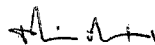


A UTILIZAÇÃO DE PROGRAMAS
PARA DETECÇÃO DE FALHAS EM MICROCOMPUTADORES

Humberto dos Santos Melim

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 OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.).

APROVADA POR:



PAULO MARIO BIANCHI FRANÇA
presidente



EBER ASSIS SCHMITZ



LUIZ ANTONIO CARNEIRO
DA CUNHA COUCEIRO

RIO DE JANEIRO , RJ-BRASIL
SETEMBRO DE 1980

MELIM, HUMBERTO DOS SANTOS

A Utilização de Programas para Detecção de Falhas em Microcomputadores [Rio de Janeiro], 1980.

VII, 76 p. 29,7 cm (COPPE-UFRJ, M. Sc., Engenharia de Sistemas e Computação, 1980.

Tese - Universidade Federal do Rio de Janeiro. Núcleo de Computação Eletrônica.

1. Programas de Teste I. COPPE/UFRJ II. Título (série).

AGRADECIMENTOS

Ao orientador, pelas valiosas sugestões apresentadas durante o desenvolvimento deste trabalho, aos demais membros da banca, que muito me honraram com sua participação, e aos colegas do NCE, pelo apoio e colaboração prestados.

RESUMO

O presente trabalho tece considerações sobre a utilização de programas de teste em microcomputadores e descreve um conjunto de testes, residentes em uma ROM de 1Kx8, destinados a detetar falhas nos diversos módulos que compõem um sistema baseado no microprocessador 8080.

ABSTRACT

This work surveys some aspects of self-test programs for microcomputers and presents a 1 Kbyte ROM resident program developed for fault detection in a 8080 based system.

ÍNDICE

I - INTRODUÇÃO	1
II - TESTES	4
2.1 - CLASSIFICAÇÃO	4
2.2 - TESTES DE PROCESSADOR	5
2.3 - TESTES DE MEMÓRIA RAM	13
2.4 - TESTES DE MEMÓRIA ROM	19
2.5 - TESTES DE PERIFÉRICOS	20
2.5.1 - PERIFÉRICOS DE SAÍDA	20
2.5.2 - PERIFÉRICOS DE ENTRADA	21
2.5.3 - PERIFÉRICOS DE ARMAZENAMENTO	21
III - DESCRIÇÃO DO MICROCOMPUTADOR POTI	23
3.1 - APRESENTAÇÃO	23
3.2 - ORGANIZAÇÃO	23
3.3 - UNIDADE CENTRAL DE PROCESSAMENTO	26
3.4 - MEMÓRIA RAM	26
3.5 - MEMÓRIA ROM	28
3.6 - PERIFÉRICO TECLADO	28
3.7 - PERIFÉRICO VÍDEO	28
3.8 - PERIFÉRICO DISCO	29
3.9 - PERIFÉRICO IMPRESSORA	30
3.10- PERIFÉRICOS ADICIONAIS	30

IV - TESTES APLICADOS AO POTI	31
4.1 - SEQUENCIAMENTO DOS TESTES	31
4.2 - TESTE DA UCP	33
4.3 - TESTE DO VÍDEO	39
4.4 - TESTE DA MEMÓRIA RAM	41
4.5 - TESTE DA MEMÓRIA ROM	42
4.6 - TESTE DO TECLADO	42
4.7 - TESTE DA IMPRESSORA	43
4.8 - TESTE DO DISCO	43
V - CONCLUSÃO	44
BIBLIOGRAFIA	46
APÊNDICE - LISTAGEM DO PROGRAMA DE TESTE	50

I - INTRODUÇÃO

A crescente utilização de microprocessadores nos últimos anos, devido ao seu baixo custo, alto desempenho e grande flexibilidade, ao mesmo tempo que ocasionou mudanças substanciais no projeto de sistemas, dificultou também os procedimentos de teste dos mesmos^{22,26,31}.

Os métodos tradicionais para encontrar bits presos em blocos combinacionais⁴ não são adequados em sistemas com microprocessadores, devido à grande complexidade e inacessibilidade dos circuitos e à existência de outros tipos de erro.

Entre as opções atualmente disponíveis está o emprego da análise de assinaturas^{20,21,30,33}, uma técnica recentemente desenvolvida e que procura caracterizar os diversos pontos de um circuito através da compressão dos dados ali presentes, amostrados a partir de sinais de referência, ou a utilização de equipamentos automáticos de teste (ATE)^{5,6,28,32,34}, especialmente desenvolvidos para este fim.

A análise de assinaturas, no entanto, necessita ser definida na fase de projeto, com o intuito de prover os sinais de referência indispensáveis à sua utilização. Requer ainda aparelhagem específica para gerar as assinaturas e compará-las com padrões já estabelecidos.

Por sua vez, o emprego de ATE vai se tornando desaconselhável até mesmo em produção de larga escala, devido ao seu alto preço, que tem se mantido aproximadamente constante ao longo do tempo, absorvendo uma parcela cada vez

maior em relação ao custo de um componente ou sistema, já que os demais gastos associados à fabricação decrescem à medida que a tecnologia evolui³.

Uma outra alternativa consiste em utilizar a potencialidade inerente dos microprocessadores para executar auto-testes^{1,3,14,18,24,29,35}. Os programas de teste, escritos na própria linguagem do processador, permitem avaliar o funcionamento do sistema. Podem ser empregados durante o desenvolvimento, teste de produção, verificação e manutenção.

Seu uso em aplicações de alta confiabilidade, tais como equipamento espacial ou militar, usinas nucleares e sistemas médicos de controle de funções vitais, permite acionar sinais de alarme ou advertência, ou comutar para sistema "back-up".

O grande número de estímulos que pode ser rapidamente aplicado com o uso desta técnica se revela uma valiosa ferramenta para manutenção, reduzindo substancialmente o MTTR do equipamento e possibilitando uma precisa verificação de seu correto funcionamento.

A constante redução do custo por bit de memórias ROM permite até mesmo a inclusão dos testes mais básicos no próprio microprocessador, como já ocorre no MC6805, da Motorola³.

Uma vez que as únicas despesas envolvidas na utilização de auto-testes consistem na produção de programas apropriados, resultam economias substanciais, por não haver necessidade de adquirir equipamentos específicos.

Para sistemas com um único microprocessador, onde

o custo, tamanho e potência consumida devem ser minimizados, o emprego destes programas surge conseqüentemente como uma excelente alternativa de teste.

É a este tópico que se prende o presente trabalho,abordando diversos métodos e detalhando a implementação de um programa de auto-teste para um microcomputador típico.

II - TESTES

2.1 - Classificação

Os testes normalmente aplicados a componentes ou sistemas lógicos a fim de verificar seu correto funcionamento se enquadram em três categorias:

I) Teste estático, no qual as correntes e tensões apropriadas são aplicadas a determinados pontos e as correntes e tensões resultantes em outros pontos são verificadas.

II) Teste de funcionamento, procedimento pelo qual se testa a lógica interna de um dispositivo, através da aplicação de padrões binários nas entradas e comparação das saídas com resultados conhecidos.

III) Teste dinâmico, onde se verificam os tempos e formas de onda correspondentes aos parâmetros esperados do circuito em teste.

Em virtude da atual complexidade dos componentes básicos de sistemas lógicos, a utilização de um teste estático permite detetar apenas os defeitos mais grosseiros, antes de se dar início a testes mais sofisticados.

Por outro lado, um dispositivo aprovado no teste de funcionamento tem grande probabilidade de estar perfeito. Deve-se notar que o teste de funcionamento é de todos o mais importante, podendo, dentro de certos limites, englobar implicitamente parte dos demais testes.

Defeitos apontados pelos testes estático e dinâmico só são importantes na medida em que prejudiquem o fun

cionamento do circuito, ao ser este incorporado em um sistema. Seu emprego procura detetar de maneira indireta qualquer discrepância que possa ocasionar problemas de funcionamento. Naturalmente, por desconhecimento do sistema ao qual se destina um componente, bem como por questões de padronização, circuitos são rejeitados após sua fabricação se deixarem de atender a qualquer de suas especificações. Entretanto, ao se tentar localizar falhas em um sistema, é no teste de funcionamento que se deve concentrar especial atenção.

