


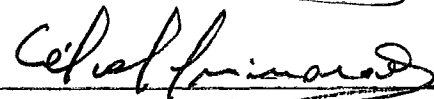
SISTEMAS DE INTERRUÇÃO.
UMA CONTRIBUIÇÃO AO COMPUTADOR G-10

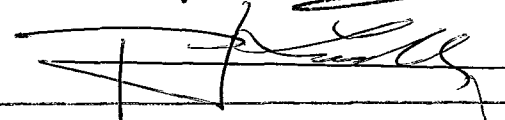
Jorge da Cunha Pereira Filho

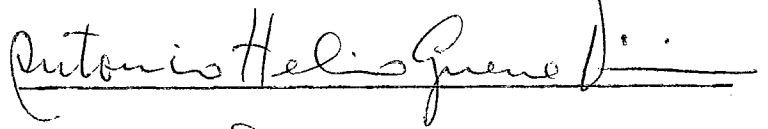
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA (M.Sc.).

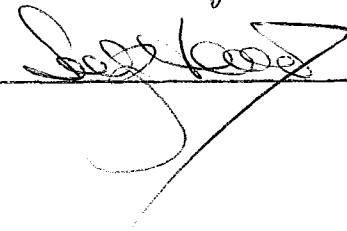
Aprovada por:











RIO DE JANEIRO
ESTADO DO RIO DE JANEIRO - BRASIL
JUNHO 1975

AGRADECIMENTOS

A minha esposa, MARLENE, pela compreensão que teve durante o curso de Mestrado e ainda na fase de elaboração desta Tese.

Ao CF(EN) CESAR MOACIR BASTOS CARDOSO, Coordenador do GTE (Grupo de Trabalho Especial), responsável pelo projeto do computador G-10 (Projeto "Guaranys") pela oportunidade que criou de aplicar os conhecimentos adquiridos no curso de Mestrado, quando contratou o projeto do "software" com a PUC/RJ, ao qual fui integrado, e ainda todo o apoio e incentivo para o desenvolvimento desta Tese.

Aos meus colegas de trabalho, do Departamento de Informática da PUC/RJ, no projeto de "software", que forneceram grande parte da bibliografia e outras informações para a elaboração desta Tese, especialmente VICTOR MORENO (Ph.D.) e FIRMO FREIRE (Ph.D.).

Aos Assessores do GTE, especialmente SERGIO POMPEIANO MOTTA (M.Sc.), por terem facilitado a obtenção de informações e incentivado o desenvolvimento desta Tese.

Aos meus professores do curso de Mestrado na COPPE/UFRJ, especialmente a LUCIANO PEREIRA (Ph.D.), orientador, JEAN PIERRE LAVELLE (Ph.D.) e CELIO GUIMARÃES (Ph.D.) pelos seguros e preciosos conselhos que nortearam a conclusão desta Tese.

A presença na Banca do Prof. ANTONIO HELIO GUERRA VIEIRA (Ph.D.), responsável pelo desenvolvimento do "hardware", e do Prof. SERGIO ROBERTO TEIXEIRA (Ph.D.), responsável pelo desenvolvimento do "software", do Computador G-10.

Aos professores NELSON MACULAN FILHO (M.Sc.) e MIGUEL JONATHAN (M.Sc.) pelo apoio recebido durante a realização do curso de Mestrado.

A todos aqueles cujo nome não foi citado mas que tornaram possível a realização desta Tese.

RESUMO

Esta Tese descreve de forma progressiva o fenômeno das interrupções. Inicialmente conceitua as interrupções e a terminologia técnica utilizada para definir alguns dos seus principais parâmetros. Mostra as classificações mais utilizadas para dividir as interrupções em grupos e as diversas maneiras usadas para atendê-las. Estabelece os conceitos de prioridade, máscara e mudança de contexto.

Vários sistemas de interrupção de computadores das mais diversas origens e finalidades são descritos, inclusive alguns que são considerados "clássicos", chegando até os mais atuais, voltados para processamento em tempo real.

Uma descrição geral do computador G-10 é feita, com enfoque maior em seu sistema de interrupções. Uma proposta para um novo sistema de interrupções para o mesmo computador é proposta. Algumas reflexões sobre a solução proposta são feitas.

ABSTRACT

This Thesis describes in a progressive way the interrupt process. Initially it treats the same notions about interruption and related technical terminology used to define some of main parameters of the process. It discusses the more commonly used classifications to distinguish among groups of interrupts and their respective solutions. It establishes some concepts of priority, mask and context change.

Several computer interrupt systems of widely differing origins and purposes are then described, including some which are considered to be "classic", concluding with the most recent, which are oriented to real-time processing.

A general description of the G-10 computer is presented, giving special emphasis to its interrupt system. Suggestions are made for a new interrupt system for this computer and some conclusions presented on the value of the proposed solution.

ÍNDICE

	PÁG.
APRESENTAÇÃO	1
INTRODUÇÃO	4
CAPÍTULO 1 - CONCEITUAÇÃO DE INTERRUPÇÃO	5
1.1. Introdução	5
1.2. Descrição Estrutural de um Computador	5
1.3. Operações de Entrada e Saída	6
1.4. Origem das Interrupções	6
1.5. Técnicas de Seleção e Atendimento de Interrupções	7
1.6. Sequência de Atendimento de Interrupções ..	8
1.7. Alterações nos Métodos Básicos de Seleção e Atendimento	9
1.8. O uso do Vetor de Interrupção	10
1.9. Mudança de Contexto para Atender a Interrupção	10
1.10. Retorno ao Contexto do Programa Interrompido	11
1.11. Tratamento de Interrupções Múltiplas	12
1.12. Características Gerais dos Sistemas de Interrupção	13
1.13. Interrupções Internas	15
1.14. Alguns Exemplos de Interrupções Internas..	17
1.15. O Ciclo das Interrupções	18
1.16. Identificação de Dispositivos nas Interrupções Externas	20
1.17. Estrutura do Sistema de Interrupção	21
1.18. Limitação das Interrupções	23
1.19. Instruções associadas ao Sistema de Interrupções	24
1.20. Ordenação das Estruturas de Prioridade....	26
1.21. Mudança de Contexto com o Uso de Máscara de Interrupção	28

	PÁG.
CAPÍTULO 2 - DESCRIÇÃO DE ALGUNS SISTEMAS DE INTERRUPTÃO....	31
2.1. O Sistema de Interrupção do Sistema IBM/ /360	31
2.2. O Sistema de Interrupções do Computador FACOM U-200	33
2.3. O Sistema de Interrupções do Computador PDP-11/45	37
2.4. O Sistema de Interrupções dos Computado <u>r</u> res NOVA	42
2.5. O Sistema de Interrupções do Computador MITRA-15	46
2.6. O Sistema de Interrupções dos Computado <u>r</u> res ARGUS 700	51
 CAPÍTULO 3 - O COMPUTADOR G-10	 58
3.1. Descrição Geral	58
3.2. Via Comum de Comunicação e Via de Inter <u>r</u> rupções.....	60
3.3. Processador Central	65
3.4. Estrutura de Entrada/Saída	69
3.5. Sistema de Interrupções	76
 CAPÍTULO 4 - UMA PROPOSTA PARA O SISTEMA DE INTERRUPTÕES DO G-10	 84
4.1. Premissas Básicas	84
4.2. Máscara de Interrupções	85
4.3. Rede de Registradores e Blocos de Con - texto	86
4.4. Mudança de Contexto	88
4.5. Convenções do Sistema	89
4.6. Organização da Memória	91
4.7. Sequência de Atendimento das Interrup - ções	92

	PÁG.
4.8. Alteração de Prioridades e Limitação das Interrupções	93
4.9. Instruções Relativas às Interrupções	93
4.10. Interrupções Múltiplas e Restauração do Contexto Anterior	94
4.11. Retorno da Interrupção	94
4.12. Inicialização do Sistema	95
4.13. Mudança de Modo e Conjunto de Instruções...	95
4.14. Chamada e Retorno Automáticos de Sub-Ro- tinas	96
4.15. Mudança de Tarefas	97
CAPÍTULO 5 - CONCLUSÕES	98
BIBLIOGRAFIA	100

APRESENTAÇÃO

O computador G-10, objeto maior desta Tese, é o primeiro projeto nacional de um computador, fruto exclusivamente do trabalho de brasileiros, que tem possibilidades de industrialização. É sem dúvida um dos mais importantes projetos tecnológicos já realizados no país e, considerando-se a "tecnologia de ponta", tem a total primazia. O esforço para sua construção culminou com a inauguração do protótipo no dia 19 de maio próximo passado, esperando-se que uma versão compactada esteja disponível no mes de setembro próximo vindouro. Ao ser concluída a fase de projeto, inicia-se nova fase de aperfeiçoamento, já voltada e orientada para a mais breve industrialização e comercialização.

As origens do computador brasileiro, remontam aos anos de 1960, na enfase que a Marinha de Guerra do Brasil, através da Diretoria de Comunicações e Eletrônica deu, não apenas à modernização dos seus equipamentos eletrônicos, mas também à nacionalização dos mesmos. A presença de sistemas de processamento de dados tácticos existentes nos modernos navios, baseados em computadores digitais, despertou a idéia de também estimular sua construção no Brasil.

Conjugou-se este desejo com a procura que, a partir de 1968, o BNDE, através do FUNTEC fazia de áreas tecnológicas para investimentos que resultassem em significativo avanço econômico para o país. Não havendo dúvidas quanto à sua importância e efeito multiplicador, iniciaram-se em 1969 os primeiros contatos entre a Marinha e o BNDE e foram feitos os acordos para a criação do **Grupo de Trabalho Especial (GTE)** para administrar o desenvolvimento do projeto. Os recursos solicitados pela Marinha foram incluídos no PND.

Em 18/02/71, o GTE foi criado, pelo decreto nº 68.267, conforme exposição de motivos do Ministro da Marinha, ad referendum do Secretário do Planejamento da Presidência da República, tendo por coordenadores um representante da Marinha e outro do BNDE iniciando-se seus trabalhos efetivamente em outubro do mesmo ano. O representante designado pela Marinha foi o Comandante GUARANY, um dos maio

res entusiastas e principais idealizadores do projeto que hoje tem o seu nome, reverenciando sua memória. Também a primeira letra "G", da sigla pela qual é conhecido o computador vem do seu nome.

