MODELO PARA ELABORAÇÃO DE TESTES DE UNIDADES COMPUTACIONAIS

Maria Alice Silveira de Brito Mochel

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 CIÊNCIAS (M.Sc.) EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Prof. Estevam Gilberto de Simone
(Presidente)

Profa. Sueli Mendes dos Santos

Stephan Kovach

Manoel Lage Pinheiro da Silva

RIO DE JANEIRO, RJ - BRASIL
DEZEMBRO DE 1979

MOCHEL, MARIA ALICE SILVEIRA DE BRITO

Modelo Para Elaboração de Testes de Unidades Computacionais - Aplicação ao Processador de Instruções do Minicomputador G11 (Rio de Janeiro) 1979.

VI, 171p. 29,7 cm (COPPE-UFRJ, M.Sc., Engenharia de Sistemas e Computação, 1979)

Tese - Univ. Fed. Rio de Janeiro. Fac. Engenharia

i. Testes I. COPPE/UFRJ II. Título

AGRADECIMENTOS

Ao Professor Estevam Gilberto De Simone por sua orientação e incentivo; a Sérgio Leite Terzella e Álvaro Abilio Neves pelo apoio teórico e inestimável, aos integrantes do projeto de desenvolvimento do mini-computador G11 graças aos quais foi possível realizar este trabalho, à COBRA Computadores e Sistemas Brasileiros S.A. pelo acesso ao projeto G11, referência indispensável na presente tese.

RESUMO

Este trabalho enfatiza o comportamento de testes de unidades computacionais sujeitas a falhas através do desenvolvimento de um sistema de detecção e diagnóstico para o processador de instruções do mini-computador G11, distribuido em três etapas a saber: levantamento de pontos, seleção dos testes necessários e avaliação de sua confiabilidade. Analisa a dificuldade de selecionar os pontos e os testes bem como determina a ordem de aplicação dos testes, sugerindo um método que contribua para a resolução destas questões, baseado no modelo descrito por Kime e Russel em 1975.

ABSTRACT

This paper studies the behavior of a test system, as applied to computing devices, which is intended for fault detection and diagnosis. The system has been applied to the G11 C.P.U., three main steps have then been considered, namely; problem identification, test selection, and realiability evaluation.

The model described by Kime and Russel (1975) has been applied as a mean to help solve all the proposed problems.

ÍNDICE

CAPÍTULO I

INTRODUÇÃO

Com o aumento da complexidade dos sistemas digitais atuais, a confiabilidade tornou-se um aspecto importante. Os sistemas digitais como dispositivos físicos estão sujeitos a falhas. A tecnologia empregada atualmente torna os sistemas digitais mais confiáveis que a tecnologia empregada inicialmente, a nível de cada componente, devido ao crescimento da integração dos circuitos. Uma falha pode ser definida como qualquer mudança não desejada no comportamento de um sistema. O ideal em termos de manutenção de um sistema digital é que a detecção, o diagnóstico e o reparo de uma falha sejam efetuados no tempo mais breve possível.

Nos sistemas digitais envolvendo processamento de tempo real como redes de distribuição telefônica ou controle de vôo aéreo chega a ser necessário o uso de monitores, que exercitam e testam o sistema periodicamente, verificando seu funcionamento, e podendo capacitar a continuação da operação sob certas falhas e até repará-las. Em alguns sistemas êste esquema deve estar incorporado para garantir a alta confiabilidade. São os casos extremos de processamento em tempo real, onde a necessidade obriga o sistema a possuir "auto-reparo".

A questão do tempo de desativação de uma máquina está diretamente ligada ao seu esquema de manutenção, que vai desde a infra-estrutura que a fábrica possui em termos de re

cursos humanos e sistemas de manutenção até sua política de manutenção que pode ser por relocação de módulos ou componentes.

Um fator importante na manutenção é o sistema de teste. Sua eficiência é medida pela capacidade de localização da falha, permitindo o reparo que pode ser feito através de reconfiguração, substituição ou correção da parte defeituosa.

Quando se pretende verificar o funcionamento de uma máquina para reparar algum defeito que esteja ocorrendo, na realidade está se tentando verificar é o estado das peças e suas conexões. Esta observação leva a considerar que o teste de uma máquina é um conjunto de testes de peças. Cada peça merecendo atenção e sendo testada em todas as situações possíveis.

As situações examinadas para cada peça, são as situações externas a peça. Por exemplo, se a peça em questão é um componente eletrônico, só interessa analisar as saídas correspondentes as entradas submetidas. Não há interesse em verificar o funcionamento interno do componente. Interessa-nos apenas ter conhecimento se os resultados esperados são obtidos. Isto é, se o componente funciona adequadamente ou não. No entanto, se a política de reparo, determinar o conserto da peça, em vez de sua substituição, após o exame externo quando houver a constatação de que a peça apresenta defeito, será a peça examinada internamente. O que limita então o detalhamento do exame de uma máquina é o nível do reparo.

No caso do processador de instruções, onde as instruções podem ser consideradas como seu revestimento, confundindo, o objeto do teste, no "processador" ou "conjunto de instruções", o detalhamento é um item importante a ser determinado, devido a quantidade de elementos envolvidos em seu interior.

Um dos processadores mais populares o "8080" da INTEL, apresenta uma coleção de 240 (duzentos e quarenta) instruções. Possibilitar esta variedade de funções em uma unidade computacional e garantir o seu desempenho, leva a supor que seu projeto não seja de simples concepção. Para o usuário as instruções se traduzem como funções que recebendo determinados valores como seus parâmetros, geram resultados esperados. Esta imagem contrasta com o comportamento complicado no interior do processador, que sugere uma rede de componentes, onde cada instrução ao ser executada, provoca a ativação de um caminho próprio, i.e., para cada instrução existe uma sequência programada de ativação de componentes.

Para conhecer o comportamento de um processador, o interior de cada instrução, é necessário estudar em detalhes, a arquitetura do processador, sua unidade de controle, sua unidade de comunicação de dados. Ao mesmo tempo, sem perder de vista o objetivo do teste, levantando apenas os elementos necessários ao teste, os elementos que podem ser reparados quando apresentarem defeito.

Desde 1950 que estudos vem sendo realizados na área de testes de sistemas digitais. A bibliografia deste assunto já é extensa e Breuer et al |1| procuraram reuni-la no livro "DIAGNOSE & RELIABLE DESIGN OF DIGITAL SYSTEMS". O peso dado a cada capítulo, bem como as próprias referências bibliográficas apontam que a maioria das teorias se voltaram para a verificação de funcionamento dos circuitos digitais. Hoje em dia, quando os componentes apresentam uma alta integração, chegando a possuir software armazenado, como a microprogramação, a tentativa razoável seria aproveitar os resultados destes estudos extrapolando-os para um nível superior, tratando os elementos microprogramados como pontos, assim como os circuitos tem seus pontos analisados. É aproximadamente dentro desta perspectiva que se desenvolverá o presente trabalho.

A teoria que ensaia os testes dos componentes com alto índice de integração é descrita por Lippmam et al |2|, e indica que os esquemas tradicionais de testes de unidades computacionais são projetados para encontrarem falhas fixas na maioria dos sistemas combinatoriais, onde pirâmides de circuitos são usados para efetuar uma tarefa específica e assim operar em caminhos que podem ser traçados seguindo o circuito específico. Deste modo estes métodos não podem satisfazer aos testes das barras bidirecionais e as falhas de software que caracterizam os sistemas baseados em microprocessadores. Uma possível solução seria então, o uso de novos algorítmos de diagnósticos desenvolvidos para serem testados nos microcomputadores.

Devido ao alto nível de integração dos microprocessadores, seus testes são complicados, pois os novos circuitos de alta integração tanto oferecem solução como ocasionam novos problemas. O artifício a utilizar é o aproveitamento da própria inteligência dos microcomputadores, notadamente sua habilidade de fazer auto-teste. Os "auto-testes" podem reduzir os custos de testes de microcomputadores e pode ter o mesmo efeito que os "testes de placas de microprocessadores".

Escritos nas próprias linguagens de microprocessadores em vez da linguagem do equipamento de teste automático, um programa de teste residente providenciará grandes volumes de dados estimulados e verificações apuradas do funcionamento do sistema. Além destas vantagens, podem ser usados em testes das fases de desenvolvimento, produção, implantação e manutenção.

Para que o sistema de teste proporcione reparo a nível de desenvolvimento, verificando todos os pontos de projeto, seu diagnóstico deve ser minucioso, fornecendo as localizações das falhas. Um exame extenso da arquitetura do processador de instruções do mini-computador G11 deve ser feito com o objetivo de levantar todos os pontos de falhas possivelmente reparáveis por reconfiguração, substituição ou reparo local. Ao mesmo tempo a implementação do teste deve levar em consideração que sua ferramenta básica está em teste e que pode não estar livre de erros exigindo que os testes mais simples sejam feitos antes, que sejam repetidas situações que utilizem o mesmo recurso para que se chegue a conclusão de onde ocorre uma falha, caso

uma situação tenha apresentado defeito. Outro aspecto considerado são os pontos que são ativados sempre para qualquer situação e que não tem possibilidade de serem identificados isoladamente, sendo então por necessidade reunidos em um ponto. É importante que no momento de descrição dos testes os pontos estejam totalmente delineados para que não haja nem redundância nem falta de testes.

Após o levantamento dos pontos e descrição dos testes deve ser feita uma avaliação dos testes no que diz respeito a diagnóstico com reparo. Vários autores |3| a |16| estudaram a questão da avaliação dos testes, e alguns critérios foram estabelecidos para que um sistema de teste tenha capacidade de ser diagnosticável com reparo. O estudo desses autores não evidenciou apenas a avaliação, foi acrescentada a teoria de testes alguma formalização.

CAPÍTULO II

LEVANTAMENTO DOS PONTOS DE FALHA

A arquitetura do processador de instruções do minicomputador G11 vai ser analisada neste capítulo com o objetivo de serem levantados todos os pontos de falha que deverão permitir reparo.

2.1 - Características Gerais do Mini-Computador G11

a) Palavras de 16 bits de comprimento, ou 2 "bytes" de 8 bits cada um.

b) Endereçamento direto de 64 K palavras de memória principal (K = 1024).

c) Processamento de palavras ou "bytes" (facilidades para a manipulação de bits individuais).

d) Estrutura modular: as funções básicas do sistema são desempenhadas por módulos de fácil reposição, que interagem assincronamente por intermédio de uma via comum de alta velocidade.

e) Possui um sistema de ponteiros dinâmicos que permite a manipulação eficiente de estruturas de dados tais como pilhas, vetores, filas, ou tabelas.

f) Conjunto de 8 registradores de 16 bits de propósito geral. Um

desses registradores é usado como contador de instruções.

g) Dois modos de processamento: supervisor e usuário.

h) Sistema de interrupções com 9 níveis de prioridade.

i) Interrupções de vários níveis através de um conjunto de posições reservadas de memória.

j) Memória de semicondutor de acesso rápido.

k) Endereçamento relativo a uma base, permitindo que as operações de relocação estática e dinâmica sejam feitas em qualquer instante do funcionamento do processador.

l) Proteção contra acessos de escrita na memória feitos fora de uma área associada a um programa.

m) Poderoso conjunto de 117 instruções de máquina micro-programadas. Uma instrução pode ter 16 ou 32 bits de comprimento.

n) Vários modos de endereçamento são disponíveis para os operandos de uma instrução. Entre esses podemos citar:
   - Operando contido na própria instrução (imediato);
   - Operando em registrador de propósito geral;
   - Endereço de operando em registrador de propósito geral, podendo esse ser pré-incrementado, ou pós-decrementado, durante o endereçamento;

- Endereço do operando especificado na própria instrução, com ou sem uso de um registrador de índice, e com ou sem pré-incremento do índice;
- Instrução especifica endereço do endereço do operando, com ou sem pós-indexação, e com pré-incremento opcional do índice.

o) Até 4 canais de entrada/saída podendo ser ligados a via comum do G-11. Cada canal pode ser do tipo seletor (transfere blocos de palavras com um dispositivo de cada vez), ou do tipo concentrador (transfere "bytes" com vários dispositivos ao mesmo tempo).

p) As instruções de entrada/saída do G-11 permitem o endereçamento de no máximo 16 dispositivos por canal. Um dispositivo pode ser um periférico, ou uma unidade de controle a qual estão ligados um ou mais periféricos.

q) Configuração do sistema: O mini-computador G-11 possui uma estrutura modular, como pode ser visto na figura 2.1. Os módulos se ligam a uma via comum bidirecional, capaz de transmitir 16 bits de dados em paralelo, chamada de Via Comum de Comunicação (VCC). Os módulos do G-11 são:
   - Processador Central (UCP)
   - Memória Principal
   - Canais de Entrada/Saída, do tipo seletor ou concentrador
   - Controlador da Via Comum de Comunicação (CVCC), responsável pelo controle de acesso e monitoramento dos diálogos que ocorrem na via.

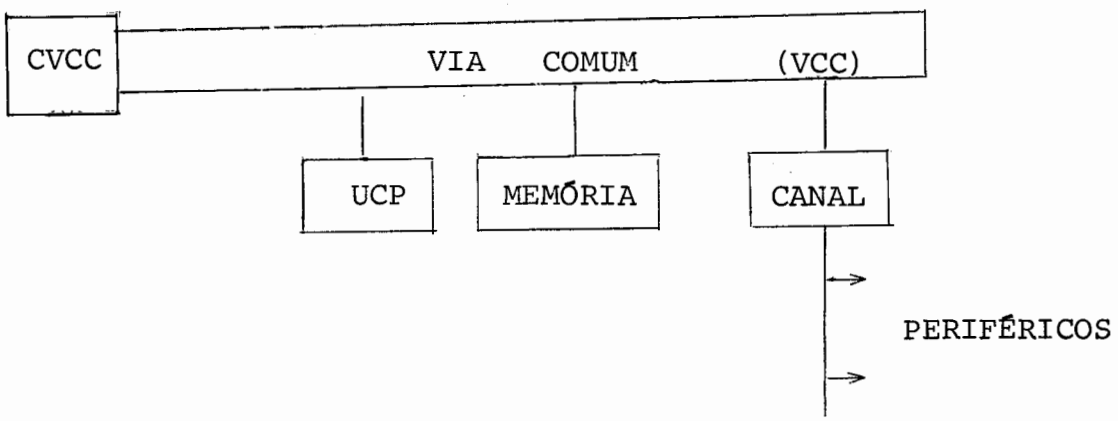


Fig. 2.1 - ESTRUTURA DO G11

2.2 - Arquitetura da Unidade Central de Processamento

O G11 é um minicomputador de propósitos gerais, implementado com lógica TTL, e composto de vários módulos interconectados por uma Via Comum de Comunicação (VCC).

A Unidade Central de Processamento (UCP) do G11 é o módulo que interpreta e executa a maioria das instruções do computador. Ela emprega aritmética inteira de complemento a 2, e é capaz de endereçar diretamente 65.536(64K) palavras de 16 bits de memória principal.

A UCP pode ser dividida em dois submódulos: A unidade de movimento de dados (UMD) e a unidade de controle microprogramada (UCM). A figura 2.2 ilustra a composição da UCP.

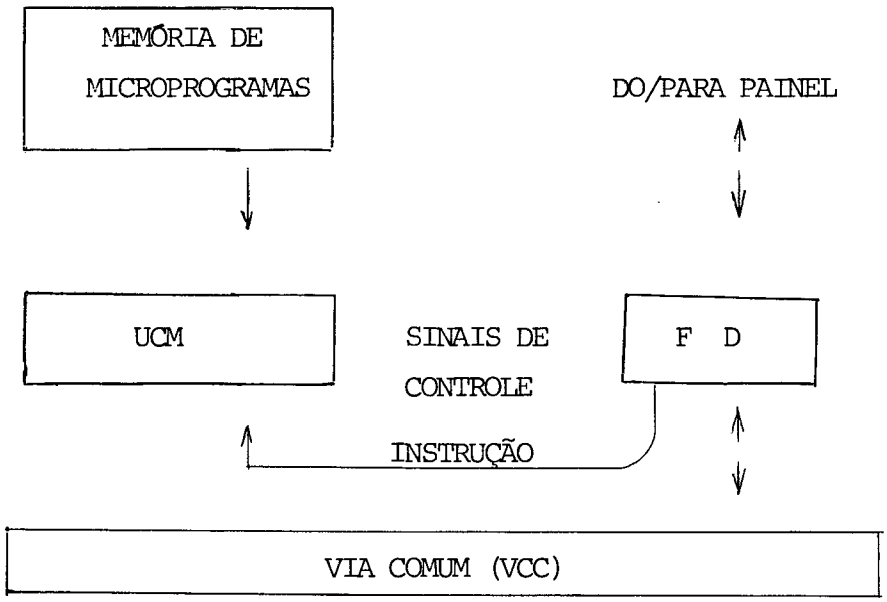


Fig. 2.2 - O PROCESSADOR CENTRAL

A UMD é composta de registradores de le bits e de uma Unidade Lógica e Aritmética (ULA), interligados de modo a realizar as operações necessárias a execução de uma instrução.

A UCM identifica a instrução e controla a sequência de passos necessários para executá-la. Ela utiliza uma memória de semi-condutor de acesso rápido, do tipo "ler-somente" (ROM), onde estão armazenados os microprogramas que, devidamente decodificados na UCM, geram os sinais para executar a instrução.

2.3 - Unidade de Controle Microprogramada

Para a descrição posterior do funcionamento da unidade de controle, cabe primeiramente definir alguns termos que são usados.

- Memória de controle: É uma unidade de armazenamento que contém os microprogramas. No caso do G-11 é uma memória do tipo "ler-somente" expandível até 4K palavras de 32 bits por palavra.

- Micro-programa: É um conjunto de micro-instruções que devem ser executados numa determinada sequência.

- Micro-instrução: É o conteúdo de uma palavra de 32 bits da memória de controle, subdividida em vários campos contendo micro-ordens.

- Campo: É cada subdivisão da micro-instrução onde estão especificadas as micro-ordens. A micro-instrução do G11 possui 10 campos.

- Micro-ordem: É um código particular de uma certa função, especificado num dos campos da micro-instrução.

- Palavra de controle: É uma palavra de 32 bits da memória de controle, contendo uma micro-instrução, cujas micro-ordens são executadas no mesmo ciclo de controle. O conceito de palavra de controle e micro-instrução geralmente se confundem.

- Sinais de controle: São todos os sinais elementares que se originam da decodificação de cada micro-ordem. Estes sinais divergem da unidade de controle para partes diferentes da máquina , permitindo ou inibindo a passagem de informação pelos elementos armazenadores da unidade de movimento de dados do Processa

dor Central (UCP) e demais circuitos, além da própria Unidade de controle.

A microprogramação dispõe das seguintes entidades para o processamento das informações:

- Rede de 16 registradores: 4 registradores de rascunho
  1 registrador de pilha de contexto
  1 registrador de limite de programa
  1 registrador de base local
  1 registrador de programa
  8 registradores de propósito geral

  (RØ a R7), sendo o registrador RØ usado como contador de instruções.

- Registrador de dados (D)

- Registrador de endereços (E)

- Registrador de instruções (I)

- Registradores deslocadores (P e Q)

- Registrador de Estado (RE)

- Unidade lógica e aritmética

- Painel

- Memória principal

- Via comum

A unidade de controle é composta dos seguintes elementos:

- Mapeador
- Seleção
- Memória de controle
- Registrador de endereços da memória de controle (EC)
- Registrador de dados da memória de controle (DC)
- 2 registradores de endereço de retorno de micro-subrorina (SØ e S1)
- Decodificadores
- Somador de "+1"
- Contador (CNTD)
- Lógica de testes de condições

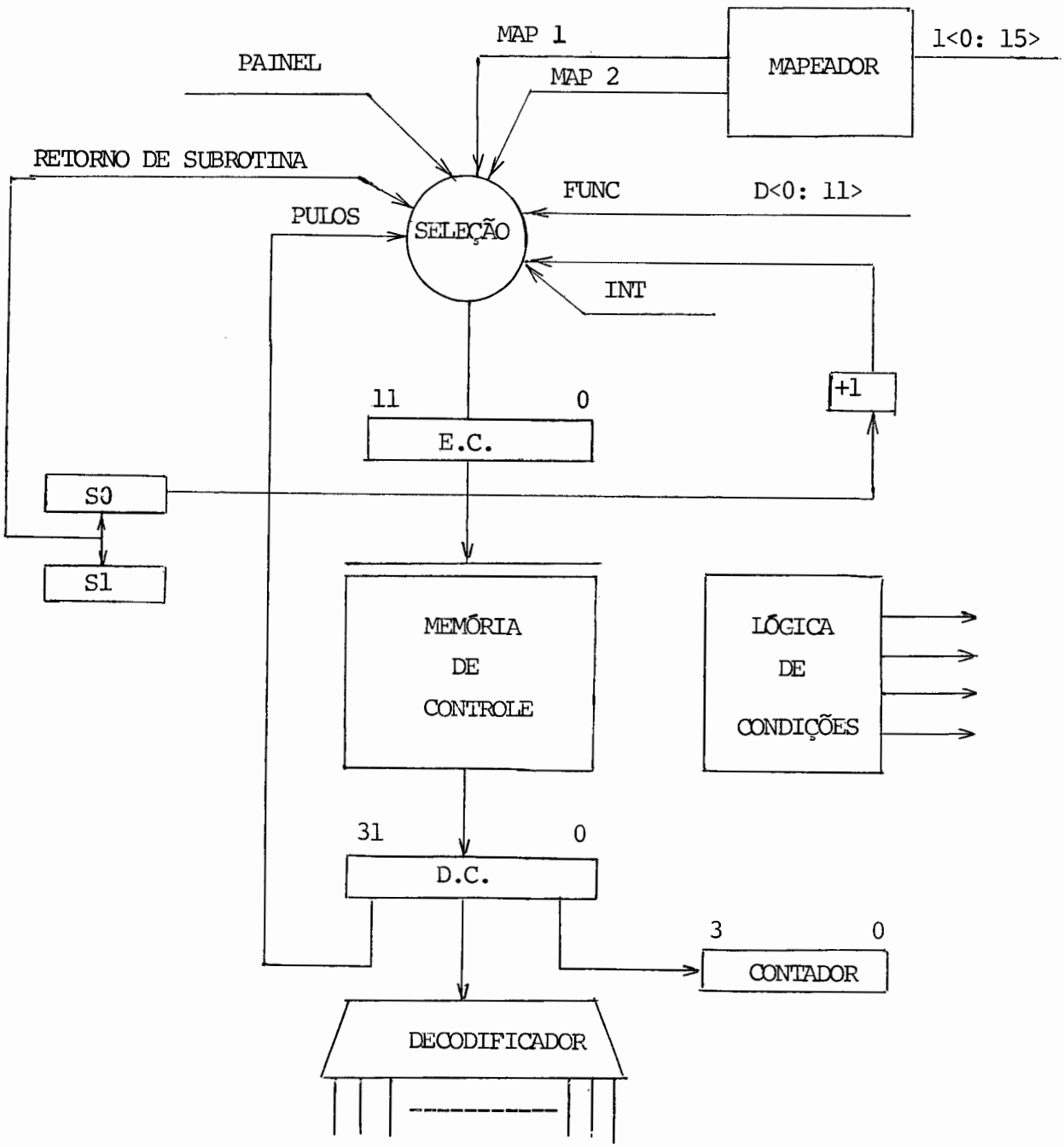


Fig. 2.3 - FLUXO DE CONTROLE

2.3.1 - Mapeador

O interfaceamento entre a unidade de movimento de dados do processador e a unidade de controle é feito pelo mapeador, cuja função é endereçar (mapear) na memória de controle um determinado microprograma de acordo com o código de operação da instrução no registrador de instruções I, botão acionado no painel, ou alguma interrupção. O mapeador, portanto, é constituído de 3 partes:

- Mapeador de instruções
- Mapeador de painel
- Mapeador de interrupções

. Mapeador de instruções

Esse mapeador consiste basicamente de duas memórias do tipo "ler-somente", cujas entradas são alimentadas pelo registrador I e cujas saídas podem ser relacionadas por certas micro-ordens para o registrador de endereços da memória de controle (EC), conforme pode ser observado na fig. 2.4.

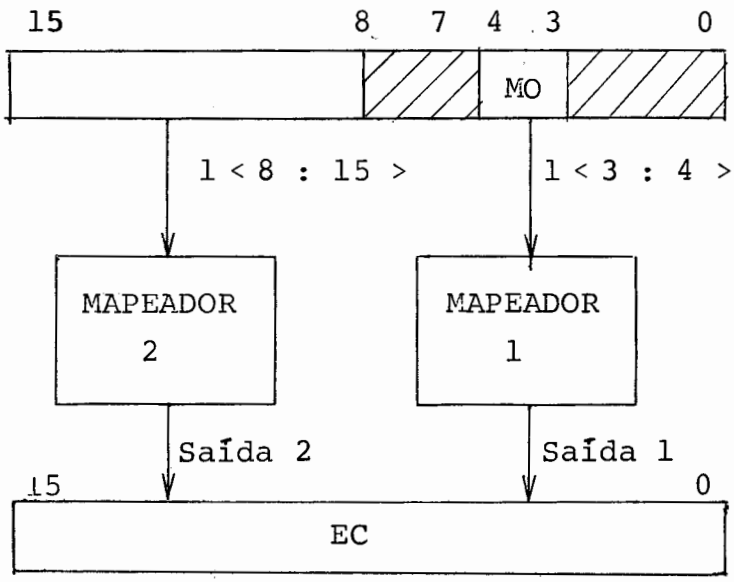


Fig. 2.4 - MAPEADOR DE INSTRUÇÕES

Ainda observando a fig. 2.4, o mapeador 1 endereça uma micro-rotina na memória de controle correspondente ao cálculo de endereço efetivo ao modo de operação M (bits < 3:4>).

Existe uma micro-rotina para cada modo de operação:

MO	MODO DE OPERAÇÃO
$(0\ 0)_2$	Direto
$(0\ 1)_2$	Indireto
$(1\ 0)_2$	Indireto pré-incrementado
$(1\ 1)_2$	Indireto pós-decrementado

Se não existir o campo MO na instrução, o mapeador 1 endereça a micro-rotina de execução da instrução se es-

ta for curta, ou permite o acionamento do mapeador 2 que endereçará uma micro-rotina de cálculo de endereço efetivo correspondente ao modo de endereçamento ME, se a instrução for longa.

O mapeador 2 tem suas entradas alimentadas pelo byte mais significativo da instrução da seguinte maneira:

- Se a instrução for longa, as entradas do mapeador 2 são: O código de operação (bits < 11:15 >), campo S/D (bit < 10 >) e campo ME (bits < 8:9 >).

- Se a instrução for curta, as entradas do mapeador 2 são: O código de operação (bits < 11:15 >), e sub-código de operação (bits < 8:10 >).

No caso de instruções longas, esse mapeador endereça primeiramente uma micro-rotina de cálculo de endereço efetivo correspondente ao modo de endereçamento ME (bits < 8:9 >).

ME	MODO DE ENDEREÇAMENTO
$(00)_2$	Direto/pré-indexado
$(01)_2$	Indireto 1 nível/pré-indexado
$(10)_2$	Indireto 2 níveis/pré-indexado
$(11)_2$	Indireto 1 nível/pós-indexado

Após a execução da micro-rotina de endereçamento ME, o mapeamento 2 endereça o micro-programa de execução da instrução.

No caso das instruções curtas, será endereçado o micro-programa de execução propriamente dita da instrução correspondente ao código e sub-código de operação.

Tanto o mapeador 1 como o mapeador 2 tem suas saídas selecionadas para o registrador EC através de micro-instruções especiais que serão descritas mais adiante.

. Mapeador do Painel

Esse mapeador tem suas entradas alimentadas por um conjunto de botões do painel frontal do G-11, e sua finalidade é a de endereçar um micro-programa que execute as funções correspondentes a cada botão do painel.

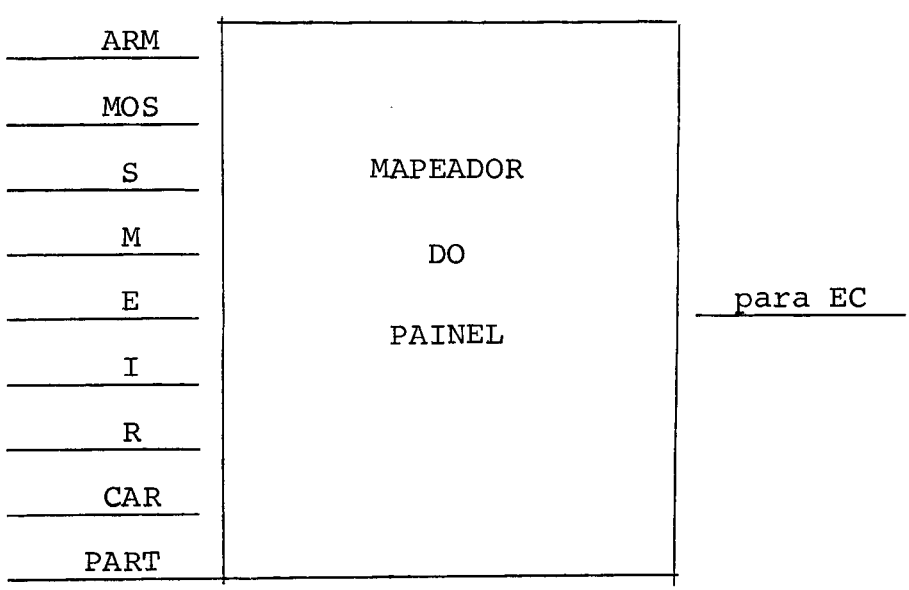


Fig. 2.5 - MAPEADOR DO PAINEL

Esse mapeador tem sua saída selecionada para o registrador EC por meio de uma micro-instrução especial que será descrita mais adiante.

. Mapeador de Interrupções

Esse mapeador tem suas entradas alimentadas pelo circuito de interrupção e no conjunto de botões do painel, e sua finalidade é a de endereçar um micro-programa de tratamento de cada interrupção que possa ocorrer.

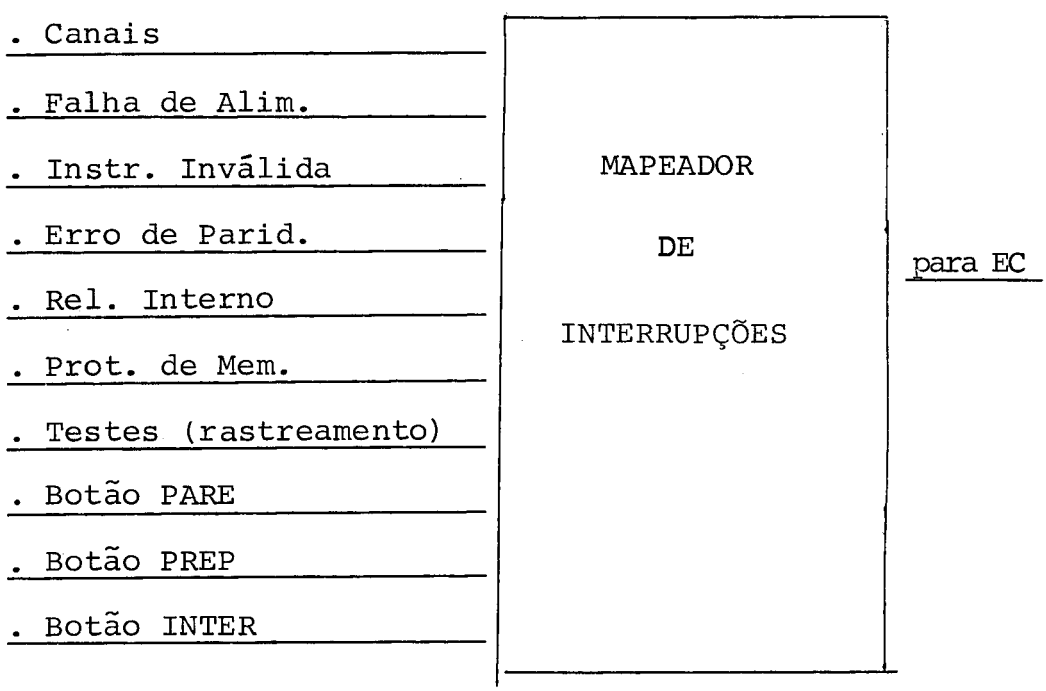


Fig. 2.6 - MAPEADOR DE INTERRUPÇÕES

Todas as interrupções na figura acima, exceto por proteção de memória e instruções inválidas são mapeadas e tratadas após o término da execução da instrução sendo executada por ocasião da ocorrência da interrupção. As duas citadas, por

serem internas, podem ocorrer durante a execução de uma instrução.

2.3.2 - Seleção

A seleção é ativada por micro-ordens específicas, permitindo a passagem para o registrador de endereços da memória de controle (EC) uma das seguintes entradas:

- As duas saídas do mapeador;
- Endereço de retorno de uma micro-subrotina para o micro-programa principal;
- Endereço contido nos bits de 19 a 30 nas micro-instruções de pulo (pulos dentro dos micro-programas ou para outras micro-rotinas);
- Endereço proveniente do mapeador de interrupção;
- Endereço proveniente do mapeador do painel;
- Endereço da micro-instrução seguinte proveniente do somador de +1;
- Saída da Unidade Lógica e Aritmética.

