAORES
C
0046      PRINT 34
0047  34   FORMAT('          VETOR QUE GUARDA O NUM DE MICROS',/, '          AINDA N
*AO REPRESENTADAS')
0048      DO 9 J=1,MA
0049      C(J)=0
0050      DO 8 Y=1,MI
0051      IF(.NOT.N(J,Y))GOTO 8
0052  111  C(J)=C(J)+1
0053      8  CONTINUE
0054  70   FORMAT(4X,'C(',I3,')=',I3,2X,'C(',I2,')=',I3,2X,'C(',I2,')=',I3)
0055  9    CONTINUE
0056      WRITE(6,70)(J,C(J),J=1,MA)
0057      MX=MA-1
0058      DO 21 J=1,MX
0059      YJ=J+1
0060      DO 10 Y=YJ,MA
0061      IF(C(Y).GE.C(J))GOTO 10
0062      A=C(J)
0063      C(J)=C(Y)
0064      C(Y)=A
0065      A=L(Y)
0066      L(Y)=L(J)
0067      L(J)=A
0068      A=TPO(Y)
0069      TPO(Y)=TPO(J)
0070      TPO(J)=A
0071      DO 22 IA=1,MI
0072      Z=N(J,IA)
0073      N(J,IA)=N(Y,IA)
0074      N(Y,IA)=Z
0075  22   CONTINUE
0076      10 CONTINUE
0077  21   CONTINUE
0078      IF(C(MA).EQ.0)GOTO 333
0079      B=0
0080      DO 12 J=1,MA
0081      IF(C(MA).NE.C(MA-J))GOTO 19
0082      B=J
0083      12 CONTINUE
0084  19   IF(B.EQ.0)GOTO 108
0085  1080 MAB=MA-B
0086      MA1=MA-1
0087      DO 23 O=MAB,MA1

```

```

0088      O1=O+1
0089      DO 13 K=O1,MA
0090      IF(TPO(K).LE.TPO(O))GOTO 13
0091      A=TPO(O)
0092      TPO(O)=TPO(K)
0093      TPO(K)=A
0094      A=C(O)
0095      C(O)=C(K)
0096      C(K)=A
0097      A=L(O)
0098      L(O)=L(K)
0099      L(K)=A
0100      DO 24 IA=1,MI
0101      Z=N(O,IA)
0102      N(O,IA)=N(K,IA)
0103      N(K,IA)=Z
0104      24 CONTINUE
0105      13 CONTINUE
0106      23 CONTINUE
0107      108 I=I+1
0108      DO 71 J=1,MA
0109      EU(J)=0
0110      71 CONTINUE
0111      TPI(I)=TPO(MA)
0112      G(I)=L(MA)
0113      EU(1)=MA
0114      IF(END.LT.MA)GOTO 331
0115      333 WRITE(6,200)END
0116      200 FORMAT(4X,'END=',I3,'MARCA O NUM DE PESQ.FEITAS')
0117      WRITE(6,15)
0118      15 FORMAT(4X,'LISTA DAS MACROS ESCOLHIDAS')
0119      WRITE(6,20)(Q,G(Q),Q,TPI(Q),Q=1,I)
0120      20 FORMAT(10X,'G(',I3,')=',I3,2X,'TPO(',I3,')=',I9,5X,'G(',I3,')=',I3
*,2X,'TPO(',I3,')=',I9)
0121      ITPI=0
0122      DO 39 JK=1,I
0123      ITPI=ITPI+TPI(JK)
0124      39 CONTINUE
0125      WRITE(6,89)ITPI
0126      89 FORMAT(//,2X,'+ A SOMA DOS TEMPOS DAS MACROS ESCOLHIDAS =',I8)
0127      RETURN
0128      END

```

```
*OPTIONS IN EFFECT* ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
*OPTIONS IN EFFECT* NAME = SCOLHE , LINECNT = 60
*STATISTICS* SOURCE STATEMENTS = 128,PROGRAM SIZE = 3608
*STATISTICS* NO DIAGNOSTICS GENERATED
```

```
0001          SUBROUTINE ORDEN(R,K)
              C
              C      SUBROUTINA ORDEN;
              C      FUNCAO:  ORDENAR UM CONJUNTO DE VALORES
              C
0002          INTEGER R(1000)
0003          INTEGER A
0004          INTEGER GG,G2
0005          GG=K-1
0006          DO 9 I=1,GG
0007             G2=I+1
0008             DO 8 J=G2,K
0009                IF(R(I).LE.R(J))GOTO 8
0010          703  A=R(I)
0011             R(I)=R(J)
0012             R(J)=A
0013          8    CONTINUE
0014          9    CONTINUE
0015          RETURN
0016          END
```