Entre várias ações do GTE, que resultaram também na criação da DIGIBRAS - Empresa Digital Brasileira S.A., criada em julho de 1973, e da COBRA - Computadores e Sistemas Brasileiros S. A., criada em julho de 1974, destaca-se a condução do projeto do computador G-10. Para tal é como era desejável a cooperação entre a Universidade e a Indústria, foram contratados para o desenvolvimento do "hardware" e do "software" básico a FDTE/USP - Fundação para o Desenvolvimento Tecnológico da Engenharia, dirigida pelo Professor HELIO GUERRA (Ph.D.) e a firma EE - Equipamentos Eletrônicos S.A. o pré-projeto foi contratado em 11/09/72 no valor de ... Cr\$ 876 mil. O projeto, em 23/03/73, no valor de Cr\$ 21.109 mil.

Posteriormente havendo necessidade de desenvolvimento de "software" complementar foi contratada a CONSULPUC Serviços Técnicos Ltda., vinculada à PUC/RJ. O pré-projeto, contratado em 02/07/73 teve um valor de Cr\$ 1.093 mil. O projeto, contratado em 03/12/73 teve um valor de Cr\$ 10.340 mil. A equipe de ambos os contratos foi constituída com base no Departamento de Informática da PUC/RJ, sob a direção do Professor SERGIO TEIXEIRA (Ph.D.).

Somados os custos acima aos dos periféricos, Cr\$ 555 mil, do Estudo de Viabilidade, Cr\$ 279 mil, e do Projetos NUTES, Cr\$ 150 mil, temos um custo total líquido de Cr\$ 34.402 mil, bem baixo ou mesmo insignificante em relação aos benefícios que certamente trará à economia brasileira.

O processador, inicialmente definido, como "computador eletrônico para emprego em operações navais", teve seu objetivo redefinido para ser um "mini-computador para aplicações científicas e de engenharia, com possibilidades de extensão e adaptação para uso em sistemas de tempo-real e comunicações".

Sua principal característica é a existência de uma via única através da qual todos os blocos do sistema interagem e intercomunicam, tais como o processador, as memórias e os canais, o que lhe dá a flexibilidade para extensões e adaptações futuras,

como acima mencionado. Outras características são o tamanho de palavra de 16 bits e mais de uma centena de instruções de máquina com 2 operandos, com diversos e variados modos de endereçamento. A memória pode chegar, futuramente a 64K, embora o modelo atual tenha 32K. Os mais diversos tipos de dispositivos, tanto de baixa, média como alta velocidade podem ser conectados a controladores, que servem de interface com as vias dos seus canais. Podem ser usados até 4 canais de qualquer um entre 2 tipos, programado ou DMA, cada um suportando até 16 controladores.

INTRODUÇÃO

A Tese está dividida em 5 capítulos dos quais se vai desde uma descrição de interrupções, em forma didática, no 1º capítulo, até a conclusão, no 5º capítulo.

A descrição das interrupções, feita no 1º capítulo, inicia com conceituações genéricas sobre computadores e operações de entrada/saída, para então mostrar a origem do fenômeno das interrupções. São abordadas as técnicas de seleção e atendimento e o uso da máscara de interrupções e do vetor de interrupções, para facilidade da troca de contexto. O mecanismo das interrupções é estudado num nível de abstração que permite mesmo aos especialistas em "software" compreendê-lo, sem se aprofundar em problemas de eletrônica. Também são classificadas as interrupções e definidos os principais tipos das internas. O ciclo das interrupções é analisado para definir seus principais parâmetros. No mesmo nível de abstração são definidas as instruções típicas.

No 2º capítulo são descritos diversos sistemas desde o clássico IBM/360, passando pelo FACOM U-200, PDP-11/45 e NOVA até chegar a modelos mais voltados para processamento em tempo-real como o MITRA-15 e ARGUS-700. Pode-se aí ter uma idéia da diversidade de soluções embora possa-se depreender os princípios básicos utilizados.

No 3º capítulo é dada uma descrição geral do computador nacional, o G-10, com especial enfoque no sistema de interrupções para o qual se pretende dar uma contribuição. Suas principais características de arquitetura são evidenciadas, chegando-se toda via a detalhes de interesse.

No 4º capítulo é proposta uma nova solução para o problema das interrupções no G-10 mantendo-se todavia o nível de abstração dos capítulos anteriores.

Comentários e conclusões sobre a solução proposta constituem o 5º capítulo.

CAPÍTULO 1

CONCEITUAÇÃO DE INTERRUPTÃO

1.1 - Introdução

O conceito de interrupção fica bem compreendido partindo-se do contexto de paralelismo, descrito de forma magistral por Lorin [1].

Embora a existência do sistema de interrupções não seja essencial à caracterização de um computador digital de propósito geral, no conceito de Beizer [2], sua presença nos tipos atuais torna-se cada vez mais imperiosa. Abreviadamente chama-se, no texto que se segue, a uma tal máquina, de computador.

Dentro do critério de divisão em níveis, proposto por Bell e Newell [3], a descrição de computadores se restringirá aos três primeiros, ou seja: "PMS", "ISP" e "register-transfer". Ainda com relação ao segundo nível, ou seja, "ISP", somente será descrito o que diretamente se relacionar ao fenômeno das interrupções.

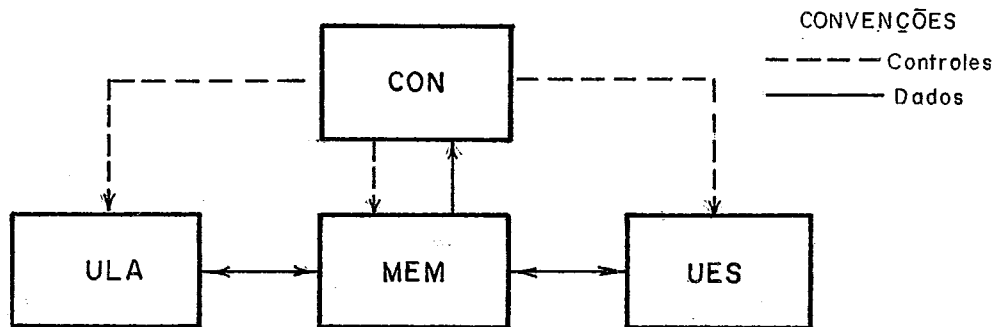
1.2 - Descrição Estrutural de um Computador

Uma descrição funcional de computador é a daquele que é capaz de simular qualquer outro, inclusive a si mesmo, segundo Beizer [2].

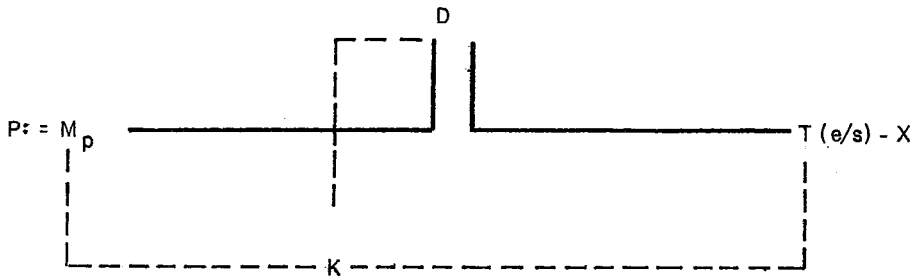
Em nível de "PMS" será adotada, todavia, a descrição estrutural, que é exemplificada para os seguintes blocos:

- a) memória ou unidade de armazenamento (MEM);
- b) unidade de controle (CON);
- c) unidade de lógica e aritmética (ULA);
- d) unidade de entrada/saída (UES).

O diagrama de blocos do computador IAS, segundo Langdon e Fregni [4] é apresentado a seguir:



Neste mesmo nível, adotando-se a notação de Bell e Newell [3] e tem-se, para o mesmo computador o diagrama "PMS" seguinte:



1.3 - Operações de Entrada e Saída

A classificação dos tipos de E/S usadas neste texto obedece a nomenclatura adotada pela maioria dos autores [5,6,7,8,9]:

- programada: quando a transferência de cada dado é feita totalmente sob controle de programa;
- DMA ("direct memory access"): quando a transferência de um bloco de dados é apenas iniciada pelo programa continuando sob controle do bloco de E/S.

1.4 - Origem das interrupções

A conclusão de uma operação de E/S programada acarreta a interrupção do programa em execução e ocorre um desvio para uma rotina de serviço de interrupção, que fará a transferência de

cada dado. Segundo Holland [6], esta é a forma de E/S mais usada nos minicomputadores, o que acarreta um número muito maior de interrupções do que nos computadores de processamento de dados ("EDP computers"), onde a transferência por blocos é a dominante. Em alguns casos o número de interrupções é proporcional ao número de "bits" transferidos, o que todavia, não é uma forma comum. Poder-se-ia compensar o alto número de interrupções num desses sistemas que transmitem os dados de forma serial, "bit" a "bit", por um sistema de interrupções muito eficiente. Porém nem sempre é este o caso e, para diminuir custos, o sistema de interrupções é quase sempre muito simples. Normalmente, nestes casos, o sistema possui um único nível de interrupção e a prioridade dos dispositivos é determinada pela ordem em que a rotina de serviços interroga o estado de cada dispositivo para determinar quem será atendido. A maioria todavia, usa um número maior de níveis que, segundo o mesmo autor [6], varia entre 16 e 256.

Mesmo usando-se a transferência tipo DMA, o processo para atender a interrupções é o mesmo.