2.3.3 - Memória de Controle

A memória de controle é constituída de uma memória "ler-somente", com expansibilidade de até 4K palavras de 32 bits. Aqui estão contidos todos os micro-programas da fase de busca, fase de cálculo de endereço efetivo, e fase de execução

propriamente dita das instruções de máquina, tratamento das interrupções, de botões do painel, e carregador absoluto.

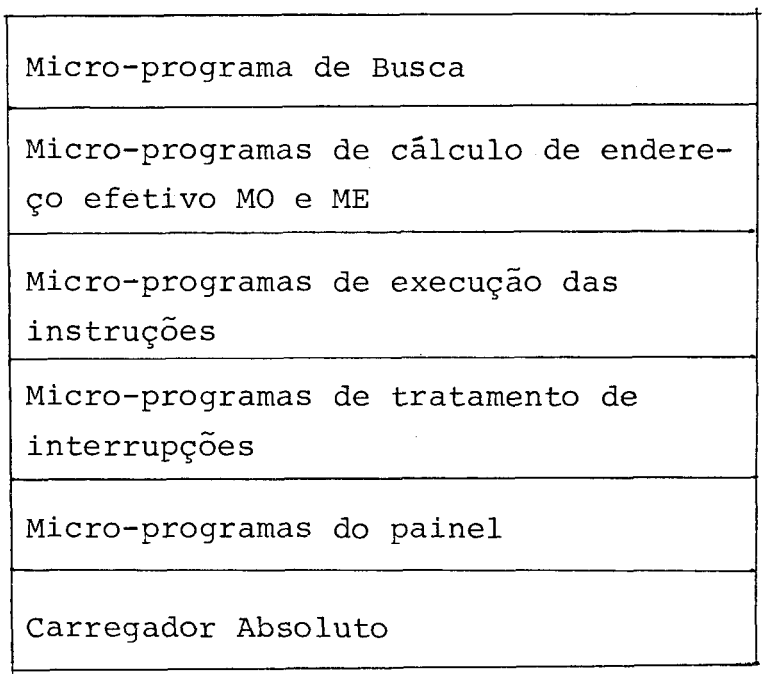


Fig. 2.6 - ORGANIZAÇÃO DA MEMÓRIA DE CONTROLE

As palavras de 32 bits de cada micro-programa são lidas sequencialmente; esta sequência pode ser descontinuada por uma micro-ordem geral de desvio, que faz com que a execução do micro-programa seja desviada para outro ponto da memória de controle, retornando ou não para o endereço de retorno, conforme o tipo da micro-ordem.

Após a leitura de uma micro-instrução, esta é armazenada no registrador de dados da memória de controle (DC). Durante a sua execução, a memória de controle é novamente acessada, automaticamente, para a leitura da micro-instrução seguinte, que só será armazenada no registrador DC após o término da execução anterior, o que se dá ao fim de um clico de controle.

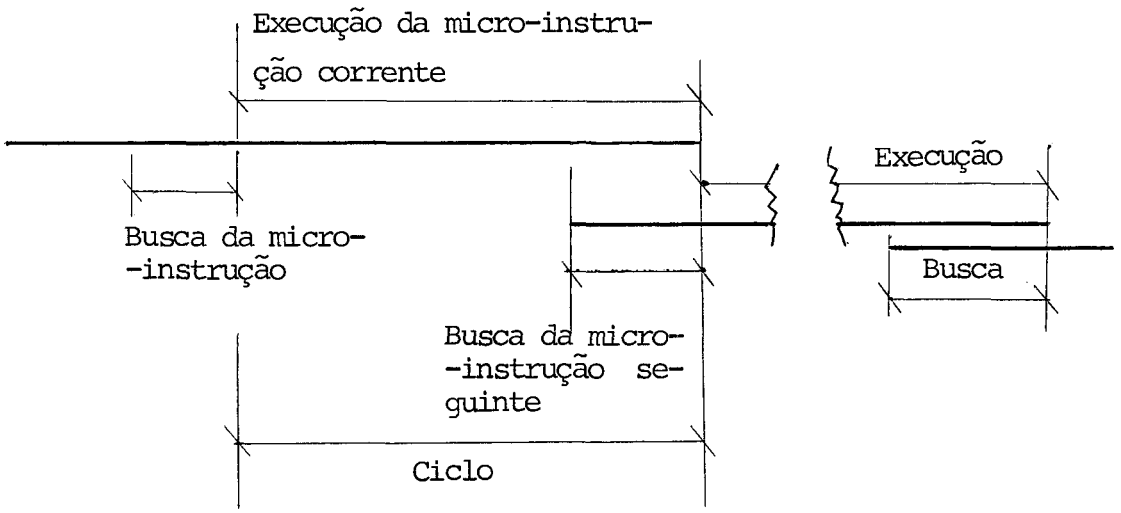


Fig. 2.7 - CICLO DE CONTROLE

2.3.4 - Registrador de Endereços de Controle (EC)

O registrador de endereços da memória de controle (EC) possui 12 bits, tendo assim a capacidade de acesso até 4K posições. Pode-se dividir esse espaço endereçável em blocos, sendo que uma versão qualquer do G11 não precisa conter todos os blocos possíveis de memória; o tamanho escolhido para os blocos, deverá ser tal que contenha o conjunto básico de instruções do G11. As micro-ordens de pulo possuem a capacidade de endereçar 4K palavras, com um endereçamento direto e absoluto não havendo portanto nenhuma espécie de paginação na memória de controle.

2.3.5 - Registrador de Dados do Controle (DC)

Ao se endereçar uma posição na memória de controle a micro-instrução é lida e transferida para o registrador

de dados (DC) na memória de controle, destruindo o seu conteúdo anterior. A seguir é iniciada a fase de execução que consiste na decodificação geral de micro-ordens.

2.3.6 - Registradores de endereço de retorno de micro-subrotina (SØ e S1)

Os registradores SØ e S1, de 12 bits, guardam o endereço de retorno quando de uma chamada de micro-subrotina. Esses dois registradores, funcionando como uma pilha, permitem até dois níveis de chamada de micro-subrotinas, no final das quais os endereços de retorno são recuperados em EC, prosseguindo-se a execução do micro-programa principal.

2.3.7 - Decodificadores

Após a micro-instrução ter sido armazenada no registrador DC, inicia-se a decodificação das micro-ordens dos campos gerando à sua saída sinais de controle, ativando os circuitos necessários para a execução da operação especificada por cada micro-ordem.

2.3.8 - Somador de "+1"

O somador de +1 é um circuito que incrementa de uma unidade o conteúdo do registrador EC fornecendo na sua saída o endereço seguinte à micro-instrução corrente. Ele é, portanto, responsável pela sequencialização normal do endereçamento

das micro-instruções, sendo inibido, apenas por uma micro-instrução geral de pulo.

2.3.9 - Contador (CNTD)

O contador de controle CNTD possui 4 bits e é carregável por uma micro-ordem especial ou pelos 4 bits menos significativos da saída da ULA.

O contador CNTD é usado em todas as operações que necessitem de uma contagem de até 16 vezes, como acontece por exemplo com as operações de deslocamento. Ele é também usado para endereçar sequencialmente os registradores da rede na Unidade de Movimento de Dados do Processador.

2.3.10 - Lógica de Testes de Condições

Este bloco fornece uma série de sinais de estado decorrentes da execução de uma instrução de máquina. Esses sinais são testáveis por micro-ordens de testes ao longo do micro-programa, gerando condições segundo as quais são executados desvios para pontos diferentes dos micro-programas que não as subsequentes.

O estado em que se encontra uma instrução durante o seu processamento é representado por condições geradas por operações efetuadas na ULA, deslocadores, do próprio código de operação da instrução, modo de endereçamento e de operação,

etc. Estas condições serão especificadas mais adiante, quando da descrição da palavra de controle do G11.

2.4 - Unidade de Movimento de Dados

Essa é a unidade onde vão ser realizadas as operações. Consiste da interconexão dos registradores com a unidade lógica e aritmética Pode ser alimentada pelo painel ou memória e é controlada pela unidade de controle.

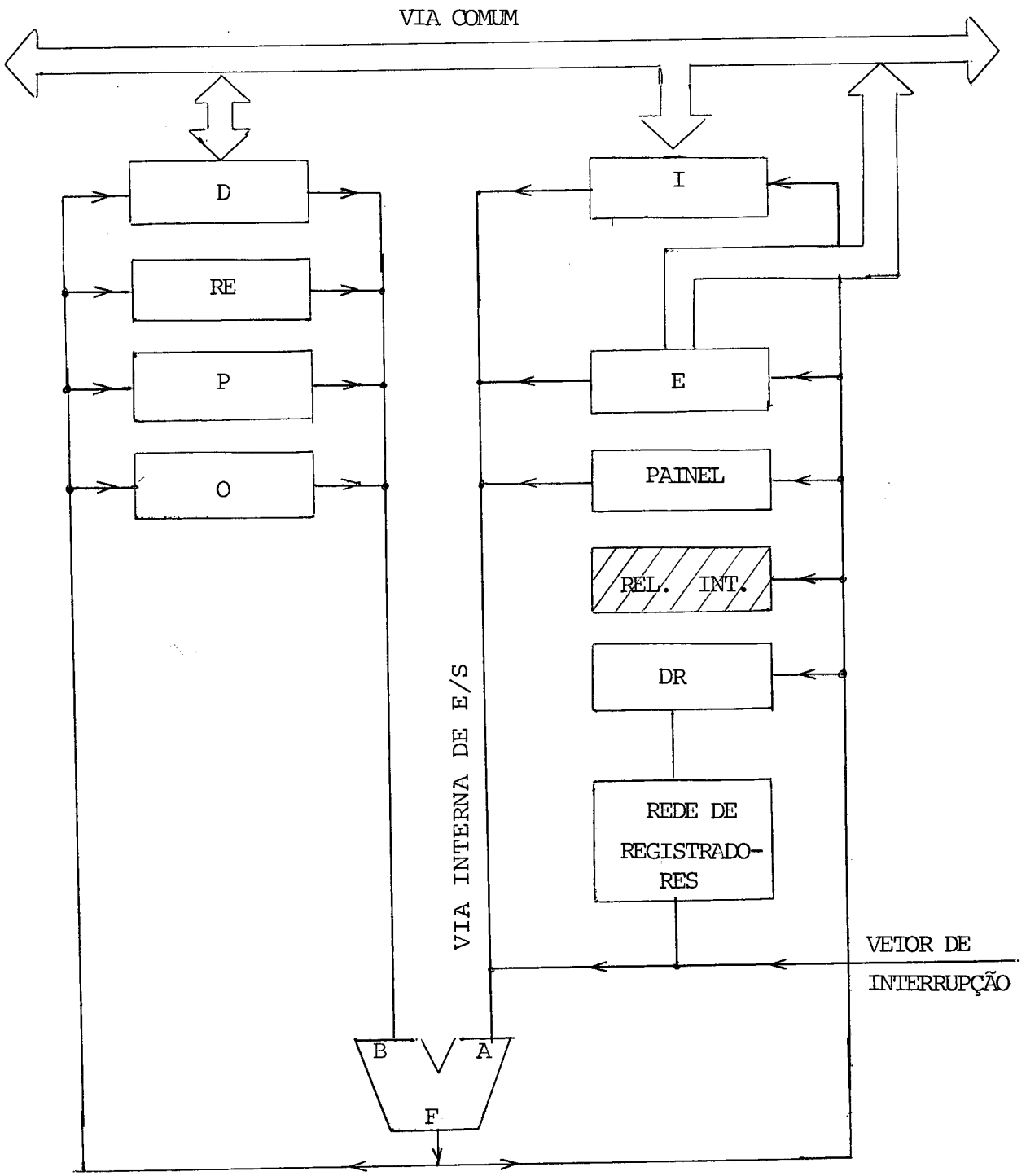


Fig. 2.8 - UNIDADE DE MOVIMENTO DE DADOS

2.4.1 - Registrador de Estado

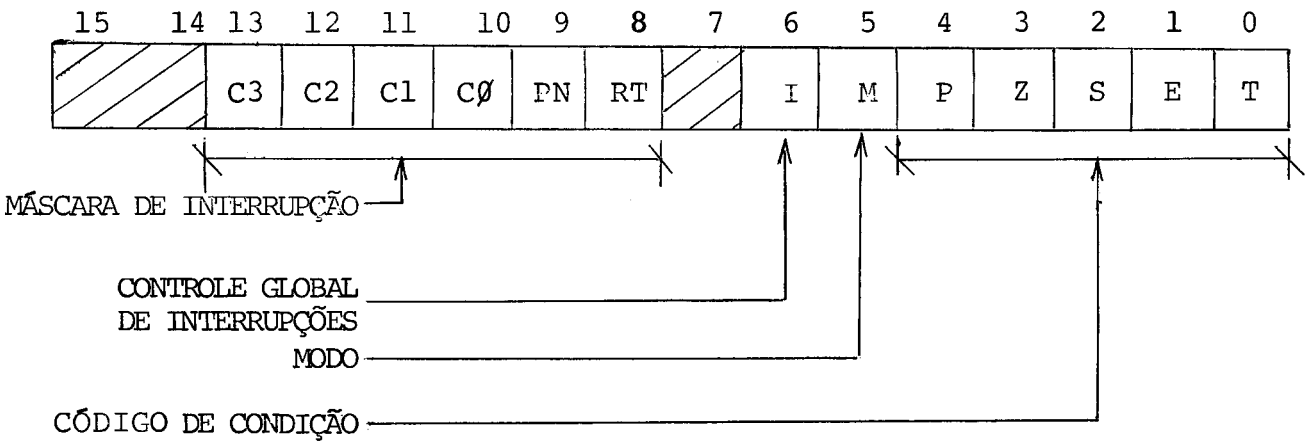


Fig. 2.9 - REGISTRADOR DE ESTADO

Todas as informações sobre o estado atual da UCP são reunidas numa palavra de 16 bits (palavra de estado) , que é mantida num registrador da UMD chamado de Registrador de Estado (RE). Essas informações consistem da máscara de interrupção, um indicador global de ocorrência de interrupções, um indicador do modo de execução (supervisor ou usuário), e os códigos de condição que descrevem o último resultado computado pela UCP.

. Máscara de Interrupção

A UCP reconhece 9 níveis de interrupções externas, com prioridades distintas. Desses, os 3 mais prioritários (níveis Ø, 1 e 2) não podem ser inibidos. Os 6 níveis restantes são controlados por uma máscara de interrupção, na qual cada bit corresponde a um dos níveis, conforme mostra a fig. 2.10.

BIT DE RE	ORIGEM DA INTERRUPÇÃO	NOME	NÍVEL
8	rastreamento	RT	3
9	painel	PN	4
1Ø	canal Ø	CØ	5
11	canal 1	C1	6
12	canal 2	C2	7
13	canal 3	C3	8

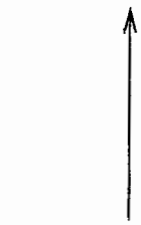


Fig. 2.10 - NÍVEIS DE INTERRUPÇÃO INIBÍVEIS

Convencionou-se que se o bit está ligado (=1) as interrupções correspondentes estão inibidas, e em caso contrário (=Ø) permitidas.

- Controle global de interrupções

A máscara de interrupção possibilita a inibição seletiva de interrupções. Entretanto, o G11 possui um mecanismo que permite a inibição de todas essas interrupções ao mesmo tempo, independente do valor da máscara. O bit I do RE é usado para essa finalidade.

Quando esse bit está ligado (I=1) o processador não aceita nenhuma interrupção dos níveis 4 a 8, inclusive esses, mesmo que a máscara de interrupções permita a ocorrência das mesmas . Quando desligado (I=Ø), é a máscara de interrupção que controla que interrupções podem ocorrer.

O bit I pode ser alterado por programa (através de instruções privilegiadas), ou automaticamente por "hardware". O "hardware" liga esse bit após a ocorrência de qualquer interrupção, ou após uma chamada de supervisor (CSUP), para permitir que o supervisor carregue uma nova máscara de interrupção. O bit é também ligado ao ser apertado o botão PREP (prepara) do painel.

- Modo

O bit de modo (M) indica se a UCP está operando no modo supervisor (M=1), ou no modo usuário (M=Ø). O modo usuário restringe o acesso a máquina, não permitindo que certas instruções (ditas privilegiadas) sejam executadas. Qualquer tentativa de executar uma dessas instruções em modo usuário acarretará numa interrupção de nível 1.

As instruções privilegiadas do G11 são as seguintes, em ordem alfabética de mnemônico:

AA/AFI/CGMI/EBC/EBD/ENT/ESP/LC/LED/LIC/LRE/LVI/MED/MINT/PARA/RINT/SAI/SBC/SBD/.

O bit M é automaticamente ligado pelo "hardware" imediatamente após uma interrupção, ou após uma chamada de supervisor (CSUP). Com isso garante-se que o programa que trata a interrupção, ou a CSUP, tem acesso irrestrito à máquina. Ele é também ligado ao se apertar o botão PREP (prepare) do painel.

- Códigos de condição:

Os códigos de condição contem informações sobre o último resultado que saiu da ULA, isto é, sobre a última instrução executada que realizou operações lógicas e/ou aritméticas.

Os bits do código de condição são ligados como se segue:

P = 1, se o resultado foi ímpar.

Z = 1, se o resultado foi zero.

S = 1, se o resultado foi negativo.

E = 1, se a operação resultou em "vai-um" do bit mais significativo.

T = 1, se a operação resultou em transbordamento.

NOTA: O transbordamento ("overflow") numa operação indica resultado errado, por não ser o mesmo representável em 16 bits.

2.4.2 - Registradores de Propósito Geral

O G11 dispõe de 8 registradores de 16 bits que podem ser usados como acumuladores em operações aritméticas, armazenamento temporário, indexadores ou ponteiros. Eles são chamados de registradores de propósito geral, e nas instruções do G11 são endereçados por um número de Ø a 7.

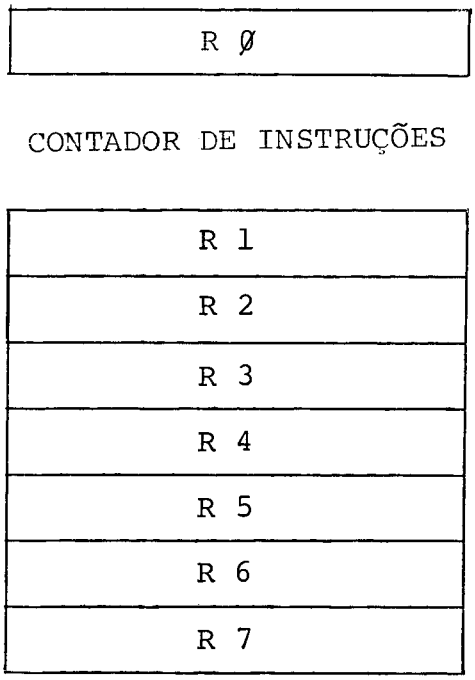


Fig. 2.11 - REGISTRADORES DE PROPÓSITO GERAL

O registrador RØ é usado como o contador de instruções (CI) da máquina e contém o endereço relativo a base de programa da última palavra da instrução sendo executada. Normalmente o programador só emprega esse registrador para endereçamento relativo ao contador de instruções.

O registrador R7 possui também um uso especial. A UCP emprega esse registrador para passar informação ao supervisor sobre certas interrupções, ou para passar o número da CSUP numa chamada de supervisor; ele é ainda endereçado implicitamente em várias instruções de E/S. Tal uso de R7 é, entretanto, totalmente transparente aos programas de usuários.

Os registradores RØ e R1 possuem restrições

quanto ao seu emprego em instruções, a saber:

. O registrador RØ não pode ser usado como indexador;

. Não existem instruções curtas nem imediatas que usem RØ e Rl como operandos;

. Ambos os registradores RØ e Rl não são salvos nem restaurados pelas instruções SALV e REST, respectivamente;

. Os registradores RØ e Rl não são usados como operandos implícitos de operações mistas (que envolvem operandos de 32 bits).

2.4.3 - Base Local

O G11 possui no seu reportório algumas instruções que usam um deslocamento não negativo de 8 bits, chamados de instruções de deslocamento curto. Esse deslocamento é usado em tempo de execução ao conteúdo do registrador de Base Local (BL), obtendo-se assim um deslocamento efetivo de 16 bits. A finalidade da base local é principalmente permitir o endereçamento de uma área de memória de até 256 palavras de tamanho por instruções curtas de 16 bits, em vez de se usar instruções longas de 32 bits, permitindo assim a economia de considerável área de código.

Um programa pode alterar e ler o valor de BL unicamente através das instruções (não privilegiadas) Restaura Registradores (REST) e Salva Registradores (SALV), respectivamente.

2.4.4 - Base de Programa

À exceção das instruções Carrega Absoluto (CGA) e Armazena Absoluto (AA), todos os endereços gerados pelas instruções do G11 são relativas a um valor fixo, que fica armazenado no registrador de Base de Programa (BP). Usualmente esse valor é igual ao endereço absoluto da primeira palavra do programa em execução.

A menos das exceções já mencionadas, todos os endereços gerados na UCP são adicionados ao conteúdo de BP para a obtenção de um endereço efetivo de 16 bits, que será usado no acesso a memória.

O registrador BP só pode ser alterado como parte das operações de restauração de contexto, ou pela UCP após uma interrupção. Não há instrução para ler o conteúdo desse registrador.

2.4.5 - Limite de Programa

O G11 usa o registrador de limite de programa (LP) em conjunto com BP para implementar um esquema de proteção de memória. O registrador LP contém o endereço relativo da última posição da área de programa (tamanho da área menos um), e a UCP impede qualquer acesso de escrita na memória fora da área delimitada pelo par BP e LP.

É importante observar que não há proteção contra leitura de palavras da memória, isto é, dados podem ser lidos, ou instruções executadas, de qualquer lugar da memória. Entretanto, acessos de escrita são impedidos (uma interrupção do processador é gerada) caso o endereço efetivo de acesso a memória seja menor que BP, ou maior que a soma de BP e LP.

O LP só pode ser alterado como parte das operações de restauração de contexto, ou pela UCP após uma interrupção. Não há instrução que permita ler o conteúdo desse registrador.

2.4.6 - Registrador Índice da Pilha de Contexto

O recebimento de uma interrupção pela UCP causa uma mudança de contexto, isto é, os registradores relevantes ao programa em execução são salvos na memória e novos valores são carregados em alguns desses registradores.

A área de memória onde deve ser salvo o contexto do programa interrompido é determinada pela UCP através do conteúdo do registrador índice da pilha de contexto (RIPC), e de uma palavra reservada de memória. A restauração desse contexto nos registradores apropriados também é feita por referência a RIPC e a mesma palavra reservada.

O registrador RIPC não pode ser lido por programa via instrução (privilegiada) ele índice da pilha de con-

texto (LIC). Ele não pode ser alterado por programa. A UCP inicia RIPC com o valor 1 (um) ao ser apertado o botão CAR (carrega) do painel.

2.4.7 - Registrador de Dados D

2.4.8 - Registradores Deslocadores P e Q

2.4.9 - Registrador I

2.4.10 - Registrador de Endereços E

2.4.11 - Painel

2.4.12 - Unidade Lógica e Aritmética

2.4.13 - Registradores de Rascunho

2.5 - Micro Instrução

A micro-instrução no G11 é constituída de 32 bits divididos em 10 campos contendo as várias micro-ordens cuja decodificação gera sinais de controle que acionarão os circuitos convenientemente da Unidade de Movimento de dados e da própria unidade de controle, numa sequência tal que os algoritmos sejam corretamente executados.

O formato da micro-instrução é o seguinte:

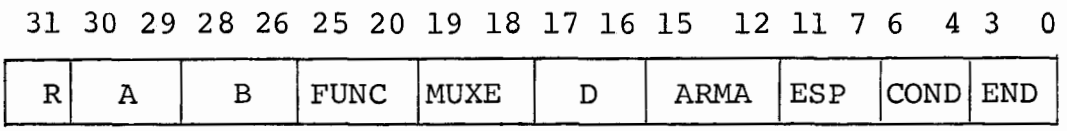


Fig. 2.12 - FORMATO DA MICRO-INSTRUÇÃO

Cada campo possui um grupo de bits codificados que determinam a operação a ser efetuada pelo controle. Cada código desses grupos corresponde a uma micro-ordem.

As operações gerais especificadas em cada campo são as seguintes:

Campo R - As micro-ordens desse campo controlam o relógio central no sentido de permitir a geração dos ciclos de controle ou de inibí-los. São usados por ocasião de um acesso à memória principal ou à via comum de comunicação, a fim de parar o processamento de micro-instruções subsequentes até que o dado acessado esteja seguramente correto no registrador de dados da Unidade de Movimento de dados.

Campo A - Este campo contém as micro-ordens que selecionam o registrador de Unidade de Movimento de dados que terá acesso à entrada A da ULA.

Estas micro-ordens funcionam em conjunto com as dos campos MUXE e END.

Campo B - Este campo contém as micro-ordens que selecionam o registrador de Unidade de Movimento de dados que terá acesso à entrada B da ULA.

Campo FUNC - Esse campo contém os códigos das funções lógicas e aritméticas da ULA, ou das operações de deslocamentos efetuados pelos registradores deslocadores P e Q, com ou sem atualização dos bits do registrador de estado S do processador.

Campo MUXE - As micro-ordens desse campo controlam a seleção e endereçamento dos registradores da rede.

Eles funcionam em conjunto com as micro-ordens dos campos A e END.

Campo D - Esse campo contém as micro-ordens que determinam o sentido de deslocamento ou a carga dos registradores deslocadores P e Q.

Estas micro-ordens funcionam em conjunto com as do campo FUNC.

Campo ARMA - As micro-ordens desse campo são responsáveis pela seleção do registrador onde será armazenado o resultado da operação realizada no ciclo de controle corrente. Esse resultado é provindo da saída F de ULA.

Essas micro-ordens eventualmente funcionam em conjunto com as dos campos MUXE e END, quando for selecionado um dos registradores da rede.

Campo ESP - Esse campo contém os códigos das micro-ordens especiais que efetuam pulos para micro-rotinas diferentes, acessos à memória ou à via comum, testes de condições, controle sobre o circuito de seleção para o registrador EC, e carga imediata do contador CNTD.

Estas micro-ordens eventualmente funcionam em conjunto com as do campo COND, e do campo END, quando for determinada a carga do contador CNTD.

Campo COND - Esse campo contém as condições que devem ser testadas pelas micro-ordens condicionais do campo ESP. Essas condições só podem ser testadas por micro-programas e nunca por instruções de máquina. A condição é dita verdadeira se ela for igual a 1.

Campo END - Esse campo contém o código dos registradores da rede, vetor de interrupção e registrador de chaves do painel endereçáveis pelas micro-ordens dos campos A e MUXE.

2.6 - Micro-Programação das Instruções

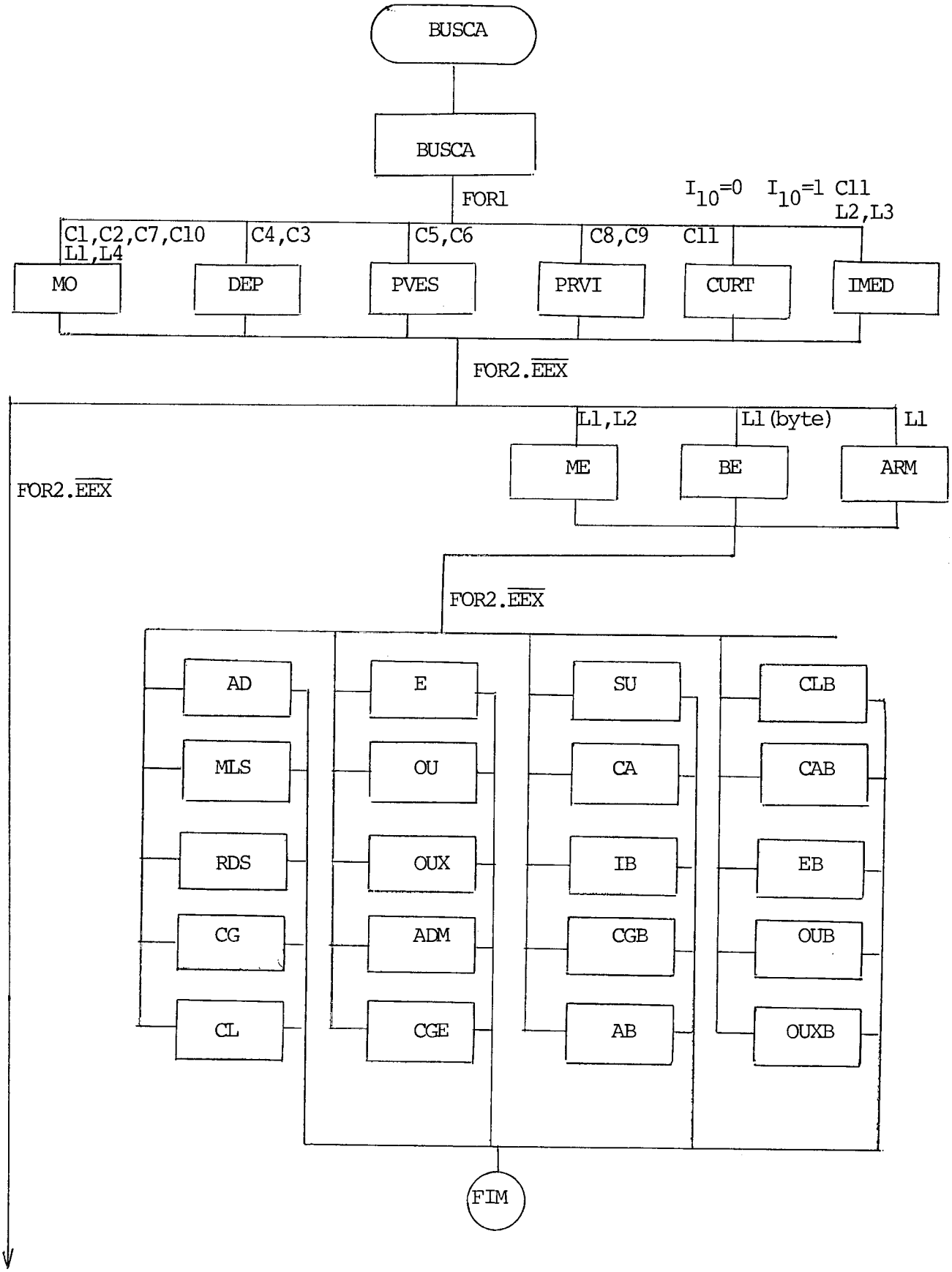
O controle micro-programado permite a implementação mais simples das instruções, visto que as operações executadas seguem a mesma filosofia em todos os casos: os sinais de controle do fluxo são gerados pela decodificação ou mesmo diretamente das palavras de controle (micro-instruções) constituintes do micro-programa.