Os testes de funcionamento podem ser realizados através de equipamentos especialmente desenvolvidos para este fim ou através da capacidade de processamento do próprio sistema. Nesta categoria se enquadram todos os testes por programa, independente da forma em que se encontrem implementados.

2.2 - Testes de Processador

O crescente custo do trabalho associado à manutenção e o amplo uso de processadores em aplicações onde se necessita confiabilidade e disponibilidade tem ocasionado o aparecimento de projetos que incluem facilidades embutidas para manutenção, a fim de rapidamente localizar, a nível de subsistema, as partes com defeito e isolar a(s) placa(s) correspondente(s), para devida substituição.

A particular implementação das facilidades de teste difere bastante, até para um mesmo fabricante, como demonstram os exemplos a seguir.

No processador KL10 da DEC^{2, 10} para sistemas de

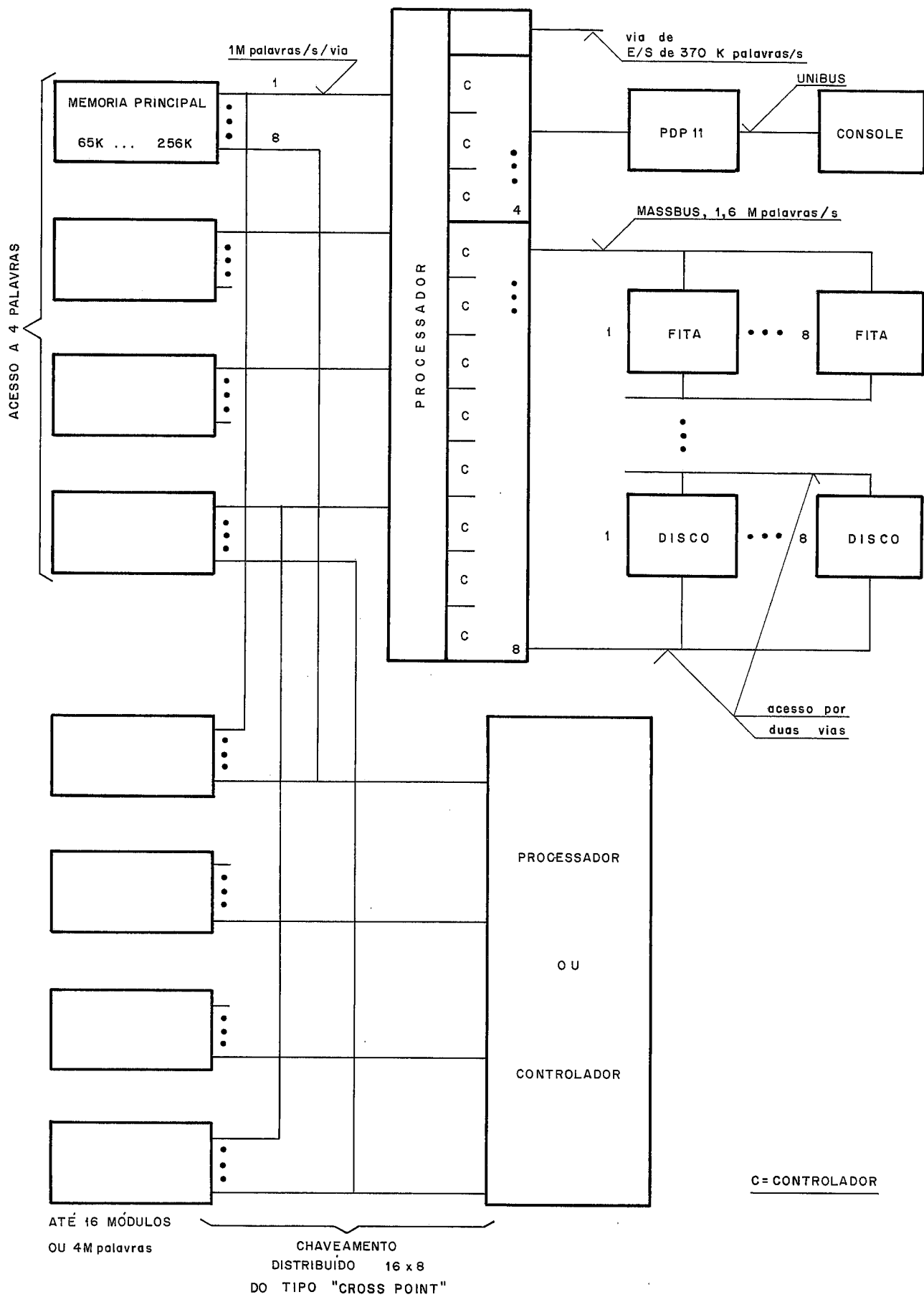


FIG 2-1 - DIAGRAMA DE BLOCOS DO KL 10.

grande porte, mostrado na figura 2-1, é possível ligar até quatro computadores PDP 11, que operam como processadores de entrada e saída. Um deles funciona também como console e processador para diagnósticos, com possibilidade de carregar microprogramas na memória de controle, examinar as trajetórias de dados e a lógica de controle do KL10 através de uma via especial, mesmo quando este se encontra completamente inoperante, podendo ainda verificar todas as vias de dados e realizar diagnóstico remoto através de linha telefônica.

Já o PDP 11/60², mostrado na figura 2-2, dispõe de uma unidade independente de microdiagnóstico, que testa as placas no processador, indicando, através de um conjunto de LEDs, o código associado a quaisquer anormalidades detectadas, e que, juntamente com o uso de um diretório, permite determinar a(s) placa(s) a substituir.

Para correta indicação, esta unidade supõe que uma pequena parte do processador, mais especificamente, o sequenciador de micropalavras, esteja operacional.

O PDP 11/70^{11, 12, 13}, cujo diagrama de blocos se vê na figura 2-3, possui um módulo acoplado à via padrão UNIBUS, onde se encontra uma ROM de 512 palavras de 16 bits, contendo um programa de diagnósticos e também a parte referente ao carregamento do sistema operacional. Este programa testa as instruções do processador, o cache, o gerenciamento de memória, o mapeador do UNIBUS, a memória principal e, parcialmente, o controlador e o periférico de UNIBUS ou MASSBUS a partir de onde o sistema operacional é carregado. A detecção de quaisquer erros provoca uma parada e o