1.5 - Técnicas de Seleção e Atendimento de Interrupções

Segundo Rinder [7], as técnicas usadas para selecionar e atender interrupções podem ser grupadas em 3 tipos básicos:

- 1) Uma única linha de pedido de interrupções é usada para todos os blocos ou dispositivos. Quando as interrupções são permitidas, uma rotina interroga cada um dos dispositivos, de acordo com uma ordem pre-estabelecida, para determinar qual será atendido. A prioridade está implícita na própria ordem em que os dispositivos se encontram incluídos numa Tabela. Esta tabela poderá ser alterada se não estiver gravada numa memória tipo ROM;
- 2) Uma única linha de pedido de interrupções é usada e há uma única linha de reconhecimento que passa através de todos os blocos ou dispositivos. A priorida-

de está implícita na própria posição ocupada pelo dispositivo na linha de reconhecimento. As unidades de maior prioridade serão interrogadas primeiro e, se não estão pedindo interrupção, permitem a passagem do sinal de reconhecimento para os blocos de mais baixa prioridade, na ordem. O dispositivo que pediu interrupção e foi reconhecido coloca seu endereço em linhas próprias, para se identificar;

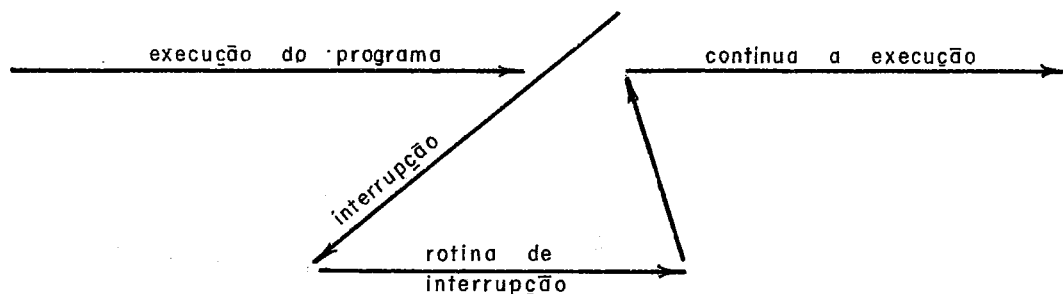
- 3) Diversas linhas de pedido de interrupção e diversas linhas de reconhecimento, uma para cada bloco ou dispositivo, são usadas. Os conflitos para interromper o PC são resolvidos por um sistema de prioridade, que pode ser por "hardware" ou usar uma memória tipo ROM ("firmware").

1.6 - Sequência de Atendimento de Interrupções

Segundo Kocher [10], a sequência de atendimento de uma interrupção, que pode ser implementada parte em "Firmware" e parte em "Software" é a seguinte:

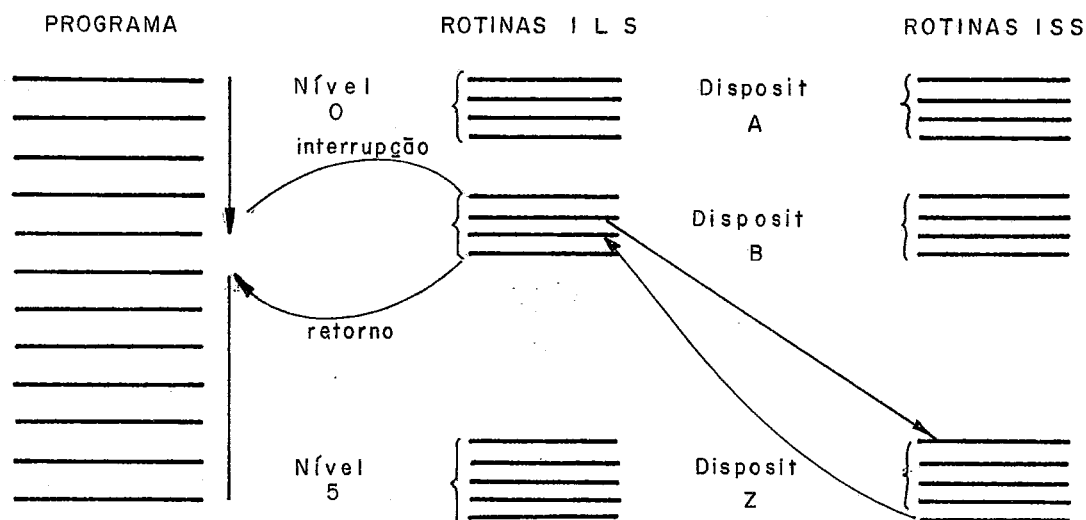
- 1) Salvar as informações correntes, contidas em todos os registradores (automaticamente ou sob controle do usuário);
- 2) Reconhecer o pedido de interrupção do dispositivo, enviando o sinal na linha de reconhecimento;
- 3) Executar as operações necessárias a atender o pedido de interrupção do dispositivo;
- 4) Restaurar o conteúdo original dos registradores;
- 5) Reassumir a execução do programa que foi interrompido.

Diagrama genérico para representar esta sequência, segundo [5] é:



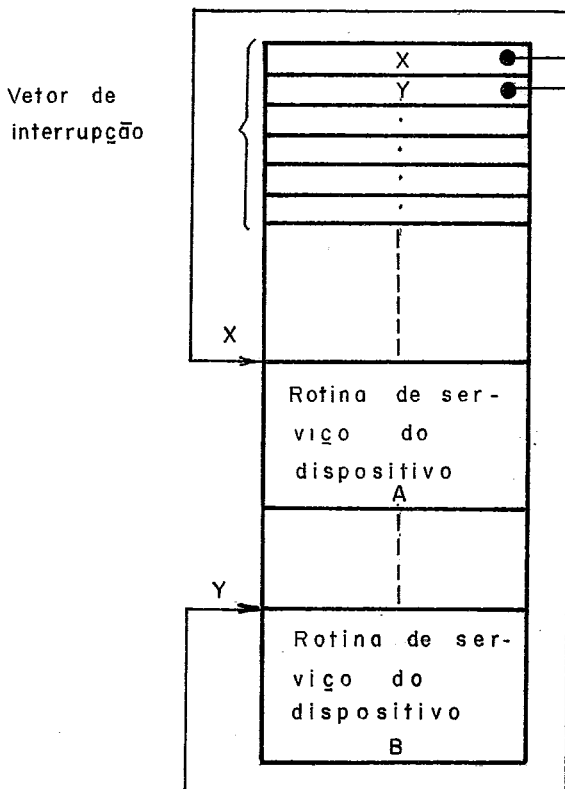
1.7 - Alterações nos Métodos Básicos de Seleção e Atendimento

Há computadores, todavia, que usam um dos 3 processos indicados anteriormente, porém com alguma alteração. Por exemplo, o computador IBM 1800 [3], usa um sistema semelhante ao terceiro tipo apresentado, com vários níveis de interrupção (24 níveis). Cada nível inclui diversos dispositivos. Há necessidade de, ao ocorrer uma interrupção, saber, dentro de um nível, qual o dispositivo que deve ser atendido. Assim, antes de desviar para uma rotina de serviço (ISS ou "Interrupt Service Subroutine") é chamada uma rotina que determinará o dispositivo a atender (ILS ou "Interrupt Level Subroutine"). A rotina ILS fará a chamada da ISS do dispositivo. Representa-se graficamente como se segue:



1.8 - O Uso do Vetor de Interrupção

O desvio para a rotina de serviço, quando feito diretamente (sem o uso de ILS), é feito por "hardware", de uma forma indireta. Haverá na memória um "Vetor de Interrupção", [11,12] que é constituído por uma sequência de posições de memória contendo, cada uma, o endereço da rotina de serviço de um dispositivo. A situação na memória é representada como no gráfico abaixo:

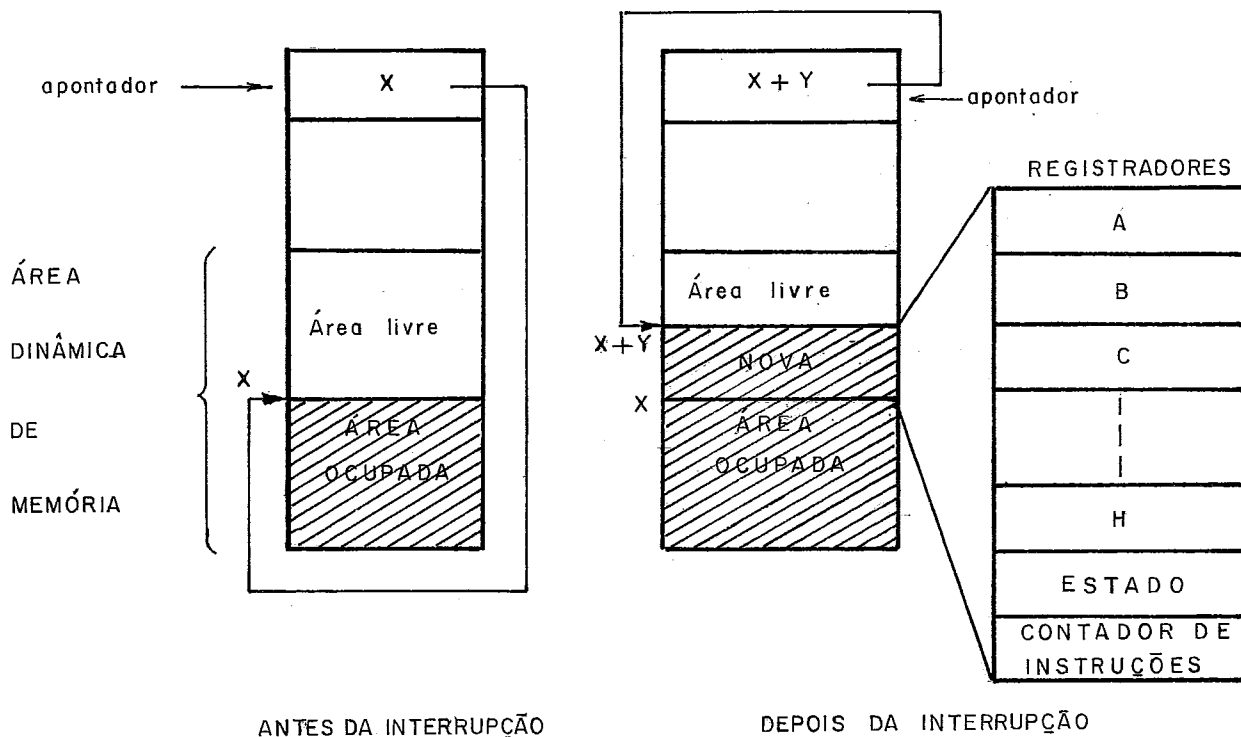


O desvio indireto para a subrotina é feito usando o Vetor de Interrupção, cujo conteúdo aponta para a rotina do dispositivo correspondente, em cada posição.

1.9 - Mudança de Contexto para Atender a Interrupção

Ao se iniciar a rotina de interrupção, quando for deixado a cargo do "software", salvar-se-ã os registradores que contém o estado corrente do programa interrompido. A área onde serão salvos os registradores pode estar contida dentro de cada rotina ou constituir uma parte comum da memória. Uma área comum permite ter um aproveitamento dinâmico da memória, sob controle de um apontador. Se são permitidas interrupções múltiplas, isto é, interrup

ções das próprias rotinas de interrupção, o uso da área dinâmica po de oferecer vantagens. Sejam Y registradores para salvar como mos trado no gráfico abaixo nas situações antes e depois:



A área dinâmica funciona como uma pilha e deve ser administrada por "software". O uso da pilha automática, administrada por "hardware" é muito semelhante. Apenas não dá ao programador a liberdade de manipulá-la, embora seja muito mais rápida.