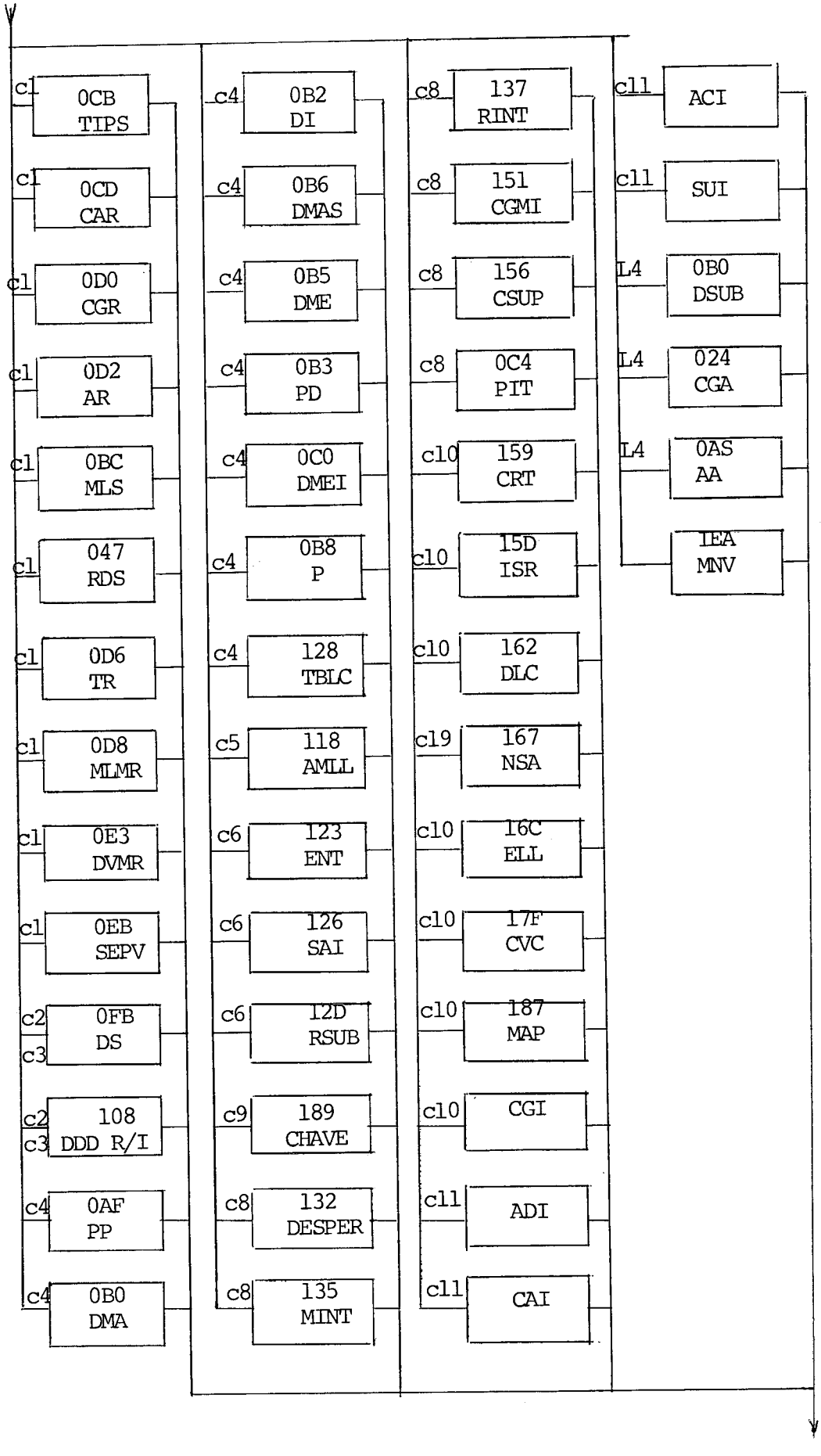
Essas operações são padronizadas para todas as instruções, permitindo assim uma facilidade na mudança ou ampliação do conjunto de instruções estabelecido, bastando para isso, configurar um micro-programa correspondente à execução da instrução modificada ou acrescentada.

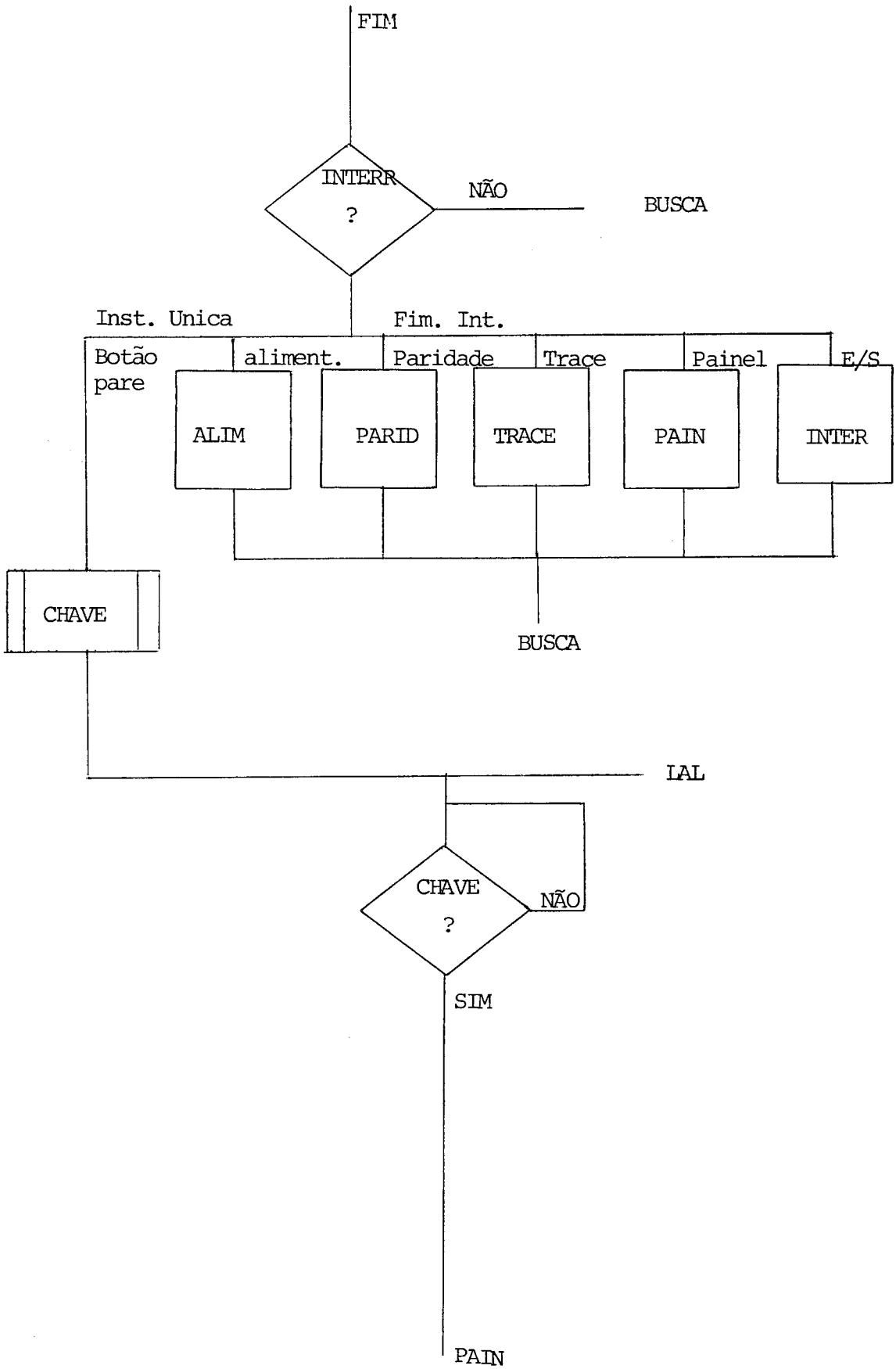
Dessa maneira torna-se possível implementar instruções especiais (no caso do computador ser adaptado a funções específicas) sem a necessidade de modificação de circuitos, desde que estas instruções não difiram dos formatos gerais das instruções do conjunto básico já existente.

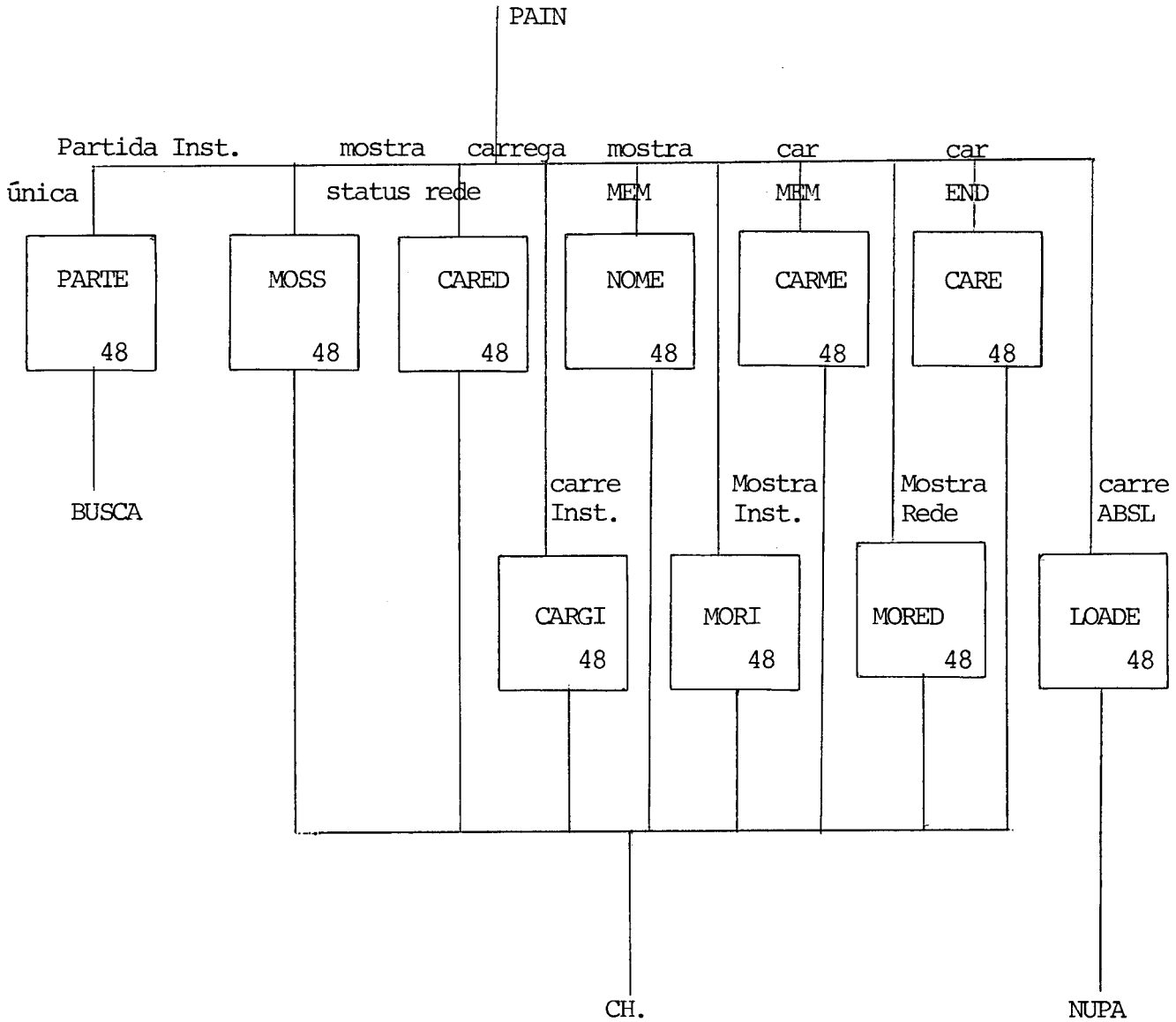
2.6.1 - Descrição Lógica da Micro-Programação das Instruções do G11

Toda a instrução passa por três fases durante a sua execução: Busca, cálculo de endereço efetivo e execução. O fluxograma seguinte representa a microprogramação das instruções.


BUSCA
BUSCA
FOR1
I_{10}=0
I_{10}=1
C11
L2,L3
C1,C2,C7,C10
L1,L4
C4,C3
C5,C6
C8,C9
C11
MO
DEP
PVES
PRVI
CURT
IMED
FOR2.$\overline{EEX}$
L1,L2
L1 (byte)
L1
ME
BE
ARM
FOR2.$\overline{EEX}$
FOR2.$\overline{EEX}$
AD
MLS
RDS
CG
CL
E
OU
OUX
ADM
CGE
SU
CA
IB
CGB
AB
CLB
CAB
EB
OUB
OUXB
FIM


c1 0CB TIPS
c1 0CD CAR
c1 0D0 CGR
c1 0D2 AR
c1 0BC MLS
c1 047 RDS
c1 0D6 TR
c1 0D8 MLMR
c1 0E3 DVMR
c1 0EB SEPV
c2 c3 0FB DS
c2 c3 108 DDD R/I
c4 0AF PP
c4 0B0 DMA
c4 0B2 DI
c4 0B6 DMAS
c4 0B5 DME
c4 0B3 PD
c4 0C0 DMEI
c4 0B8 P
c4 128 TBLC
c5 118 AMLL
c6 123 ENT
c6 126 SAI
c6 12D RSUB
c9 189 CHAVE
c8 132 DESPER
c8 135 MINT
c8 137 RINT
c8 151 CGMI
c8 156 CSUP
c8 0C4 PIT
c10 159 CRT
c10 15D ISR
c10 162 DLC
c19 167 NSA
c10 16C ELL
c10 17F CVC
c10 187 MAP
c10 CGI
c11 ADI
c11 CAI
c11 ACI
c11 SUI
L4 0B0 DSUB
L4 024 CGA
L4 0AS AA
IEA MNV


FIM
INTERR
?
NÃO
BUSCA
Inst. Unica
Fim. Int.
Botão
pare
aliment.
Paridade
Trace
Painel
E/S
ALIM
PARID
TRACE
PAIN
INTER
CHAVE
BUSCA
LAL
CHAVE
?
NÃO
SIM
PAIN


PAIN
Partida Inst.
única
mostra
status rede
carrega
mostra
MEM
car
MEM
car
END
PARTE
48
MOSS
48
CARED
48
NOME
48
CARME
48
CARE
48
BUSCA
carre
Inst.
Mostra
Inst.
Mostra
Rede
carre
ABSL
CARGI
48
MORI
48
MORED
48
LOADE
48
CH.
NUPA

Descrição Lógica das Micro-Subrotinas:

BUSCA, ME, BE, PEG, INIC, INICD, MGUA, MO, DEP, PVES, PRVI , CURT, IMED, GUAB.

Micro-Rotina BUSCA:

RØ ← endereço de próxima instrução

se instrução é curta: D ← I ∩ $(ØØFF)_{16}$; vai para FOR1.

se instrução é longa: procede leitura da próxima palavra; vai para FOR1.

Micro-Rotina ME:

Para ME = ØØ |Ø1|1Ø

D ← segundo operando;

SP1 ← endereço do segundo operando (já indexado se for o caso);

P ← Rj;

vai para EEX.FOR2.

Micro-Rotina BE:

Para BE = 00|01|10

D ← primeiro operando;

P ← SP2 ← segundo byte com sinal expandido;

SP1 ← endereço da palavra do byte;

vai para EEX.FOR2

Micro-Rotina PEG:

Se não indireto (M0=00): D ← (R_j); vai para fim.

Se indireto (M0≠00): D ← C((B) + (Q)); vai para fim.

Micro-Rotina INIC:

SPØ ← D;

INICD: CALL PEG;

SP3 ← Q;

Se curta: Q ← R_i; vai para fim.

Se longa: Q ← D; D ← SPØ; vai para fim.

OBS: A micro-rotina INICD é uma parte da micro-rotina INIC.

Micro-Rotina MGUA:

Se direto: P ← (Q); R_j ← D; vai para fim.

Se indireto: P ←(Q); vai para GUAB.

Micro-Rotina M0:

Se formatos C1, C2, C7, C10:

Se M0=00: D ← I ∩ $(00FF)_{16}$; vai para FOR2.

Se M0=01: Q ← endereço do 1º operando (conteúdo de R_j);

D ← I ∩ $(00FF)_{16}$; vai para FOR2.

Se M0=10: $R_j \leftarrow R_j + 1$; Q ← endereço do 1º operando (conteúdo de R_j);

D ← I ∩ $(00FF)_{16}$; vai para FOR2.

Se M0=11: Q ← endereço do 1º operando (conteúdo de R_j);

$R_j \leftarrow (R_j) - 1$; D ← I ∩ $(00FF)_{16}$; vai para FOR2.

Se formato L1, L4:

Se M0=00: D ← segunda palavra da inst;

Se M0=01: Q ← endereço do primeiro operando (conteúdo de R_j);

D ← segunda palavra da inst;

Se M0=10: $R_j \leftarrow R_j + 1$;

Q ← endereço do primeiro operando;

D ← segunda palavra da instrução;

Se M0=11: Q ← endereço do primeiro operando;

$R_j \leftarrow R_j - 1$;

D ← segunda palavra da instrução;

vai para FOR2.

Micro-Rotina DEP:

Se formato C4 (PULA):

SPØ ← RE;

D ← I∩$(00FF)_{16}$;

vai para FOR2.

Se formato L4 (DESVIO):

SPØ ← RE;

D ← segunda palavra da instrução;

vai para FOR2.

Micro-Rotina PVES:

Se modo supervisor:

D ← I ∩ $(00FF)_{16}$;

Q ← D com bit 7 expandido;

vai para FOR2.

Se modo Usuário:

vai para PRIV.

Micro-Rotina PRVI:

P ← 001F;

se instrução não é CSUP ou PIT: se usuário: vai para PRIV.

Se modo supervisor ou I = (CSUP ou PIT): Q ← D com bit 7 expandido;

vai para FOR2.

Micro-Rotina CURT:

(Cll com $I_{10} = 0$)

P ← (BL) + (D);

E ← (P) + (B);

Lê,

D ← operando;

vai para FOR2.

Micro-Rotina IMED:

Se Cll com $I_{10} = 1$ ou C3: D ← I∩$(00FF)_{16}$;

vai para FOR2.

Se L2: D ← segunda palavra da instrução

vai para FOR2.

Micro-Rotina GUAB:

Se P > LD → Proteção;

E ← B + P;

ESC;

vai para fim.

2.7 - Relação dos Pontos de Falha

1 - Unidade de Controle:

1.1 - Mapeador:

1.1.1 - Mapeador de Instruções:

1.1.1.1 - Mapeador 1:

1.1.1.1.1 - Quando instrução curta sem M0

1.1.1.1.2 - Quando instrução curta com M0

1.1.1.1.3 - Quanto instrução longa sem M0

1.1.1.2 - Mapeador 2

As suas duas alternativas já foram ativadas nos testes do mapeador 1, apenas não se combinou o cálculo de M0 e ME.

1.1.1.2.1 - Ativa o mapeador 2 para calcular ME após ter calculado M0

1.1.2 - Mapeador de painel:

1.1.2.1 - Função ARM

1.1.2.2 - Função MOS

1.1.2.3 - Função S

1.1.2.4 - Função M

1.1.2.5 - Função E

1.1.2.6 - Função R

1.1.2.8 - Função CAR

1.1.2.9 - Função PART

1.1.3 - Mapeador de interrupções:

1.1.3.1 - Canais

1.1.3.2 - Falha de alimentação

1.1.3.3 - Erro de paridade

1.1.3.4 - Proteção de memória

1.1.3.5 - Testes

1.1.3.6 - Botão PARE

1.1.3.7 - Botão PREP

1.1.3.8 - Botão INTER

1.2 - Seleção para (EC)

1.2.1 - Seleção das duas saídas do mapeador para (EC)

1.2.2 - Seleção do endereço de retorno de uma micro-subrotina para o micro-programa principal;

1.2.3 - Endereço contido nos bits de 19 a 30 nas micro-instruções de pulo (pulos dentro dos micro-programas ou para outras micro-rotinas).

1.2.4 - Endereço proveniente do mapeador de interrupção

1.2.5 - Endereço proveniente do mapeador do painel

1.2.6 - Endereço da micro-instrução seguinte proveniente do somador de + 1;

1.2.7 - Saída da Unidade Lógica e Aritmética

1.3 - Memória de Controle

1.3.1 - Micro-programa de Busca

1.3.2 - Micro-subrotinas de cálculo de endereço efetivo M0

1.3.3 - Micro-subrotinas de cálculo de endereço efetivo ME

1.3.4 - Micro-subrotinas de preparação de operando DEP

1.3.5 - Micro-subrotinas de preparação de operando PVES

1.3.6 - Micro-subrotinas de preparação de operando CURT

1.3.7 - Micro-subrotinas de preparação de operando IMED

1.3.8 - Micro-subrotinas de cálculo de endereço efetivo BE

1.3.9 - Micro-subrotina de preparação de operando PRVI

1.3.10 - Micro-subrotina auxiliar PEG

1.3.11 - Micro-subrotina auxiliar INIC

1.3.12 - Micro-subrotina auxiliar GUAB

1.3.13 - Micro-subrotina auxiliar MGUA

1.3.14 - Micro-programas de funções das instruções (68 módulos)

1.3.15 - Micro-programas de tratamento de interrupções

1.3.16 - Micro-programas do painel

1.3.17 - Carregador absoluto

1.4 - Registrador de endereços de controle (EC)

1.5 - Registrador de dados do controle (DC)

1.6 - Registradores de endereço de retorno de micro-subrotinas (SØ e S1)

1.7 - Decodificadores

1.8 - Somador de "+1"

1.9 - Contador (CNTD)

1.10 - Lógica de testes de condições

2 - Unidade de Movimento de Dados:

2.1 - Registrador de Estado

2.2 - Registradores de propósito geral

2.3 - Base local

2.4 - Base de programa

2.5 - Limite de programa

2.6 - Registrador índice de pilha de contexto

2.7 - Registradores de rascunho (4)

2.8 - Registrador de dados D

2.9 - Registradores deslocadores P e Q

2.10 - Registrador I

2.11 - Registrador de endereços E

2.12 - Painel

2.13 - Unidade lógica e aritmética

3 - Conjunto de Instruções do Processador G11

CAPÍTULO III

TESTES DOS PONTOS DE FALHAS DO PROCESSADOR DE INSTRUÇÕES DO MINI-COMPUTADOR G11

Os testes tem dois objetivos principais: detecção e diagnóstico com reparo da falha. Ao mesmo tempo seu uso deve ser facilitado de forma a auxiliar o operador de teste particularmente na fase de manutenção quando é necessário devolvera máquina funcionando no tempo mais rápido. Quanto maior for a substituição de operações manuais por operações automáticas mais rápido se processa a manutenção. A implementação do teste é desenvolvida levando em conta este aspecto.

Geralmente o teste de manutenção diagnostica a placa com defeito, no caso do G11 seria indicado o defeito na Unidade de Controle ou Unidade de Movimento de Dados. No entanto este teste está sendo projetado para as fases de desenvolvimento, produção, implementação e manutenção, o que exige uma indicação mais minuciosa do defeito. O ponto defeituoso pode ser um daqueles relacionados na seção 2.7 do capítulo anterior. Alguns pontos são estendidos no caso de haver necessidade e recursos disponíveis enquanto que outros são reunidos por falta de recursos de teste. O instrumento básico de teste é um programa formado por instruções do próprio conjunto de instruções do processador do mini-computador G11. O uso deste instrumento tomado como básico, exige cuidado especial para que o teste tenha capacidade de identificar pelo menos uma falha quando várias falhas ocorrem

simultaneamente. Esse ponto defeituoso pode ser então substituído ou reparado e o teste prosseguir com a finalidade de identificar as restantes da mesma forma que a primeira falha, conforme sugere Breuer et al |1|. Seguindo esta filosofia, se n falhas ocorrem simultaneamente, no máximo n iterações são necessárias para reparar as n falhas. O objetivo do projeto é obter um sistema que tenha a capacidade de ser diagnosticável com reparo das n falhas relacionadas na seção 2.7. Como foi observado anteriormente, esse número pode crescer ou decrescer de acordo com os obstáculos existentes. Na próxima seção, serão relacionados os testes necessários para cada ponto e os recursos que cada teste de cada ponto exige, recursos estes que precisam estar livres de erro.

3.1 - Relação dos Testes para cada Ponto Sujeito a Falha

Para cada ponto deve ser feito um teste completo*. Cada teste utiliza como recursos outros pontos que devem estar livres de erro para que o ponto em questão seja diagnósticado. Algumas medidas de precaução podem reduzir a frequência de fracassos nos testes que dependem de recursos que ainda não foram verificados. Uma das medidas é a ordem de realização dos testes de acordo com a quantidade de pontos envolvidos. Outra medida é a seleção de dois testes para cada ponto que tenham como característica o mínimo de coincidência entre os pontos utilizados.

---

* ver definição de teste completo no capítulo 4.

Relação dos testes na ordem em que os pontos aparecem na seção 2.7.

1 - Testes da Unidade de Controle

1.1 - Testes do mapeador

1.1.1 - Testes do mapeador de instruções

1.1.1.1 - Testes do mapeador 1

1.1.1.1.1 - Quando instrução curta sem M0

Testes do ponto 1.1.1.1.1:

1 - Execução de uma instrução curta pertencente ao formato C11 com bit 10 de I = 0.

Instrução "AC R, E"

Pontos ativados quando a instrução AC é executada:

- Micro-subrotinas da memória de controle: BUSCA; CURT; funções;
- Unidade de Controle;
- UMD;

2 - Execução de uma instrução curta pertencente ao formato C11 com bit 10 de I = 1

Instrução "CGI R, I"

Pontos ativados quando à instrução CGI é executada:

- Micro-subrotinas de memória de controle: BUSCA, IMED; funções;
- Unidade de Controle

- UMD

Pontos distintos entre os dois testes selecionados para o ponto 1.1.1.1.1.

1 - Função: O primeiro teste usa função AC

O segundo teste usa função CGI

2 - Micro-subrotinas de preparação de operando:

O primeiro teste usa CURT e o segundo usa IMED.

1.1.1.1.2 - Quando instrução curta com M0

1 - Execução de uma instrução curta pertencente ao formato C1 variando o campo M0 da instrução com todos os valores de $(00)_2$ a $(11)_2$.

Instrução "AR R, RR"

Pontos ativados quando a instrução AR é executada: Micro-subrotinas da memória de controle: BUSCA, M0; funções:

- Unidade de Controle
- UMD

2 - Execução de uma instrução curta pertencente ao formato C10 variando o campo M0 da instrução com todos os valores de $(00)_2$ a $(11)_2$.

Instrução CGMU RR

Pontos ativados quando a instrução CGMU é executada: Micro-subrotinas da memória de controle: BUSCA; M0; funções:

- Unidade de Controle
- UMD

Pontos distintos entre os dois testes selecionados para o ponto 1.1.1.1.2:

1 - Funções: O primeiro teste usa função AR

O segundo teste usa função CRT

1.1.1.1.3: Quando instrução longa sem M0

1 - Execução de uma instrução longa pertencente ao formato L2 variando o campo ME da instrução com todos os valores de $(00)_2$ a $(11)_2$.

Instrução "AI I, RM"

Pontos ativados quando a instrução "AI I, RM" é executada: Micro-subrotinas da memória de controle: BUSCA; ME; função de arm

- Unidade de Controle
- UMD

2 - Execução de uma instrução longa pertencente ao formato L3.

Instrução "DI RM"

Pontos ativados quando a instrução DI é executada: Micro-subrotinas da memória de controle: BUSCA; DEP; função DI

- Unidade de Controle
- UMD

Pontos distintis entre os dois testes selecionados para o ponto 1.1.1.1.3:

1 - Função: O primeiro teste ativa função de armazenamento
O segundo teste ativa função DI

2 - Micro-subrotinas de preparação de operando:

O primeiro teste ativa ME e o segundo ativa DEP.

1.1.1.2 - Teste do mapeador 2 (combinação de M0 e ME)

1 - Execução de uma instrução longa pertencente ao formato Ll combinando todas as variações dos campos M0 e ME de instrução.

Instrução "A RR, RM"

Pontos ativados quando a instrução A é executada: Micro-subrotinas da memória de controle: BUSCA; M0; ME; função:
- Unidade de controle
- UMD

2 - Execução de uma instrução longa pertencente ao formato Ll combinando todas as variações dos campos M0 e ME da instrução.

Instrução "CG RR, RM"

Pontos ativados quando a instrução CG é executada:

- Micro-subrotinas de memória de controle: BUSCA; M0; ME ; Função.
- Unidade de Controle
- UMD

Pontos distintos entre os dois testes relacionados para o ponto 1.1.1.2:

1 - Funções: O primeiro teste ativa função de armazenamento
O segundo teste ativa função de carregamento

1.1.2 - Testes do mapeador de painel

1.1.2.1 - Testes da função ARM, realizados através de operação manual da função ARM.

Pontos ativados quando a operação manual ARM é realizada:

- Micro-subrotinas de memória de controle: PAIN
- Unidade de Controle
- UMD

1.1.2.2 - Testes da função MOS, realizados através de operação manual da função MOS.

Pontos ativados quando a operação manual MOS é realizada:

- Micro-subrotina de memória de controle: PAIN
- Unidade de Controle
- UMD

1.1.2.3 - Testes da função S (mostra reg. de estado), realizados através de operações manuais de ARM, MOS, EX (de instrução Armazenada em reg. I).

Pontos ativados quando a operação manual da função S é realizada:

- Micro-subrotina de memória de controle: PAIN
- Unidade de controle
- UMD

1.1.2.4 - Testes da função M, realizados através de operações manuais de ARM, MOS

Pontos ativados quando a operação manual da função S é realizada:

- Micro-subrotina de memória de controle: PAIN
- Unidade de controle
- UMD

1.1.3 - Testes do mapeador de interrupção

1.1.3.1 - Testes de interrupção por canal de E/S; não são realizados no sistema de testes do processador.

1.1.3.2 - Teste de interrupção por falha de alimentação.

1.1.3.3 - Testes de interrupção por erro de paridade; são realizados no sistema de testes de memória.

1.1.3.4 - Testes de interrupção por proteção de memória, realizados através de execução de instruções de desvio para endereços fora dos limites do programa.

Pontos ativados quando interrupção por falha de alimentação ou por proteção de memória ocorre:

- Micro-subrotinas da memória de controle
- Unidade de Controle
- UMD

1.1.3.5 - Testes de interrupção por rastreamento, realizado através de operação manual de rastreamento quando máquina está processando.

Pontos ativados quando interrupção por rastreamento é realizada: levando em conta que o rastreamento depende de um programa em execução, a ativação deste ponto depende da maioria dos recursos da máquina e é aconselhável sua verificação, somente depois do programa de teste ter passado um número razoável de vezes.

1.1.3.6 - Teste do botão PARE, coincide com teste da função de painel PARE. É realizado através de operação manual.

Pontos ativados quando a operação manual da função PARE é realizada:

- Botão PARE; função de painel PARE
- Unidade de Controle
- UMD

1.1.3.7 - Teste do botão PREP, coincide com teste da função de painel PREP. É realizado através de operação manual.

Pontos ativados quando a operação manual da função PREP é realizada:

- Botão PREP; função de painel PREP
- Unidade de Controle
- UMD

1.1.3.8 - Teste do botão INTER, coincide com teste da função de painel INTER. É realizado através de operação manual.

Pontos ativados quando a operação manual da função INTER é realizada:

- Botão INTER; Função de painel INTER
- Unidade de Controle
- UMD

1.2 - Testes de seleção para (EC)

1.2.1 - Testes de seleção das duas saídas do mapeador de instruções através da aplicação dos testes selecionados para o mapeador de instruções.

1.2.2 - Teste da seleção do endereço de retorno de uma micro-subrotina para o micro-programa principal;

A observação da estrutura da micro-programação mostra que o uso de micro-subrotinas é intenso, a execução de cada instrução vai ativar no mínimo três micro-subrotinas principais (BUSCA, prepara operandos, execução da função) , portanto este ponto fica confundido com o restante dos outros pontos, utilizados por todas as instruções e quase todas as funções, pertencentes a unidade de controle.

1.2.3 - Testes da seleção do endereço contido nos bits de 19 a 30 nas micro-instruções de pulo (pulos dentro dos micro-programas ou para outras micro-subrotinas).

Este ponto apresenta o mesmo problema apontado no teste do ponto 1.2.2; fica impossível identificar através da instrução a origem do defeito, pois todas as instruções vão precisar ativar micro-instruções de pulo nas suas execuções.

1.2.4 - Teste da seleção do endereço proveniente do mapeador de interrupção.

Este ponto deve estar agregado aos pontos de interrupção do mapeador: canais, falha de alimentação, instrução inválida, etc ...

É impossível distinguir a falha na seleção e nas demais.

1.2.5 - Teste da seleção do endereço proveniente do mapeador do painel.

Este ponto deve estar agregado aos pontos de teste do mapeador do painel, pela mesma razão apontada no teste 1.2.4.

1.2.6 - Teste da seleção do endereço da micro-instrução seguinte proveniente do somador de +1;

Este ponto vai ser comum a todas as instruções ele é um recurso básico, se por acaso nenhuma instrução funcionar, um dos pontos com defeito é este. Não há meio de cercar este ponto.

1.2.7 - Teste da seleção da saída da unidade lógica e aritmética, através da ativação de todas as funções lógicas e aritméticas.

1.3 - Teste da memória de controle que é ativado, i.e., das micro-subrotinas armazenados;

1.3.1 - Testes da micro-subrotina de BUSCA;

1.3.1.1 - Testes do micro-programa de BUSCA de instrução curta;

- Execução de uma instrução curta pertencente ao formato C1 ; instrução "AR R, RR"

Pontos ativados quando a instrução AR é executada:

- Micro-subrotinas da memória de controle: BUSCA, M0; função de ARM
- Unidade de Controle
- UMD

2 - Execução de uma instrução curta pertencente ao formato C11 ; Instrução "CGI R, I"

Pontos ativados da memória de controle: BUSCA; IMED; função carrega

- Unidade de Controle
- UMD

1.3.1.2 - Teste do micro-programa de Busca de instrução longa;

1 - Execução de uma instrução longa pertencente ao formato Ll ; Instrução "A RR, RM"

Pontos ativados da memória de controle: BUSCA, M0; ME, função de ARM

- Unidade de Controle
- UMD

2 - Execução de uma instrução longa pertencente ao formato Ll; Instrução "CG RR, RM"

Pontos ativados da memória de controle: BUSCA; M0; ME; função de CG

- Unidade de Controle
- UMD

1.3.2 - Testes das micro-subrotinas de cálculo de endereço efetivo M0.

Execução de duas instruções curtas pertencentes aos formatos Cl e C10, variando o campo M0 de $(00)_2$ a $(11)_2$.