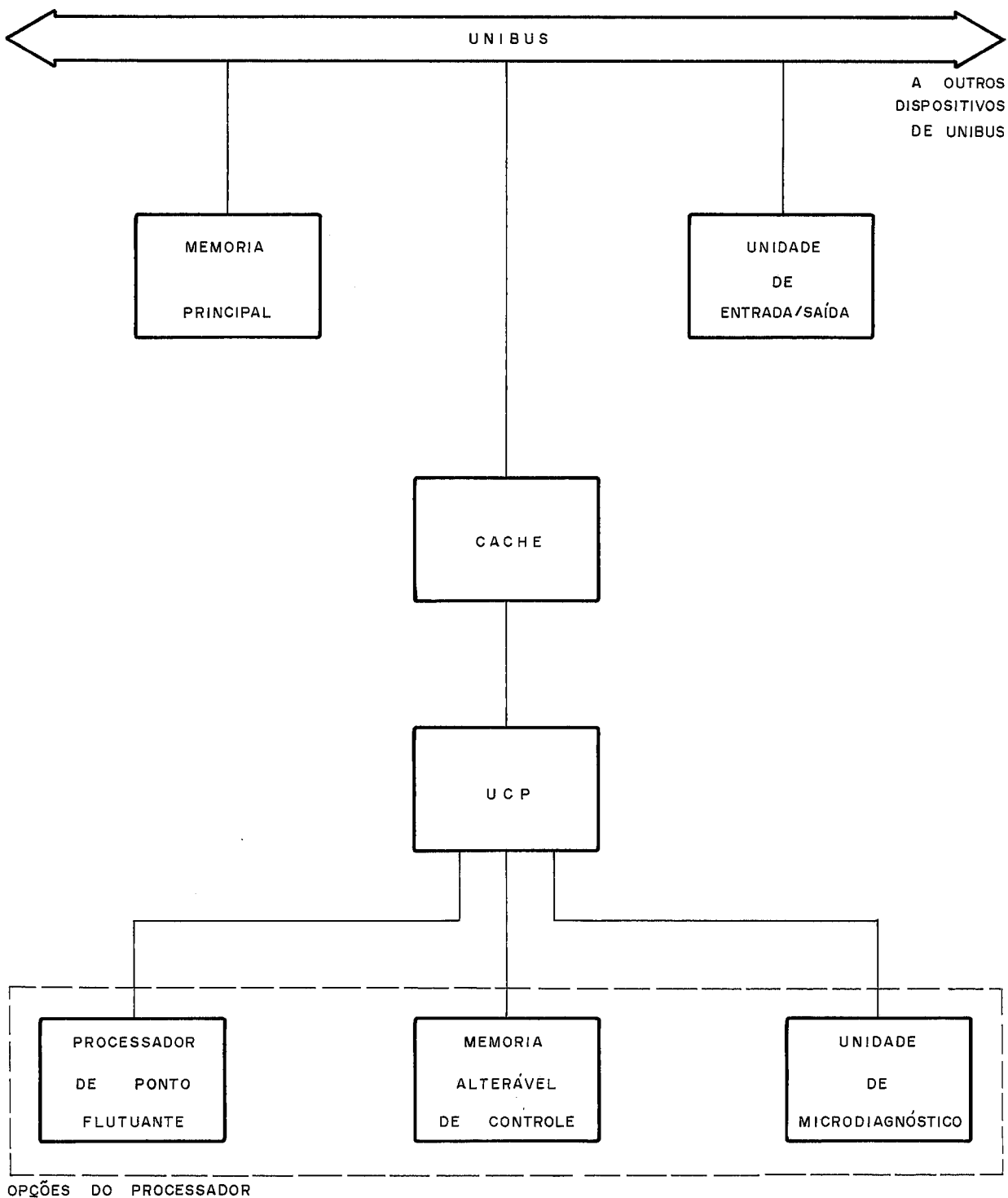


FIG. 2-2 - DIAGRAMA DE BLOCOS DO PDP 11/60.

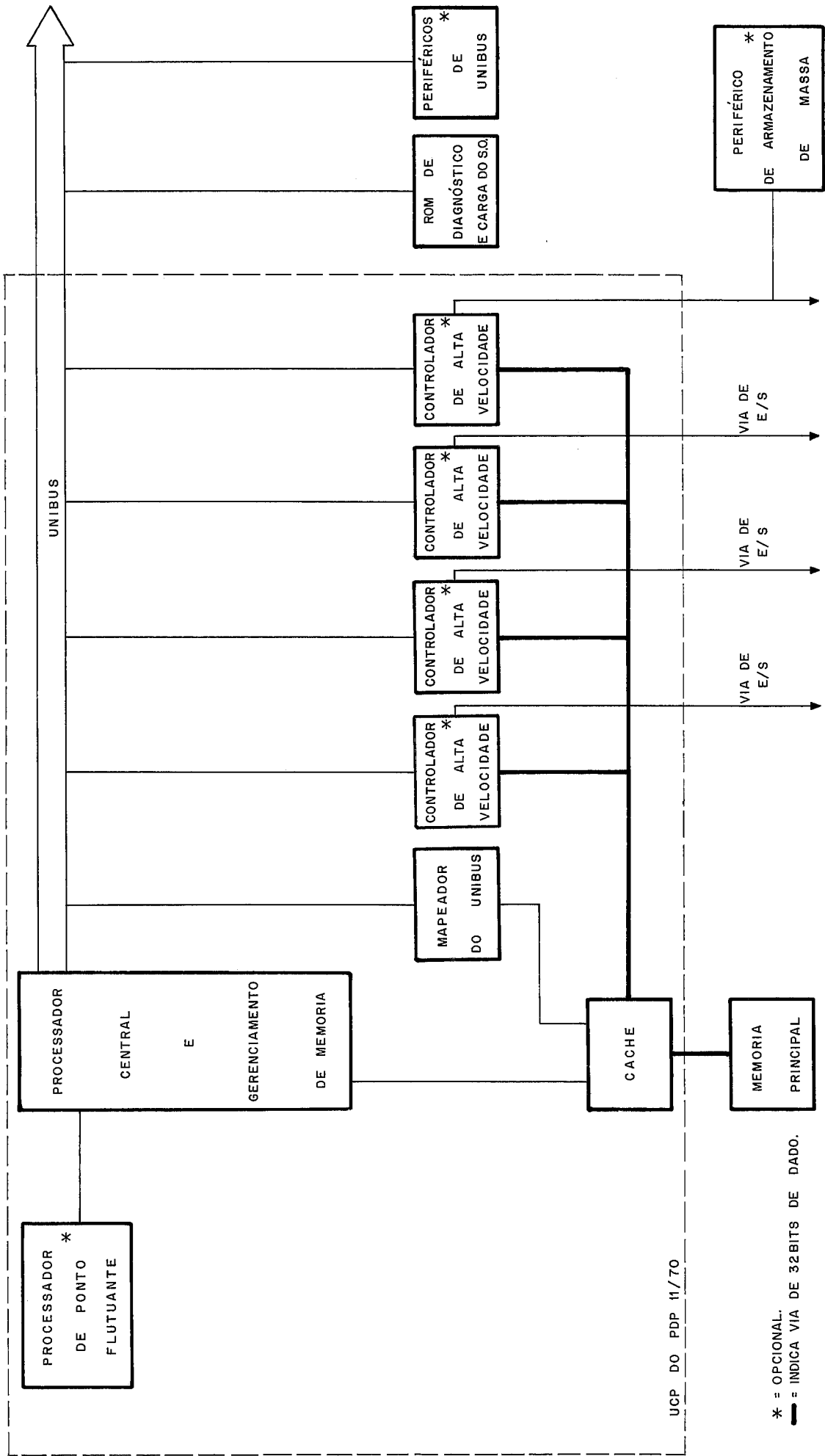


FIG. 2-3 - DIAGRAMA DE BLOCOS DO PDP 11/70.

endereço correspondente, disponível no painel, indica o ti
po do erro encontrado.

O VAX 11/780^{2, 8}, apresentado na figura 2-4, volta a empregar esquema semelhante ao do KL10, com uma memória de controle que pode ser carregada com o microprograma normal ou de diagnóstico, através do subsistema console, constituído por um microcomputador LSI-11 com disco flexível e capacidade para diagnóstico remoto e auto-teste, refletindo a evolução tecnológica da idéia original.

A IBM, em seu recente modelo 4341^{9, 17}, leva ainda mais adiante a mesma idéia, como ilustra a figura 2-5, va
lendo-se de um microprocessador em separado para suporte e manutenção, que não apenas é capaz de carregar microprogramas de operação e diagnóstico, realizar diagnóstico remoto e auto-teste, como ainda registra em diskette o es
tado do processador principal, em caso de erro, para subseq
uente análise, uma vez que este dispõe de lógica específ
icamente incluída em sua arquitetura para permitir o a
cesso a seus estados internos. Este método permite até mesmo a d
eteção da causa de erros intermitentes, pois as informaç
ões são gravadas já na primeira ocorrência da anormalidad
e, acompanhada de dados adicionais relativos a condições ambientais de temperatura e de voltagem de alimentação, de forma que não se faz necessário reconstruir o erro para localizar o problema.