Existem soluções intermediárias, isto é, parte dos registradores são salvos automaticamente numa pilha, por "hardware" e outra parte fica a cargo do programador, por "software". Neste caso, o programador pode escolher quais os registradores (normalmente os de trabalho) devem ser salvos e em que área, se na área da rotina ou numa área dinâmica.

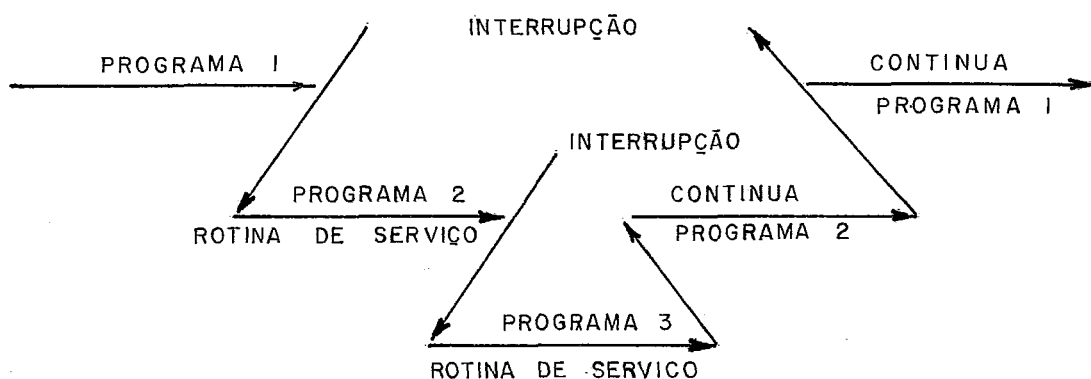
1.10 - Retorno ao Contexto do Programa Interrompido

Quando se vai retornar da rotina de serviço, os mes-

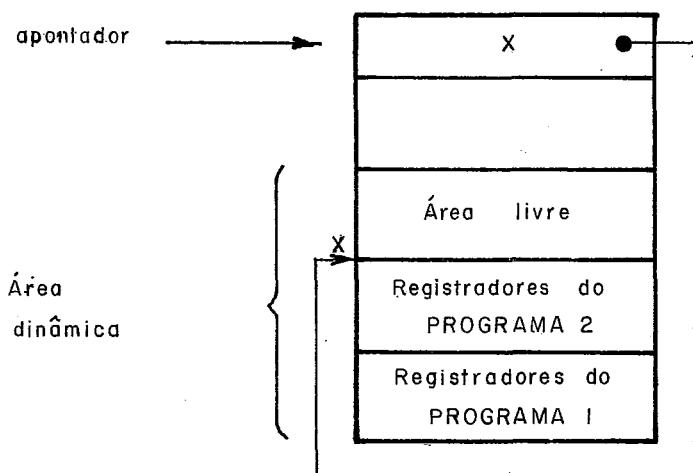
mos registradores que foram salvos devem ser restaurados, antes de ser reassumida a execução do programa interrompido (exatamente na instrução, que seria executada, pois o contador de instruções será restaurado, com o valor que tinha antes da interrupção). O apontador da pilha é decrementado.

1.11 - Tratamento de Interrupções Múltiplas

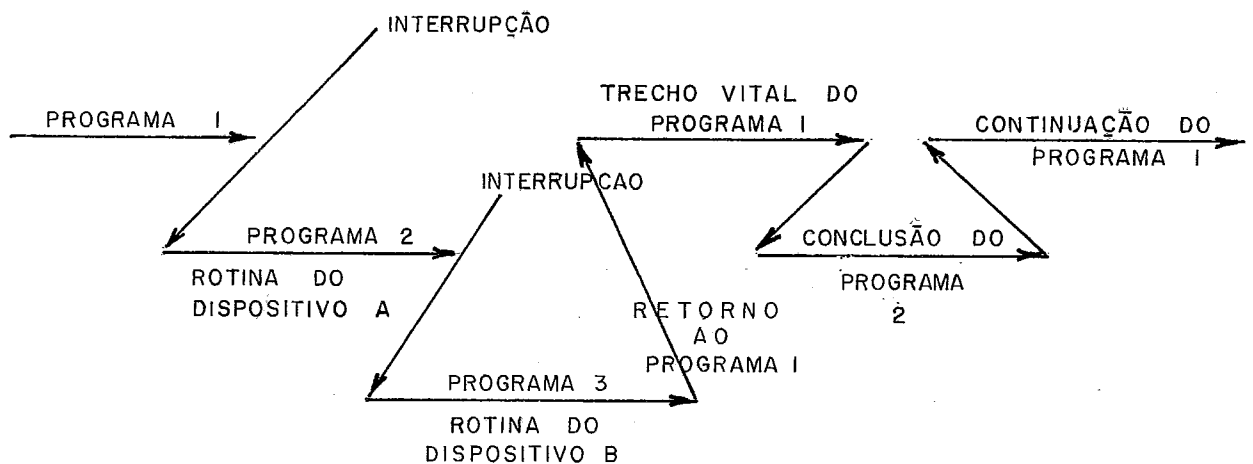
As interrupções múltiplas ficarão embutidas, o que se apresenta-se graficamente, segundo [5], como a seguir:



Executando-se o programa 3, usando-se uma área dinâmica, ela terá como conteúdo os registradores dos programas 1 e 2.



O uso da pilha automática obriga o retorno sequencial das interrupções, na ordem inversa em que foram embutidas. Toda - via, pode ser desejável um retorno não sequencial. Neste caso deve ser usada a área dinâmica que permite esta flexibilidade, conju gando-se o seu uso com o de uma tabela de controle que indique quais os programas que a estão utilizando. As interrupções serão atendi das neste caso como exemplificado no gráfico abaixo, conforme [5]:



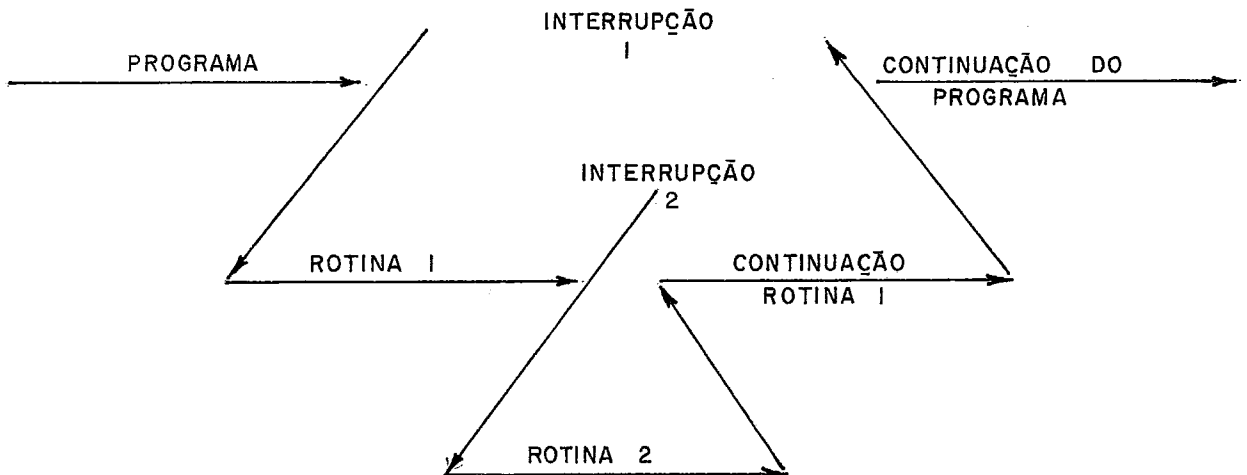
Esta flexibilidade é importante para alguns sistemas em tempo real, se bem que, conseqüentemente, consuma mais tempo.

1.12 - Características Gerais dos Sistemas de Interrupção

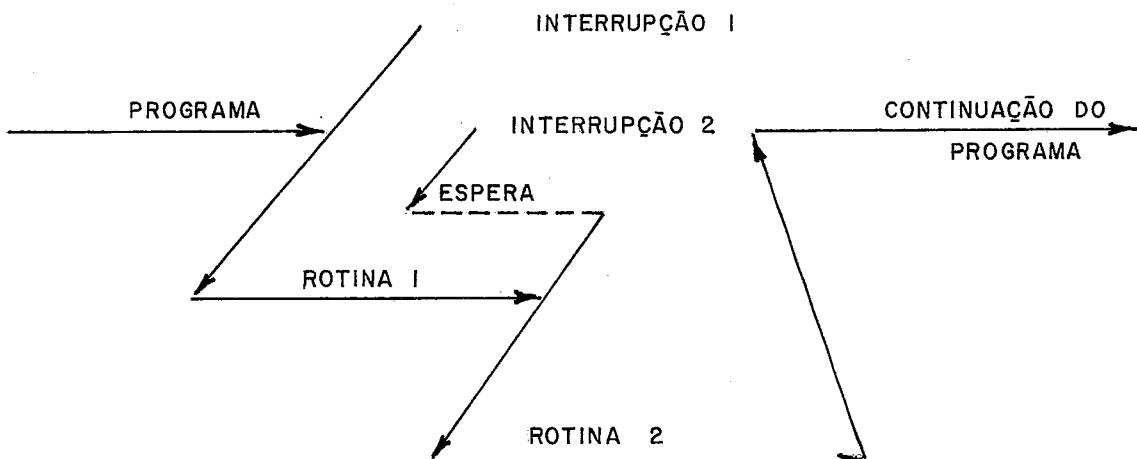
Uma característica de todos os sistemas de interrupção, segundo Holland [6] é a de que ao se atender uma interrupção, todos os demais pedidos são inibidos, durante a fase de passagem para a rotina de serviço ou no retorno. Pelo menos a primeira instrução da rotina de serviço será executada, antes que uma nova interrupção possa ocorrer, para permitir que instruções do tipo "inibir interrupções" e "salvar registradores de trabalho", quando for o caso.