1 - Instrução "AR R, RR"

2 - Instrução "CGMU RR"

Pontos ativados quando as instruções AR e CGMU são executadas:

- Micro-subrotinas da memória de controle: BUSCA, MO; funções
- Unidade de Controle
- UMD

Execução de duas instruções longas pertencentes aos formatos L1 e L4 fazendo variar os campos M0 e ME de $(00)_2$ a $(11)_2$.

1 - Instrução "A RR, RM" pertencente ao formato L1.

2 - Instrução "CGA RR, RM" pertencente ao formato L4

Pontos ativados quanto as instruções são executadas:

- Micro-subrotinas da memória de controle: BUSCA, MO, ME , funções
- Unidade de Controle
- UMD

1.3.3 - Testes das micro-subrotinas de cálculo de endereço efetivo ME;

Execução de duas instruções longas pertencentes aos formatos L1 e L4 fazendo variar os campos M0 e ME de $(00)_2$ a $(11)_2$;

1 - Instrução "A RR, RM" pertencente ao formato L1

2 - Instrução "AI RR, RM" pertencente ao formato L2

Pontos ativados quando as instruções são executadas:

- micro-subrotinas da memória de controle: BUSCA; M0; ME ; função
- Unidade de Controle
- UMD

1.3.4 - Testes da micro-subrotina de preparação de operando DEP;

Execução de duas instruções pertencentes aos formatos C4 e L4;

1 - Instrução "PI" pertencente ao formato C4

2 - Instrução "DI" pertencente ao formato L4

1.3.5 - Testes de micro-subrotina de preparação de operando PVES; este ponto não é testado neste sistema, é agregado ao sistema de testes de periféricos;

1.3.6 - Testes de micro-subrotina de preparação de operando CURT;

Execução de duas instruções pertencentes ao formato C11 (curto, quando I10 = Ø).

1 - Instrução "AC"

2 - Instrução "CGC"

1.3.7 - Testes de micro-subrotinas de preparação de operando IMED:

Execução de uma instrução pertencente ao formato C11 e outra instrução pertencente ao formato L2.

1 - Instrução "CGI"

2 - Instrução "AI"

1.3.8 - Teste de micro-subrotina de cálculo de endereço efetivo BE:

Execução de duas instruções pertencentes ao formato longo L1 variando o campo BE de $(00)_2$ e $(11)_2$.

1 - Instrução "AB"

2 - Instrução "CGB"

1.3.9 - Testes da micro-subrotina de preparação de operando PRVI.

OBS: Apenas as instruções PARE e CSUP serão testadas neste sistema.

1 - Instrução "PARE"

2 - Instrução "CSUP"

Os pontos relacionados anteriormente na seção 2.7 de 1.3.10 a 1.3.13 não merecem teste especial devido as instruções não possuirem capacidade de ativar estas micro-rotinas de forma aparente.

1.3.14 - Testes dos micro-programas de funções das instruções: Cada função lógica e aritmética será testada esgotando todas as situações.

1.3.15 - Testes dos micro-programas de tratamento de interrupções. Estes pontos estão sendo testados juntamente com o mapeador de interrupções e seleção. É impossível identificar cada etapa separadamente.

1.3.16 - Testes dos micro-programas do painel estes pontos estão sendo testados juntamente com funções do painel, mapeador, seleção, pelo mesmo motivo apontado em 1.3.15.

1.3.17 - Testes do carregador absoluto; após seu carregamento , alguns endereços selecionados com planejamento devem ser mostrados via painel.

1.4 - Registrador de endereços de controle (EC)

Este ponto é comum a todas as operações, seu mau funcionamento vai prejudicar toda a operação. Sua identificação automática é impossível.

Os pontos (1.5), (1.6), (1.7), (1.8), (1.9) e (1.10) apresentam as mesmas dificuldades apontadas para o ponto (1.4).

2 - Teste da Unidade de Movimento de dados;

Os testes dos pontos da unidade de movimento de dados serão realizados por um conjunto de operações manuais permitidas através do painel.

3 - Teste do conjunto de instruções

Os testes das instruções é realizado em duas partes. A primeira é a verificação intensiva das funções e a segunda a ativação de todas as instruções por uma situação apenas.

3.2 - Análise dos Testes Relacionados Quanto a Capacidade de Localização de Falhas

A observação dos testes do conjunto de pontos relacionados pertencentes a UCP e sujeitos a reparo quando apresentarem defeito, mostra que certos pontos tem características que merecem atenção especial.

Na análise foram encontradas quatro classes de pontos cujas características são relacionadas abaixo:

1 - Pontos que são ativados por qualquer operação manual ou automática. O mau funcionamento destes pontos compromete o funcionamento inteiro da UCP.

2 - Pontos que são ativados por qualquer operação automática. O mau funcionamento destes pontos compromete todas as execução das instruções.

3 - Pontos que são ativados pela mesma situação externa. Se um ou mais pontos desses apresenta mau funcionamento, é impossível localizar qual ou quais estão apresentando defeito.

4 - As instruções, cada instrução ativa uma série de pontos ao ser executada, podem ser consideradas como caminho de pontos, esses caminhos precisam ser testados, além disso os micro-programas de funções são particulares a uma pequena parcela de instruções.

Estas características sugerem uma nova relação de pontos e portanto uma nova organização do teste.

Nova Relação dos pontos baseada nas características apontadas:

Primeira classe de pontos: aqueles que são ativados por qualquer operação manual ou automática:

1 - Registradores de endereço de retorno de micro-subrotinas (SØ e S1).

2 - Seleção do endereço de retorno de uma micro-subrotina para o micro-programa principal.

3 - Seleção do endereço contido nos bits de 19 a 30 nas micro-instruções de pulo (pulos dentro dos micro-programas ou para outras micro-subrotinas).

4 - Seleção do endereço da micro-instrução seguinte proveniente do somador de +1.

5 - Somador de +1

6 - Parte comum da seleção

7 - Registrador de endereço de Controle (EC)

8 - Registrador de dados de controle (DC)

9 - Decodificadores

10 - Contador (CNTD)

11 - Lógica de testes de condições

12 - Seleção da saída de ULA e parte da ULA para operação manual

Segunda classe de pontos: aqueles que são ativados por qualquer operação automática. O mau funcionamento destes pontos compromete a execução das instruções. Não estão incluídos aqui os pontos relacionados no item anterior.

1 - Mapeador de instruções (parte comum a todas as instruções)

2 - Seleção das duas saídas do mapeador de instruções para (EC)

3 - Micro-programa de Busca

4 - Unidade de movimento de dados, parte da ULA para operação automática.

Terceira classe de pontos: aqueles que são ativados pela mesma situação externa. Situação provocada pelo programador (manual ou através de instrução). Se um ou mais pontos desses apresenta mau funcionamento, é impossível localizar a falha particular, apenas apontar o conjunto de possíveis pontos com falha.

1 - Situação externa: instrução curta sem M0

Pontos que são ativados apenas por esta situação externa

1. Parte do mapeador 1
2. Micro-subrotinas de preparação de operando CURT/IMED/PVES/PRVI/DEP (apenas pulos).

2 - Situação externa: instrução curta com M0

Pontos que são ativados apenas por esta situação externa

1. Parte do mapeador 1
2. Micro-subrotinas de cálculo de endereço efetivo M0

3 - Situação externa: instrução longa sem M0

Pontos que são ativados apenas por esta situação externa

1. Parte do mapeador 1
2. Micro-subrotinas de cálculo de endereço efetivo ME/DEP (desvios)/IMED.

4 - Situação externa: instrução longa com M0 e ME/BE

Pontos que são ativados apenas por esta situação externa

1. Parte do mapeador 2
2. Micro-subrotinas de cálculo de endereço efetivo M0
3. Micro-subrotinas de cálculo de endereço efetivo ME/BE

5 - Situação externa: pressão do botão ARM no painel

Pontos que são ativados apenas por esta situação

1. Mapeador de painel da função ARM
2. Micro-programa do painel que trata função ARM

6 - Situação externa: pressão do botão MOS no painel

Pontos que são ativados apenas por esta situação

1. Mapeador de painel da função MOS
2. Micro-programa do painel que trata função MOS

7 - Situação externa: pressão do botão S no painel

Pontos que são ativados apenas por esta situação

1. Mapeador de painel para S
2. Micro-programa do painel que trata S

8 - Situação externa: pressão do botão M no painel

Pontos que são ativados apenas por esta situação

1. Mapeador de painel para M
2. Micro-programa do painel que trata M

9 - Situação externa: pressão do botão E no painel

Pontos que são ativados apenas por esta situação

1. Mapeador de painel para E
2. Micro-programa de painel que trata E

10 - Situação externa: pressão do botão I no painel

Pontos que são ativados apenas por esta situação

1. Mapeador de painel para I
2. Micro-programa de painel que trata I

11 - Situação externa: pressão do botão R no painel

Pontos que são ativados apenas por esta situação

1. Mapeador do painel para R
2. Micro-programas do painel que trata R

12 - Situação externa: pressão do botão CAR

Pontos que são ativados apenas por esta situação

1. Mapeador de painel para CAR
2. Micro-programa de painel que trata CAR

13 - Situação externa: pressão do botão PART

Pontos que são ativados apenas por esta situação

1. Mapeador de painel para PART
2. Micro-programa de painel para PART

14 - Situação externa: qualquer operação com o painel

Pontos que são ativados

1. Mapeador de painel (parte comum ativada por todas as situações que ativam as partes anteriores).
2. Seleção do endereço proveniente do mapeador de painel.
3. Parte comum dos micro-programas de tratamento das funções de painel
4. Registrador de painel da U.M.D.

15 - Situação externa: ocorrência de E/S

Pontos ativados apenas por esta situação

1. Mapeador de interrupção por CANAIS
2. Micro-programa que trata de interrupção por CANAIS

16 - Situação externa: ocorrência de falha de alimentação

Pontos ativados apenas por esta situação

1. Mapeador de interrupção por falha de alimentação
2. Micro-programa que trata de interrupção por falha de alimentação.

17 - Situação externa: ocorrência de erro de paridade

Pontos ativados apenas por esta situação

1. Mapeador de interrupção por erro de paridade
2. Micro-programa que trata de interrupção por erro de paridade.

18 - Situação externa: tentativa de acesso a endereço fora dos limites permitidos.

1. Mapeador de interrupção por proteção de memória.

2. Micro-programa que trata de interrupção por proteção de memória.

19 - Situação externa: interrupção de rastreamento

Pontos ativados apenas por esta situação

1. Mapeador de interrupção por rastreamento
2. Micro-programa que trata de interrupção por rastreamento

20 - Situação externa: pressão do botão PARE no painel

Pontos ativados apenas por esta situação

1. Mapeador de interrupção por PARE
2. Micro-programa que trata de interrupção por PARE

21 - Situação externa: pressão do botão PREP no painel

Pontos ativados apenas por esta situação

1. Mapeador de interrupção por PREP
2. Micro-programa que trata de interrupção por PREP

22 - Situação externa: pressão do botão INTER no painel

Pontos ativados apenas por esta situação

1. Mapeador de interrupção pelo botão INTER
2. Micro-programa que trata de interrupção pelo botão INTER

23 - Situação externa: qualquer interrupção

Pontos ativados em qualquer interrupção

1. Mapeador de interrupção (parte comum que é ativada pelas situações que ativam os pontos anteriores).
2. Seleção de endereço proveniente do mapeador de interrupção.

3. Micro-subrotina de tratamento de interrupções (parte comum que é ativada pelas situações que ativam os pontos anteriores).

24 - Carregador absoluto

Quarta classe de pontos:

1 - Todas as micro-subrotinas de funções

2 - Todas as instruções

Nesta última relação, os pontos ainda podem não ser localizados ou não ser reparáveis, mas a indicação do grupo de falhas dá orientação para o reparo ou substituição.

A observação do comportamento da UCP para cada falha em cada classe orienta a organização do teste.

Qualquer falha na primeira classe compromete o funcionamento completo da UCP. Se nada funciona, operações por painel, instruções, algum, alguns ou todos pontos pertencentes a primeira classe estão com defeito.

Na segunda classe, é necessário observar que, a unidade de movimento de dados não é completamente utilizada por cada instrução, como porém cada instrução utiliza sempre alguns elementos e outros elementos são utilizados de forma variada, vai ser necessário testar estes elementos, e já que o pon-

tos das outras classes precisam da unidade porque combinam instruções de forma que toda a unidade é ativada, é aconselhável testar toda a unidade neste momento.

O mapeador de instruções e a seleção das duas saídas do mapeador de instrução podem ser testados com operação manual, através de um armazenamento por painel de uma instrução em I.

O micro-programa de Busca é testado por operação automática. Se qualquer instrução armazenada não funcionar, o erro deve ser neste ponto.

Os pontos da terceira classe de 5 a 14, 16 , 18 a 23 e 24 são verificados por intervenção do operador.

O ponto 17 da terceira classe não é testado neste sistema, sua verificação é feita no teste de memória.

O ponto 15 da terceira classe não é testado neste sistema, sua verificação é feita no teste de periférico.

Os pontos de 1 a 4 são testados logo depois dos testes através de programa, que exigem intervenção do operador, são os testes das instruções usadas como auxiliares no res-

tante do programa de teste*.

As micro-subrotinas de funções pertencentes a quarta classe são testadas através de execução das instruções mais simples, em termos de preparação de operando, que as ativam. Duas instruções são escolhidas para cada ponto.

As instruções restantes não precisam ser verificadas em uma ordem determinada.

A seguir é apresentada a descrição dos testes, os pontos podem sofrer nova reclassificação devido ainda a questão de localização.

3.3 - Descrição dos Testes

Introdução:

Três elementos são acrescentados aqui:

1. Instrumentos de testes dos pontos;

2 - Lógica de implementação de cada teste;

3 - Padrões de teste;

---

*São as instruções básicas de funcionamento de UCP tais como PARA, EX, LC, LRE, CAR, carrega endereço, ..., e que permitem a comunicação com o operador para a execução do programa. É necessário que estas instruções estejam livres de erro. Podem ser testadas manualmente ou através de uma "giga de testes" (micro-processador com programas para executar as instruções que seriam executadas manualmente).

Os instrumentos de teste são funções de painel e instruções que podem participar do teste de um determinado ponto somente se estiverem "livres de erros", i.e., já tiverem sido testadas.

A lógica de implementação de teste do ponto varia com a disponibilidade de instrumentos "livres de erro". E são apresentadas ao longo das descrições de cada teste.

Os padrões de teste dependem diretamente de dois fatores:

1 - Recomendação de projeto da máquina;

2 - Lógica de implementação da micro-programação para as funções lógicas a aritméticas;

A arquitetura do minicomputador G11 apresenta todos os registradores da Unidade Lógica e Aritmética divididos em placas, o byte esquerdo fica na placa 1 e o byte direito na placa 2, esta característica exige um cuidado especial na transmissão de bit entre os bits 7 e 8.

Outras recomendações são: mudança de cada bit de Ø para 1 e de 1 para Ø; influência de vizinhança de bits ; mudança de sinal (transbordo).

A análise do algoritmo empregado em cada função indica a faixa de valores em que uma entrada se enquadra pa-

ra ativar um caminho lógico no algoritmo. Levantando todos os caminhos do algoritmo, consequentemente todas as faixas, com segurança pode se selecionar um dado para cada caminho.

Testes da Primeira Classe:

Os pontos desta classe precisam ser testados por instrumentos que foram classificados nas segunda e terceira classes, as funções do painel, a ativação dos caminhos de algumas instruções, a UMD. Esses pontos que não pertencem a primeira classe, por falta de recursos para os testes vão tornar-se da primeira classe. São todos os pontos da segunda classe e os pontos 5, 6, 7, 8, 9, 10, 11, 13, 14, 21, 22, 23, 24, 25 da terceira classe.

Os testes dos pontos pertencentes a nova classe 1 são testados nas etapas seguintes.

1 - Teste da rede de registradores, reg. I e reg. E de UMD com funções ARM e MOS;

2 - Teste M(E);

3 - Mapeador de instruções, seleção das duas saídas do mapeador; micro-programa de busca e função PART; ULA;

4 - Teste de RE através de uma instrução que provoque mudança de condições;

5 - Teste dos registradores P e Q;

6 - Teste da função PREP;

7 - Teste da função INTER;

8 - Carregador absoluto, CAR

Convenção:

Di: configuração binária do padrão de dados i.

Ej: configuração binária do endereço j;

Rede de registradores: RØ, Rl, ..., Rl5

E: Registrador de endereço

M(Ej): palavra da memória cujo endereço é Ej.

Relação dos padrões de dados Di, i variando de 1 a 4

$D_1 = (0000\ \ 0000\ \ 0000\ \ 0000)_2$

$D_2 = (1111\ \ 1111\ \ 1111\ \ 1111)_2$

$D_3 = (0101\ \ 0101\ \ 0101\ \ 0101)_2$

$D_4 = (1010\ \ 1010\ \ 1010\ \ 1010)_2$

Relação dos padrões de endereço Ej, j variando de 1 a 19

$E_1 = (0000\ \ 0000\ \ 0000\ \ 0000)_2$

$E_2 = (0000\ \ 1111\ \ 1111\ \ 1111)_2$

$E_3 = (0000\ \ 0101\ \ 0101\ \ 0101)_2$

$E_4 = (0000\ \ 1010\ \ 1010\ \ 1010)_2$

$E_5 = (0001\ \ 1111\ \ 1111\ \ 1111)_2$

- Limite superior a 8 K palavras na memória

- Limite inferior do segundo módulo de 8 K da memória

$E_6 = (0010\ \ 0000\ \ 0000\ \ 0000)_2$

$E_7 = (0011 \quad 1111 \quad 1111 \quad 1111)_2$

- Limite superior de 16 K palavras na memória

- Limite inferior do terceiro módulo de 8 K da memória

$E_8 = (0100 \quad 0000 \quad 0000 \quad 0000)_2$

$E_9 = (0101 \quad 1111 \quad 1111 \quad 1111)_2$

- Limite superior de 24 K palavras na memória

- Limite inferior do quarto módulo de 8 K na memória

$E_{10} = (0110 \quad 0000 \quad 0000 \quad 0000)_2$

$E11 = (0111 \quad 1111 \quad 1111 \quad 1111)_2$

- Limite superior de 32 K palavras na memória

⋮

- Limite inferior do oitavo módulo de 8 K na memória

$E_{18} = (1110 \quad 0000 \quad 0000 \quad 0000)_2$

$E_{19} = (1111 \quad 1111 \quad 1111 \quad 1111)_2$

- Limite superior de 64 K palavras na memória

Etapa 1:

Fluxo de operações manuais

i ← 1

Chaves ← Di

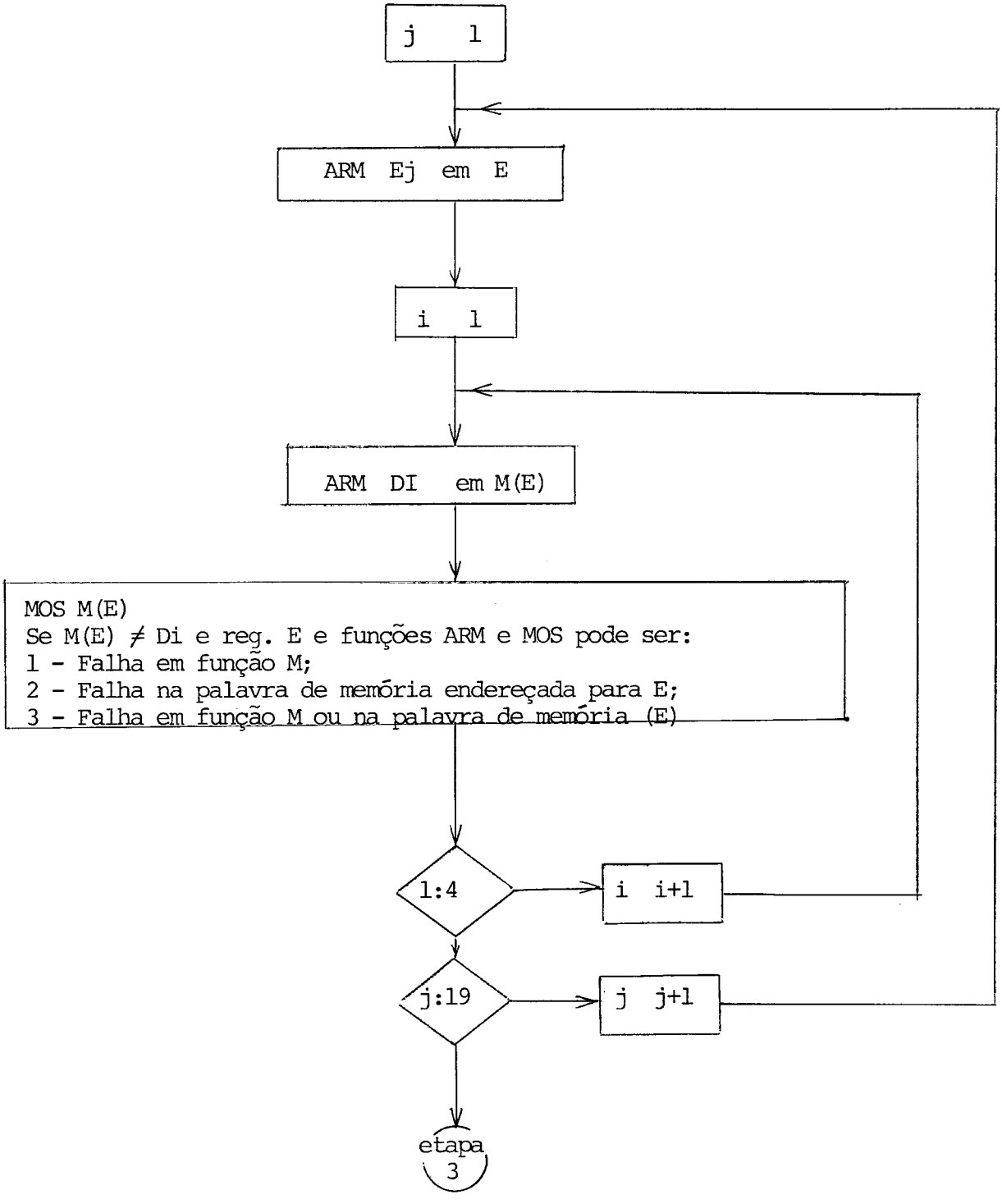
ARM em rede de registradores, reg. I e Reg. E

MOS rede de registradores, reg I e reg. E se todos registradores ≠ Di: ARM ou MOS tem falha - se algum ou alguns registradores ≠ Di: apenas o reg. tem falha

1 : 4 = etapa (2) i = 1 +1

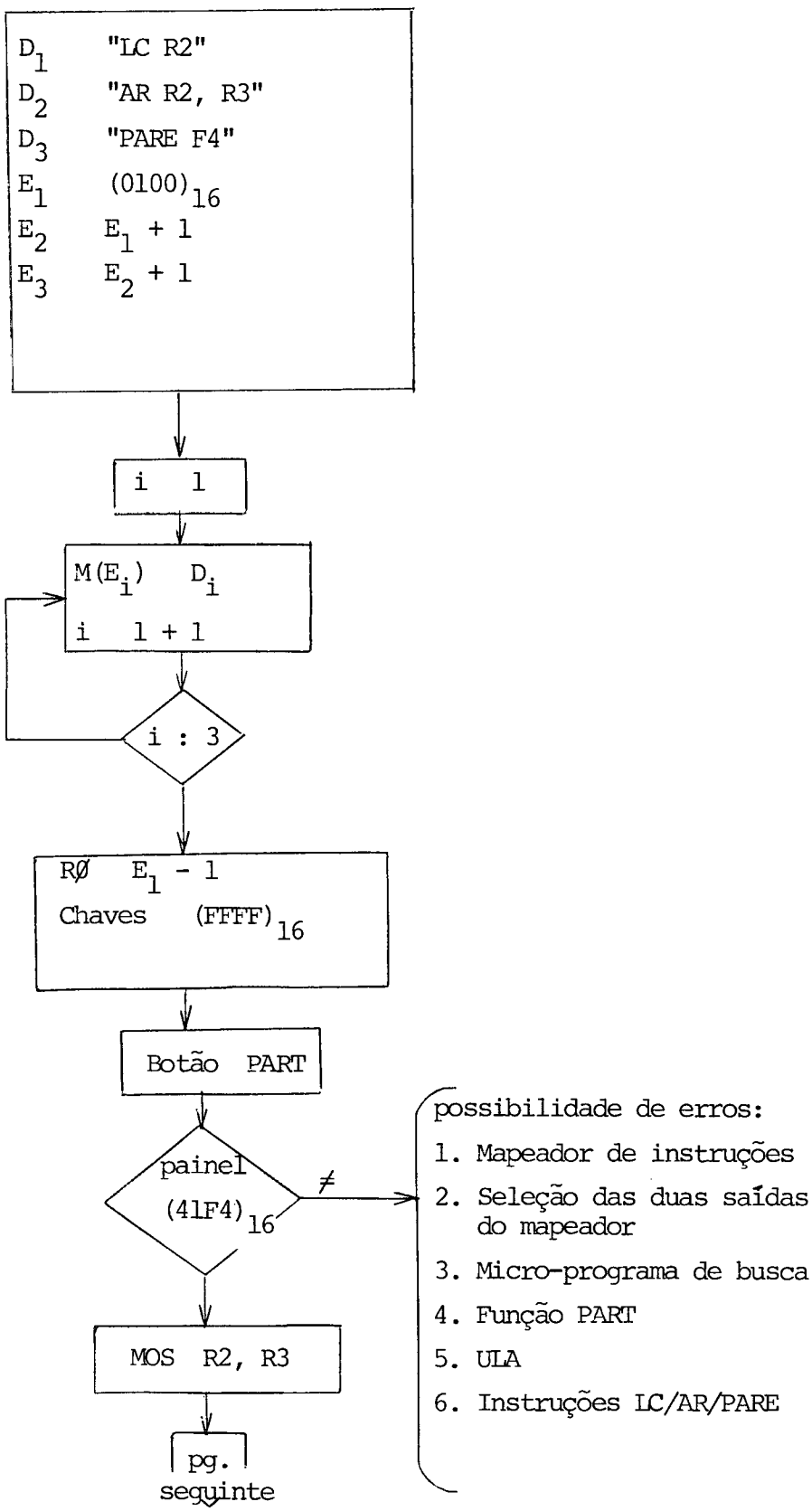
Etapa 2:

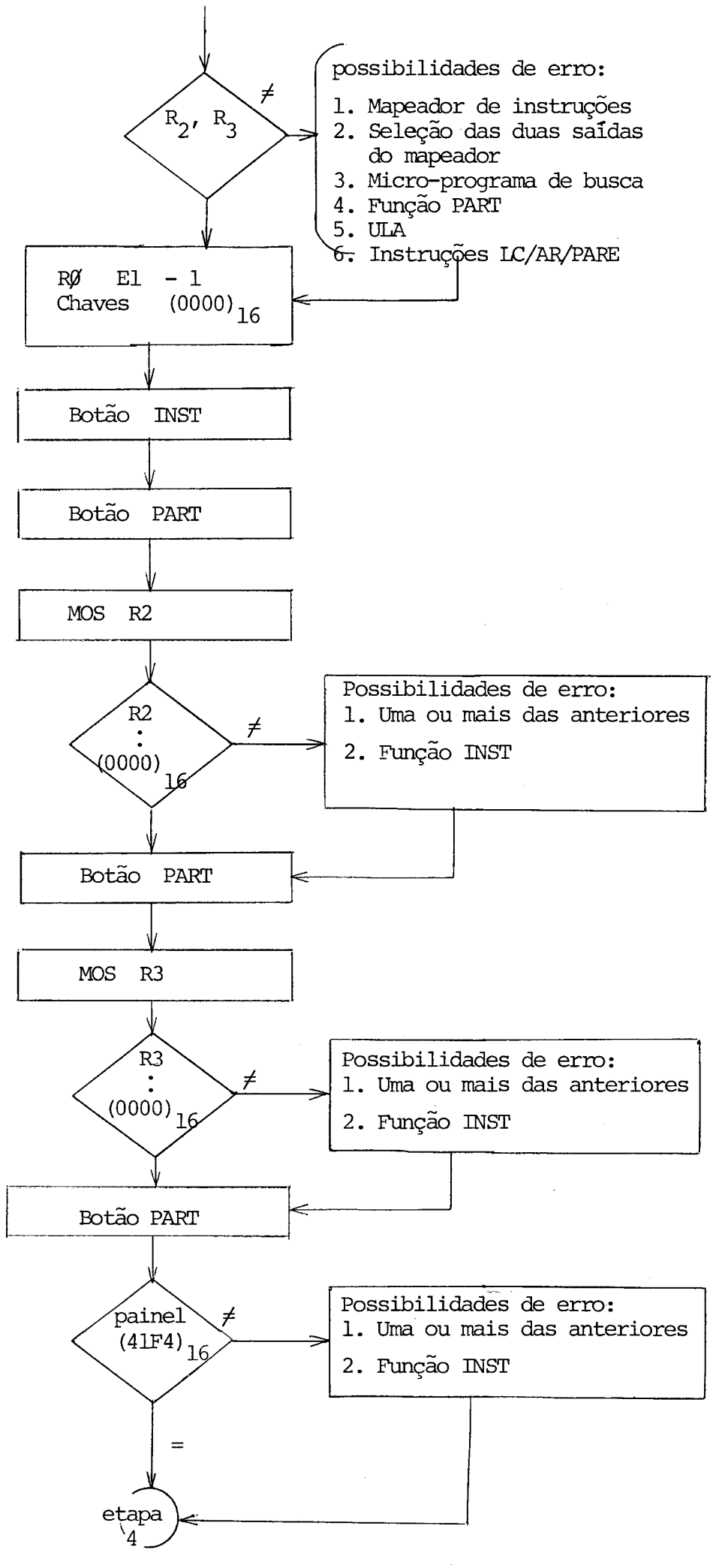
Fluxo de operações:



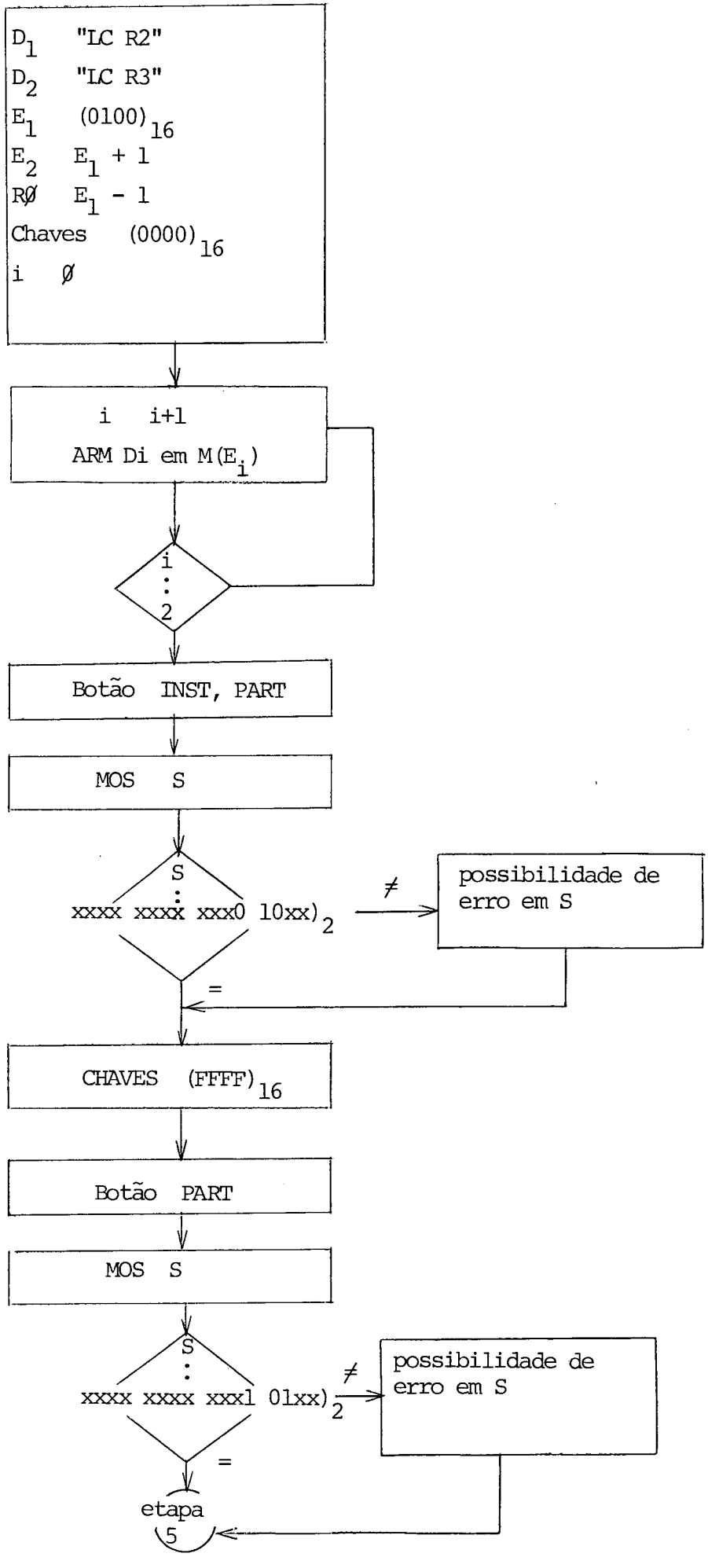
Etapa 3:

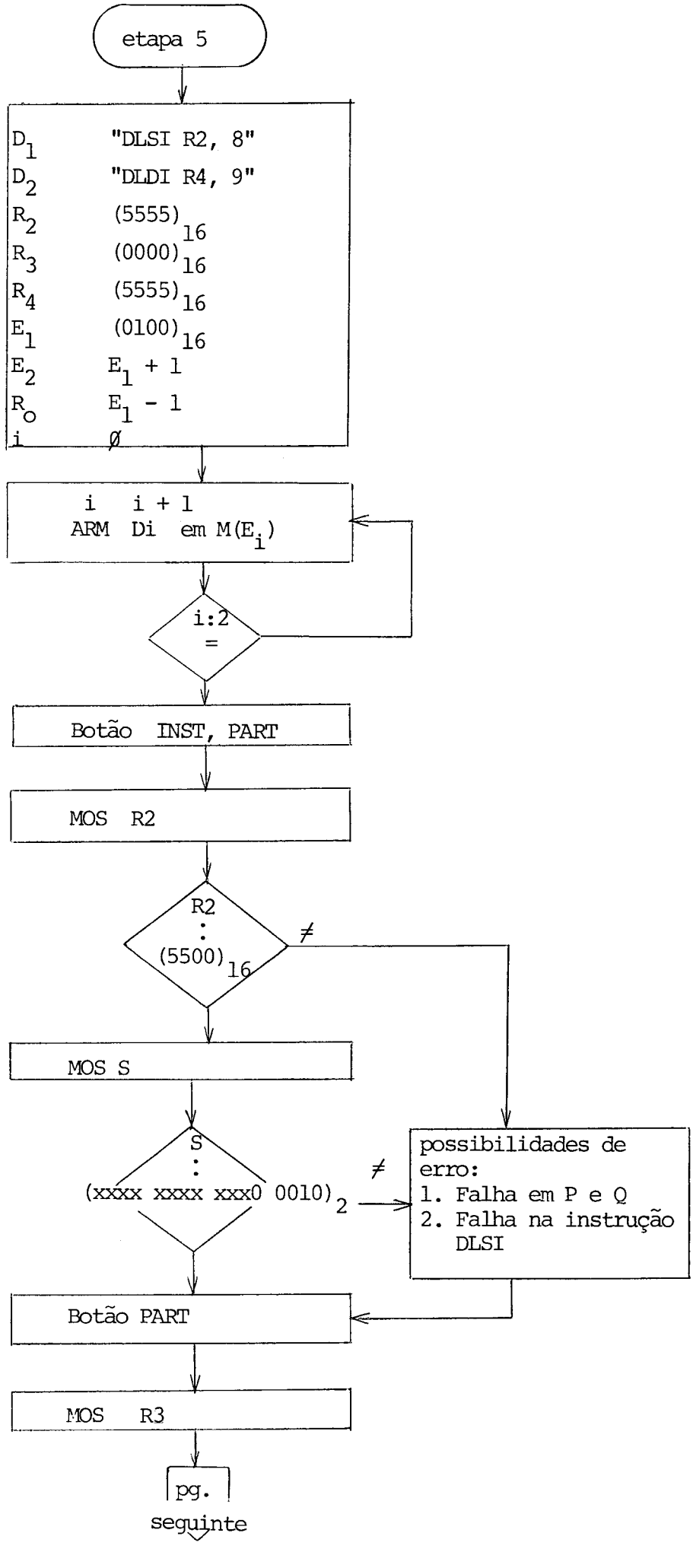
Fluxo de operação

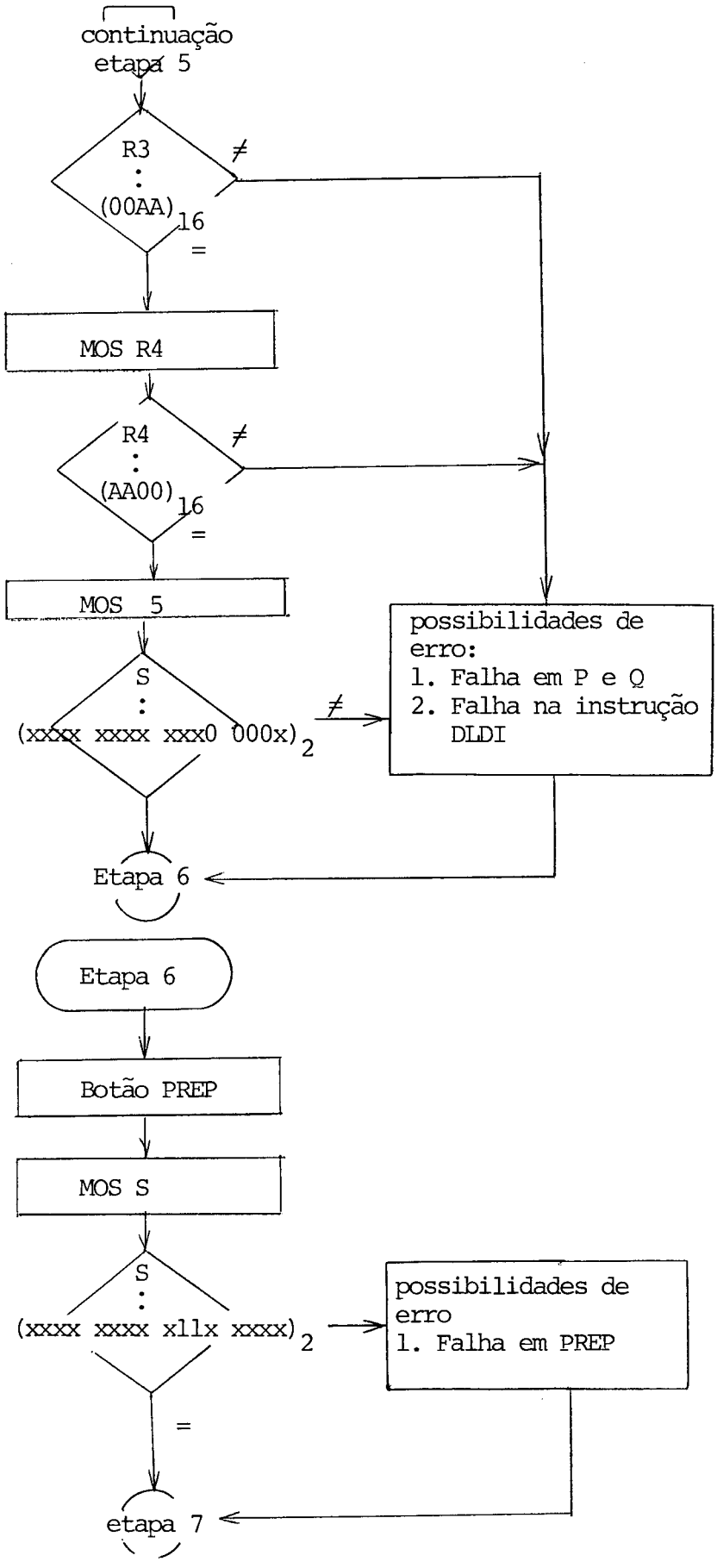


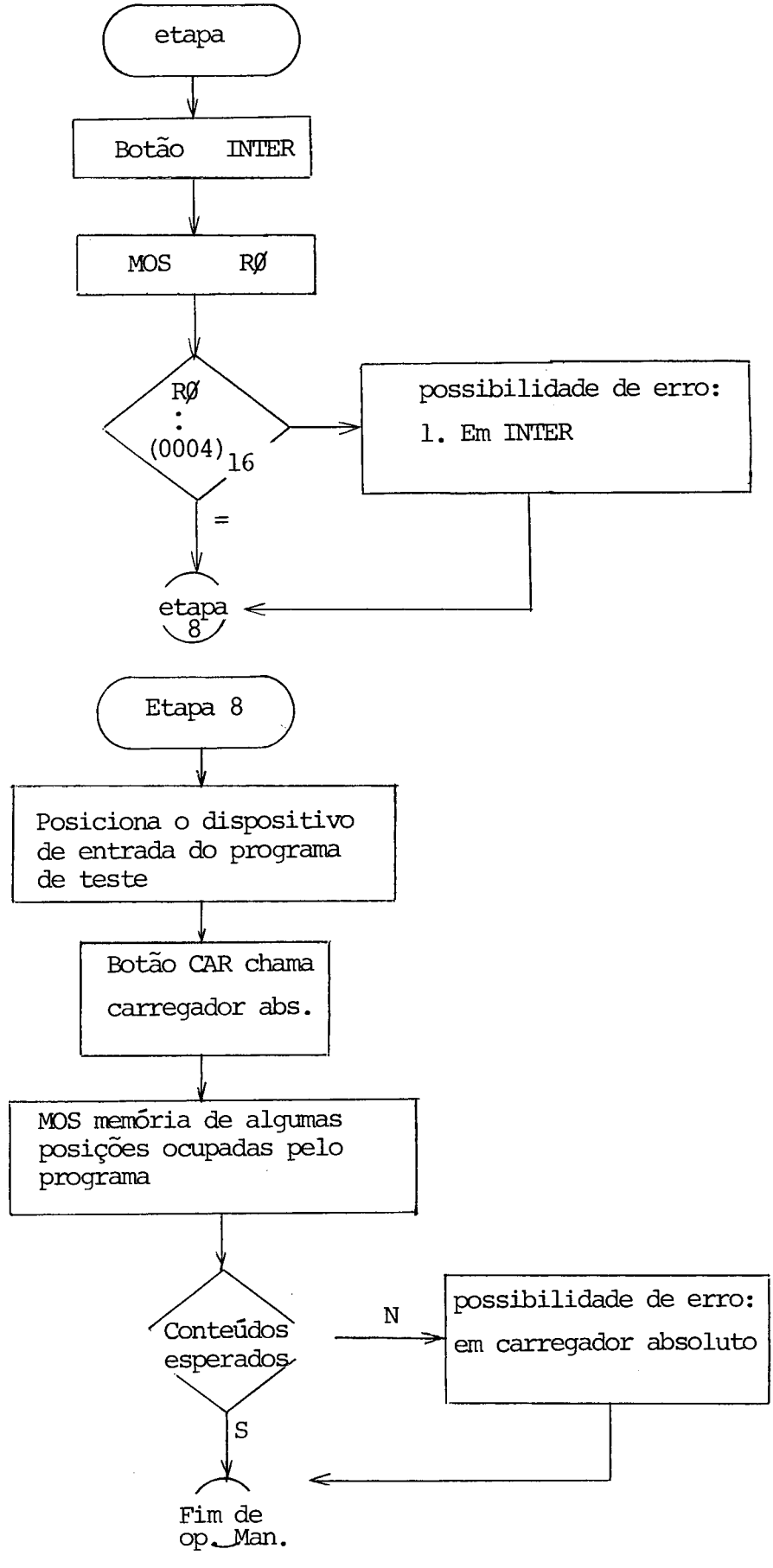

possibilidades de erro:
1. Mapeador de instruções
2. Seleção das duas saídas do mapeador
3. Micro-programa de busca
4. Função PART
5. ULA
6. Instruções LC/AR/PARE
≠
R2, R3
RØ E1 - 1
Chaves (0000)16
Botão INST
Botão PART
MOS R2
R2 : (0000)16
≠
Possibilidades de erro:
1. Uma ou mais das anteriores
2. Função INST
Botão PART
MOS R3
R3 : (0000)16
≠
Possibilidades de erro:
1. Uma ou mais das anteriores
2. Função INST
Botão PART
painel (41F4)16
≠
Possibilidades de erro:
1. Uma ou mais das anteriores
2. Função INST
=
etapa 4

Etapa 4:


D1 "LC R2"
D2 "LC R3"
E1 (0100)16
E2 E1 + 1
RØ E1 - 1
Chaves (0000)16
i Ø
i i+1
ARM Di em M(Ei)
i
:
2
Botão INST, PART
MOS S
S
:
xxxx xxxx xxx0 10xx)2
≠
possibilidade de erro em S
=
CHAVES (FFFF)16
Botão PART
MOS S
S
:
xxxx xxxx xxx1 01xx)2
≠
possibilidade de erro em S
=
etapa 5


etapa 5
D1 "DLSI R2, 8"
D2 "DLDI R4, 9"
R2 (5555)16
R3 (0000)16
R4 (5555)16
E1 (0100)16
E2 E1 + 1
Ro E1 - 1
i ø
i i + 1
ARM Di em M(Ei)
i:2 =
Botão INST, PART
MOS R2
R2 : (5500)16
≠
MOS S
S : (xxxx xxxx xxx0 0010)2
≠
possibilidades de erro:
1. Falha em P e Q
2. Falha na instrução DLSI
Botão PART
MOS R3
pg. seguinte


continuação
etapa 5
R3
:
(00AA)16
≠
=
MOS R4
R4
:
(AA00)16
≠
=
MOS 5
S
:
(xxxx xxxx xxx0 000x)2
≠
possibilidades de erro:
1. Falha em P e Q
2. Falha na instrução DLDI
Etapa 6
Etapa 6
Botão PREP
MOS S
S
:
(xxxx xxxx x11x xxxx)2
possibilidades de erro
1. Falha em PREP
=
etapa 7


etapa
Botão INTER
MOS RØ
RØ
:
(0004)16
=
possibilidade de erro:
1. Em INTER
etapa
8
Etapa 8
Posiciona o dispositivo
de entrada do programa
de teste
Botão CAR chama
carregador abs.
MOS memória de algumas
posições ocupadas pelo
programa
Conteúdos
esperados
N
possibilidade de erro:
em carregador absoluto
S
Fim de
op. Man.

Análise dos resultados dos testes dos pontos da primeira classe:

Se todas as etapas não são bem sucedidas: pode ser falha em algum dos seguintes pontos:

1 - Registradores de retorno de micro-subrotinas (SØ e S1);

2 - Seleção do endereço de retorno de uma micro-subrotina para o micro-programa principal;

3 - Seleção do endereço contido nos bits de 19 a 30 nas micro-instruções de pulo (pulos dentro dos micro-programas ou para outras micro-subrotinas);

4 - Seleção do endereço da micro-instrução seguinte proveniente do somador de +1;

5 - Somador de +1;

6 - Parte comum da seleção:

7 - Registrador de endereço de controle (EC);

8 - Registrador de dados de controle (DC);

9 - Decodificadores;

10 - Contador (CNTD);

11 - Lógica de testes de condições;

12 - Seleção da saída da ULA e ULA;

Se algumas etapas funcionam e outras não, os pontos com falhas podem ser aqueles acrescentados a primeira classe. A seguir é dada uma lista da participação dos outros pontos nas etapas.

PONTOS	UTILIZAÇÃO NAS ETAPAS							
	1	2	3	4	5	6	7	8
ARM	X	X	X	X	X			
MOS	X	X	X	X	X		X	X
Rede de registradores	X							
RØ	X		X	X			X	
R_1	X							
R_2	X		X	X	X			
R_3	X		X	X	X			
R_4	X				X			
R_5	X							
R_6	X							
R_7	X							
Reg. I	X		X	X	X			
Reg. E	X	X	X	X	X			X
Reg. S (RE)			X	X	X	X		X
M		X	X	X	X			X
M(E) Palavra da memória		X	X	X	X			X
Mapeador de instrução			X	X	X			
Seleção das duas saídas do mapeador			X	X	X			
Micro-programa de busca			X	X	X			
Função PART			X	X	X			
ULA			X	X	X			X
Reg. P e Q					X			
Função PREP						X		
Função INTER							X	
Função INST			X	X	X			
Inst. LC R2			X	X				
Inst. AR R2, R3			X					
Inst. PARE F4			X					
Inst. LC R3				X				
Inst. DLSI R2, 8					X			
Inst. DLAI R4, 9						X		
CAR								X
Periférico de entrada								X

A lista anterior auxilia na determinação das falhas dos pontos acrescentados a primeira classe de pontos.

Na etapa 8:

Se alguma das etapas 1, 2, 3, 4, 5, 7 funcionou significa que MOS não tem falha;

Se alguma das etapas 1, 2, 3, 4, 5 funcionou significa que reg. E não tem falha;

Se alguma das etapas 2, 3, 4, 5 funcionou significa que M e M(E) não tem falha;

Se alguma das etapas 3, 4, 5 funcionou significa que ULA não tem falha;

Os pontos CAR e periférico de entrada aparecem apenas em etapa 8, se os pontos relacionados anteriormente não tem falhas e etapa 8 não foi bem sucedida, "CAR" ou periférico deve ter falha;

Na etapa 7:

Se alguma das etapas 1, 2, 3, 4, 5 funcionou significa que MOS não tem falha;

Se alguma das etapas 1, 3, 4 funcionou significa que RØ não tem falha;

O ponto INTER aparece apenas na etapa 7;

Se MOS e RØ não tem falha e a etapa 7 não é bem sucedida significa que INTER tem falha;

Na etapa 6:

Se alguma das etapas 4, 5 funcionou, significa que S não tem falhas;

O ponto PREP aparece apenas na etapa 6;

Se S não tem falha e a etapa 6 não é bem sucedida significa que PREP tem falha;

Na etapa 5:

Se RØ, R2, R3 e R4 funcionaram em etapa 1 significa que estes registradores não tem falhas;

Se alguma das etapas 1, 2, 3, 4, 5 funcionou significa que ARM , MOS e reg. E não tem falha;

Se alguma das etapas 1, 3, 4 funcionou significa que reg I não tem falha;

Se alguma das etapas 2, 3, 4 funcionou significa que M e M(E) não tem falhas;

Se alguma das etapas 3, 4 funcionou significa que mapeador de instruções, seleção das duas saídas do mapeador, micro-programa de BUSCA, função PART, ULA, função INST não tem falhas;

Reg. P e Q, inst. DLSI R2, 8 e DLDI R4, 9 aparecem apenas na etapa 5, se os pontos relacionados anteriormente não tem falhas, significa que a falha está em um dos pontos que aparecem apenas na etapa 5;

Na etapa 4, teste de registrador de estado:

Se registradores RØ, R2, R3 funcionaram na etapa 1, não tem falhas.

Se alguma das etapas 1, 2, 3 funcionou significa que ARM, MOS , Reg. E não tem falhas;

Se alguma das etapas 1, 3 funcionou significa que reg. I não tem falhas;

Se alguma das etapas 2, 3 funcionou significa que M e M(E) não tem falhas;

Se etapa 3 funcionou significa que: Mapeador de instruções, seleção das duas saídas do mapeador, micro-programa de busca, função PART, ULA, função INST, inst. LC R2, não tem falhas;

O reg. S é verificado neste etapa, se os pontos que foram usados nas etapas anteriores não tem falhas e a etapa 4 não foi bem sucedida significa que a falha está em S (reg. de estado);

Na etapa 3:

Se os reg. RØ, R2, R3, reg. I foram bem sucedidos na etapa 1 , então não tem falhas;

Se alguma das etapas 1, 2 funcionou significa que ARM, MOS e reg. E não tem falhas;

Se a etapa 2 funcionou significa que M e M(E) não tem falhas;

Reg. de estado RE, mapeador de instruções; seleção das duas saídas do mapeador; micro-programa de busca; função PART; ULA; função INST; instruções: LC R2; AR R2, R3; PARE F4; aparecem nesta etapa, se os pontos relacionados anteriormente não tem falhas e esta etapa não é bem sucedida, significa que a falha deve estar em algum ponto que aparece nesta etapa;

Na etapa 2:

ARM, MOS e reg. E não tem falhas;

M e M(E) aparecem nesta etapa, se os pontos relacionados anteriormente não tem falhas e esta etapa não é bem sucedida, significa que a falha deve estar em algum ponto que aparece nesta etapa;

Na etapa 1:

Se alguma das etaps 2, 3, 4, 5 funcionou significa que ARM , MOS e reg. E não tem falhas;

Se alguma das etapas 3, 4, 5 funcionou significa que R2, R3 , reg. I não tem falhas;

Se alguma das etapas 3, 4 funcionou significa que RØ não tem falhas;

Se a etapa 1 não é bem sucedida, proceder as etapas seguintes para localizar as falhas.

Fica concluído nesta análise que os pontos que apresentam falhas são localizados pela combinação das etapas onde eles aparecem e que o mapa de utilização de pontos por etapa é suficiente para o operador de testes localizar as falhas. A descrição das análises de cada etapa mostra como usar o mapa.

Essa terceira organização dos pontos, vai reclassificar os pontos restantes da terceira classe para a segunda classe e os pontos da quarta classe para a terceira.

Segunda Classe de Pontos:

1 - Situação externa: instrução curta sem MO:

pontos que são ativados apenas por esta situação externa

1. Parte do mapeador 1
2. Micro-subrotinas de preparação de operando: CURT/IMED/PVES/PRVI/DEP (apenas pulos).

2 - Situação externa: instrução curta com MO:

pontos que são ativados apenas por esta situação externa

1. Parte do mapeador 1
2. Micro-subrotinas de cálculo de endereço efetivo MO

3 - Situação externa; instrução longa sem MO:

pontos que são ativados apenas por esta situação externa

1. Parte do mapeador 1
2. Micro-subrotinas de cálculo de endereço efetivo: ME/DEP (desvios)/IMED

4 - Situação externa: instrução longa com MO e ME/BE:

pontos que são ativados apenas por esta situação externa

1. Parte do mapeador 2
2. Micro-subrotinas de cálculo de endereço efetivo MO
3. Micro-subrotinas de cálculo de endereço efetivo ME/BE

5 - Situação externa: ocorrência de falhas de alimentação:

pontos ativados por esta situação

1. Mapeador de interrupção por falha de alimentação
2. Micro-programa que trata de interrupção por falha de alimentação.

6 - Situação externa: tentativa de acesso a endereço fora dos limites permitidos.

pontos ativados apenas por esta situação

1. Mapeador de interrupção por proteção de memória
2. Micro-programa que trata de interrupção por proteção de memória.

7 - Situação externa: interrupção de rastreamento.

pontos ativados apenas por esta situação

1. Mapeador de interrupção por rastreamento
2. Micro-programa que trata de interrupção por rastreamento

8 - Situação externa: pressão do botão INTER no painel:

pontos ativados apenas por esta situação

1. Mapeador de interrupção pelo botão INTER
2. Micro-programa que trata de interrupção pelo botão INTER

OBS: Para as descrições dos testes de 2ª. e 3ª. classes: Estes testes incluem apenas a instrução que ativa a micro-instrução em teste com seus padrões de entrada, a forma de implementação é flexível de acordo com o programa.

Testes dos pontos da segunda classe:

Teste do ponto 1, instrução curta sem MO: Este ponto é subdividido em 4 pontos devido as micro-subrotinas de preparação de operando não terem dependência entre si.

Micro-subrotina CURT:

AC	R, E	$R=R2=(5555)_{16}$	$E = (AAAA)_{16}$
CGC	R, E	$R=R2=(AAAA)_{16}$	$E = (5555)_{16}$

Micro-subrotina IMED:

CGI	R, I	$R=R2=(FFFF)_{16}$	$I = (00)_{16}$
DLSI	R, I	$R=R2=(5555)_{16}$	$I = (08)_{16}$

Micro-subrotina PVES: As instruções que ativam esta micro-subrotina são testadas no teste de periféricos.

Micro-subrotina PRVI:

PIT	R, E	$R=R2=(FFF1)_{16}$	negativo
		$R=R2=(000F)_{16}$	positivo
		$R=R2=(0000)_{16}$	zero
PARE	I	$I = (F4)_{16}$	

Teste do ponto 2, instrução curta com M0:

com M0 = $(ØØ)_2$; $(Ø1)_2$; $(1Ø)_2$; $(11)_2$

AR R, RR R=R2=$(5555)_{16}$

RR=R3=$(AAAA)_{16}$

com M0 = $(ØØ)_2$; $(Ø1)_2$; $(1Ø)_2$; $(11)_2$

LRE R2 Re = $(xxxx\ xxxx\ xxx1\ 0100)_2$

Teste do ponto 3, subdividido em 2 pontos:

- longa de desvio:

    DI RM

    DD RM

- longa IMED:

    ADM RM, I RM = $(5555)_{16}$

    I = $(01010)_2$ → RM = $(555A)_{16}$

Teste do ponto 4, subdividido em 2 pontos:

com M0 = $(00)_2$; $(01)_2$; $(10)_2$; $(11)_2$ e ME = $(00)_2$; $(01)_2$; $(10)_2$; $(11)_2$

- Longa com M0 e ME:

    CGE RR, RM

    OU RR, RM; RR = $(AAAA)_{16}$

    RM = $(5555)_{16}$; RR = $(FFFF)_{16}$

- Longa com M0 e BE:

com todos os BE's

AB R, RM R=R2 = $(55)_{16}$

byte de RM = $(AA)_{16}$ byte de RM = $(55)_{16}$

com todos os BE's

CGB R, RM R=R2 = $(0055)_{16}$

byte de RM = $(AA)_{16}$ → R=R2=$(FFAA)_{16}$

Teste do ponto 5:

Depois que o programa foi processado um número de vezes considerado suficiente, o operador de teste pode provocar uma falha de alimentação.

Após esta ação o processador deve ficar no estado parado com painel = $(41FF)_{16}$ e RØ = $(ØØØØ)_{16}$.

Teste do ponto 6:

Depois que o programa foi processado um número de vezes considerado suficiente, o operador de teste pode armazenar um desvio para endereço fora dos limites permitidos e armazenar em RØ o endereço desse desvio, pressionar PART.

Após esta ação o processador deve ficar no estado parado com painel = $(41FF)_{16}$ e RØ = $(ØØØ1)_{16}$.

Teste do ponto 7:

Depois que o programa for processado um número de vezes considerado suficiente, o operador de teste pode pressionar o botão TEST.

Após esta ação o processador deve ficar no estado parado com painel = $(41FF)_{16}$ e RØ = $(0003)_{16}$.

Teste do ponto 8:

Depois que o programa for processado um número de vezes considerado suficiente, o operador de teste pode pressionar o botão INTER.

Após esta ação o processador fica no estado parado com painel = $(41FF)_{16}$ e RØ = $(0004)_{16}$.

Testes dos pontos de terceira classe:

Os primeiros pontos a serem testados são as micro-subrotinas de funções, cujos testes são realizados através das instruções que as ativam. A lógica de implementação de cada função é examinada com o objetivo de tomar conhecimento de todos seus diferentes caminhos lógicos cujos percursos são provocados pela ocorrência de situações diferentes. As situações são traduzidas em padrões de dados.

Se alguma instrução já participou de algum teste anteriormente ; seu teste será repetido porque o objetivo dos testes anteriores era verificar outros elementos, isto é, a instrução era considerada apenas um instrumento. Se a instrução é considerada um ponto, todo o caminho da instrução está sendo testado, ela merece um teste completo, aliás é necessário, sua omissão causa dificuldade na localização da falha do outro ponto, fica uma dúvida se é o ponto ou o instrumento de teste no caso a instrução que apresenta falha.

São sessenta e oito micro-subrotinas de funções no processador do mini-computador G11, estas micro-subrotinas devem ser testadas em todos os padrões que provoquem caminhos diferentes pela micro-subrotina, resultando no final uma lista muito grande de padrões. Com o objetivo de exemplificar o teste completo das micro-subrotinas de funções vão ser relacionadas apenas algumas micro-subrotinas.

O restante das instruções serão testadas apenas com o objetivo de verificar o caminho através de seu ciclo de execução , "BUSCA", preparação de operando, EXECUÇÃO, o que permite que seu teste seja completo através de um padrão de dados apenas. O padrão de dados para estas instruções deve ser um padrão escolhido na relação de padrões levantados para o teste da função que é ativada para a instrução.

1 - Teste da micro-subrotina de funções CAR através da instrução CAR R, RR (compara aritmético registradores)

Padrões de teste:*

R Em hexadecimal	RR Em hexadecimal	RE Em binário (P Z S E T)	OBSERVAÇÃO
3FFF	C001	X 0 0 X 0	+ com - cuja sub. é +
7FFF	0001	X 0 0 X 0	+ com + cuja sub. é -
0001	0001	X 1 0 X 0	+ com + cuja sub. é Ø
5555	5556	X 0 1 X 0	+ < com + > cuja sub. é -
FFFE	0001	X 0 1 X 0	- com + cuja sub. é -
8000	0001	X 0 0 X 0	**- com + cuja sub. é + (Téoric. errada)
7FFF	FFFF	X 0 1 X 0	**+ com - cuja sub. é - (Téoric. errada)

**As duas deveriam acusar transbordo.

2 - Teste da micro-subrotina de função CGR através da instrução CGC R, RR (carrega registrador)

Padrões de teste:

R	RR	R	P Z S E T	OBSERVAÇÃO
0000	AAAA	AAAA	0 0 1 0 0	Ø → - com 1's alternados
AAAA	5555	5555	1 0 0 0 0	- → + com 1's alternados de forma a todos Ø → 1 e todos 1 → Ø

*Os números representados em hexodecimal quando ultrapassam o valor $(7FFF)_{16}$ são considerados negativos, i.e., o bit mais a esquerda apresenta o valor $(1)_2$.

3 - Teste da micro-subrotina de função AR através da instrução AR R, RR (armazena registrador)

Padrões de teste

R	RR	RR	PZSET	OBSERVAÇÃO
5555	AAAA	5555	xxxxx	Todos bits Ø → 1 e todos bits 1→

4 - Teste da micro-subrotina de função MLS através da instrução MLSR R, RR (multiplica simples registradores)

Padrões de teste:

R cador	RR cando	R produto	PZSET		OBSERVAÇÃO
0001	8000	8000	001X0	+ X - = -	
0002	C000	8000	001X0	+ X - = -	
FFFE	4000	8000	001X0	- X + = -	
FFFF	8001	7FFF	100X0	- X - = +	
0001	7FFF	7FFF	100X0	+ X + = +	
8001	FFFF	7FFF	100X0	- X - = +	
7FFF	0001	7FFF	100X0	+ X + = +	
0002	8000	0000	010X1	+ X - = Ø	com transbordo
0003	C000	4000	000X1	+ X - = +	com transbordo
FFFE	5000	6000	000X1	- X + = +	com transbordo
FFFE	8001	FFFE	001X1	- X - = -	com transbordo
0002	7FFF	FFFE	001X1	+ X + = -	com transbordo
8001	FFFE	FFFE	001X1	- X - = -	com transbordo
7FFF	0002	FFFE	001X1	+ X + = -	com transbordo
Casos intermediários					
0281	0033	7FB3	100X0	+ X + = +	
FFAB	0181	802B	101X0	- X + = -	
0282	0033	7FE6	000X0	+ X + = +	
FFAA	017D	8002	001X0	- X + = -	
017D	FFAA	8002	001X0	+ X - = -	
0172	FFAA	83B4	001X0	+ X - = -	
FFAA	FFAA	1CE4	000X0	- X - = +	

5 - Teste de micro-subrotina TR através de instrução TR R, RR (troca registradores)

R	RR	R	RR	OBSERVAÇÃO
5555	AAAA	AAAA	5555	Todos Ø → 1; todos 1 → Ø

6 - Teste da micro-subrotina SEPV através da instrução SEPB R,RR (separa bytes em registradores)

R	RR	R	RR
AAAA	5555	0055	0055

7 - Teste da micro-subrotina DS

Esta micro-subrotina é ativada por 4 instruções:

DLS R, R; DAS R, R; DLSI, R, I; DASI R, I

Vão ser feitos dois testes completos um para lógico e outro para aritmético.

O teste com deslocamento lógico vai ser feito na instrução DLS R, R e o teste feito com deslocamento aritmético vai ser DASI R, I

Padrões para DLS R dest., R orig:

Rdest	Rorig.	Rdest	PZSET	OBSERVAÇÃO
5555	+8	5500	00010	extensão = 1; byte esquerdo = byte direito.
AAAA	+15	0000	01010	zero palavra
AAAA	0	AAAA	00100	não desloca
5555	-5	02AA	00010	zero bits a esquerda
AAAA	7	5500	00010	caso não extremo
5555	-8	0055	10000	byte direito = byte esquerdo
AAAA	-15	0001	10000	bit Ø = bit 15

Padrões para DASI R, I

Rdest	I	Rdest.	PZSET	OBSERVAÇÃO
AAAA	0	AAAA	00100	não desloca
AAAA	+15	0000	01011	bit 15 = bit Ø
AAAA	-15	FFFF	10100	bit Ø = bit 15; todos os bits à esquerda = bit 15
5555	+7	AA80	00111	caso não extremo
5555	-8	0055	10000	byte direito = byte esq., todos bits a esquerda = bit 15
AAAA	-5	FD55	00110	caso não extremo

8 - Teste da micro-subrotina DDD:

Esta micro-subrotina é ativada por 4 instruções:

DLD R, R; DAD R, R; DLDI R, I; DADI R, I

Da mesma forma que na micro-subrotina DS, são feitos dois testes completos, um para lógico e outro para aritmético.