```
*OPTIONS IN EFFECT* ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
*OPTIONS IN EFFECT* NAME = ORDEN , LINECNT = 60
*STATISTICS* SOURCE STATEMENTS = 16,PROGRAM SIZE = 492
*STATISTICS* NO DIAGNOSTICS GENERATED
```

```
0001          SUBROUTINE REDUN(R,K)
              C
              C      SUBROUTINA REDUN;
              C      FUNCAO:TIRAR OS TERMOS REDUNDANTES DE UM CONJUNTO
              C          DE VALORES ORDENADOS.
              C
0002          INTEGER R(1000)
              C
              C      TIRA OS TERMOS REDUNDANTES
              C
0003          1      KG=K-1
0004          DO 5 I=1,KG
0005          J=I+1
0006          IF(R(I).NE.R(J))GOTO 5
0007          DO 21 KI=J,K
0008          R(KI-1)=R(KI)
0009          21      CONTINUE
0010          R(K)=0
0011          K=K-1
0012          GOTO 1
0013          5      CONTINUE
0014          RETURN
0015          END
```

C8C1 B4C5 B2C5 C505 C387 C3A4 D19F D19F
 388 8 260
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C8E6 BEE6 BEC9 C51D C387 C3A4 D19F D19F
 393 8 260
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C8E6 BF66 BF4B C51F C387 C3A4 D19F D19F
 398 8 340
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 EDEA AF2B 57C3 ACC3 C387 C3A4 D19F D19F
 403 5 220
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 30C2 C387 C3A4 D19F D19F
 405 5 220
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 3104 C387 C3A4 D19F D19F
 407 5 220
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 3146 C387 C3A4 D19F D19F
 409 7 250
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C8CF 3CD1 3CD7 C387 C3A4 D19F D19F
 413 7 250
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C8CD 3CDB 3CDD C387 C3A4 D19F D19F
 417 7 350
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 EDE7 3102 57E5 D19F C387 C3A4 D19F
 421 6 240
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C6A7 1CCE C387 C3A4 D19F D19F
 424 10 280
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 C2A9 CC04 CB23 1A12 DE94 1E14 C387 C3A4 D19F D19F
 431 10 340
 MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR
 FB09 ED99 FOC1 FBOF 2783 82B7 C387 C3A4 D19F D19F
 VETOR REALOCAAO
 00D1 041B 05C0 05C4 1A12 1CCE 1E11 1E14 1E18 1E59 1E5D 2783 300F 305D 30C2 3102 3104 3146 3CD1 3CD7
 3CDB 3CDD 4282 42A1 4C0A 4C0C 5649 56A0 57A2 57C3 57C6 57CC 57E5 57EB 57EC 59CA 59D0 59E2 5A5B 5A67
 5A9B 5AAE 5BD5 5FE0 6822 6C0E 6DC9 7022 704C 7081 709E 752C 781A 781D 79F2 80F1 8191 825A 8264 82B7
 860C 86DB 86E1 891D 9A8F 9A97 9AAA 9AAB 9AAD 9CCA 9D45 A4D2 A6CE AC0D AC18 AC49 AC55 ACC3 ADA0 ADA8
 ADC0 ADC4 ADD0 ADD3 ADDC ADE5 ADEA AEC0 AF02 AF2B AF44 B08B B140 B2C5 B3A8 B3AA B4C5 B84F B851 B8D4
 B957 BAC6 BEC9 BEE6 BF4B BF66 C080 C121 C191 C1C0 C24A C281 C285 C287 C289 C28B C28D C2A7 C2A9 C31F
 C35B C387 C3A4 C3CC C402 C404 C414 C461 C4C5 C505 C50F C51D C51F C54C C560 C58C C59F C618 C61C C6A7
 C708 C80B C82F C84D C84E C85E C866 C868 C89E C8C1 C8CD C8CF C8E6 C904 C908 C911 C94E C9A0 C9BE C9CB
 CA44 CA47 CA59 CB19 CB23 CB4E CB8E CBDA CC02 CC04 CC05 CC10 CC14 CC1F CD88 D183 D187 D18F D190 D198
 D19E D19F D3B8 D80B DA9D DB4E DB67 DB88 DCCC DD1F DE94 DE9C DEDA DF0D DF0E E34B E388 E38D E3C0 E3E9
 E5C8 E8C5 E8CF E8D8 E8EA E91B E91D EAE8 EB05 EC24 EC2A EC4F EC56 ECD3 ECD5 ECE1 ED15 ED16 ED57 ED89

25	9	290																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
C9BE	C281	DB4E C94E C285 CB8E C287 D190 D187																	
33	4	1900																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
CB4E	C28D	D190 D187																	
36	5	220																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FAF2	C85E	C289 D190 D187																	
40	5	220																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FAF2	C89E	C28B D190 D187																	
44	5	0																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
D198	C1C0	D198 D190 D187																	
48	3	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
781A	D190	D187																	
50	4	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
C1C0	781D	D190 D187																	
53	6	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
5A5B	891D	DA9D 6822 00D1 00D1																	
59	3	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
5A9B	D190	D187																	