Num sistema de vários níveis, as interrupções serão

organizadas de uma forma hierárquica, para permitir estabelecer um critério de prioridade. Sem um sistema hierarquizado de prioridades é impossível decidir a que nível atender, quando ocorrem pedidos em 2 níveis, simultaneamente. Ou, tendo atendido um pedido e na ocorrência de um segundo, durante a execução da rotina de serviço, também não é possível decidir sobre atender ou não o novo pedido. Segundo o mesmo Holland [6] a maioria dos sistemas não permite a ocorrência de uma nova interrupção de nível igual ou inferior, enquanto aquela estiver sendo atendida e até que se retorne da rotina de serviço. Atendendo-se a uma interrupção, ao ocorrer um pedido de um dispositivo cuja prioridade é maior, deve-se poder atendê-lo. Representa-se graficamente segundo [5], por:



O pedido de um dispositivo de prioridade igual ou inferior, ficará pendente até que termine a execução da rotina que atende ao nível atual. Representa-se graficamente segundo [5], por :

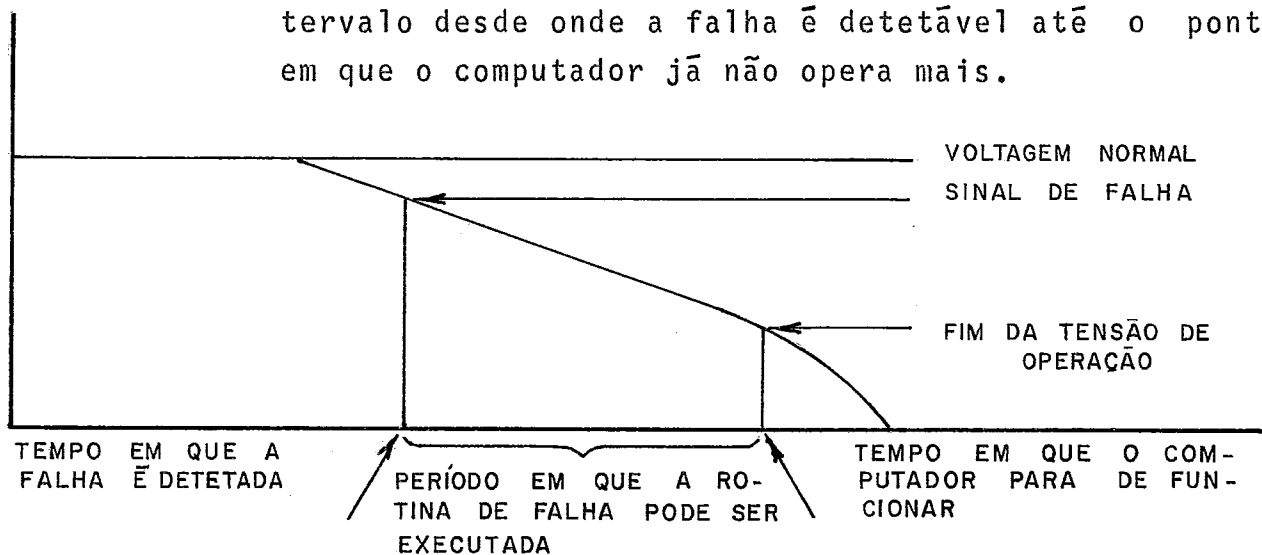


1.13 - Interrupções Internas

Interrupções não são geradas exclusivamente pelos dispositivos periféricos. Há as que são geradas pelo processador central ou por dispositivos diretamente ligados a ele e podem ser consideradas internas. Os tipos mais importantes, segundo Kocher [10] são:

- 1) Falha de Alimentação/Recuperação: Um dispositivo controla a voltagem da alimentação de força do computador. Ao cair a voltagem ocorre uma interrupção.

A rotina de interrupção para a máquina, porém de uma forma ordenada. Segundo [5], o tempo disponível para executar a rotina é de 1ms, coincidindo com o intervalo desde onde a falha é detetável até o ponto em que o computador já não opera mais.



Ao retornar a tensão da alimentação a níveis operacionais, um computador bem projetado gera uma interrupção interna e haverá um desvio para a rotina de recuperação. Esta restaura todas as condições anteriores à falha e o processamento continuará normalmente. Desta maneira, nem mesmo o operador terá que

se preocupar com a ocorrência de uma falha de alimentação. Em alguns computadores, todavia, a entrada na rotina de recuperação terá que ser feita manualmente, através do painel, pelo operador, pois o computador estará no estado "parado" e não haverá o sinal de interrupção para recuperação.

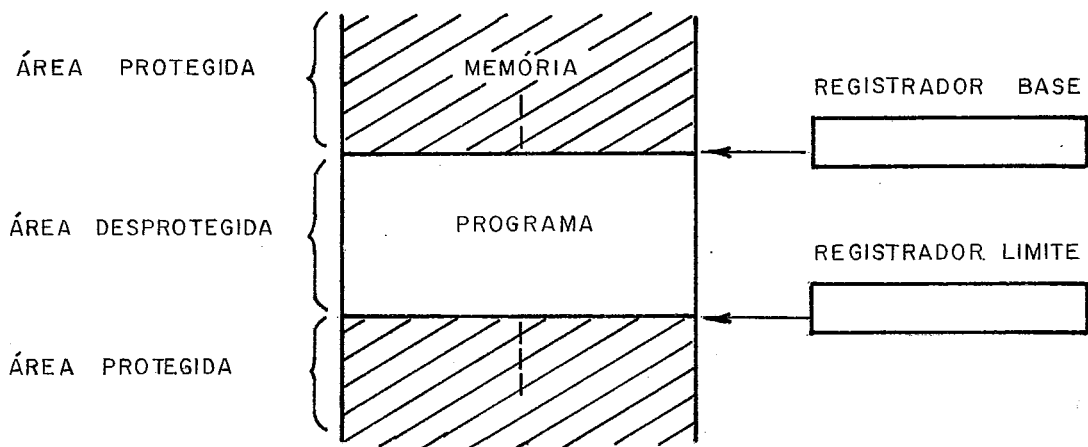
Usualmente o nível de interrupção para falha de alimentação/recuperação é o mais alto de todos e não pode ser inibido, desarmado ou mascarado;

- 2) Relógios: São classificados nos tipos de "tempo relativo" ou "tempo real". No primeiro caso é feita a contagem do número de ciclos do processador. Estes relógios geram interrupções a intervalos de tempo determinados pelo programador ou em uma determinada hora do dia, respectivamente. O uso dos relógios permite fazer a divisão do tempo em "fatias" ("time slicing") ou limitar a execução de programas, etc.

O uso do relógio de tempo relativo permite obter o tempo real indiretamente. A cada interrupção, o sistema operacional incrementará um contador de tempo de um certo valor, correspondente ao número de ciclos executados.

- 3) Proteção de Memória: Em computadores que possuem um sistema de proteção de memória, como por exemplo registradores base e limite, haverá uma área protegida que não pode ser usada pelo programa, e outra desprotegida à qual é permitida o acesso. Se um programa tenta ter acesso fora de sua área de ação, ocorrerá uma interrupção.

O programa em execução ficará com sua ação circunscrita aos endereços situados entre a base e o limite. Este tipo de erro também não é, normalmente, inibível, desarmável ou mascarável. Há alternativas para desarmá-lo em estado supervisor, em algumas máquinas.



- 4) Erros internos: Ocorrência de erros de paridade na memória ou nos registradores: erros de aritmética, tais como divisão por zero e "over-flow"/"under-flow", códigos inválidos de instruções. Estas interrupções também, normalmente são não inibíveis, desarmáveis ou mascaráveis.
- 5) Interrupção pelo Painel: Através de uma chave do painel o operador pode interromper o programa em execução.
- 6) Chamada de Supervisor: Execução desta instrução.

Alguns dispositivos de interrupção são fornecidos opcionalmente pelo fabricante, como é o caso da falha de alimentação/recuperação ou do relógio de tempo real. Outros tipos de interrupção podem não estar implementados em alguns computadores.

1.14 - Alguns Exemplos de Interrupções Internas

Os computadores HP-série 2100 usam as interrupções internas de [13]:

- Falha de Alimentação;
- Proteção de Memória;

- Paridade na Memória.

Os computadores VARIAN série 620 usam as interrupções internas de [14] :

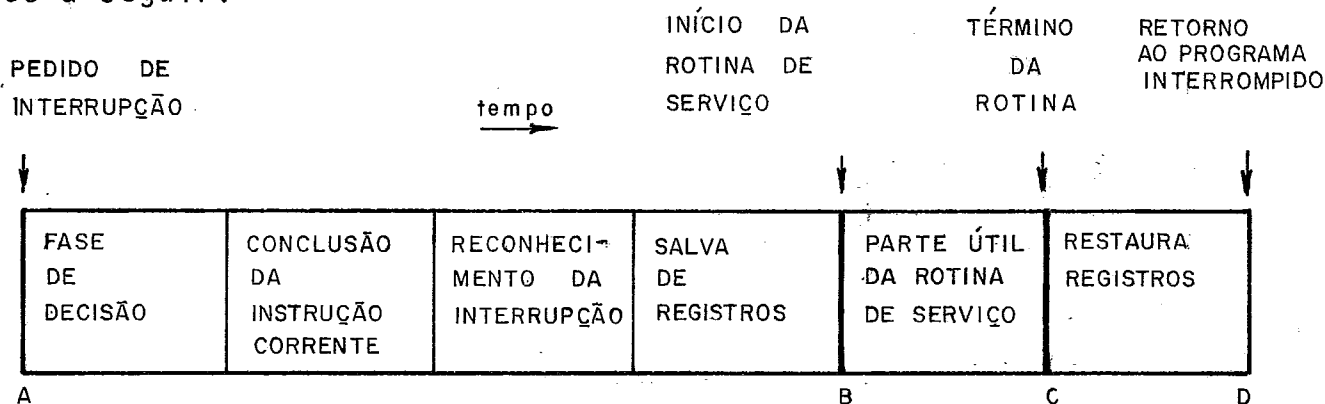
- Falha de Alimentação/Recuperação;
- Relógio de Tempo Real;
- Proteção de Memória.

Todavia, no VARIAN, são todas opcionais, dependendo da existência de dispositivos especiais.

1.15 - O Ciclo das Interrupções

Chama-se de "ciclo de interrupção" a sequência, já apresentada anteriormente, que vai desde o pedido de interrupção até o retorno ao programa interrompido.

Representa-se graficamente a sequência como no gráfico a seguir:



A parte do ciclo de interrupção que não inclui instruções úteis, ou seja, que não produza diretamente a ação pedida pela interrupção, no caso AB + CD, é a fase do "não faz nada" segundo Beizer [2] ou também chamada de "overhead", segundo Kocher e Van Gelder [10,15]. É o preço que a interrupção paga para ser atendida. Há, todavia, uma outra medida de tempo importante [8,10,15] que é o "tempo de resposta" ou "latência" e é a demora até a execução da primeira instrução útil. O tempo útil de execução está re-

presentado por BC.