O teste de deslocamento lógico é feito com a instrução DLD R,R e o teste de deslocamento aritmético duplo imediato com DADI R, I.

Padrões para DLD R,R

(R-1)(dest)	R(dest)	R(orig)	(R-1)(dest)	R(dest)	PZSET	OBS:
FFFF	0000	0	FFFF	0000	00100	não desloca
AAAA	AAAA	15	5555	0000	00010	ext.p/esq.
5555	5555	15	AAAA	8000	00100	ext.p/esq.
0000	AAAA	+7	0055	5500	00000	caso não extremo
AAAA	AAAA	-15	0001	5555	10000	ext.p/dir.
5555	5555	-15	0000	AAAA	00010	exp.p/dir.
AAAA	0000	-9	0055	5500	00000	caso não extremo

Padrões para DADI R,I

(R-1)(det)	R(dest)	I	(R-1)dest	R(dest)	PZSET	OBS:
FFFF	0000	0	FFFF	0000	00100	não desloca
AAAA	AAAA	15	5555	0000	00011	ext.p/esq. (-)
5555	5555	15	AAAA	8000	00101	ext.p/esq. (+)
0000	AAAA	+7	0055	5500	00000	não extremo
AAAA	AAAA	-15	FFFF	5555	10100	ext.p/dir. (-)
5555	5555	-15	0000	AAAA	00010	ext.p/dir. (+)
AAAA	0000	-9	FFD5	5500	00100	não extremo

9 - Teste da micro-subrotina PP através da instrução PP E (pula se par)

O teste pela instrução PP E é desenvolvido no programa nas situações descritas a seguir:

Pula se par para trás com P de RE = Ø
Pula se par para trás com P de RE = 1
Pula se par para frente com P de RE = Ø
Pula se par para frente com P de RE = 1

10 - Teste da micro-subrotina DMA através da instrução PMA E (pula se maior):

O teste pela instrução PMA E é desenvolvido no programa nas situações descritas a seguir:

Pula se maior para trás com Z de RE = 1 S de RE = 0
Pula se maior para trás com Z de RE = 0 S de RE = 1
Pula se maior para trás com Z de RE = 0 S de RE = 0
Pula se maior para frente com Z de RE=0 S de RE = 0
Pula se maior para frente com Z de RE=0 S de RE = 1
Pula se maior para frente com Z de RE=0 S de RE = 0

11 - Teste da micro-subrotina DI através da instrução PI E (pula se)

O teste pela instrução PI E é desenvolvido no programa nas situações descritas a seguir:

Pula se igual para trás com Z de RE = Ø

Pula se igual para trás com Z de RE = 1

Pula se igual para frente com Z de RE=Ø

Pula se igual para frente com Z de RE=1

12 - Teste da micro-subrotina DMAI através da instrução PMAI E (pula se maior ou igual)

Os testes pela instrução PMAI E são desenvolvidos no programa nas situações descritas a seguir:

Pula se maior ou igual para trás com Z = Ø S = 1

Pula se maior ou igual para trás com Z = Ø S = Ø

Pula se maior ou igual para trás com Z = 1 S = Ø

Pula se maior ou igual para frente com Z = Ø S = 1

Pula se maior ou igual para frente com Z = 0 S = 0

Pula se maior ou igual para frente com Z = 1 S = 0

13 - Teste da micro-subrotina DME através da instrução PME E (pula se menor)

Os testes pela instrução PME E são desenvolvidos no programa nas situações descritas a seguir:

Pula se menor para trás com Z = Ø S = 0

Pula se menor para trás com Z = Ø S = 1

Pula se menor para trás com Z = 1 S = 0

Pula se menor para frente com Z = 0 S = 0

Pula se menor para frente com Z = 0 S = 1

Pula se menor para frente com Z = 1 S = 0

14 - Teste de micro-subrotina PD através de instrução PD (pula se diferente)

Os testes pela instrução PD E são desenvolvidos no programa nas situações descritas a seguir:

Pula se diferente para trás com Z = Ø

Pula se diferente para trás com Z = 1

Pula se diferente para frente com Z = Ø

Pula se diferente para frente com Z = 1

CAPÍTULO IV

AVALIAÇÃO DO SISTEMA DE TESTES PROPOSTO

4.1 - Apresentação do Modelo de Determinação de Detetabilidade e Diagnosticabilidade com Reparo de um Sistema Digital com t Componentes Sujeito a Falha, Concluído por Kime e Russel |3|.

A maioria dos modelos iniciais de análise de diagnóstico tem considerado o sistema a ser dividido em um número de subsistemas isolados sob a suposição de que cada subsistema pode ser testado completamente pela combinação de outros subsistemas. Cada teste então definido envolve a aplicação controlada de estímulos ao subsistema sob teste e a análise dos resultados obtidos, proporciona a avaliação do subsistema verificado, como uma falha externa ou o subsistema está defeituoso. Entretanto, se algum dos subsistemas envolvidos no teste está com defeito, o subsistema sendo testado seria avaliado como defeituoso quando não é ou tendo falha externa quando realmente está defeituoso.

A primeira consideração séria sobre os efeitos de testes inválidos para processo de diagnóstico foi feita por Preparata et al |8|. Há uma limitação da classe de sistemas considerados onde cada subsistema sozinho é capaz de testar outro. Condições necessárias e condições suficientes são dadas para tais sistemas serem diagnosticáveis para no máximo t falhas presentes no subsistema. Extensões da teoria são dadas

por Preparata |9| e Seshagiri |10|. Resultados semelhantes são obtidos por Vancura e Kime |11| para sistemas onde mais do que um subsistema pode ser necessário para testar outro; entretanto, a avaliação da capacidade de interconecção ainda é arbitrada segundo a necessidade.

A seguir estão relacionados resultados gerais pertencentes a "diagnosticabilidade por níveis de sistema" para que seja providenciada uma base mais unificada para consideração do projeto e análise de diagnóstico de sistemas digitais. Depois de uma breve descrição do modelo e exposição do assunto, os conceitos detectabilidade e diagnosticabilidade são também estabelecidos.

- Modelo de Diagnóstico

O modelo empregado compreende os outros mencionados embora difira deles em pontos de vista. Melhor do que falando de certos subsistemas testando outros, o modelo é formulado em termos de "falhas", "testes" e de suas relações. Uma ampla interpretação é dada para falhas e testes. Por exemplo, uma falha pode ser definida como qualquer condição que cause o mau funcionamento de uma parte particular do sistema tal como um computador num complexo de multi-computadores ou a unidade aritmética de um computador. Um teste pode ser qualquer combinação de procedimentos de hardware e software usados para determinar se uma falha ocorreu. Assim um teste poderia ser uma coleção de padrões de entrada para uma rede lógica combinacional ou

um programa de verificação externa completo para um sub-sistema maior. Exatamente o objeto de uma falha ou um teste é dependente do nível de análise feita para o diagnóstico.

Associada a um sistema "S" há a idéia da possibilidade de ocorrer uma série $F = \{f_1, f_2, \ldots f_n\}$ de n falhas e de serem aplicados uma série $T = \{t_1, t_2, \ldots, t_p\}$ de p testes de passagem de defeito em S. A presença de falhas simultâneas é possível. A série $F = \{F^1, F^2, \ldots, F^{2n}\}$ consiste de todas subséries de F, e cada padrão $F^k \varepsilon F$ é tolerável em S.

Uma representação da matriz Booleana F de padrões de falhas é dada pela ordem $2^n \times n$ tendo $F_i^k = 1$ se falha f_i é um elemento de padrão de falha F^k; $F_i^k = \emptyset$ se falha f_i não é um elemento de padrão de falha F^k.

Em certas situações onde falhas atingem um "nível baixo", tal como numa rede lógica com falhas definidas como linhas s-a-0 ou s-a-1, um teste particular pode falhas para um número simples de ocorrências e como consequência ser aproveitado como teste para cada dessas falhas. Entretanto em diagnósticos onde as falhas estão em um "nível superior do sistema", é assumido que dois subsistemas não são completamente verificados pelo mesmo teste. Acrescentando, de um ponto de vista do diagnóstico, é razoável considerar somente testes que falhem pelo menos em um defeito presente. Assim cada teste em T é assumido a ser um teste completo para um e somente um defeito em F.

Do mesmo modo, falhas indetectáveis (e portanto sistemas não diagnosticáveis trivialmente) são eliminados da consideração pela suposição que pelo menos um teste completo existe em T para cada falha em F.

Um teste t_j é um teste completo para falha f: se t_j sempre falha quando f_i sozinho está presente em S e sempre passa, para todas as falhas de F ausentes. A série de testes que são completos para a falha f_i é denotada por $t(f_i)$. A notação pode ser extendida para definir:

$$t(f_i, f_j, \ldots, f_k) = t(f_i)\ t(f_j) \ldots t(f_k)$$

Uma representação alternativa dessa informação é obtida por uma matriz Booleana C de ordem $n \times p$ chamada a "matriz completa de testes" ou simplesmente a "matriz C". A matriz C tem:

$C_j^i = 1$ se teste t_j é completo para falha f_i

$C_j^i = \emptyset$ se teste t_j não é completo para falha f_i

Se um teste é definido como <u>completo para falhas simples pode se tornar incompleto num meio de falhas múltiplas</u>. Também, acrescentando, o teste sendo completo para uma falha particular, pode ser incompleto para uma ou mais outras falhas. Esses mecanismos de invalidação de testes são incluídos no modelo pela associação em cada padrão de falha F^k uma série

$T(F^k)$ de testes que são inválidos na presença de F^k, i.e., testes que não produzem necessariamente informações indicativas do estado real do sistema devido aos efeitos de falha em F^k. Mais precisamente, um teste t_j é inválido de padrão de falha F^k se sua presença causa t_j ser imprognosticável ou desconfiável no sentido de que t_j poderia passar sempre através de $t_j \in t(F^k)$ ou poderia falhar se $t_j \notin t(F^k)$. Considere, por exemplo, a situação na qual $t_j \notin t(F^k)$ mas t_j seja um teste incompleto por uma ou mais falhas em F^k, i.e., t_j detecte alguma das falhas associadas ao mau funcionamento. Logo não pode ser determinado a priori se t_j passará ou falhará. Do mesmo modo, se há presença de falhas no hardware usado para efetuar o teste t_j; então t_j poderia passar ou falhar desatento a quaisquer falhas realmente presentes para as quais t_j é completo. Se o teste t_j é conhecido para:

- sempre passar se $t_j \notin t(F^k)$ e
- sempre falhar se $t_j \in t(F^k)$, então t_j é válido na presença de F^k.

Consistente com a definição de um teste completo é a condição que:

Nenhum teste é inválido pela falha para o qual ele é completo |i.e.

se $t_j \in t(f_i)$, então $t_j \notin T(f_i)$|.

Um dos principais componentes do modelo é uma matriz que define a relação entre os padrões de falha e as consequências do teste. A "tabela de falhas generalizada" ou "matris G" tem dimensões $2^n \times p$ e seu elemento G_j^k assume valores conforme as situações:

$G_j^k = \emptyset$, se teste t_j é conhecido para sempre passar na presença de F^k;

$G_j^k = 1$, se teste t_j é conhecido para sempre falhar na presença de F^k;

$G_j^k = X$, se teste t_j tem resultado desconhecido na presença de F^k;

estas situações podem ser formalizadas de acordo com a condição consistente com a definição de um teste completo: se $t_j \varepsilon t(f_i)$, então $t_j \notin T(f_i)$

$G_j^k = \emptyset$, se $t_j \notin t(F^k)$ e $t_j \notin T(F^k)$

$G_j^k = 1$, se $t_j \varepsilon t(F^k)$ e $t_j \notin T(F^k)$

$G_j^k = X$, se $t_j \varepsilon T(F^k)$.

A principal suposição aqui é que um teste que é válido na presença de F^k, pode falhar para F^k presente somente se é completo para alguma falha em F^k. Essa suposição é razoável a nível de sistema, embora não seja válida para um nível de rede de circuito para falhas to tipo "stuck-at". Uma linha G^k contendo q entradas com valor X (devido a q testes sen-

do inválidos na presença de falhas padrões F^k) representa (2^q binário) p-tuplas que são todos dos resultados da série de testes que podem ser obtidos pela aplicação dos testes $t_1, t_2, \ldots, t_p$ a S. Desse modo, é suposto que os resultados de testes inválidos são independentes, i.e., não são deterministicamente correlacionados com cada um dos outros, portanto que um teste que passa implica em outro que falha, etc.

Dirigindo atenção ao processo de invalidação de teste, os sistemas a serem considerados satisfazem a condição que:

$T(F^i \cup F^i) \supseteq T(F^i) \cup T(F^j)$ para todo F^i, $F^j \varepsilon F$ e são chamados de sistemas semimórficos. Se um sistema semimórfico satisfaz a condição mais restritiva em que $T(F^i \cup F^j) = T(F^i) \cup T(F^j)$ para todo F^i, $F^j \varepsilon F$, então o sistema é levado a ser chamado de mórfico. A condição mórfica aparece naturalmente se uma certa série de falhas devem estar ausentes em ordem para um teste particular a ser avaliado. Os sistemas considerados por |5|, |7| e |10| são mórficos e a condição é necessária para um número de resultados apresentados aqui também. A condição semimórfica menos restritiva é útil para sistemas nos quais duas ou mais falhas simultâneas são necessárias em ordem para invalidar um teste. Assim existe uma situação, por exemplo, quando um teste é efetuado usando a saída de um aprovador em uma estrutura triplamente redundante na qual mais de um subsistema suprindo o avaliador pode estar com defeito. Em adição a consideração direta de sistemas semimórficos, uma aproximação particular seria usa-

da para representá-los usando um modelo mórfico. Essa aproximação é baseada na utilização de $T(f_i)$ para cada $f_i \in F$ e então construindo $T(F^k) = \bigcup T(f_i)$ sobre todos $f_i \in F^k$. Sempre que uma referência ao sistema S_M é feita como uma aproximação mórfica a um sistema S, essa técnica é assumida.

Se um sistema é mórfico (ou possivelmente uma aproximação mórfica para um sistema semimórfico), então dois componentes padrões adicionais são avaliáveis. A coleção de n séries de testes invalidados associados com as n falhas em F é representada pela "matriz de invalidação de testes" ou "matriz T" |9|. A matriz T é uma matriz Booleana de dimensões $n \times p$ cujo elemento T^i_j assume os valores de acordo com as situações a seguir relacionadas:

$T^i_j = 1$ se a falha f_i invalida o teste t_j e

$T^i_j = \emptyset$ se a falha f_i não invalida o teste t_j.

Um componente gráfico que representa o modelo rigorosamente, providencia um instrumento efetivo de análise e visualização dos sistemas. O gráfico de diagnósticos D de um sistema S é chamado um gráfico dirigido que tem um vértice para cada falha em F e uma aresta dirigida no sentido, do vértice associado a falha f_i para o vértice associado a falha f_j se e somente se f_i invalida pelo menos um teste que é completo para f_j (i.e. se e somente se $T(f_i) \cap t(f_j) \neq \emptyset$). Os vértices e arestas de D são chamados da seguinte maneira:

1) Cada vértice é chamado internamente pela falha associada a ele;

2) Uma aresta dirigida do vértice f_i para o vértice f_j é chamada de "os testes completos para f_j que são invalidados pela falha f_i" |i.e., os testes na série $T(f_i) \cap t(f_j)$| e

3) Cada vértice é chamado externamente pelos testes completos , para f_i que não são invalidados por qualquer falha em $\mathcal{F}$ |i.e., os testes na série $t(f_i) - T(\mathcal{F})$| .

A última classificação providenciada está incluída de tal modo que o grafico de diagnóstico possa representar adequadamente um sistema no qual alguns elementos possam ser testados manualmente ou possam ser testados por sistemas livres de falha.

Neste ponto da apresentação, podem ser empregadas as duas notações:

Para modelo de diagnóstico matriz G, $S = (\mathcal{F}, \mathcal{T}, F, G)$ e para gráfico de diagnóstico $S = (\mathcal{F}, \mathcal{T}, F, D)$. Observe que o modelo de gráfico apenas caracteriza completamente os sistemas que são mórficos.

- Detectabilidade e Diagnosticabilidade

O interesse principal é a capacidade da série de testes $\mathcal{T}$, detectar e identificar (localizar) as falhas em F sob a condição que o máximo de t falhas podem estar presentes simultaneamente. Um sistema tem a falha detectável por t se e somente se algum teste em $\mathcal{T}$ falhará exatamente para o número de falhas previstos presentes, que pode ser 1 ou até o número previsto por t. Está implícita a suposição de que nenhum teste falhará se o sistema está livre de falhas. O tipo de diagnosticabilidade considerada envolve reparo de falha |12| que é análoga a diagnosticabilidade sequencial de PREPARATA et al |8|. Um sistema é diagnosticável com reparo da falha por t se e somente se existe uma sequência de aplicações da série de testes $\mathcal{T}$ e reparos de falhas identificados que permite todas as falhas orinalmente presentes a serem identificadas providenciado o número de falhas originalmente presentes não exceder t. Note que a definição não permite o reparo desnecessário de um equipamento bom, e que "reparo" é usado como um termo genérico significando reconfiguração, recolocação e reparo realmente. Equivalentemente, um sistema é diagnosticável com reparo da falha por t se uma aplicação da série de teste é suficiente para identificar pelo menos uma falha presente (que pode então ser reparada e a série de teste reaplicada) providenciado o número de falhas presentes que não excedem a t.

As definições seguintes estão numa forma mais prática baseadas sobre o modelo pelo seguinte teorema. A

notação $\sqcap$ não vai ser usada para representar o operador de interseção cúbica definido em |11|.

TEOREMA 1

a) Um sistema S = ($\mathcal{F}$, $\mathcal{T}$, F, G) tem as falhas detectáveis por t se e somente se, para:

$F^k \varepsilon F(t)$, $F^k \neq \emptyset$ implica $G^k_j = 1$ para algum $t_j \varepsilon \mathcal{T}$

b) Um sistema S = ($\mathcal{F}$, $\mathcal{T}$, F, G) tem as falhas diagnosticáveis com reparo se e somente se, para qualquer

$F^i, F^j, \ldots, F^k \varepsilon F(t)$, $F^i \cap F^j \cap F^k = \emptyset$ implica

$G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$.

Demonstração:

Pela definição da matriz G, $G^k_j = 1$ se e somente se teste t_j falha exatamente para o padrão de falhas F^k presentes em S.

Assim a parte a) segue da definição da detectabilidade de falhas por t. A contrapositiva da parte b) é provocada. Primeiro note que uma linha de G representa todos comportamentos da série de testes para o correspondente padrão de falhas presentes no sistema. Tomando $F^i, F^j, \ldots, F^k \varepsilon F(t)$ e assumindo que $G^i \sqcap G^j \sqcap \ldots \sqcap G^k \neq 0$. Então cada um de

F^i, F^j, ..., F^k pode dar origem ao mesmo comportamento na série de testes. Se aquele comportamento de teste ocorre, é impossível determinar que padrão de falha está realmente presente. Portanto o sistema não é diagnosticável com reparo da falha por t . Reciprocamente, assumir que $G^i \sqcap G^j \sqcap \ldots \sqcap G^k \neq 0$ implica $F^i \cap F^j \cap \ldots \cap F^k \neq \emptyset$. Depois de cada aplicação da série de teste, um comportamento resulta que poderia ter sido próprio para a presença de qualquer um padrão de falhas em alguma série, dita $\{F^i, F^j, \ldots, F^k\} \subseteq F(t)$, determinável da matriz G . Mas existe, pela suposição, alguma falha f_i comum a cada padrão de falha naquela série, tal que f_i pode ser identificada e reparada. Portanto o sistema é diagnosticável com reparo da falha por t.

O teorema 1 pode ser usado diretamente para determinar detectabilidade e diagnosticabilidade da tabela de falhas generalizadas abordadas em |9|, |12|. Entretanto, oferece observação limitada dentro de fatores que afetam diagnosticabilidade e seu uso direto como um instrumento de projeto interativo pode ser um tanto inconveniente. Seu uso nesse documento seria apenas como base para a demonstração de maior legibilidade dos resultados adotados.

Uma relação evidente existe entre detectabilidade e diagnosticabilidade: uma condição necessária para um sistema S ser diagnosticável com reparo da falha por t é que seja detectável da falha por t.

- Índice de fechamento do sistema

Um parâmetro relacionado a invalidação de teste, o índice de fechamento do sistema, é definido aqui.

É uma medida da capacidade de executar testes válidos na presença de falhas. Como é evidenciado na próxima seção, o índice é em algum sentido um aspecto de qualidade no diagnóstico.

Uma padrão de falhas F^k é fechado se e somente se todo teste completo em T para cada falha em F^k é inválido na presença de F^k. Equivalentemente, F^k é fechado se e somente se $t(F^k) \subseteq T(F^k)$. A cardinalidade (tamanho) do menor padrão de falhas fechado em F é o "índice de fechamento do sistema", denotado por c(S). Desde que todo teste inválido sobre falhas no padrão possa passar, o índice de fechamento pode ser imaginado como o tamanho do menor padrão de falhas potencialmente não detectável. Pela convenção, $c(S) = \infty$ se nenhum padrão de falhas fechado existe. A condição para que uma falha não possa invalidar um teste completo da própria implica em que $2 \leq c(S) \leq n$ se é finito. Um sistema é dito a ser fechado se c(S) é finito, e completamente fechado se F é um padrão de falhas fechado.

O índice de fechamento pode ser determinado pelo exame da estrutura do gráfico de diagnóstico. Antes da apresentação dos resultados, algumas definições da teoria de gráficos são necessárias. Um gráfico dirigido é cíclico se ele

contem pelo menos um ciclo direto e acíclico do contrário. O perímetro de um gráfico dirigido D é o comprimento de seu menor ciclo dirigido, e é denotado por g(D). Se D é acíclico, então $g(D) = \infty$ por convenção. O raio de um gráfico dirigido D, denotado por r(D), é o menor número de vértices que atingem qualquer vértice em D.

Dois resultados que tratam das relações entre gráfico de sistemas de testes mórficos e seus índices de fechamento.

TEOREMA 2

Se $S = (F, T, F, D)$ é mórfico, então $c(S) \geq g(D)$. Esse resultado indica que se um sistema mórfico tem um padrão de falhas fechado, então seu gráfico de diagnóstico é cíclico. Reciprocamente, se um gráfico de diagnóstico é acíclico, então o sistema, se mórfico, tem $c(S) = \infty$.

TEOREMA 3

Se $S = (F, T, F, D)$ é mórfico e completamente fechado, então $c(S) \leq r(D)$.

Um algoritmo de programação linear zero-um para determinação de c(S) para sistemas mórficos é dado em |12|. Em adição, um método algébrico booleano para determinação de todos padrões de falhas fechadas e c(S) para sistemas semimórfi-

cos é dado em |12|.

Observação: Tanto os dois teoremas anteriores como os teoremas , lemas e corolário que vem a seguir estão demonstrados em |3|.

- Condições para diagnosticabilidade

Nesta seção, são mostrados resultados que relacionam o índice de fechamento de um sistema e sua diagnosticabilidade. A natureza dos resultados demonstram claramente o significado do índice. Também algumas relações fundamentais entre detetabilidade e diagnosticabilidade são estabelecidas.

O primeiro resultado mostra que a detetabilidade de um sistema é dependente somente de seu índice de fechamento.

TEOREMA 4

Um sistema S é "detetável de t falhas" se e somente se $c(S) \geq t + 1$. Para o caso especial de $t = n$, S é "detetável de n falhas" se e somente se $c(S) = \infty$.

Como foi notado previamente, a "detetabilidade de t falhas" é necessária para "diagnosticabilidade de t falhas". Assim a condição $c(S) \geq t + 1$ é necessária para a "diagnosticabilidade de t falhas com reparo". Uma condição extrema

relacionando detetabilidade e diagnosticabilidade para sistemas mórficos segue.

TEOREMA 5

Uma condição necessária para um sistema S ser "diagnosticável de t falhas com reparo" é que S seja "detetável de 2 t falhas" i.e., $c(S) \geq 2t + 1$.

Este resultado pode ser extendido aos sistemas semimórficos se a condição mórfica permanece para todos padrões de falha em $F(t) \subseteq F$. O sistema definido pela restrição de F para F(t) é $S(t) = (F, T, F(t), \hat{G})$ onde $\hat{G}$ é a parcela de G definida pelos padrões de falhas em F(t).

COROLÁRIO 1

Uma condição necessária para um sistema S semimórfico, para o qual S(t) é mórfico, ser "diagnosticável de t falhas com reparo" é para S_M, a aproximação mórfica para S, ter $c(S_M) \geq 2t + 1$.

Não é verdade, em geral, que as condições estabelecidas necessariamente para "diagnosticabilidade de t falhas" sejam também suficiente. Uma condição suficiente muito geral pode ser obtida com o auxílio dos dois lemas seguintes.

LEMA 1

Tomando $S = (F, T, F, G)$ ser semimórfico e tomando $F^i, F^j, \ldots, F^k \in F$ serem dois ou mais padrões de falhas tais que $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$. Se $|F^i \cup F^j \cup \ldots \cup F^k| < c(S)$, então $G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$.

LEMA 2

Tomando $\{F^i, F^j, \ldots, F^k\} \subseteq F(t)$ por qualquer série de dois ou mais padrões de falhas que é o mínimo em relação a propriedade que $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$. Então $|F^i \cup F^j \cup \ldots \cup F^k| \leq \lfloor\{(t+2)/2\}^2\rfloor$.

Os seguintes resultados estabelecem outras relações fundamentais entre detetabilidade e diagnosticabilidade: mostra que um sistema semimórfico com um grau suficientemente amplo de detetabilidade é automaticamente diagnosticável de t falhas com reparo.

TEOREMA 6

Uma condição suficiente para um sistema semimórfico S ser "diagnosticável de t falhas com reparo" é que S seja "detetável de $\gamma(t)$ falhas", i.e., $c(S) \geq \gamma(t) + 1$, onde $\gamma(t) = \lfloor\{(t+2)/2\}^2\rfloor$. Para o caso especial de $t = n$, S é "diagnosticável de n falhas com reparo" se e somente se $c(S) = \infty$, ou equivalentemente, se e somente se S é "detetável de n falhas".

Portanto, pelo Teorema 5, qualquer sistema mórfico que tem um gráfico de diagnóstico acíclico é "diagnosticável de n falhas com reparo". Tais sistemas aparecem, por exemplo da aplicação do diagnóstico do "carregador absoluto". Nessa aproximação uma pequena porção do sistema é garantida a ser livre de falhas por teste manual ou pela preparação ultra confiável, e é então usado para testar outra parte. A qualquer tempo durante a verificação, subsistemas que já foram testados e reparados se necessários são usados para testar outros, com a continuação do processo até que o sistema inteiro é verificado. Desde que nenhuma volta seja feita, tais sistemas orientam gráficos de diagnósticos acíclicos.

Usando o Teorema 4 e o fato de que $\gamma(t) = 2t$ para $t = 1, 2$ e 3, o seguinte corolário é imediato.

COROLÁRIO 2:

Para $t = 1, 2$ e 3 um sistema mórfico S é "diagnosticável de t falhas com reparo" se e somente se $c(S) \geq 2t + 1$.

É provado em |12| que o conjunto de condições dos teoremas 5 e 6 constituem a mais geral relação entre detetabilidade e diagnosticabilidade com reparo pelos sistemas mórficos. Isto é, baseado sobre o conhecimento do grau de detetabilidade de um sistema S sozinho, é possível decidir que S é (não é) "diagnosticável de t falhas com reparo" se e somente se S é

"detetável de $\gamma(t)$ falhas" (é não "detetável de 2t falhas"). As condições que são ambas necessariamente e suficiente para diagnosticabilidade com reparo para sistemas mórficos e $t \geq 4$ não são conhecidas (outras do que aquelas do teorema 1). Entretanto, informações adicionais podem algumas vezes ser utilizadas para avaliar diagnosticabilidade (ver |12| e |7|, teorema 4|.

Uma classe especial de sistemas para os quais alguns resultados interessantes tem sido obtidos é aquela de sistemas em que exista precisamente um teste completo para cada falha. Assim um sistema é chamado um sistema de teste-simples-por-falha (TSPF). Como o teorema 7 afirma, sistemas TSPF são particularmente para analisar via seus gráficos de diagnóstico.

TEOREMA 7:

Se D é o gráfico de diagnóstico de uma aproximação mórfica S_M de um semimórfico (TSPF) sistema S, então $c(S) \leq g(D)$.

Assim, pelo teorema 2, se $S = (F, T, F, D)$ é um sistema mórfico TSPF, então $c(S) = g(D)$. Consequentemente, um sistema mórfico TSPF S é detetável de t falhas se e somente se $g(D) \geq t + 1$. O principal resultado de diagnosticabilidade para sistemas TSPF requer o seguinte lema.

LEMA 3

Tomando S = (F, T, F, D) ser um sistema mórfico TSPF tal que D consiste de um ciclo simples. Se $g(D) \leq \gamma(t)$, então S não é diagnosticável de t-falhas com reparo.

O principal resultado mostra que a condição suficiente para a diagnosticabilidade de t falhas do teorema 6 é também necessária para sistemas mórficos TSPF.

TEOREMA 8:

Um sistema TSPF S é diagnosticável com reparo se e somente se $c(S) \geq \gamma(t) + 1$.

Para sistemas mórficos TSPF, a condição do teorema 8 é equivalente a $g(D) \geq \gamma(t) + 1$. Esse resultado pode ser extendido a sistemas semimórficos sob a mesma condição usada para desenvolver o Corolário 1.

COROLÁRIO 3:

Um sistema semimórfico TSPF S para o qual S(t) é mórfico e diagnosticável de t falhas com reparo se e somente se $c(S_M) \geq \gamma(t) + 1$.

- Conclusões

Pelo emprego de um modelo orignalmente proposto em |11|, um parâmetro chamado closure index é definido como que caracteriza de forma fundamental a capacidade de execução de testes válidos na presença de falhas. Resultados gerais são dados para permitir a determinação de detectabilidade e diagnosticabilidade com reparo de um sistema contendo no máximo t falhas sobre a base do parâmetro definido.

O modelo usado em |3| é completamente relacionado a vários outros, e essas relações são discutidas em alguns detalhes em |11|, |12|. Em particular os modelos de Preparata et al |7| e Seshagiri |6| são casos especiais do gráfico de diagnóstico no qual cada aresta representa um teste diferente . Além disso, os resultados em |4| - |6| são limitados a sistemas mórficos. Como uma consequência, esses resultados relacionados podem ser legivelmente estabelecidos usando os resultados mais gerais incluídos. Por exemplo, o principal resultado em "condições para diagnosticabilidade" de |7| é o uso imediato dos teoremas 6 e 8. A diagnosticabilidade do sistema hierárquico em |6|, um sistema que tem um simples gráfico de diagnóstico acíclico, segue do teorema 6.

O modelo presente tem limitações inerentes como assinaladas em Kime e Russel |3|. Uma parcela dessas limitações não são repartidas pelos modelos menos restritivos tais como a tabela de falhas generalizadas em |8|, |11|. Contudo, quando usando o modelo menos restritivo, maior esforço é requerido para obter informação e pouco discernimento na interação entre os componentes durante o diagnóstico é obtido. Assim, o modelo presente tem mérito em que providencia discernimento que é útil em projeto de sistemas diagnosticáveis. Além disso, é capaz de manipular uma ampla classe de sistemas maior do que outros modelos avaliáveis mesmo com suas limitações.

4.2 - Avaliação do Sistema de Testes do Processador de Instruções do Mini-Computador G11 de Acordo com o Modelo Proposto por Kime e Russel |3|

As variáveis do sistema narradas no capítulo 3, os pontos e testes, serão referenciados segundo a terminologia adotada na apresentação do modelo, com o objetivo de seguir a orientação proposta.

Os pontos que se encontram na última classificação de testes serão tratados como falhas e serão novamente relacionados, no sentido de possibilitar o preenchimento das matrizes.

Falhas da Primeira Classe:

Estas falhas possuem uma característica bas-

tante especial, que é comprometer o funcionamento de qualquer operação manual ou automática ou a realização dos testes das falhas pertencentes as classes restantes. Em termos de diagnósticos de falhas pertencentes as outras classes, as falhas de primeira classe exercem as mesmas influências e por este motivo serão agrupadas em uma única falha representada por f_1. O teste completo de f_1 será considerado o conjunto de testes dedicados as falhas pertencentes a primeira classe e representado por t_1.

As falhas correspondentes aos pontos anteriores 2.1 e 2.3 serão desdobrados. Os demais pontos terão apenas a substituição de sua representação por falha. A seguir está a relação das falhas correspondentes aos pontos.

Relação de Falhas:

f_1: Primeira classe de pontos considerada como uma falha

$f_{2.1}$: Todos os prefixos $f_{2.1}$ são relativos as micro-subrotinas de preparação de operando para as instruções curtas sem M0.