Embora cada um dos sistemas descritos, visto como um todo, apresente capacidade de auto-teste, quem melhor ca
racteriza este aspecto é o PDP 11/70, que utiliza as pr
óprias instruções de seu único processador para realizar

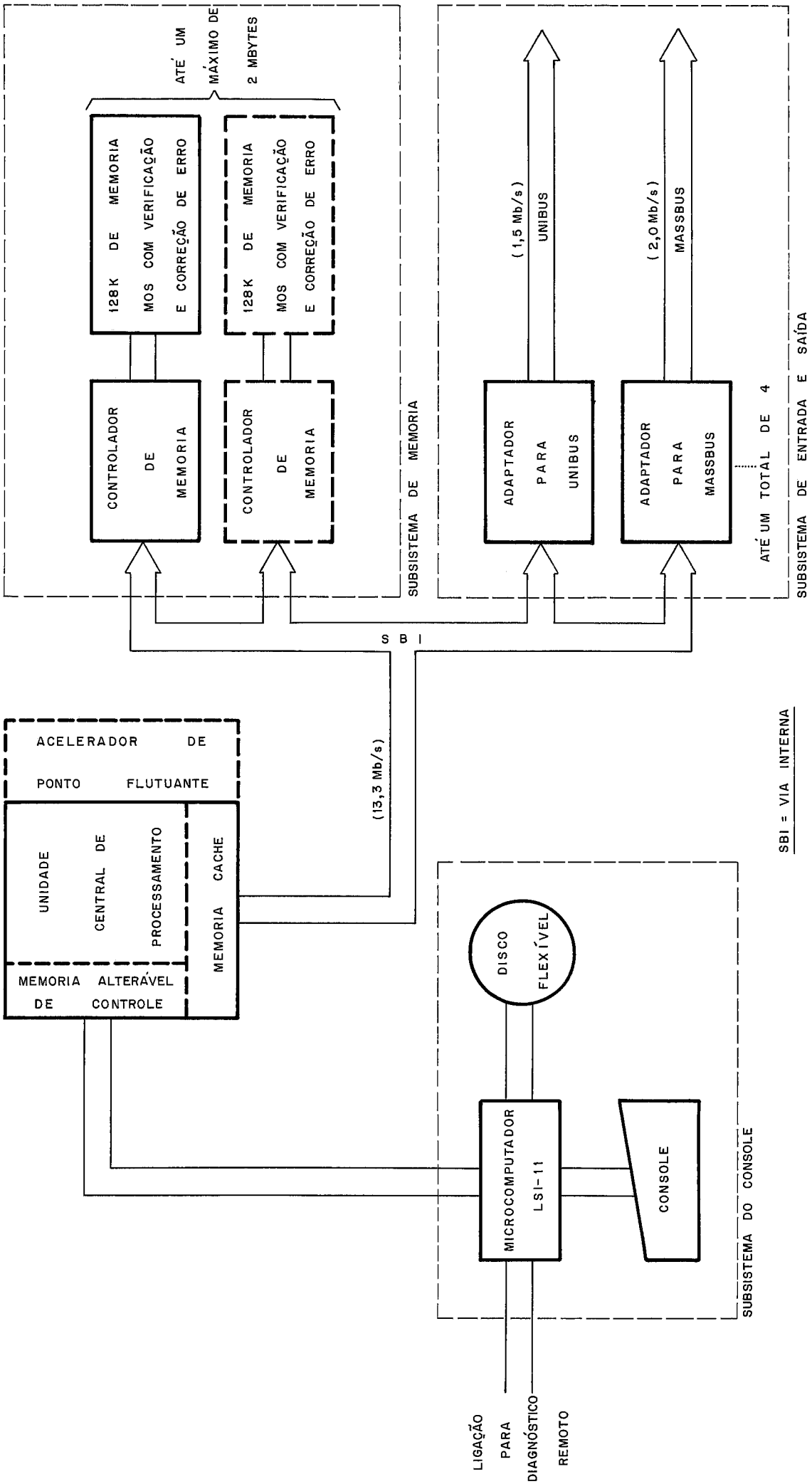


FIG. 2-4 — DIAGRAMA DE BLOCOS DO VAX 11/780.

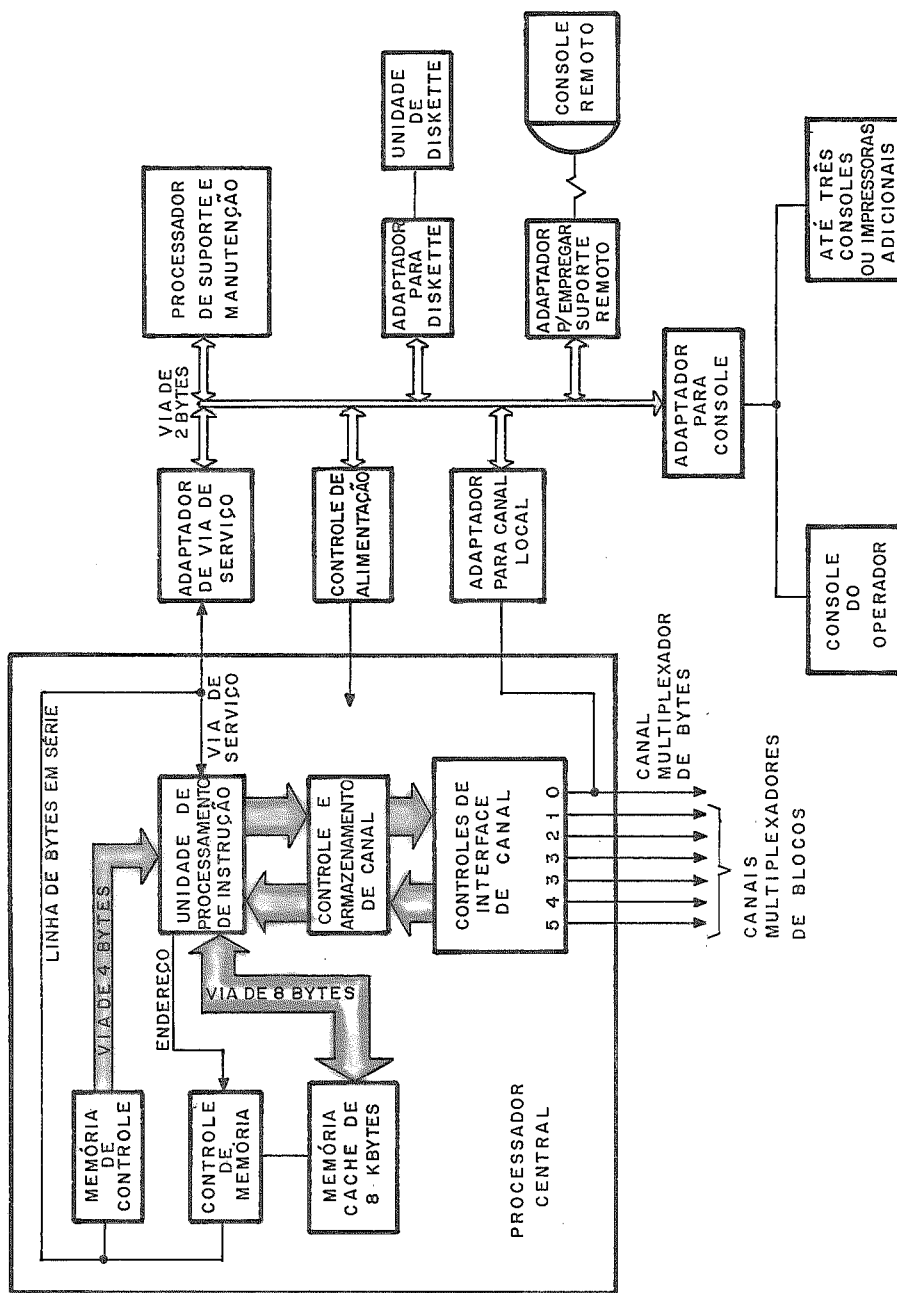


Fig. 2-5 - Diagrama de Blocos do 4341

auto-diagnóstico.

Deve-se notar que há um problema fundamental com programas de auto-teste: uma falha pode anular o efeito de outra, de modo que ambas passem despercebidas. Para contornar este problema, pode-se testar inicialmente as instruções mais confiáveis e a seguir usá-las para verificar as mais complexas.