Ambas medidas, segundo Kocher [10] são importantes para se avaliar se o sistema se aproxima do ótimo ou não. Obviamente com um "tempo de resposta" vagaroso e um "overhead" muito alto, o número de dispositivos que podem ser atendidos diminui, bem como o volume do processamento do sistema.

Segundo os mesmos autores [10,15] um tempo de resposta muito lento pode levar o sistema a um estado de "saturação". Ocorre a "saturação" quando o número de interrupções pendentes cresce sempre. Isto significa que o número de interrupções ocorre numa velocidade tal que o sistema não consegue atendê-las. Algumas interrupções nunca serão atendidas. O problema da saturação é, portanto, de vital interesse para sistemas em tempo real.

Define-se o tempo de "latência" ou "tempo de resposta", segundo Vachon [8], como sendo o tempo desde que o dispositivo solicitou um serviço ao processador até o momento em que de fato é atendido, ou seja, no caso de uma transferência de dado, esta foi concluída. Segundo Vachon [8] um sistema bem projetado deve estabelecer um compromisso entre o tempo em que um dado está disponível e a "latência" máxima. Se o máximo tempo de "latência" ultrapassa o tempo de disponibilidade de um dado, ele será perdido, por "over-run".

Com Milstead [16], diz-se que a latência é devida a funções que podem ser implementadas em "hardware" e/ou "software" e que incluem:

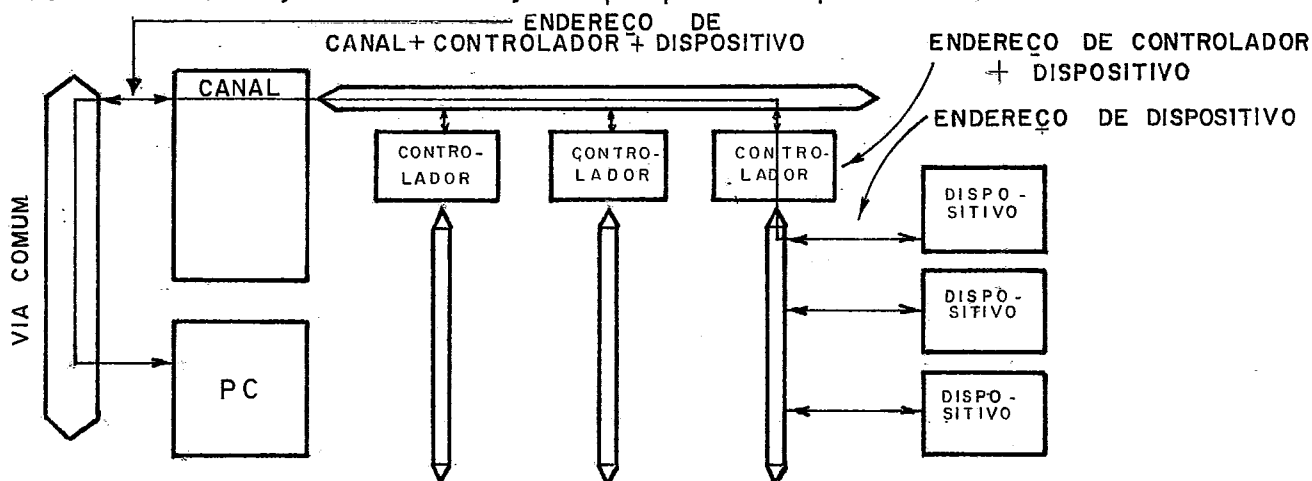
- a) Terminação normal da instrução que está sendo executada pelo processador central;
- b) Inibir todos os pedidos pendentes no sistema de interrupções;
- c) Determinar a fonte da interrupção;
- d) Salvar o valor corrente de todos os registradores;

- e) Permitir que parte das interrupções possam ocorrer , desinibindo-as, conforme o sistema de níveis de prioridades;
- f) Transferir o controle para uma rotina de serviço do dispositivo que está interrompendo.

Nem todos os computadores tem todos ítems da sequência enumerada acima, mas as que existirem consumirão tempo, que de terminará uma maior ou menor "latência". O tempo de "latência" é importante para o sistema inteiro, devido a sua aplicação, por exemplo em tempo real, ou ser importante apenas para um ou outro dispositivo dentro do sistema.

1.16 - Identificação de Dispositivos nas Interrupções Externas

A estrutura do sistema de interrupções seria muito simples se houvesse um único dispositivo. Todavia tal não acontece e o número de interrupções provenientes dos dispositivos é muito grande. A concorrência de todos eles pelo uso do processador central, num sistema de vários níveis leva ao problema de determinar prioridades. Cada vez que um pedido de interrupção é aceito , o dispositivo reconhecido deve se identificar através do seu endereço. Este endereço é composto, pois temos vários dispositivos conectados a um controlador e vários controladores conectados a um canal. Portanto, o endereço inclui o endereço do bloco (canal), do controlador e, finalmente, do próprio dispositivo.



Junto com o endereço o dispositivo envia, também, um código que identifica a causa da interrupção, para permitir a rotina de serviço tomar uma determinada ação entre várias alternativas. Uma maneira semelhante, porém muito mais cara, é ter um sinal para cada causa.

Se o código da interrupção não estiver incluído, ou o sinal da causa, a rotina de serviço terá que fazer um ou mais testes, para conhecer o estado dos dispositivos e identificar a interrupção.

Se o endereço incluir apenas o canal e controlador, a rotina de serviço terá que determinar qual dos dispositivos causou a interrupção, sendo necessário também fazer o teste do estado de cada um dos dispositivos.

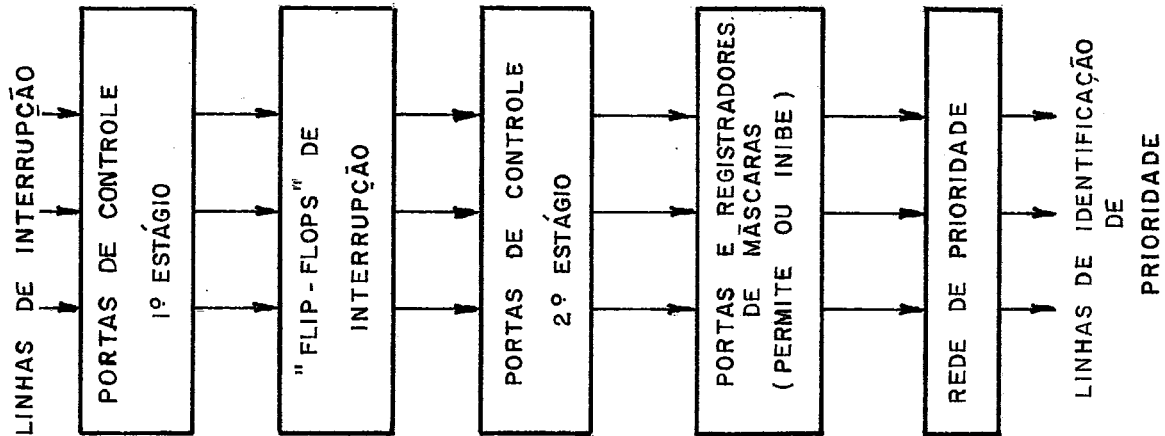
Quando o endereço for apenas do canal, os testes incluirão os controladores e dispositivos.

Portanto, quanto mais completo o endereço, menor o "overhead" para identificar o dispositivo e a causa da interrupção. Normalmente, segundo Beizer [2], a maioria dos computadores inclui no endereço o número do canal, controlador e dispositivo, deixando para ser identificada por "software" apenas a causa da interrupção, através do teste do registrador de estado daquele dispositivo.

1.17 - Estrutura do Sistema de Interrupção

Para atender ao problema da prioridade entre interrupções os circuitos lógicos são estruturados de maneira a sempre resolver os conflitos. A estrutura de atendimento de interrupções, com prioridade, tem o aspecto geral mostrado no gráfico abaixo, segundo Beizer [2].

Os sinais, nas linhas de interrupção, quase sempre tem a forma de um pulso ao invés de uma senoide.



Estes sinais ou pulsos são amostrados pelo 1º estágio de portas e os que estiverem ativos num determinado instante passarão e ligarão os seus "flip-flops".

Um pulso dura um pouco mais que a execução de uma instrução. Ao fim de cada instrução as portas são abertas uma vez, para atualizar os "flip-flops". Depois e cada vez que os "flip-flops" são atualizados as portas do 2º estágio são abertas.

Assim, os sinais que são enviados para a rede de prioridades, através da máscara, permanecerão estáveis durante a fase de decisão.

O conteúdo dos "flip-flops" passam, abertas as portas do 2º estágio, através da máscara, para a rede de prioridades.

A máscara é um conjunto de portas que permite ou inibe cada um dos sinais de interrupção que serão apresentados à rede. São os sinais permitidos serão conhecidos pela rede. A máscara é controlada pelo programador, dando-lhe a possibilidade de alterar o atendimento das interrupções. Isto equivale a alterar, por programa, o sistema de prioridades, dinamicamente.

A rede de prioridades resolve o conflito entre os sinais que lhe foram apresentados e deixa passar o de maior priorida

de para o processador. Quando o processador está em condições de atender a interrupção, ele envia o sinal de reconhecimento e, ao mesmo tempo, limpa o "flip-flop" que causou a interrupção.

Esta mesma estrutura será usada em blocos do tipo canal e até mesmo em controladores com vários dispositivos.

Num sistema dotado de tais estruturas o sinal enviado ao canal por um controlador corresponderá sempre ao do dispositivo de maior prioridade. Analogamente o sinal enviado por um canal ao processador central corresponderá sempre ao do controlador de maior prioridade. Já no processador central será atendido o sinal do canal de maior prioridade. Este exemplo foi o de um sistema com 3 níveis ou estágios de estruturas de prioridades: nos controladores, nos canais e no processador central. Desta forma o processador estará atendendo sempre ao canal, controlador e dispositivo de maior prioridades.

1.18 - Limitação das Interrupções

Segundo Kocher [10] em um minicomputador as interrupções são limitadas de várias maneiras que podem ser grupadas em:

- 1) Inibir todos os pedidos de interrupção;
- 2) Inibir algumas interrupções através do uso de um registrador de "máscara". Nesta máscara, cada "bit" corresponde a um determinado nível de interrupção ou a um determinado dispositivo. Quando, por exemplo o "bit" é 1(um) as interrupções serão permitidas e quando é 0(zero) serão inibidas.
- 3) O processador ativa ou desativa "flip-flops" em cada controlador que "desarmarão" ou "inibirão" os pedidos de interrupção deste controlador. Diz-se que a interrupção está "inibida" quando o dispositivo deixa seu pedido pendente em um "flag", embora ela não seja enviada. Estará "desarmada" quando não há pos-

sibilidade de ativar o "flag" e de enviar o pedido. Quando a interrupção é "permitida", se há um "flag" ativado, o pedido será enviado ao processador.