76	18	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FADD	DB67	C2A7 5A67 D18F ADC0 F3A8 CC1F E5C8	C285	E3E9	C866	C4C5	D3B8	DB67	C2A7	5A67	D18F								
78	19	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FOC6	FCCD	DB67 C2A7 5A67 D18F ADC0 F3A8 CC1F	E5C8	C285	E3E9	C866	C4C5	D3B8	DB67	C2A7	5A67	D18F							
81	21	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
F125	FD05	FOED E8EA DB67 C2A7 5A67 D18F ADC0	F3A8	CC1F	E5C8	C285	E3E9	C866	C4C5	D3B8	DB67	C2A7	5A67						
D18F																			
86	21	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
F125	C31F	EAE8 EB05 DB67 C2A7 5A67 D18F ADC0	F3A8	CC1F	E5C3	C285	E3E9	C866	C4C5	D3B8	DB67	C2A7	5A67						
D18F																			
91	19	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FCC1	E3E9	DB67 C2A7 5A67 D18F ADC0 F3A8 CC1F	E5C8	C285	E3E9	C866	C4C5	D3B8	DB67	C2A7	5A67	D18F							
94	19	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FOC4	FCC5	DB67 C2A7 5A67 D18F ADC0 F3A8 CC1F	E5C8	C285	E3E9	C866	C4C5	D3B8	DB67	C2A7	5A67	D18F							
97	19	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FOC5	FCEA	DB67 C2A7 5A67 D18F ADC0 F3A8 CC1F	E5C8	C285	E3E9	C866	C4C5	D3B8	DB67	C2A7	5A67	D18F							
100	23	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FODF	C35B	F125 E91D FOE8 E8C5 DB67 C2A7 5A67	D18F	ADC0	F3A8	CC1F	E5C8	C285	E3E9	C866	C4C5	D3B8	DB67						
C2A7 5A67 D18F																			
110	19	270																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			

704C	C908	AC18	5649	AC49	9AAB	EC24	5649	9AAA	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67	D18F	
120	16	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FOC8	EDA3	ECD5	F109	F1A4	ED15	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67	D18F				
127	16	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FOD0	EDA3	F156	ECE1	F197	ED57	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67	D18F				
134	14	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
F11D	EDA1	F158	ED16	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67	D18F						
139	11	60																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FECE	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67	D18F									
141	11	60																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FB16	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67	D18F									
143	21	400																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
CB19	C191	CC04	5AAE	86DB	7081	9AAB	EC2A	56A0	9A8F	EC4F	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67
D18F																			
155	14	60																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
C9A0	C560	C59F	DD1F	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67	D18F						
160	11	60																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FAD6	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67	D18F									
162	11	90																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
F124	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67	D18F									
164	12	60																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
E34B	FB57	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67	D18F								
168	12	90																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
F124	F124	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67	D18F								
171	11	90																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FB3A	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67	D18F									
173	13	90																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
FAD5	C58C	DCCC	DB67	C2A7	5A67	D18F	B3AA	D19E	DB67	C2A7	5A67	D18F							
180	5	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
9D45	DB67	C2A7	5A67	D18F															
182	5	210																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
B957	DB67	C2A7	5A67	D18F															
192	45	520																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
C121	D80B	4C0A	ADC0	C708	C404	5FE0	80F1	C708	C402	5BD5	ADE5	42A1	05C0	CC05	ED89	59D0	FF91	041B	E388
G868	4C0A	ADD3	C708	C080	C618	ADDC	4282	CC05	ED8F	86E1	59CA	FF93	860C	E38D	CA47	7022	DB67	C2A7	5A67
D18F B3A8 C911 C191 D183																			
230	10	350																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			
F901	ECD5	DB67	C2A7	5A67	D18F	B3A8	C911	C191	D183										
233	10	350																	
MICROS QUE COMPOE O MICROPROGRAMA DA MACRO ANTERIOR																			