O PDP 11/70, por exemplo, supõe o correto funcionamento do contador de programas (PC) e da instrução HALT, e aplicando sequências de instruções, organizadas em grupos funcionais, compara os resultados obtidos com os esperados, de modo a, na ocorrência de qualquer discrepância, executar a instrução HALT e determinar, pelo endereço onde se encontra, o grupo de instruções que deixou de operar, e, indiretamente, a parte da lógica que corresponde às mesmas.

A suposição inicial pode ser facilmente verificada executando-se a instrução HALT e observando-se a evolução do conteúdo do PC, através do acompanhamento passo a passo das instruções, nos indicadores do painel frontal.

2.3 - Testes de Memória RAM

Uma vez que se torna totalmente impraticável aplicar e verificar todos os padrões em uma pastilha de memória semicondutora, em virtude do extenso número de configurações possíveis, há que se fazer uma seleção dos padrões a utilizar, com base na estrutura interna da mesma e dos tipos mais comuns de defeito, entre os quais pode-se mencionar²³ :

D1 - Fuga em uma das entradas, que necessita mais

do que a corrente especificada para funcionamento.

D2 - Decodificador aberto, impedindo a utilização de parte da matriz de armazenamento.

D3 - Múltiplo endereçamento, onde os dados são lidos de ou escritos em outras células além da que estiver especificada pelo endereço.

D4 - Inabilidade em manter os dados entre os ciclos de restauração ("REFRESH"), em memórias dinâmicas.

D5 - Circuitos abertos ou em curto, devido à metalização insuficiente da excessiva em uma das fases de fabricação.

D6 - Sensibilidades a padrões, o que implica em menor confiabilidade para determinadas configurações de dados armazenados.

D7 - Defeito em algum elemento interno da pastilha em relação à sua especificação.

D8 - Tempo de acesso lento, causado por cargas armazenadas nos circuitos de saída ou por longas linhas.

D9 - Recuperação lenta no amplificador de leitura, notada pela tendência do amplificador de permanecer no mesmo estado após uma longa série de bits similares, devido à carga imprópriamente acumulada.

D10 - Recuperação lenta de escrita, provocando um acréscimo no tempo de acesso para a realização de uma leitura imediatamente após uma escrita.

A organização intena de memórias semicondutoras consiste de u'a matriz com dois decodificadores, cada um recebendo parte das linhas de endereço e ativando a linha e a coluna correspondente ao mesmo. A célula que estiver

na interseção será selecionada para escrita ou leitura quando os sinais apropriados forem ativados. Normalmente a matriz de células é quadrada, havendo, entretanto, variações na geometria devido ao processo de fabricação e ao projeto dos semicondutores empregados.

Com o propósito de descobrir um ou vários dos possíveis defeitos citados anteriormente, vários métodos de teste tem sido propostos^{7,19,23,27}, entre os quais:

T1 - Teste de Seleção Múltipla de Endereço:

Os e ls alternados são escritos em posições sucessivas, começando com o endereço zero. Cada célula é então lida e verificada, enquanto se utiliza uma sequência de endereço, complemento do endereço, endereço, endereço mais um, e assim por diante. Esta sequência do endereçamento testa todas as possíveis trocas de N bits de endereço, onde N é o número de linhas de endereço da memória. Ao final deste procedimento, a memória é lida sequencialmente e o padrão de dados inicialmente gravado é conferido.

T2 - Teste de Deslocamento da Diagonal:

Este teste inicia com a memória contendo zeros em todas as células, exceto os da diagonal da matriz, que contém ls. As células são todas lidas e verificadas, pela ordem, resultando longas sequências de zeros seguidas por um bit 1. A matriz é então carregada com a diagonal de ls deslocada uma célula para a direita, e novamente verificada. Este processo se repete até que a diagonal retorne à posição inicial, quando então o dado 1 já foi armazenado em cada célula de memória. O teste é a seguir repetido com uma diagonal de zeros e com as demais células em 1.

T3 - Teste de 1s e 0s Marchando:

A memória é inicialmente preenchida com um padrão e, a seguir, lida sequencialmente, a partir do endereço zero e até o final da mesma. A cada endereço, o dado correspondente é conferido, complementado e reescrito na mesma posição. Ao se atingir o final, a memória é novamente percorrida, em ordem reversa, seguindo o mesmo procedimento até o endereço inicial, quando então o padrão inicial é invertido e todos os passos novamente executados.

T4 - Teste de Paridade Armazenada:

Similar ao anterior, mas consome menos tempo. Consiste em preencher a memória de forma que cada célula contenha a paridade do endereço a que corresponde. As células são da mesma forma lidas em ordem crescente e verificadas, sendo, ao final, todos os passos repetidos para a paridade inversa.

T5 - Teste de 1s e 0s Caminhando:

A memória é inicializada com um mesmo padrão. A seguir, cada célula é testada, escrevendo-se na mesma o inverso do padrão inicial, verificando todas as demais células para averiguar se permaneceram inalteradas e voltando à célula sob teste para confirmar se contém o inverso das demais. A célula é então reescrita com o padrão original e a sequência de operações, repetida para a próxima, até que todas tenham sido testadas. O procedimento é a seguir executado para a memória inicializada com o padrão inverso.

T6 - Teste de 1s e 0s Galopando:

Análogo ao anterior, com a diferença que, durante a verificação das demais células, a posição em teste é lida

após a leitura de cada uma delas. Da mesma forma que no teste anterior, a sequência é repetida para cada célula e todo o procedimento é executado também com o padrão inverso.

T7 - Teste Ping-Pong:

Consiste de uma variação simplificada de T6, com um único passo, que, ao invés de verificar o estado de todas as demais células durante o teste de cada uma delas, apenas inverte os bits de endereço, um por vez, e verifica a célula correspondente, voltando, após cada leitura, para a célula em teste.

T8 - Teste de Recuperação de Escrita:

Este teste verifica o efeito de se escrever dados, cada escrita seguida por uma leitura, para todos os possíveis pares de células. A memória inicia com qualquer valor; a célula em teste recebe um padrão e, para cada uma das demais células, segue-se a sequência: escrever o inverso do padrão na célula selecionada, ler a célula em teste, inverter o padrão da célula selecionada e ler novamente a célula em teste. Quando toda a sequência terminar em relação à primeira célula, testa-se a segunda, e assim por diante. Ao final, todo o procedimento é repetido com o padrão inverso do que foi inicialmente adotado nas células em teste.

O número de acessos exigido pelos testes, para u'a memória de tamanho N , é de, respectivamente, $5N$, $2N(\sqrt{N}+3)$, $10N$, $4N$, $2N(N+3)$, $2N(2N+1)$, $N(2\log_2 N+3)$, $2N(4N-3)$, incluindo o carregamento inicial da memória com os padrões que se fizerem necessários e supondo a matriz quadrada, no

caso de T2.

Uma vez que o ciclo de instrução de microprocessadores é bastante superior ao ciclo de leitura ou escrita de memória, torna-se impossível testar por programa acessos com a máxima taxa permitida. Desta forma, as falhas D9 e D10 não serão normalmente detetadas por esta modalidade de teste. Ocorre ainda que a falha D1 só poderá ser detetada se aparecer em extensão suficiente para prejudicar o funcionamento da memória.

Outro ponto a considerar no emprego de testes por programa é que a sequência de endereços aplicada tem pouco significado. Uma vez que as linhas correspondentes voltam sempre ao seu estado quiescente, entre dois ciclos consecutivos de memória. Desta forma, testes semelhantes a T1 perdem sua utilidade nestes casos.

As falhas D2, D3, D5, D6, D7 e D8 são todas detetadas pelos testes T2, T3, T4, T5, T6, T7 e T8. Os procedimentos mais complexos verificam melhor alguns aspectos, como, por exemplo, D6, com a consequente desvantagem de exigirem um número de acessos mais elevado.