$f_{2.1.1}$: Micro-subrotina de preparação de operando CURT

$f_{2.1.2}$: Micro-subrotina de preparação de operando IMED

$f_{2.1.3}$: Micro-subrotina de preparação de operando PRVI

$f_{2.1.4}$: Micro-subrotina de preparação de operando DEP

$f_{2.2}$: Todos os prefixos $f_{2.2}$ são relativos as micro-subrotinas de preparação de operando das instruções curtas com M0, como só existe uma micro-subrotina de preparação de operando é considerado como ponto de falha o próprio prefixo f ....

$f_{2.3}$: Todos os prefixos $f_{2.3}$ são relativos as micro-subrotinas de preparação de operando das instruções longas sem M0.

$f_{2.3.1}$: Micro-subrotina de preparação de operando das instruções longas de desvio.

$f_{2.3.2}$: Micro-subrotina de preparação de operando das instruções longas imediatas.

$f_{2.4}$: Todos os prefixos $f_{2.4}$ são relativos a micro-subrotina de preparação de operando das instruções longas com M0 e ME ou BE.

$f_{2.4.1}$: Micro-subrotinas de preparação de operando das instruções longas com M0 e ME.

$f_{2.4.2}$: Micro-subrotinas de preparação de operando das instruções longas com M0 e BE.

$f_{2.5}$: Falha de interrupção por falha de alimentação

$f_{2.6}$: Falha de interrupção por falha de proteção

$f_{2.7}$: Falha de interrupção por teste

$f_{2.8}$: Falha de interrupção manual

$f_{3.1.1}$: Todas as falhas de prefixo $f_{3.1.1}$ são subordinadas a falha $f_{2.1.1}$.

$f_{3.1.1.1}$: Falha da função ACI (armazena curta).

$f_{3.1.2}$: Todas as falhas de prefixo $f_{3.1.2}$ são subordinadas a falha $f_{2.1.2}$.

$f_{3.1.2.1}$: Falha de função CGI (carrega imediato)

$f_{3.1.2.2}$: Falha da função ADI

$f_{3.1.2.3}$: Falha da função CAI

$f_{3.1.2.4}$: Falha da função SUI

$f_{3.1.3}$: Todas as falhas de prefixo $f_{3.1.3}$ são subordinadas a falha $f_{2.1.3}$.

$f_{3.1.3.1}$: Falha da função CHAVE

$f_{3.1.3.2}$: Falha da função DESPER

$f_{3.1.3.3}$: Falha da função MINT

$f_{3.1.3.4}$: Falha da função RINT

$f_{3.1.4}$: Todas as falhas de prefixo $f_{3.1.4}$ são subordinadas a falha $f_{2.1.4}$.

$f_{3.1.4.1}$: Falha da função PP

$f_{3.1.4.2}$: Falha da função PD

$f_{3.1.4.3}$: Falha da função P

$f_{3.1.3.5}$: Falha da função CGMI

$f_{3.1.3.6}$: Falha da função CSUP

$f_{3.1.4}$: Todas as falhas de prefixo $f_{3.1.4}$ são subordinadas a falha $f_{2.1.4}$.

$f_{3.2}$: Todas as falhas de prefixo $f_{3.2}$ são subordinadas a falha $f_{2.2}$.

$f_{3.2.1}$: Falha da função TIPS

$f_{3.2.2}$: Falha da função CAR

$f_{3.2.3}$: Falha da função CGR

$f_{3.2.4}$: Falha da função AR

$f_{3.2.5}$: Falha da função MLS

⋮

$f_{3.2.11}$: Falha da função CRT

$f_{3.2.12}$: Falha da função ISR

⋮

$f_{3.2.17}$: Falha da função MAP

$f_{3.3.1}$: Todas as falhas de prefixo $f_{3.3}$ são subordinadas a falha $f_{2.3.1}$.

$f_{3.3.1.1}$: Falha da função DMA

$f_{3.3.1.2}$: Falha da função DI

$f_{3.3.1.3}$: Falha da função DMAI

$f_{3.3.1.4}$: Falha da função DME

$f_{3.3.1.5}$: Falha da função DMEI

$f_{3.3.2}$: Todas as falhas de prefixo $f_{3.3.2}$ são subordinas a falha $f_{2.3.2}$.

$f_{3.3.2.1}$: Falha da função ADM

$f_{3.4.1}$: Todas as falhas de prefixo $f_{3.4.1}$ são subordinadas a falha $f_{2.4.1}$.

$f_{3.4.1.1}$: Falha da função AD

$f_{3.4.1.2}$: Falha da função E

$f_{3.4.1.3}$: Falha da função SU

$f_{3.4.1.4}$: Falha da função MLS

$f_{3.4.1.5}$: Falha da função OU

⋮

$f_{3.4.1.10}$Falha da função CL

$f_{3.4.2}$: Todas as falhas de prefixo $f_{3.4.2}$ são subordinadas a falha $f_{2.4.2}$.

$f_{3.4.2.1}$: Falha da função CLB

$f_{3.4.2.2}$: Falha da função CAB

⋮

$f_{3.4.2.8}$: Falha da função OUXB

f_4: Falhas de prefixo f_4 são as falhas das instruções que não foram incluídas nos testes das funções, mas cujas conexões necessitam de testes.

Assim como as falhas de prefixo f_3 dependem das falhas de prefixo f_2, as falhas de prefixo f_4 também dependem das falhas de prefixo f_2 e a mesma notação vai ser usada na relação de falhas de prefixo f_4.

Relação dos testes completos para cada falha

A cada falha está vinculado um teste completo. Os testes tem a mesma notação que as falhas, a única diferença é que no lugar de f aparece t.

Por exemplo: Teste de falha f_1 é chamado de t_1.

Denominando o sistema de testes desenvolvidos no capítulo 3 de S. Considerando que S teste as n falhas do conjunto $F = \{f_1, f_{2.1.1}, \ldots, f_4 \ldots\}$ pelo conjunto de testes completos $T = \{t_1, t_{2.1.1}, \ldots, t_4 \ldots\}$, é preciso que seja verificada a validade do sistema junto a localização e reparo de falhas.

A avaliação então compreende o preenchimento das três matrizes e a análise de seu conteúdo de acordo com as exigências relacionadas no modelo descrito neste capítulo.

Preenchimento das matrizes em mapas anexos

Matriz F: Matriz de padrões de falhas.

Matriz C: Matriz dos testes completos de falhas.

Matriz T: Matriz dos testes inválidos por falhas.

Matriz G: Matriz generalizada de falhas.

MATRIZ DE CONFIGURACOES DE FALHAS F

F(I,J) = 0, INDICA QUE NAO HA' FALHA EM J
F(I,J) = 1, INDICA QUE HA' FALHA EM J

OBS. ESTA MATRIZ POSSUI (2 ** N) LINHAS, ESTAO REPRESENTADAS NESTE MAPA APENAS AS CONFIGURACOES NECESSARIAS A AVALIACAO DO SISTEMA , CUJO CRITERIO DE SELECAO FOI O DE EVITAR REPETICAO DE COMPORTAMENTOS.

```
        /          F A L H A S
        /
        / 12222222222222333333333333333333...33...333333333333...333...34...
        /  ..........................................................................
CONFIGU-/  11112334456781111111111111122222...22...233333344444...444...4.....
RACOES  /  .... ....    ..................... ..................................
        /  1234 1212    12222333333344412345...11...111111211111...122........
  DE    /               ..............      ...12...7.............................
        /               11234123456123      ...  ... 12345112345...112........
FALHAS  /                                                          0
```

```
F01     / 10000000000000000000000000000000...00...000000000000...000...00...
F02     / 01000000000000000000000000000000...00...000000000000...000...00...
F03     / 00100000000000000000000000000000...00...000000000000...000...00...
F04     / 00010000000000000000000000000000...00...000000000000...000...00...
F05     / 00001000000000000000000000000000...00...000000000000...000...00...
F06     / 00000100000000000000000000000000...00...000000000000...000...00...
F07     / 00000010000000000000000000000000...00...000000000000...000...00...
F08     / 00000001000000000000000000000000...00...000000000000...000...00...
F09     / 00000000100000000000000000000000...00...000000000000...000...00...
F10     / 00000000010000000000000000000000...00...000000000000...000...00...
F11     / 00000000001000000000000000000000...00...000000000000...000...00...
F12     / 00000000000100000000000000000000...00...000000000000...000...00...
F13     / 00000000000010000000000000000000...00...000000000000...000...00...
F14     / 00000000000001000000000000000000...00...000000000000...000...00...
F15     / 00000000000000100000000000000000...00...000000000000...000...00...
F16     / 00000000000000010000000000000000...00...000000000000...000...00...
F17     / 00000000000000001000000000000000...00...000000000000...000...00...
F18     / 00000000000000000100000000000000...00...000000000000...000...00...
F19     / 00000000000000000010000000000000...00...000000000000...000...00...
F20     / 00000000000000000001000000000000...00...000000000000...000...00...
F21     / 00000000000000000000100000000000...00...000000000000...000...00...
F22     / 00000000000000000000010000000000...00...000000000000...000...00...
F23     / 00000000000000000000001000000000...00...000000000000...000...00...
F24     / 00000000000000000000000100000000...00...000000000000...000...00...
F25     / 00000000000000000000000010000000...00...000000000000...000...00...
F26     / 00000000000000000000000001000000...00...000000000000...000...00...
F27     / 00000000000000000000000000100000...00...000000000000...000...00...
F28     / 00000000000000000000000000010000...00...000000000000...000...00...
F29     / 00000000000000000000000000001000...00...000000000000...000...00...
F30     / 00000000000000000000000000000100...00...000000000000...000...00...
F31     / 00000000000000000000000000000010...00...000000000000...000...00...
F32     / 00000000000000000000000000000001...00...000000000000...000...00...
F33     / 00000000000000000000000000000000...00...000000000000...000...00...
  .     / ..........................................................................
  .     / ..........................................................................
  .     / ..........................................................................
F34     / 00000000000000000000000000000000...10...000000000000...000...00...
F35     / 00000000000000000000000000000000...01...000000000000...000...00...
  .     / ..........................................................................
  .     / ..........................................................................
  .     / ..........................................................................
F36     / 00000000000000000000000000000000...00...100000000000...000...00...
```

MATRIZ DE CONFIGURACOES DE FALHAS F

F(I,J) = 0, INDICA QUE NAO HA' FALHA EM J
F(I,J) = 1, INDICA QUE HA' FALHA EM J

```
                /          F A L H A S
                /
                / 12222222222222333333333333333333333...33...333333333333...333...34...
                /  ..........................................................................
CONFIGU-/  1111233445678111111111111111122222...22...233333344444...444...4....
RACOES  /  .... ....    ..................... ..............................
        /  1234 1212    122223333334412345...11...111111211111...122........
  DE    /               ..............      ...12...7.........................
        /               11234123456123      ...  ... 12345112345...112........
FALHAS  /                                                          0
--------------------------------------------------------------------------------------------
F37     / 00000000000000000000000000000000000...00...010000000000...000...00...
F38     / 00000000000000000000000000000000000...00...001000000000...000...00...
F39     / 00000000000000000000000000000000000...00...000100000000...000...00...
F40     / 00000000000000000000000000000000000...00...000010000000...000...00...
F41     / 00000000000000000000000000000000000...00...000001000000...000...00...
F42     / 00000000000000000000000000000000000...00...000000100000...000...00...
F43     / 00000000000000000000000000000000000...00...000000010000...000...00...
F44     / 00000000000000000000000000000000000...00...000000001000...000...00...
F45     / 00000000000000000000000000000000000...00...000000000100...000...00...
F46     / 00000000000000000000000000000000000...00...000000000010...000...00...
F47     / 00000000000000000000000000000000000...00...000000000001...000...00...
  .     / ..........................................................................
  .     / ..........................................................................
  .     / ..........................................................................
F48     / 00000000000000000000000000000000000...00...000000000000...100...00...
F49     / 00000000000000000000000000000000000...00...000000000000...010...00...
F50     / 00000000000000000000000000000000000...00...000000000000...001...00...
  .     / ..........................................................................
  .     / ..........................................................................
  .     / ..........................................................................
F51     / 00000000000000000000000000000000000...00...000000000000...000...10...
F52     / 00000000000000000000000000000000000...00...000000000000...000...01...
  .     / ..........................................................................
  .     / ..........................................................................
  .     / ..........................................................................
F53     / 01000000000000100000000000000000000...00...000000000000...000...00...
F54     / 00100000000000010000000000000000000...00...000000000000...000...00...
F55     / 00100000000000001000000000000000000...00...000000000000...000...00...
F56     / 00100000000000000100000000000000000...00...000000000000...000...00...
F57     / 00010000000000000010000000000000000...00...000000000000...000...00...
F58     / 00010000000000000001000000000000000...00...000000000000...000...00...
F59     / 00000100000000000000000000010000000...00...000000000000...000...00...
F60     / 00000100000000000000000000001000000...00...000000000000...000...00...
F61     / 00000001000000000000000000000000000...00...000000100000...000...00...
F62     / 00000000100000000000000000000000000...00...000000010000...000...00...
F63     / 00000000100000000000000000000000000...00...000000001000...000...00...
F64     / 00000000100000000000000000000000000...00...000000000100...000...00...
F65     / 00000000100000000000000000000000000...00...000000000010...000...00...
F66     / 00000000010000000000000000000000000...00...000000000000...010...00...
F67     / 00000000010000000000000000000000000...00...000000000000...001...00...
F68     / 00000000010000000000000000000000000...00...000000000000...000...10...
F69     / 11000000000000000000000000000000000...00...000000000000...000...00...
F70     / 00100000000000011000000000000000000...00...000000000000...000...00...
F71     / 00010100000000000000100000010000000...00...000000000000...000...00...
--------------------------------------------------------------------------------------------
```

MATRIZ DE TESTES INVALIDOS

T(I,J) = 0, INDICA QUE TESTE J E' VALIDO P/ CONF. I
T(I,J) = 1, INDICA QUE TESTE J E' INVALIDO P/ CONF. I

CONFIGURACOES DE FALHAS	TESTES
	12222222222222333333333333333333...33...333333333333...333...34...
	.....
	11112334456781111111111111122222...22...233333344444...444...4....
	....
	1234 1212 122223333334441234 5...11...111111211111...122.....
	....12...7.....
	11234123456123 ... ... 12345112345...112.....
	0
F01	01111111111111111111111111111111...11...111111111111...111...11...
F02	00000000000000100000000000000000...00...000000000000...000...00...
F03	00000000000000011110000000000000...00...000000000000...000...00...
F04	00000000000000000001111110000000...00...000000000000...000...00...
F05	00000000000000000000000011100000...00...000000000000...000...00...
F06	00000000000000000000000000011111...11...100000000000...000...00...
F07	00000000000000000000000000000000...00...011111000000...000...00...
F08	00000000000000000000000000000000...00...000000100000...000...00...
F09	00000000000000000000000000000000...00...000000011111...100...00...
F10	00000000000000000000000000000000...00...000000000000...011...10...
F11	00000000000000000000000000000000...00...000000000000...000...00...
F12	00000000000000000000000000000000...00...000000000000...000...00...
F13	00000000000000000000000000000000...00...000000000000...000...00...
F14	00000000000000000000000000000000...00...000000000000...000...00...
F15	00000000000000000000000000000000...00...000000000000...000...00...
F16	00000000000000000000000000000000...00...000000000000...000...00...
F17	00000000000000000000000000000000...00...000000000000...000...00...
F18	00000000000000000000000000000000...00...000000000000...000...00...
F19	00000000000000000000000000000000...00...000000000000...000...00...
F20	00000000000000000000000000000000...00...000000000000...000...00...
F21	00000000000000000000000000000000...00...000000000000...000...00...
F22	00000000000000000000000000000000...00...000000000000...000...00...
F23	00000000000000000000000000000000...00...000000000000...000...00...
F24	00000000000000000000000000000000...00...000000000000...000...00...
F25	00000000000000000000000000000000...00...000000000000...000...00...
F26	00000000000000000000000000000000...00...000000000000...000...00...
F27	00000000000000000000000000000000...00...000000000000...000...00...
F28	00000000000000000000000000000000...00...000000000000...000...00...
F29	00000000000000000000000000000000...00...000000000000...000...00...
F30	00000000000000000000000000000000...00...000000000000...000...00...
F31	00000000000000000000000000000000...00...000000000000...000...00...
F32	00000000000000000000000000000000...00...000000000000...000...00...
F33	00000000000000000000000000000000...00...000000000000...000...00...
.	.....
.	.....
.	.....
F34	00000000000000000000000000000000...00...000000000000...000...00...
F35	00000000000000000000000000000000...00...000000000000...000...00...
.	.....
.	.....
.	.....
F36	00000000000000000000000000000000...00...000000000000...000...00...

MATRIZ DE TESTES INVALIDOS T

T(I,J) = 0, INDICA QUE TESTE J E' VALIDO P/CONF. I
T(I,J) = 1, INDICA QUE TESTE J E' INVALIDO P/CONF. I

CONFIGURACOES DE FALHAS	TESTES
	12222222222222333333333333333333...33...333333333333...333...34...
	.....
	11112334456781111111111111122222...22...233333344444...444...4....
	....
	1234 1212 12222333333444123 45...11...111111211111...122.....
	....12...7.....
	11234123456123 ... ... 12345112345...112.....
	0
F37	00000000000000000000000000000000...00...000000000000...000...00...
F38	00000000000000000000000000000000...00...000000000000...000...00...
F39	00000000000000000000000000000000...00...000000000000...000...00...
F40	00000000000000000000000000000000...00...000000000000...000...00...
F41	00000000000000000000000000000000...00...000000000000...000...00...
F42	00000000000000000000000000000000...00...000000000000...000...00...
F43	00000000000000000000000000000000...00...000000000000...000...00...
F44	00000000000000000000000000000000...00...000000000000...000...00...
F45	00000000000000000000000000000000...00...000000000000...000...00...
F46	00000000000000000000000000000000...00...000000000000...000...00...
F47	00000000000000000000000000000000...00...000000000000...000...00...
.	.....
.	.....
.	.....
F48	00000000000000000000000000000000...00...000000000000...000...00...
F49	00000000000000000000000000000000...00...000000000000...000...00...
F50	00000000000000000000000000000000...00...000000000000...000...00...
.	.....
.	.....
.	.....
F51	00000000000000000000000000000000...00...000000000000...000...00...
F52	00000000000000000000000000000000...00...000000000000...000...00...
.	.....
.	.....
.	.....
F53	00000000000000100000000000000000...00...000000000000...000...00...
F54	00000000000000011110000000000000...00...000000000000...000...00...
F55	00000000000000011110000000000000...00...000000000000...000...00...
F56	00000000000000011110000000000000...00...000000000000...000...00...
F57	00000000000000000011111100000000...00...000000000000...000...00...
F58	00000000000000000011111100000000...00...000000000000...000...00...
F59	00000000000000000000000000011111...11...100000000000...000...00...
F60	00000000000000000000000000011111...11...100000000000...000...00...
F61	00000000000000000000000000000000...00...000000100000...000...00...
F62	00000000000000000000000000000000...00...000000011111...100...00...
F63	00000000000000000000000000000000...00...000000011111...100...00...
F64	00000000000000000000000000000000...00...000000011111...100...00...
F65	00000000000000000000000000000000...00...000000011111...100...00...
F66	00000000000000000000000000000000...00...000000000000...011...10...
F67	00000000000000000000000000000000...00...000000000000...011...10...
F68	00000000000000000000000000000000...00...000000000000...011...10...
F69	11111111111111111111111111111111...11...111111111111...111...11...
F70	00000000000000011110000000000000...00...000000000000...000...00...
F71	00000000000000000011111100011111...11...100000000000...000...00...

MATRIZ GENERALIZADA DE TESTES G

G(I,J) = 0, INDICA QUE TESTE J NAO ACUSA FALHA NA CONFIGURACAO I
G(I,J) = 1, INDICA QUE TESTE J ACUSA FALHA NA CONFIGURACAO I
G(I,J) = X, INDICA QUE TESTE J TEM RESULTADO IMPREVISTO NA CONFIGURACAO I

CONFIGURACOES DE FALHAS	TESTES
	1222222222222333333333333333333...33...333333333333...333...34...
	.....
	1111233445678111111111111122222...22...233333344444...444...4....
	....
	1234 1212 122223333333444123 45...11...111111211111...122.....
	....
	...12...7.....
	11234123456123 ... ... 12345112345...112.....
	0
F01	1XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XX...XXXXXXXXXXXX...XXX...XX...
F02	0100000000000X0000000000000000000...00...000000000000...000...00...
F03	0010000000000000XXXX00000000000000...00...000000000000...000...00...
F04	0001000000000000000XXXXXX00000000...00...000000000000...000...00...
F05	0000100000000000000000000XXX00000...00...000000000000...000...00...
F06	0000010000000000000000000000XXXXX...XX...X00000000000...000...00...
F07	0000001000000000000000000000000000...00...0XXXXX000000...000...00...
F08	0000000100000000000000000000000000...00...000000X00000...000...00...
F09	0000000010000000000000000000000000...00...0000000XXXXX...X00...00...
F10	0000000001000000000000000000000000...00...000000000000...0XX...X0...
F11	0000000000100000000000000000000000...00...000000000000...000...00...
F12	0000000000010000000000000000000000...00...000000000000...000...00...
F13	0000000000001000000000000000000000...00...000000000000...000...00...
F14	0000000000000100000000000000000000...00...000000000000...000...00...
F15	0000000000000010000000000000000000...00...000000000000...000...00...
F16	0000000000000001000000000000000000...00...000000000000...000...00...
F17	0000000000000000100000000000000000...00...000000000000...000...00...
F18	0000000000000000010000000000000000...00...000000000000...000...00...
F19	0000000000000000001000000000000000...00...000000000000...000...00...
F20	0000000000000000000100000000000000...00...000000000000...000...00...
F21	0000000000000000000010000000000000...00...000000000000...000...00...
F22	0000000000000000000001000000000000...00...000000000000...000...00...
F23	0000000000000000000000100000000000...00...000000000000...000...00...
F24	0000000000000000000000010000000000...00...000000000000...000...00...
F25	0000000000000000000000001000000000...00...000000000000...000...00...
F26	0000000000000000000000000100000000...00...000000000000...000...00...
F27	0000000000000000000000000010000000...00...000000000000...000...00...
F28	0000000000000000000000000001000000...00...000000000000...000...00...
F29	0000000000000000000000000000100000...00...000000000000...000...00...
F30	0000000000000000000000000000010000...00...000000000000...000...00...
F31	0000000000000000000000000000001000...00...000000000000...000...00...
F32	0000000000000000000000000000000100...00...000000000000...000...00...
F33	0000000000000000000000000000000010...00...000000000000...000...00...
.	.....
.	.....
.	.....
F34	0000000000000000000000000000000001...10...000000000000...000...00...
F35	0000000000000000000000000000000000...01...000000000000...000...00...
.	.....
.	.....
.	.....
F36	0000000000000000000000000000000000...00...100000000000...000...00...

```
-------------------------------------------------------------------------------
MATRIZ GENERALIZADA DE TESTES G
-------------------------------
G(I,J) = 0, INDICA QUE TESTE J NAO ACUSA FALHA NA CONFIGURACAO I
G(I,J) = 1, INDICA QUE TESTE J ACUSA FALHA NA CONFIGURACAO I
G(I,J) = X, INDICA QUE TESTE J TEM RESULTADO IMPREVISTO NA CONFIGURACAO I
-------------------------------------------------------------------------------
        /          TESTES
        /
        / 122222222222223333333333333333333...33...333333333333...333...34...
        /  ..................................................................
CONFIGU-/  11112334456781111111111111122222...22...233333344444...444...4....
RACOES  /  .... ....    ................... .................................
        /  1234 1212    1222233333344412345...11...111111211111...122........
  DE    /               ..............     ...12...7.........................
        /               11234123456123     ...  ... 12345112345...112........
FALHAS  /                                                         0
-------------------------------------------------------------------------------
F37     / 000000000000000000000000000000000...00...010000000000...000...00...
F38     / 000000000000000000000000000000000...00...001000000000...000...00...
F39     / 000000000000000000000000000000000...00...000100000000...000...00...
F40     / 000000000000000000000000000000000...00...000010000000...000...00...
F41     / 000000000000000000000000000000000...00...000001000000...000...00...
F42     / 000000000000000000000000000000000...00...000000100000...000...00...
F43     / 000000000000000000000000000000000...00...000000010000...000...00...
F44     / 000000000000000000000000000000000...00...000000001000...000...00...
F45     / 000000000000000000000000000000000...00...000000000100...000...00...
F46     / 000000000000000000000000000000000...00...000000000010...000...00...
F47     / 000000000000000000000000000000000...00...000000000001...000...00...
   .    / ...................................................................
   .    / ...................................................................
   .    / ...................................................................
F48     / 000000000000000000000000000000000...00...000000000000...100...00...
F49     / 000000000000000000000000000000000...00...000000000000...010...00...
F50     / 000000000000000000000000000000000...00...000000000000...001...00...
   .    / ...................................................................
   .    / ...................................................................
   .    / ...................................................................
F51     / 000000000000000000000000000000000...00...000000000000...000...10...
F52     / 000000000000000000000000000000000...00...000000000000...000...01...
   .    / ...................................................................
   .    / ...................................................................
   .    / ...................................................................
F53     / 01000000000000X000000000000000000...00...000000000000...000...00...
F54     / 001000000000000XXXX00000000000000...00...000000000000...000...00...
F55     / 001000000000000XXXX00000000000000...00...000000000000...000...00...
F56     / 001000000000000XXXX00000000000000...00...000000000000...000...00...
F57     / 0001000000000000000XXXXXX00000000...00...000000000000...000...00...
F58     / 0001000000000000000XXXXXX00000000...00...000000000000...000...00...
F59     / 0000100000000000000000000000XXXXX...XX...X00000000000...000...00...
F60     / 0000100000000000000000000000XXXXX...XX...X00000000000...000...00...
F61     / 000000010000000000000000000000000...00...000000X00000...000...00...
F62     / 000000001000000000000000000000000...00...0000000XXXXX...X00...00...
F63     / 000000001000000000000000000000000...00...0000000XXXXX...X00...00...
F64     / 000000001000000000000000000000000...00...0000000XXXXX...X00...00...
F65     / 000000001000000000000000000000000...00...0000000XXXXX...X00...00...
F66     / 000000000100000000000000000000000...00...000000000000...0XX...X0...
F67     / 000000000100000000000000000000000...00...000000000000...0XX...X0...
F68     / 000000000100000000000000000000000...00...000000000000...0XX...X0...
F69     / 1XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XX...XXXXXXXXXXXX...XXX...XX...
F70     / 001000000000000XXXX00000000000000...00...000000000000...000...00...
F71     / 0001010000000000000XXXXXX000XXXXX...XX...X00000000000...000...00...
-------------------------------------------------------------------------------
```

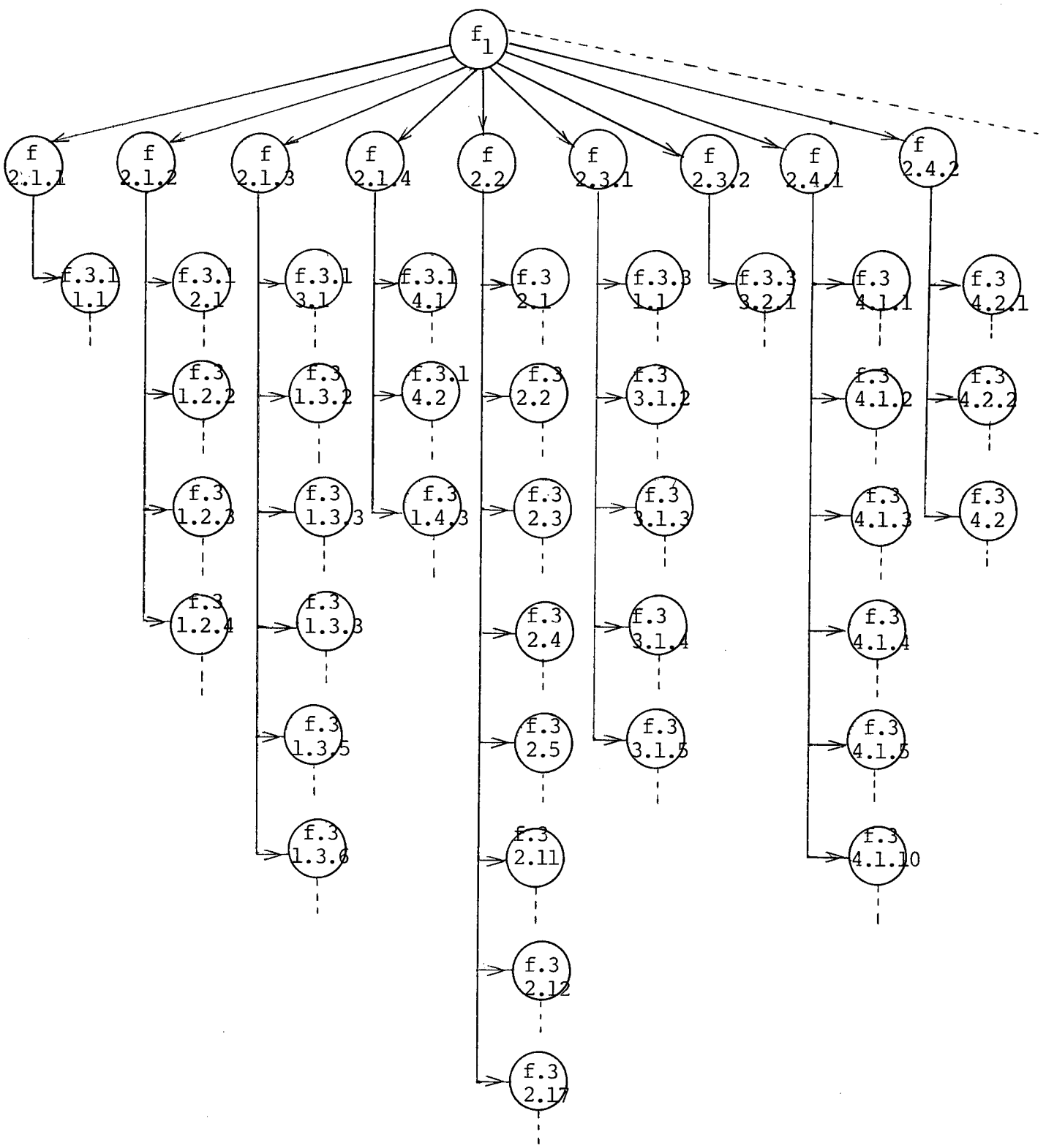


Figura 4 - Gráfico de diagnóstico de Falhas D

A matriz C não foi preenchida devido a cada falha f ser testada por um teste completo t produzindo uma matriz de diagonal principal igual 1.

As matrizes F, T e G não foram preenchidas para as 2^n linhas como sugere o modelo, a omissão da maioria das linhas mereceu um critério baseado na repetição de comportamento dos testes. O número de padrões de falhas preenchido é suficiente para a análise de sistema, já que os casos mais extremos constam nas apresentações das matrizes.

Análise do comportamento do sistema refletido nas matrizes T e G

1 - Falhas detetáveis por t:

Segundo o teorema 1:
Para qualquer $F^k \varepsilon F(t)$, $F^k = \emptyset$ implica em $G^k_j = 1$ para algum $t_j \varepsilon T$.

De acordo com a matriz G obtida, todas as linhas contém o valor "1" uma vez, o que permite concluir que o sistema S é detetável.

2 - Falhas diagnosticáveis com reparo:

Para qualquer $F^i, F^j, \ldots, F^k \varepsilon F(t)$, $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$ implica em $G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$, para todos padrões que não tem falhas em comum a intersessão de seus resultados de

testes será vazia:

Por exemplo:

$F1 \cap F2 \cap \ldots \cap F52 = \emptyset$

$G1 \sqcap G2 \sqcap \ldots \sqcap G52 = \emptyset$

$F1 \cap F53 \cap F54 = \emptyset$

$G1 \sqcap G53 \sqcap G54 = \emptyset$

Para todas as combinações onde $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$ a intercessão cúbica de G^1, G^j, G^k resulta em $\emptyset$, o que permite concluir que o sistema S é diagnosticável com reparo.

3 - Características do sistema:

a - Condições mórfica:

$T(F^i \cup F^j) = T(F^i) \cup T(F^j)$ para todo F^i, $F^j \in F$.