Naturalmente "inibir/permitir" e "desarmar/armar" são possíveis através de instruções de E/S e seu uso fica a cargo do programador.

1.19 - Instruções associadas ao Sistema de Interrupções

Segundo Beizer [2] as instruções associadas as estruturas de interrupção são classificadas em alguns tipos básicos:

- 1) Inibe/Permite: são instruções utilizadas para inibir ou permitir todas as interrupções. Excluem-se, naturalmente, aquelas que já se disse anteriormente, nunca são "inibidas", "desarmadas" ou "mascaradas", como por exemplo falha de alimentação. Esta instrução atua sobre o 2º estágio de portas de controle da estrutura de interrupções. O efeito de uma instrução do tipo "inibe" é não permitir que as portas se abram. Os pedidos chegam, através do 1º estágio de portas, até os "flip-flops". Porém não conseguem ultrapassar o 2º estágio de portas.

Para o processador central tudo ocorre como se não houvesse nenhum pedido e os programas continuam correndo sem interrupções. Assim que é dada uma instrução "permite" o processador passa novamente a atender interrupções, anulando o efeito da instrução "inibe" anterior.

A instrução "inibe" normalmente é a primeira em uma rotina de serviço de interrupção, especialmente em computadores que não possuem um comando do tipo "seletivo" ou por "máscara". Antes de deixar a rotina, ou a partir de um ponto onde já se pode "permitir" outras interrupções, deve ser dado o comando "permite".

A necessidade destas instruções é óbvia quando se considera que ocorrendo um novo pedido de interrupção, antes que os registradores do primeiro programa (interrompido) tenham sido salvos, seu conteúdo será perdido. A execução da instrução "inibe" garante a execução, sem interrupção, da rotina de serviço até o ponto em que for desejável e/ou necessário, quando se rã executada a instrução "permite". Ambas as instruções, normalmente, são do tipo "não-interrompível", isto é, garantem que a instrução seguinte também se rã executada.

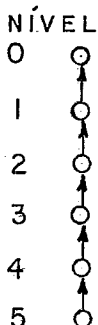
- 2) **Altere Registro de Máscara:** O registrador de Máscara de Interrupção é usado para "inibir/permitir" interrupções específicas. Normalmente a interrupção será inibida se o "bit" cuja posição lhe corresponde não for 1(um), no registro de máscara. O uso da máscara substitui praticamente todas as demais instruções para "inibir/permitir" interrupções, devido a sua grande flexibilidade. Limpar o registrador corresponde a instrução "inibe", para todos os níveis e enchendo-o de 1s(uns) tem-se a correspondência com a instrução "permite". A grande vantagem de usar um registrador é que pode-se operar normalmente com ele, tendo disponível para isto todo o conjunto de instruções aplicáveis a registradores. Pode-se, portanto, "ligar" e "desligar" certos "bits", limpar, carregar, mover ou armazenar, fazer operações lógicas e aritméticas e executar desvios com base no seu conteúdo.
- 3) **Inibe/Permite Seletivo:** São Instruções que "inibem/permitem" somente os níveis de interrupção designados na própria instrução. Variações incluem não só a seleção de um grupo como também a daqueles de menor prioridade. A flexibilidade deste tipo de instrução a coloca numa escala entre as instruções do tipo "inibe/permite" e "altere registro de máscara".

- 4) Seleção Prioridade Seguinte: Esta instrução só é usada para terminar uma rotina de serviço, iniciada em virtude de uma interrupção. Conforme o computador, ao se entrar numa rotina de serviço, alguns níveis de interrupção já estavam previamente inibidos, ou pelo uso de uma instrução "inibe seletiva" ou pelo uso de uma "máscara". Esta instrução permite também "seletivamente" a ocorrência de interrupções assim que se retorna ao programa interrompido. Cumulativamente, funciona como um "retorno de interrupção", restaurando os registradores do programa interrompido.

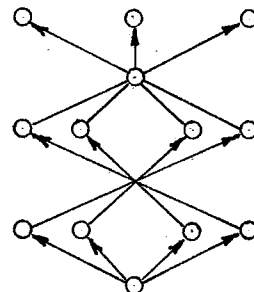
1.20 - Ordenação das Estruturas de Prioridade

Segundo Beizer [2] até agora as estruturas de prioridade de interrupção descritas obedecem a uma ordenação estrita ou quase estrita. Isto significa que entre duas quaisquer interrupções existe uma relação precedência, pois diz-se que uma é de prioridade maior, menor que ou igual a outra. Reduz-se estas relações a grafos em que cada uma das linhas é representada por um nó e a aresta indica a existência de uma relação direta de prioridade, apontando da menor para a maior.

Os sistemas descritos até agora são representados pelos grafos que se seguem, dependendo de se ter um nível de prioridade para cada linha ou se várias linhas tem o mesmo nível de prioridade:

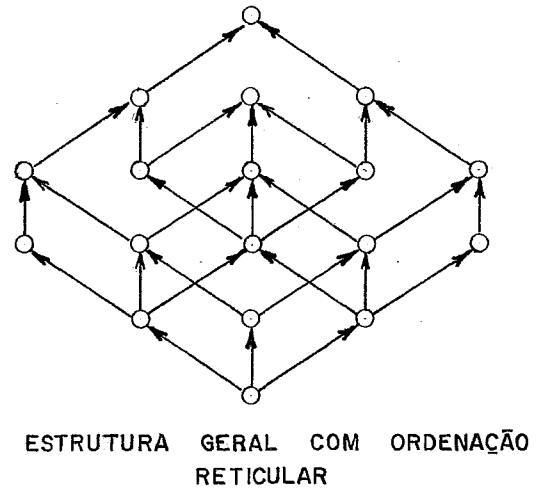
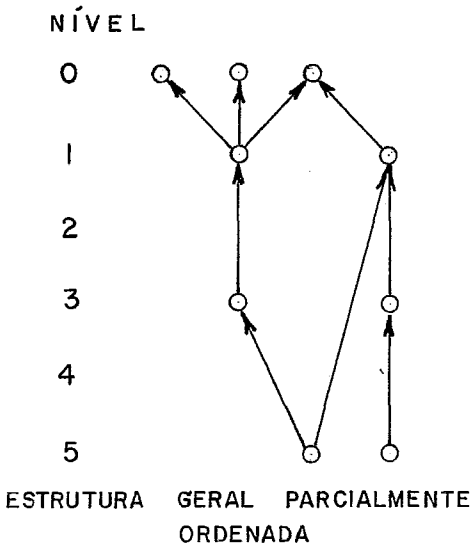


GRAFO DE UMA LINHA POR
NÍVEL DE ORDENAÇÃO ESTRITA



GRAFO DE VÁRIAS LINHAS
POR NÍVEL ORDENAÇÃO
QUASE-ESTRITA

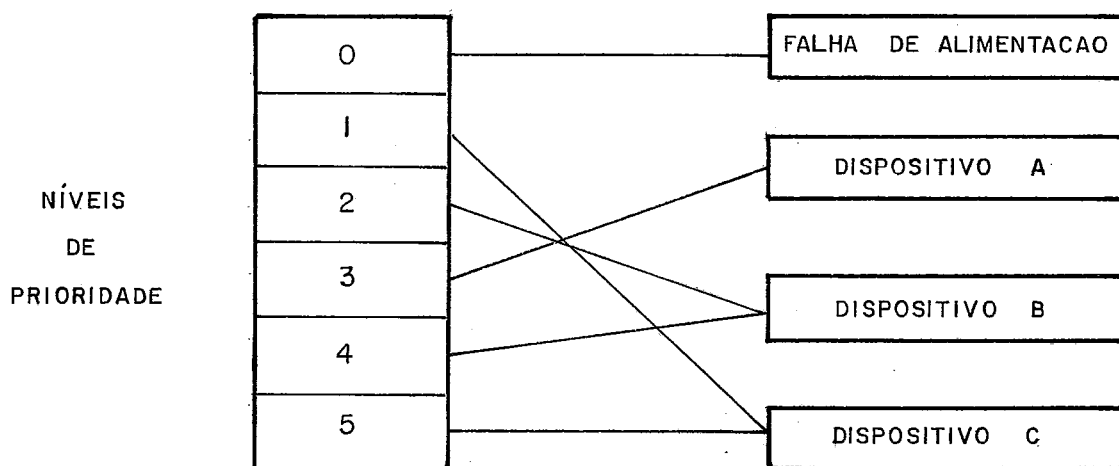
Estas estruturas, todavia, são absolutamente gerais como as dos grafos abaixo:



Qualquer estrutura é descrita de forma matricial com dimensões $u \times u$, sendo u o número de linhas ou nós. Cada elemento a_{ij} da matriz terá um dos valores 0 ou 1, ou seja, a matriz é binária. O valor é 1 se existe uma relação de prioridade tal que o nó i é diretamente inferior ao nó j . Será 0 se não existe tal relação. A implementação de um sistema matricial permite dinamicamente reestruturar a relação de prioridades.

A flexibilidade em mudar prioridades em alguns casos é alcançada, segundo Kocher [10] atribuindo-se a um mesmo dispositivo prioridades diferentes. Haverá uma linha para cada um dos níveis de prioridade que se deseje atribuir ao mesmo dispositivo. "inibindo" ou "permitindo" cada um destes níveis, o dispositivo muda de prioridade. Um exemplo hipotético é mostrado abaixo.

No exemplo apresentado se os níveis 1 e 2 estiverem inibidos, o dispositivo A terá maior prioridade que B e C. Permitindo o nível 1 e inibindo o nível 5, o dispositivo A terá menor prioridade do que C e maior apenas do que B. Permitindo os níveis 1 e 2 e inibindo os níveis 4 e 5, o dispositivo A terá menor prioridade que os dispositivos B e C.



1.21 - Mudança de Contexto com o Uso da Máscara de Interrupção

Entre todos os mecanismos usados para limitar e/ou alterar prioridades um merece especial atenção: a "máscara" de interrupções. Para usar a "máscara" eficientemente é necessário ter um "registrador de máscara" ou incluir a "máscara" no "registrador de estado".