A falha D4 necessita de um procedimento específico para ser detetada, consistindo no preenchimento da memória com determinado conteúdo e sua verificação, após decorrido um intervalo superior ao tempo de varredura das colunas ou linhas da matriz de memória das pastilhas constituintes. Para este caso, pode-se fazer uso de, por exemplo, sequências de endereços crescentes ou decrescentes, dados pseudo-aleatórios ou padrões básicos.

Deve-se testar ainda a independência entre bits

de uma mesma palavra de memória, a fim de assegurar a inexistência de acoplamentos unidirecionais ou bidirecionais entre os mesmos.

Quando as pastilhas constituintes apresentam apenas um bit de largura, esta verificação pode ser realizada em uma única palavra de cada conjunto de pastilhas e é útil para localizar curto-circuitos que estejam presentes na placa. Nos demais casos, a verificação deve se estender ao longo de toda a memória.

O emprego de um padrão circulante, correspondente à largura da pastilha, com um bit 1 e os demais em 0, bem como o inverso desta configuração, se mostra bastante adequado e simples de gerar para este tipo de teste.

Um outro método mais eficiente consiste em escrever e conferir K máscaras com metade dos bits em 1 e a outra metade em 0, sendo a máscara I constituída por I repetições de $N/(2^I)$ bits 1 e $N/(2^I)$ bits 0, com I variando de 1 a $\log_2(N)$, onde N é a menor potência de 2 maior ou igual à largura da palavra. Esta sequência de máscaras deteta qualquer acoplamento bidirecional entre os bits de uma palavra e, se complementada com o inverso das máscaras, cobre igualmente os acoplamentos unidirecionais.

2.4 - Testes de Memória ROM

Para verificar o conteúdo de memórias ROM, é usual o cálculo de algum tipo de informação redundante com os dados gravados e posterior comparação com o valor esperado.

Tem-se, por exemplo, o LRCC (Longitudinal Redundanç

cy Check Code), que consiste na paridade longitudinal para cada um dos bits da palavra, o CRCC (Cyclic Redundancy Check Code), obtido através de um polinômio gerador aplicado aos bits de cada palavra, e a soma acumulada em um determinado número de bits, entre as formas mais empregadas de informação redundante para um bloco de dados.

Todos estes métodos permitem detetar trocas simples de bits e uma percentagem de trocas múltiplas, servindo como uma primeira verificação do correto conteúdo de uma ROM.

Quando se dispõe de um periférico de armazenamento, pode-se guardar em um arquivo uma cópia do conteúdo de cada uma das ROMs e, através da comparação palavra a palavra, detetar quaisquer trocas de bits que ocorram.

2.5 - Testes de Periféricos

A idéia básica em testes de periféricos consiste em procurar utilizá-los de maneira a garantir seu funcionamento em condições normais de operação.

2.5.1 - Periféricos de Saída

Em periféricos de saída, é usual o procedimento de gerar um conjunto de linhas de modo a ter em cada posição todos os caracteres utilizados pelo menos uma vez.

Testes mais específicos dependem do tipo de periférico de saída; em dispositivos com partes mecânicas, tais como teletipos e impressoras, costuma-se ainda verificar o alinhamento das partes móveis através da impressão de linhas de tamanho crescente ou decrescente, com vértice de

convergência em uma das bordas ou no centro; a regularidade do espaçamento entre caracteres por meio da repetição do mesmo símbolo em toda uma linha; espaçamento entre linhas, troca de folha, etc. Em vídeos, testa-se também a memória RAM associada aos mesmos.

2.5.2 - Periféricos de Entrada

Para periféricos de entrada, deve-se gerar uma massa de dados para ser lida e conferida, visual ou automaticamente. A primeira alternativa pode ser vantajosamente empregada no caso de teclados, onde o eco visual dos caracteres enviados ou o valor do código gerado facilita sobremaneira a detecção de quaisquer anormalidades. Já a segunda se mostra mais adequada para leitoras de cartões ou de fitas de papel, onde é possível verificar a correspondência entre as perfurações e sua detecção ótica ou mecânica, usando um conjunto de caracteres fixo e preestabelecido.

Em qualquer dos casos, informações disponíveis, no decorrer dos testes, provenientes dos registros de estado do periférico em teste, constituem valiosos elementos para a verificação do correto funcionamento dos mesmos.

2.5.3 - Periféricos de Armazenamento

Em periféricos de armazenamento, tais como discos, diskettes, fitas, memórias de bolhas, K7, etc., a execução de um maior número de comandos é normalmente necessária para verificar seu perfeito funcionamento. Entretanto, uma vez que estes periféricos são usualmente empregados para man

ter informações, este tipo de teste costuma residir em um dos periféricos e ser selecionado e executado apenas quando o núcleo do sistema estiver em ordem.

III - DESCRIÇÃO DO MICROCOMPUTADOR POTI

3.1 - Apresentação

O POTI é um microcomputador integralmente desenvolvido pelo Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro. Consiste, em sua configuração mínima, de uma unidade de processamento com microprocessador 8080, memórias (ROM e RAM) interfaces e periféricos: teclado, vídeo, impressora e disco.

Dispõe de um sistema operacional em disco (SOCO) e de um compilador para uma linguagem de alto nível (PLTI), semelhante ao PL/1, o que lhe permite desenvolver e executar programas localmente. Este compilador foi implementado através de um interpretador, a fim de propiciar maior portabilidade do software.

3.2 - Organização

Os diversos componentes do sistema se interligam através de uma via padrão de 44 linhas, descritas na tabela 3-1.

As linhas, com exceção de INTA IN, INTA OUT, HOLDA IN e HOLDA OUT, são conectadas em paralelo a todas as interfaces. As quatro linhas mencionadas são ligadas em cadeia, a partir da UCP. As saídas INTA OUT e HOLDA OUT de uma interface vão as linhas INTA IN e HOLDA IN da próxima. As interfaces que não se utilizam de interrupção e DMA tem estes sinais simplesmente conectados da forma descrita, diretamente no painel de interconexões.

A figura 3-1 apresenta a configuração do sistema

16	Linhas de Endereço	A00 - A15
8	Linhas de Dado	D0 - D7
6	Linhas de Alimentação	+12 V +5 V (2 LINHAS) 0 V (2 LINHAS) -12 V
5	Linhas de Sincronismo	I/OR I/OW MEMR MEMW NOREADY
2	Linhas de Controle	REFRESH RESET
3	Linhas para Interrupção	INT INTA IN INTA OUT
3	Linhas para DMA	HOLD HOLDA IN HOLDA OUT
1	Linha de Referência	Ø2 TTL