$T(F^k) = \cup T(f_i)$ sobre todo $f_i \in F^k$

Ex. 1: F^1 e F^2

$T(F^1) = 0 \quad 1 \quad 1 \quad \ldots\ldots \quad 1$

$T(F^2) = 0 \quad 0 \quad \ldots\ldots \quad 1 \quad 0 \quad \ldots$

$T_{69} = T(F^i \cup F^2) = 0 \quad 1 \quad 1 \quad \ldots \quad 1 \ldots$

$T(F_1) \cup T(F_2) = \quad 0 \quad 1 \quad 1 \quad \ldots \quad 1 \ldots \quad = T(F^1 \cup F^2)$

Ex. 2: F54 e F55

$T(F54) = 0 \ldots 0\ 1\ 1\ 1\ 1\ 0 \ldots\ldots 0$

$T(F55) = 0 \ldots 0\ 1\ 1\ 1\ 1\ 0 \ldots\ldots 0$

$F(70) = F54 \cup F55$

$T(F70) = 0 \ldots 0\ 1\ 1\ 1\ 1\ 0 \ldots 0 = T(F54) \cup T(F55)$

Ex. 3: F58 e F59

$T(F58) = 0 \ldots 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \ldots$

$T(F59) = 0 \ldots 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0 \ldots$

$F71 = F58 \cup F59 = T(F71) = T(F58) \cup T(F59)$

A observação da matriz T permite concluir que a aplicação da união a qualquer par de linhas T^i, T^j apresenta como resultado uma linha igual a linha $T(F^i \cup F^j)$.

Observando a segunda condição $T(F^k) = \cup T(f_i)$ $f_i \in F^k$

Ex. 1:

F54

$T(F54) = 0 \ldots 0\ 1\ 1\ 1\ 1\ 0 \ldots$

$T(f_{2.1.2}) = 0 \ldots 0\ 1\ 1\ 1\ 1\ 0 \ldots$

$T(f_{3.1.2.1}) = 0 \ldots 0\ 0\ 0\ 0\ 0\ 0 \ldots$

$T(F54) = T(f_{2.1.2} \cup f_{3.1.2.1}) = 0\ 0 \ldots 0\ 1\ 1\ 1\ 1\ 0 \ldots$

Para todos F^k a condição $T(F^k) = \cup T(f_i)$ sobre todos $f_i \in F^k$ ocorre no sistema S permitindo concluir que o sistema é mórfico.

Esta última condição admite a representação do sistema em um gráfico.

4 - Determinação do índice de fechamento:

a - Levantamento dos padrões de falhas F^k considerados fechados:

Condição para um padrão de falhas F^k ser fechado é que $t(F^k) \subseteq T(F^k)$.

Para Fl:

$t(Fl) = t_1$

$t(Fl) = t_{2.1.1} \cdots t_4$

$t(Fl) \not\subseteq T(Fl)$

Para F2:

$t(F2) = t_{2.1.1}$

$T(F2) = t_{3.1.1.1}$

$t(F2) \not\subseteq T(F2)$

Para F65:

$t(F65) = t_{2.4.1}$

$T(F65) = t_{3.4.1.1}, t_{3.4.1.2}, t_{3.4.1.3}, t_{3.4.1.4}, t_{3.4.1.5}, \cdots t_{3.4.1.10}$.

$t(F65) \not\subseteq T(F65)$

Para qualquer padrão do sistema S, $t(F^k) \not\subseteq T(F^k)$ i.e., o sistema S não contém padrão de falhas F^k que seja fechado.

Este resultado permite concluir que: 1) O índice de fechamento do sistema c(S) é maior que n, portanto $c(S) = \infty$; 2) O gráfico representativo do sistema é um gráfico acíclico.

CAPÍTULO V

CONCLUSÕES

O levantamento dos testes para o processador de instruções do mini-computador G11 seguiu um curso cuja tendência era obter uma relação ordenada de testes de acordo com a dependência de falhas.

O desenvolvimento neste caminho além de apresentar êxito na questão da necessidade dos testes apareceram numa sequência cujos instrumentos estivessem livres de erro, proporcionou também uma racionalização dos mesmos evitando que algum ponto recebesse algum teste específico por mais de uma vez , sem necessidade.

Por outro lado, repetir a classificação dos pontos por três vezes, até se alcançar uma sequência que capacitasse o sistema a localizar e dar condições de reparo as falhas, mostrou-se uma tarefa exaustiva.

Acompanhando o processo de reclassificação , verifica-se que as condições de localização e raparo são perceptíveis dando margem a transformar o processo realizado quase que a sentimento em um método mais operacional que gere a lista final de testes na ordem necessária.

Tomando por bases as condições de deteção , diagnóstico e reparo enumerados por Kime e Russel |3| transcri-

tas no capítulo 4, pode ser criado um método que vai auxiliar em grande parte o trabalho de desenvolvimento de testes. O método em linhas gerais pode ser expresso como segue:

A partir da lista primitiva de falhas acompanhada das necessidades de recursos de seus testes completos correspondentes, preenche-se as matrizes

F - matriz de configuração de falhas
T - matriz de testes inválidos
G - matriz generalizada de testes

maiores detalhes das matrizes encontram-se no capítulo 4.

As matrizes são então, submetidas a avaliação, caso não haja concordância com as condições necessárias a deteção e diagnósticos com reparo, o sistema passa por uma análise através das próprias matrizes, com a finalidade de serem obtidas novas falhas e novos testes para que novamente através do preenchimento das matrizes se repita o processo. Desta forma , aplica-se o processo sucessivamente, até que o sistema atinja a capacidade de detetar e diagnosticar qualquer falha que permita reparo.

Este método sugerido pode ser dividido em quatro etapas principais:

ETAPA 1 - Preenchimento das matrizes

ETAPA 2: Análise das matrizes observando o comportamento dos testes segundo os critérios de localização e reparo de falhas.

Se a análise apresenta resultado satisfatório deve proceder a última etapa que é determinação da sequência dos testes.

ETAPA 3: A partir da última avaliação gerar nova lista de pontos com base na necessidade de reunir certos pontos ou estender outros bem como acrescentar ou anular testes completos. Proceder novamente a partir da etapa 1.

ETAPA 4: Determinação da sequência final observando a necessidade das falhas estarem livres de erro para que os testes sejam realizados e validados.

Com o auxílio deste processo ou outro que tenha o objetivo de otimizar e determinar a sequência de testes o trabalho de desenvolvimento dos testes fica bastante facilitado ao menos nesta parte, restando maior peso a determinação de cada teste completo.

É possível se criar um processo automático, um programa cuja lógica siga a filosofia do processo esboçado nas quatro etapas principais, sua implementação não é muito complexa, é quase uma tradução das exigências de detetabilidade e diagnosticabilidade relacionados no capítulo 4.

APÊNDICE 1

CONJUNTO DE INSTRUCOES DO PROCESSADOR DO MINICOMPUTADOR G11

DESCRICAO DA INSTRUCAO	FMTO.	SINTAXE		SEMANTICA	ESTADO
ARMAZENA	L1	A	RR,RM	(RM)<-(RR)	
ARMAZENA ABSOLUTO	L3	AA	RR,RM	(RM)<-(RR)	
ARMAZENA BYTE	L1	AB	R,RM	(RM)<-(R(0:7))	
ARMAZENA CURTO	C11	AC	R,E	(BL+D)<-(R)	
ARMAZENA REGISTRADOR	C1	AR	RR,RM	(RR)<-(R)	
CARREGA	L1	CG	RR,RM	(RR)<-(RM)	PZSET
CARREGA ABSOLUTO	L3	CGA	RR,RM	(RR)<-(RM)	PZSET
CARREGA BYTE	L1	CGB	R,RM	(R<0:7>)<-(RM) (R<8:15>)<-SINAL	PZSET
CARREGA CURTO	C11	CGC	R,E	(R)<-(BL+D)	PZSET
CARREGA INDICE DA PILHA DE CONTEXTO	C10	CGIC	RR	(IPC)<-(RR)	
CARREGA IMEDIATO	C11	CGI	R,I	(R<0:7>)<-I	PZSET
CARREGA MASCARA DE INTERRUPCAO	C8	CGMI	I	(RE<8:15>)<-I	
CARREGA MENOS UM	C10	CGMU	RR	(RR)<- -1	PZSET
CARREGA REGISTRADOR	C1	CGR	R,RR	(R)<-(RR)	PZSET
LE CHAVES	C10	LC	RR	(RR)	PZS
LE REGISTRADOR DE ESTADO	C10	LRE	RR	(RR)<-(RE)	PZS
INSERE BYTE	L1	IB	R,RM	(R<0:7>)<-(RM)	PZSET
REDUZ A BYTE DA DIREITA	C10	RBD	RR	(RR<8:15>)<-0	PZSET
RESTAURA REGISTRADOR	C10	REST	RR	REGS SAO REST.	
SALVA REGISTRADOR	C10	SALV	RR	REGS SAO SALV. OBS.: SE MO=00 OU 01, APENAS BL E' REST. OU SALV. SE MO=10 OU 11 TODOS REGS SAO REST. OU SALV.	
SEPARA BYTES EM REGISTRADOR	C1	SEPB	R,RR	(R<0:7>)<-(RR<8:15>) (R<8:15>)<-0 (RR<8:15>)<-0	
TROCA BYTES EM REGISTRADOR	C10	TB	RR	(RR<0:7>)<->(RR<8:15>)	PZSET
TROCA REGISTRADORES	C1	TR	R,RR	(R)<->(RR)	

INSTRUCOES DE COMPARACAO

DESCRICAO DA INSTRUCAO	FMTO.	SINTAXE		SEMANTICA	ESTADO
COMPARA ARITMETICO	L1	CA	RR,RM	(RR)-(RM)	PZSET
COMPARA ARITMETICO BYTE	L1	CAB	R,RM	(R)-(RM)	PZSET
COMPARA ARITMETICO CURTO	C11	CAC	R,E	(R) - (8L+D)	PZSET
COMPARA ARITMETICO IMEDIATO	C11	CAI	R,I	(R)-I	PZSET
COMPARA ARITMETICO REGISTRADORES	C1	CAR	R,RR	(R)-(RR)	PZSET
COMPARA LOGICO	L1	CL	RR,RM	(RR)-(RM)	PZSET
COMPARA LOGICO BYTE	L1	CLB	R,RM	(R)-(RM)	PZSET
CARREGA BYTE	L1	CGB	R,RM	(R<0:7>)<-(RM)	PZSET

INSTRUCOES DE DESVIO

DESVIA SE DIFERENTE	L4	DD	RM	(RO)<- RM	
DESVIA SE IGUAL	L4	DI	RM	(RO)<- RM	
DESVIA SE MAIOR	L4	DMA	RM	(RO)<- RM	
DESVIA SE MAIOR OU IGUAL	L4	DMAI	RM	(RO)<- RM	
DESVIA SE MENOR	L4	DME	RM	(RO)<- RM	
DESVIA SE MENOR OU IGUAL	L4	DMEI	RM	(RO)<- RM	
DESVIA ITERANDO	LA	DIT	R,RM	SE (R) = 0 SEGUE SE (R) < 0 ENTAO (R) <- (R) + 1 (RO) <- RM SE (R) > 0 ENTAO (R) <- (R) - 1 (RO) <- RM	
DESVIA SE PAR	L4	DP	RM	(RO) <- RM	
DESVIA PARA SUBROTINA	L3	DSUB	RR,RM	(RO) <- RM (RR) <- END.RETORNO	
RETORNA DE SUBROTINA	L3	RSUB	R,I		

INSTRUCOES DE PULO

DESCRICAO DA INSTRUCAO	FMTO.	SINTAXE		SEMANTICA	ESTADO
PULA	C4	P	E	(RO) <- (RO) + D	
PULA SE DIFERENTE	C4	PD	E	(RO) <- (RO) + D	
PULA SE IGUAL	C4	PI	E	(RO) <- (RO) + D	
PULA SE MAIOR	C4	PMA	E	(RO) <- (RO) + D	
PULA SE MAIOR OU IGUAL	C4	PMAI	E	(RO) <- (RO) + D	
PULA SE MENOR	C4	PME	E	(RO) <- (RO) + D	
PULA SE MENOR OU IGUAL	C4	PMEI	E	(RO) <- (RO) + D	
PULA ITERANDO	C9	PIT	R,E	SE (R) = 0 SEGUE SE (R) < 0 ENTAO (R) <- (R) + 1 (RO) <- (RO) - D SE (R) > 0 ENTAO (R) <- (R) - 1 (RO) <- (RO) - D	
PULA SE PAR	C4	PP	E	(RO)<- (RO) + D	

INSTRUCOES DE DESLOCAMENTO

DESLOCA ARITMETICO SIMPLES	C2	DAS	R,R		PZSET
DESLOCA ARITMETICO SIMPLES IMEDIATO	C3	DASI	R,I		PZSET
DESLOCA ARITMETICO DUPLO	C2	DAD	R,R		PZSET
DESLOCA ARITMETICO DUPLO IMEDIATO	C3	DADI	R,I		PZSET
DESLOCA LOGICO SIMPLES	C2	DLS	R,R		PZSET
DESLOCA LOGICO SIMPLES IMEDIATO	C3	DLSI	R,I		PZSET
DESLOCA LOGICO DUPLO	C2	DLD	R,R		PZSET
DESLOCA LOGICO IMEDIATO	C3	DLDI	R,I		PZSET

INSTRUCOES LOGICAS

DESCRICAO DA INSTRUCAO	FMTO.	SINTAXE		SEMANTICA	ESTADO
E	L1	E	RR,RM	(RR) <- (RR)&(RM)	PZSET
E BYTE	L1	EB	R,RM	(R) <- (R)&(RM)	PZSET
E REGISTRADORES	C1	ER	R,RR	(R) <- (R)&(RR)	PZSET
OU	L1	OU	RR,RM	(RR) <- (RR) V (RM)	PZSET
OU BYTE	L1	OUB	R,RM	(R) <- (R) V (RM)	PZSET
OU REGISTRADORES	C1	OUR	R,RR	(R) <- (R) V (RR)	PZSET
OU EXCLUSIVO	L1	OUX	RR,RM	(RR) <- (RR)OUX(RM)	PZSET
OU EXCLUSIVO BYTE	L1	OUXB	R,RM	(R) <- (R)OUX(RM)	PZSET
OU EXCLUSIVO REGISTRADORES	C1	OUXR	R,RR	(R) <- (R)OUX(RR)	PZSET
COMPLEMENTA	C10	CMPL	RR	INVERTE BITS DE(RR)	PZSET

INSTRUCOES DE MAPEAMENTO

MAPEIA ESTADO SE EXTENSAO	C10	ME	RR	(RR) <- (RE(E))	PZSET
MAPEIA ESTADO SE DIFERENTE	C10	MD	RR	(RR) <- (RE(Z))	PZSET
MAPEIA ESTADO SE IGUAL	C10	MI	RR	(RR) <- (RE(Z))	PZSET
MAPEIA ESTADO SE MAIOR	C10	MMA	RR	(RR) <- (RE(S))	PZSET
MAPEIA ESTADO SE MAIOR OU IGUAL	C10	MMAI	RR	(RR) <- (RE(S,Z))	PZSET
MAPEIA ESTADO SE MENOR	C10	MME	RR	(RR) <- (RE(S))	PZSET
MAPEIA ESTADO SE MENOR OU IGUAL	C10	MMEI	RR	(RR) <- (RE(S,Z))	PZSET
MAPEIA ESTADO SE TRANSBORDO	C10	MT	RR	(RR) <- (RE(T))	PZSET

INSTRUCOES ARITMETICAS

DESCRICAO DA INSTRUCAO	FMTO.	SINTAXE		SEMANTICA	ESTADO
ADICIONA	L1	AD	RR,RM	(RR)<- (RR)+ (RM)	PZSET
ADICIONA CURTO	C11	ADC	R,E	(R) <- (R) + (8L+D)	PZSET
ADICIONA EXTENSAO	C10	ADE	RR,RM	(RR)<- (RR)+(RE<E>)	PZSET
ADICIONA IMEDIATO	C11	ADI	R,I	(R) <- (R) + I	PZSET
ADICIONA A MEMORIA	L2	ADM	RM,I	(RM)<- (RM)+ I	PZSET
ADICIONA REGISTRADORES	C1	ADR	R,RR	(R) <- (R) + (RR)	PZSET
DECREMENTA	C10	DEC	RR	(RR)<- (RR) - 1	PZSET
DIVIDE MISTO REGISTRADORES	C1	DVMR	R,RR	(R)<-(R-1,R)/(RR) (R-1) <- RESTO	PZSET
DIVIDE SIMPLES	L1	DVS	RR,RM	(RR)<- (RR)/RM	PZSET
DIVIDE SIMPLES REGISTRADORES	C1	DVSR	R,RR	(R)<-(R)/(RR)	PZSET
INCREMENTA	C10	INC	RR	(RR)<-(RR)+1	PZSET
MULTIPLICA MISTO REGISTRADORES	C1	MLMR	R,RR	(R-1,R) <-(R)*(RR)	PZSET
MULTIPLICA SIMPLES	L1	MLS	RR,RM	(RR)<- (RR)*(RM)	PZSET
MULTIPLICA SIMPLES REGISTRADORES	C1	MLSR	R,RR	(R) <- (R)*(RR)	PZSET
RESTO DA DIVISAO	L1	RDV	RR,RM	(RR) <- RESTO DE (RR)/(RM)	PZSET
RESTO DA DIVISAO EM REGISTRADORES	D1	RDVR	R,RR	(R) <- RESTO DE ((R) / (RR))	PZSET
SUBTRAI	L1	SU	RR,RM	(RR) <- (RR) - (RM)	PZSET
SUBTRAI EXTENSAO	C10	SUE	RR	(RR) <- (RR) - (RE<E>)	PZSET
SUBTRAI IMEDIATO	C11	SUI	R,I	(R) <- (R) - I	PZSET
SUBTRAI REGISTRADORES	C1	SUR	R,RR	(R) <- (R) - (RR)	PZSET
TROCA SINAL	C10	NEG	RR	(RR) <- -(RR)	PZSET
VALOR ABSOLUTO	C10	VA	RR	SE (RR) < 0 ENTAO (RR) <- -(RR)	

INSTRUCOES ESPECIAIS

DESCRICAO DA INSTRUCAO	FMTO.	SINTAXE		SEMANTICA	ESTADO
CHAMADA SUPERVISOR	C8	CSUP	I	-128 <= I <= 127 R7 <- NUM.CHAMADA	
ESPERA	C8	ESP	I	ESPERA INTERRUPCAO	
EXECUTA	C10	EX	RR	EXECUTA INSTRUCAO EM (RR)	IMPZSET
INIBE INTERRUPCOES	C8	IINT			
PARA	C8	PARA	I	PARA UCP PAINEL <- I	
PERMITE INTERRUPCOES	C8	PINT			
RETORNA DE INTERRUPCAO	C8	RINT			IMPZSET

INSTRUCOES DE ENTRADA E SAIDA

ENTRA BLOCO COMPACTADO	C5	EBC	C,P		
ENTRA BLOCO DESCOMPACTADO	C5	EBD	C,P		
SAI BLOCO COMPACTADO	C5	SBC	C,P		
SAI BLOCO DESCOMPACTADO	C5	SBD	C,P		
ENTRA DADO	C7	ENT	RR,C,P	(RR) <- (C,P)	
SAI DADO	C7	SAI	RR,C,P	(C,P)<- (RR)	
ARMAZENA FUNCAO NA INTERFACE	C6	AFI	C,P	(C,P) <- (R7)	
LE ESTADO DO DISPOSITIVO	C6	LED	C,P	(R7) <- (C,P)	
LE VETOR DE INTERRUPCAO	C6	LVI	C,P	(R7) <- (C,P)	
MODIFICA ESTADO DO DISPOSITIVO	C6	MED	C,P	(C,P) <- (R7)	

NOTAS

1. LEGENDAS

BL = BASELOCAL
C = CANAL
E = EXPRESSAO RELOCAVEL
I = OPERANDO IMEDIATO
IPC= INDICE DA PILHA DE CONTEXTO
P = PERIFERICO
R = REGISTRADOR DE USO GERAL
RE = REGISTRADOR DE ESTADO
RM = REFERENCIA A MEMORIA
RR = REFERENCIA A REGISTRADOR
$ = REFERENCIA AO CONTADOR DE INSTRUCAO

2. SINTAXE DAS INSTRUCOES ASSEMBLER

2.1. REFERENCIA A MEMORIA

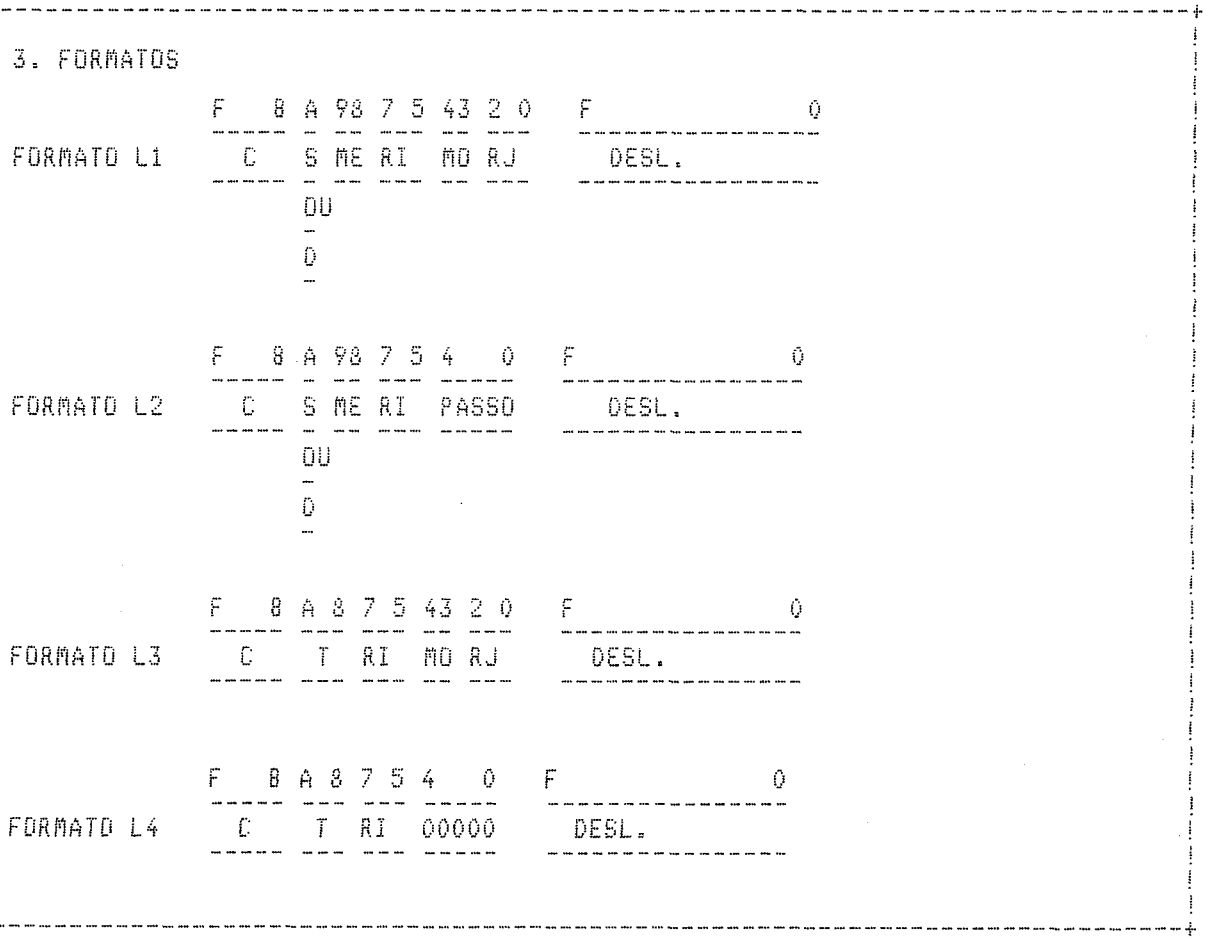
- DIRETO NAO INDEXADO: E
- DIRETO INDEXADO: E(R)
- INDIRETO NAO INDEXADO: (E)
- DIRETO,INDEXADO,PRE-INCREMENTANDO O INDICE: E(+R)
- INDIRETO, POS INDEXADO, PRE-INCREMENTANDO O INDICE: (E) (+R)

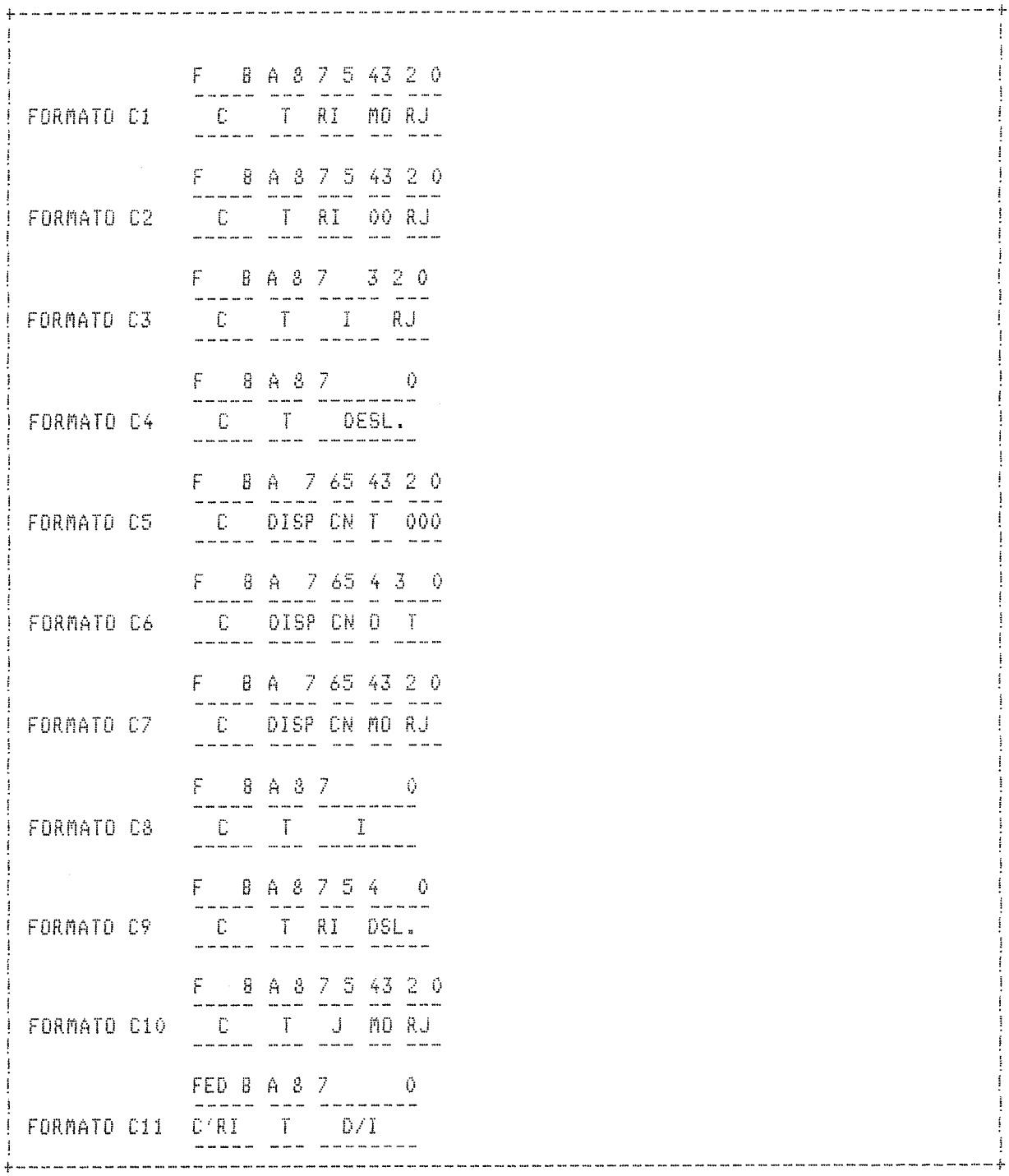
2.1.1. OBSERVACAO: NAS INSTRUCOES DE DESVIO CARREGA E ARMAZENA ABSOLUTO SO' EXISTE RM DIRETO NAO INDEXADO E DIRETO INDEXADO.

2.2. REFERENCIA A REGISTRADOR

- DIRETO: R
- INDIRETO: (R)
- INDIRETO PRE-INCREMENTADO: (R+)
- INDIRETO POS-DECREMENTADO: (R)-

3. FORMATOS


F 8 A 98 7 5 43 2 0 F 0
FORMATO L1
C S ME RI MD RJ DESL.
OU
-
D
-
F 8 A 98 7 5 4 0 F 0
FORMATO L2
C S ME RI PASSO DESL.
OU
-
D
-
F 8 A 8 7 5 43 2 0 F 0
FORMATO L3
C T RI MD RJ DESL.
F B A 8 7 5 4 0 F 0
FORMATO L4
C T RI 00000 DESL.


F B A 8 7 5 43 2 0
FORMATO C1
C T RI MD RJ
F B A 8 7 5 43 2 0
FORMATO C2
C T RI 00 RJ
F B A 8 7 3 2 0
FORMATO C3
C T I RJ
F B A 8 7 0
FORMATO C4
C T DESL.
F B A 7 65 43 2 0
FORMATO C5
C DISP CN T 000
F B A 7 65 4 3 0
FORMATO C6
C DISP CN 0 T
F B A 7 65 43 2 0
FORMATO C7
C DISP CN MD RJ
F B A 8 7 0
FORMATO C8
C T I
F B A 8 7 5 4 0
FORMATO C9
C T RI DSL.
F B A 8 7 5 43 2 0
FORMATO C10
C T J MD RJ
FED B A 8 7 0
FORMATO C11
C'RI T D/I

REFERÊNCIAS BIBLIOGRÁFICAS

| 1| Melvin A. Breuer, Arthur D. Friedman - Diagnose e Reliable Design of Digital Systems, California, Computer Science Press, Inc. Digital System Design Series, 1976.

| 2| Michael D. Lippman, Eduard S. Donn - Design Forethought Promotes Easier Testing of Microcomputer Boards, Tluke Trendar Corp., Mountain View, California, Electronics, 18 de janeiro de 1979, páginas 113-119.

| 3| Jeffrey D. Russel, Charles R. Kime - System Fault Diagnosis: Closure and Diagnosability With Repair, Berkeley , California, IEEE TRANSACTIONS ON COMPUTERS, Vol. C-24, no. 11, novembro de 1975, páginas 1078-1089.

| 4| R.E. Forbes, D.H. Rutherford, C.B. Stieglitz, L.M. Tung - A Self-Diagnosable Computer, Fall Joint Comput. Conf. , AFIPS Conf. Proc. Vol. 27, Washington, D.C., Spartan , 1965, páginas 1073-1086.

| 5| P.W. Agnew, D.H. Rutherford, R.J. Suhocki, C.M. Yen, D.E. Muller - An Architectural Study for a Self-Repairing Computer, U.S. Air Force Space Division, Final Tech. Doc. Rep. SSD-TR-65-159, AD 474976, novembro de 1965.

| 6| C.V. Ramamoorthy - A Structural Theory of Machine Diagnosis, Spring Joint Comput. Conf. de 1967, AFIPS Conf. Proc., Vol. 30, Washington, D.C.: Thompson, 1967, páginas 743-756.

| 7| F.P. Preparata, G. Metze, R.T. Chien - On The Connection Assignment Problem of Diagnosable Systems, IEEE Transactions Electron Comput., Vol. Ec-16, dezembro de 1967, páginas 848-854.

| 8| F.P. Preparata - Some Results on Sequentially Diagnosable Systems, Proc. Hawaii Int. Conf. Syst, Sci., Univ. of Hawaii Press, Honolulu, Hawaii, janeiro de 1968, páginas 623-626.

| 9| N. Seshagiri - Completely Self-Diagnosable Digital Systems, Int. J. Syst. Sci., Vol. 1, janeiro de 1971, páginas 235-246.

|10| R.P. Vancura, C.R. Kime - On Numerical Bounds in Diagnosable Systems, Dig. 1972 Int. Symp. Fault-Tolerant Computing, IEEE Computer Society Publications, junho de 1972, páginas 148-153.

|11| C.R. Kime - An Analysis Model for Digital System Diagnosis, IEEE Trans. Computer., Vol. C-19, novembro de 1970 , páginas 1063-1073.

|12| J.D. Russel - On The Diagnosability of Digital Systems , Ph.D. Dissertion, Univ. Wisconsin, Madison, Wis., julho de 1973.

|13| D.L. Dietmeyer, P.R. Schneider - Identification of Symmetry Redundancy, and Equivalence of Boolean Functions , IEEE Trans. Electron. Comput., Vol. EC-16, dezembro de 1967, páginas 804-817.

|14| J.C. Bassett, C.R. Kime - Improved Procedures for Determining Diagnostic Resolution, IEEE Trans. Comput., Vol. C-21, abril de 1972, páginas 385-388.

|15| F. Harary - Graph Theory, Reading, Mass.: Addison-Wesley , 1969.

|16| F.J. Hackl, R.W. Shirk - An Integrated Approach to Automated Computer Maitenance, 1965 IEEE Conf. Rec. Switching, Theory and Logical Design, outubro de 1965, páginas 289-302.

|17| Manual de Arquitetura do Processador de Instruções do Micro-Computador G11, Computadores e Sistemas Brasileiros S/A (COBRA), Rio de Janeiro, 1978.

|18| Manual de Microprogramação do Processador de Instruções do Mini-Computador G10, Computadores e Sistemas Brasileiros S/A (COBRA), Rio de Janeiro, 1977.