TABELA 3-1 - Linhas da via Padrão

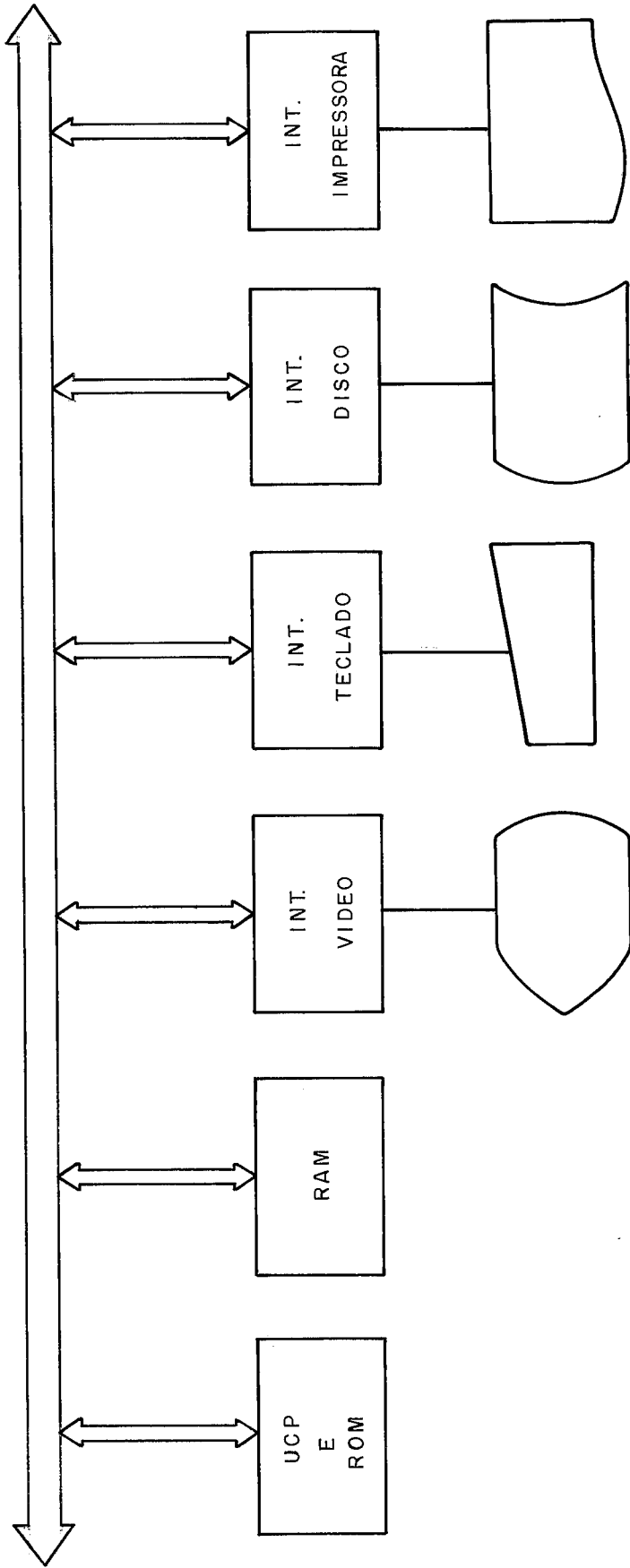


Fig. 3-1
Arquitetura do Poti (Configuração mínima)

básico, com 16 Kbytes de memória RAM.

3.3 - Unidade Central de Processamento

A UCP do sistema é constituída por um microprocessador 8080, com capacidade de atender pedidos de interrupção e de acesso direto à memória. A largura das palavras que o processador opera em paralelo é de 8 bits(1 byte) e sua capacidade de endereçamento é de 65536 palavras. O ciclo de máquina é de 500 ns, suas 72 instruções levam de 4 a 17 ciclos para execução, havendo necessidade de um ciclo adicional para cada escrita em memória, devido ao protocolo envolvido.

A figura 3-2 ilustra a alocação de endereços para os registros de entrada e saída, por onde se faz a comunicação com os periféricos, bem como a distribuição do espaço de endereçamento do processador.

3.4 - Memória RAM

A memória principal é organizada em bytes, denominando-se bit 7 ao mais significativo e bit 0 ao menos significativo bit de cada byte.

O sistema pode conter um máximo de 48 Kbytes de memória RAM, ocupando os endereços físicos entre 4000 e FFFF(HEX). Utilizam-se em sua implementação pastilhas de memória dinâmica, sendo o conteúdo das mesmas restaurado periodicamente pela UCP, à razão de um acesso a cada instrução executada, durante os intervalos em que a via não estiver sendo usada pela instrução corrente, não havendo portanto degradação no desempenho da UCP.

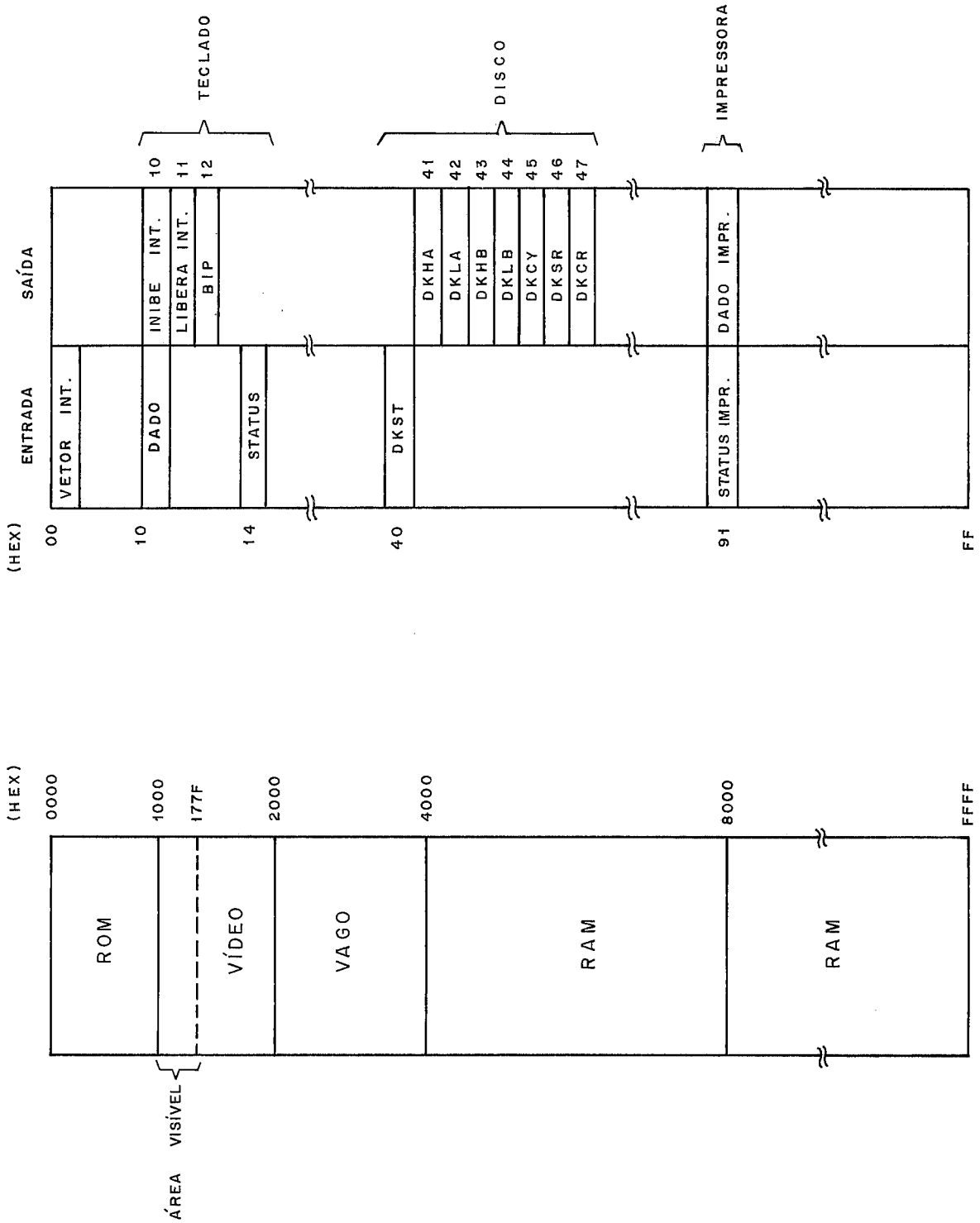


Fig. 3-2
Alocação dos Endereços dos Registros e de Espaço de Endereçamento

3.5 - Memória ROM

Os endereços entre 0000 e 1000(HEX) estão reservados para 4 Kbytes de memória ROM, onde estão gravados o interpretador da linguagem PLTI, algumas rotinas auxiliares e os procedimentos de inicialização do sistema e de tratamento de interrupção.

Ao ser ligado ou inicializado o sistema, a UCP passa a executar instruções a partir do endereço zero de seu espaço de endereçamento, onde se encontra o procedimento de carga do sistema operacional(partida fria).

3.6 - Periférico Teclado

O teclado compreende 52 teclas, sendo 47 codificadas, 2 de função, 2 teclas de deslocamento e 1 tecla de controle; sua saída apresenta código ASCII de 7 bits, em paralelo.

O código de saída, juntamente com informações de estado, sofre um processo de serialização para ser transmitido à interface de teclado, que por sua vez os restaura, recuperando a forma original. Este procedimento diminui o número de fios necessários à interligação do teclado com sua interface.

A interface de teclado é a única que pode gerar interrupções, desde que habilitada através de um de seus registros.

3.7 - Periférico Vídeo

A memória de vídeo ocupa os endereços entre 1000 e 2000(HEX), podendo ser acessada por qualquer instrução,

da mesma forma que uma palavra da memória principal. É também constituída por pastilhas de memória dinâmica, que, entretanto, não necessitam de restauração externa, uma vez que são constantemente lidas para manter visível, na tela, os caracteres correspondentes aos códigos ASCII presentes nos primeiros 1920 endereços (24 linhas por 80 caracteres). O restante dos 4 Kbytes é utilizado de modo normal, como memória do sistema, devendo-se notar que, durante a ocorrência de acessos pelo controlador de vídeo, a UCP pode ter que esperar até 88 us para completar um acesso à memória do vídeo, o que representa, no pior caso, uma degradação média de 15% na velocidade de execução das instruções.

3.8 - Periférico Disco

A interface de disco inclui circuitos de DMA e um controlador/formatador para até oito unidades, cada uma com capacidade formatada de 2,4 Mbytes.

As unidades aceitam discos removíveis, tipo cartucho, com duas superfícies magnéticas, 203 trilhas por superfície e 12 setores por trilha. Os setores são delimitados por fendas detetáveis opticamente, sendo cada setor formatado com um preâmbulo de 26 bytes zero, 1 bit de sincronismo, 2 bytes de cabeçalho, 512 bytes de dado, 2 bytes de soma acumulada dos dados e um postâmbulo com 2 bytes zero, pela ordem. O cabeçalho contém o número do cilindro correspondente, que é automaticamente conferido durante leituras ou escritas. Da mesma forma, a soma acumulada consiste de informação redundante com a área de dados, e é verificada

da durante leituras e gerada durante escritas, a fim de propiciar maior confiabilidade nas transferências de dados.

Há comandos de leitura, verificação e escrita de setor, além de posicionamento, inicialização e proteção da unidade contra escrita. Durante a área de dados do setor, ocorrem entre 1 e 512 transferências de dado por DMA entre a memória principal e a interface de disco, sem intervenção da UCP, que apenas inicia o comando. O sentido da transferência depende do comando em execução.

3.9 - Periférico Impressora

O sistema comporta uma impressora do tipo série, com matriz de pontos, velocidade até 165 caracteres por segundo, 132 caracteres por linha, código ASCII de 7 bits, impressão de caracteres normais ou alongados e armazenamento interno para uma linha. A impressora se conecta ao sistema por meio de uma interface paralela.

3.10 - Periféricos Adicionais

A configuração descrita se refere ao sistema básico. Outros periféricos podem ser conectados através da via padrão, desde que se utilize uma interface apropriada para cada um deles.

IV - TESTES APLICADOS AO POTI

4.1 - Sequenciamento do Testes

O programa de teste desenvolvido para o POTI reside em uma ROM de 1 Kbytes, que substitui a primeira ROM do sistema quando se deseja testá-lo ou verificá-lo. Este procedimento permite que o programa tenha início ao se ligar o sistema, sem necessidade de chaves para comutar do modo de operação normal para manutenção; evita ainda o dispêndio de uma ROM para cada unidade e possibilita sua implantação mesmo em sistemas já definidos, sem necessidade de quaisquer alterações em circuitos e interconexões.

O programa foi implementado no sentido de inicialmente testar a parte mais central do sistema, e, a seguir, fazendo uso das partes já verificadas, ir testando os demais módulos, como mostra a figura 4-1.

De acordo com esta filosofia, o programa tem início com o teste da UCP, onde, em caso de erro, o microprocessador para. Em funcionamento normal, um sinal audível (BIP) é emitido ao final do teste, para indicar que a parte central se encontra em perfeita ordem.

O próximo módulo a ser examinado é o vídeo, que é submetido a dois tipos de teste: um, visual, para a parte de varredura e geração de caracteres, e outro, automático, para a memória de armazenamento.

Deste ponto em diante, o vídeo, já verificado, está disponível para emitir quaisquer mensagens referentes aos demais módulos. A primeira mensagem, quando não ocorrem falhas, indica que a UCP e o vídeo estão funcionando.

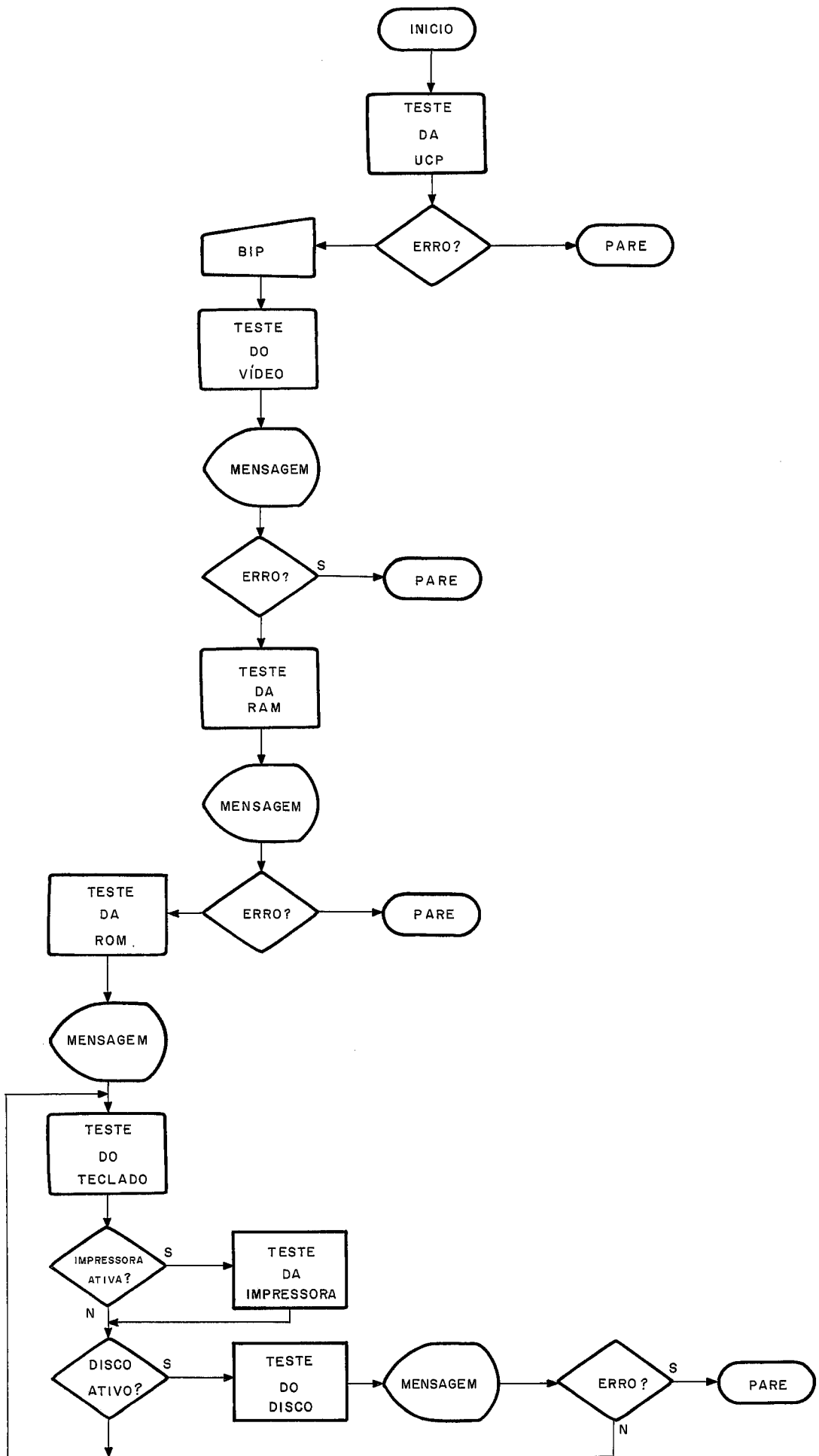


FIG. 4-1 - FLUXOGRAMA DO PROGRAMA DE TESTE.