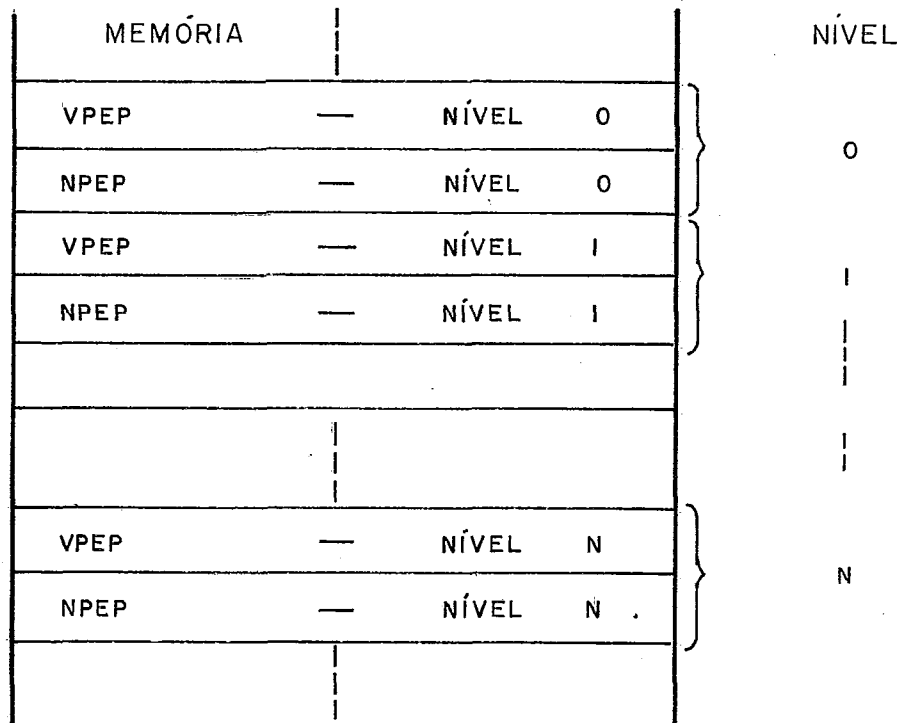
Na maioria dos computadores que a utilizam, a "máscara" está incluída no registrador de "estado". Distingue-se entre o estado do processador e o estado do programa, caso em que teríamos um "registrador de estado do processador" e um "registrador de estado do programa". Neste caso também a máscara será dividida para controlar as interrupções do sistema ou do programa ficando uma parte em cada registrador.

Ao conjunto de palavras, contendo o "contador de instruções", o "registrador de estado do programa" e o "registrador de estado do processador" chama-se de "Palavra de Estado do Programa" ("Program Status Word") ou, abreviamente PEP ("PSW").

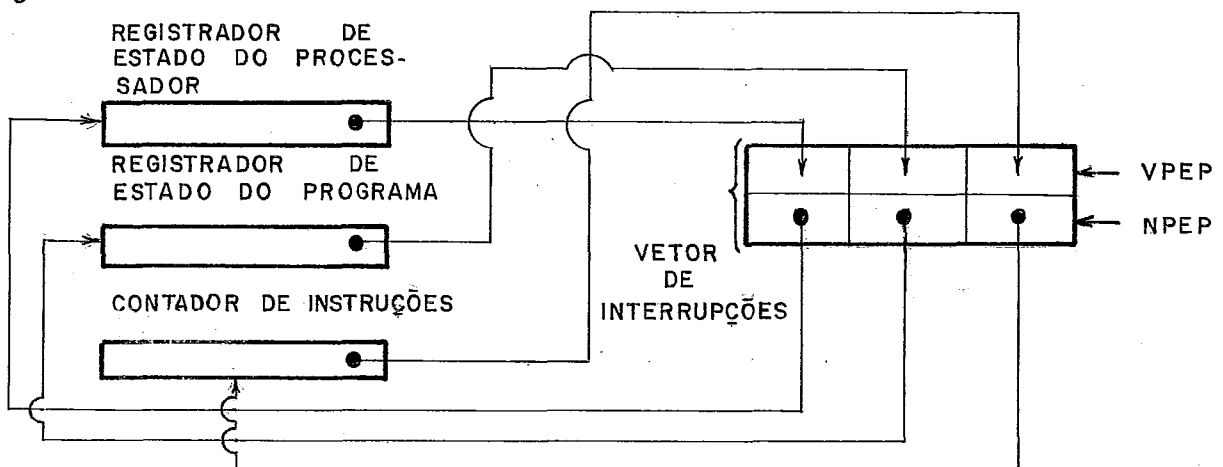
A cada nível de interrupção estão associadas 2 PEPs: uma "velha", ou VPEP ("OPSW", de "Old PSW") e outra "nova", ou NPEP

("NPSW", de "New PSW").

Em um Vetor de Interrupções (VI) a cada nível correspondem 2 posições: uma para a VPEP e outra para a NPEP. O vetor fica numa área de memória como mostrado esquematicamente abaixo:



Ao ocorrer uma interrupção de um determinado nível, o processador salvará a PEP corrente no local destinado a VPEP e colocará nos registros necessários o conteúdo da NPEP, por "hardware". Representa-se, esquematicamente, o que ocorre no seguinte gráfico:



Outros registradores de importância serão eventualmente incluídos na PEP, e ela se tornará muito mais extensa, chamando-se então de "Contexto". Normalmente os demais registradores e os de trabalho serão salvos em uma pilha ou área dinâmica de memória.

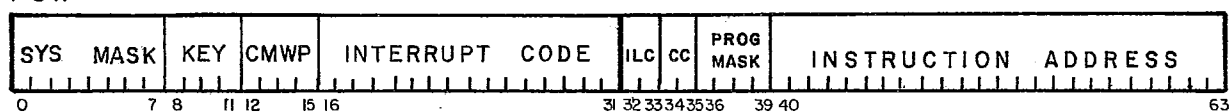
Com este mecanismo, ao se reiniciar o processamento, o "contador de instruções" estará carregado com o endereço da primeira instrução da rotina de serviço e os "estados" do programa e do processador serão os necessários à execução da rotina. Como as "máscaras" de interrupção estão incluídas nos "registradores de estado", a rotina se iniciará já com um sistema próprio e adequado de prioridades para atendimento das interrupções, obtido através dos "bits" das "máscaras", estando algumas "inibidas" e outras "permitidas".

Ao se retomar a execução do programa interrompido, os registradores serão restaurados e, especificamente, os de "estado" e "contador de instruções", a partir da VPEP do nível de interrupção do qual se está retornando.

CAPÍTULO 2DESCRIÇÃO DE ALGUNS SISTEMAS DE INTERRUPTÃO2.1 - O Sistema de Interrupções do Sistema IBM/360

Os computadores IBM/360 tem, segundo Bell e Newell, a seguinte PEP [3]:

PSW



Ocupa 2 palavras de 32 bits, num total de 64 bits, assim distribuídos:

bits

- | | | |
|---------|---------------|--|
| 0 - 7 | - SYS MASK | - Canais multiplexos, seletores de 1 a 6 e outras interrupções externas. |
| 8 - 11 | - KEY | - Chave de proteção de memória. |
| 12 - 15 | - CMWP | - Conjunto de caracteres, "machine check", estado de espera, estado de problema. |
| 16 - 31 | - INTERR.CODE | - Código de interrupção. |
| 32 - 33 | - ILC | - Código do tamanho da instrução. |
| 34 - 35 | - CC | - Código de condição. |

36 - 39 - PROG. MASK - "Over-flow", de ponto fixo, "over-flow"; decimal, "under-flow", de expoente e significância.

40 - 63 - INSTR.ADRESS - Contador de instruções.

Hã 5 níveis ou classes de interrupções, que são: entrada/saída, programa, chamada de supervisor, externa e "machine check".

Os "bits" das "máscaras" do sistema, de programa e de "machine check" são usados para inibir ou permitir interrupções.

A máscara do sistema é usada para entrada/saída e interrupções externas. A máscara de programa é usada para 4 entre as 15 interrupções de programa. A máscara de "machine check" para este tipo de interrupção. As demais interrupções não podem ser mascaradas.

As interrupções de E/S são aquelas que são pedidas por periféricos. Neste caso o endereço do canal e controlador são registrados na VPEP.

As interrupções de programa são 15 possíveis e incluem, entre outras, as seguintes: "over-flow", divisão inválida, perda de significância e "under-flow" em expoentes (num total de 8). As 7 restantes envolvem: endereço inválido, instrução privilegiada e outras análogas.

A interrupção de "chamada de supervisor" é provocada por uma instrução especial, quando o usuário quer retornar o controle ao supervisor, passando do estado de "problema" para o de "supervisor".

As interrupções externas são as provocadas pela tecla do painel, o relógio, outros processadores ou dispositivos especiais.

A interrupção de "machine check" é aquela provocada

por um erro de "hardware".

As interrupções são atendidas somente após a execução de cada instrução e na seguinte ordem de prioridade:

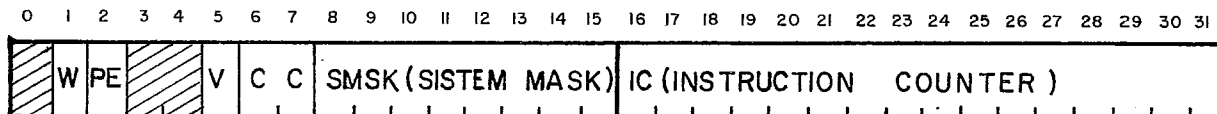
- 1 - "machine check"
- 2 - programa ou chamada de supervisor
- 3 - externas
- 4 - entrada/saída

O vetor de interrupções ocupa as primeiras posições de memória, entremeadado por outras informações.

<u>Endereço</u>	<u>Tamanho (bytes)</u>	<u>Propósito</u>
0	8	"IPL PSW"
8	8	"IPL CCW1"
16	8	"IPL CCW2"
24	8	"External old PSW"
32	8	"Sup call old PSW"
40	8	"Program old PSW"
48	8	"Machine check old PSW"
56	8	"Input/output old PSW"
64	8	"Channel status word"
72	4	"Channel address word"
76	4	"Unused"
80	4	"Timer"
84	4	"Unused"
88	8	"External new PSW"
96	8	"Supervisor call new PSW"
104	8	"Program new PSW"
112	8	"Machine check new PSW"
120	8	"Input/output new PSW"

2.2 - Sistema de Interrupções do Computador FACOM U-200

No computador FACOM U-200 [17] a PEP é constituída como representado graficamente a **seguir**:



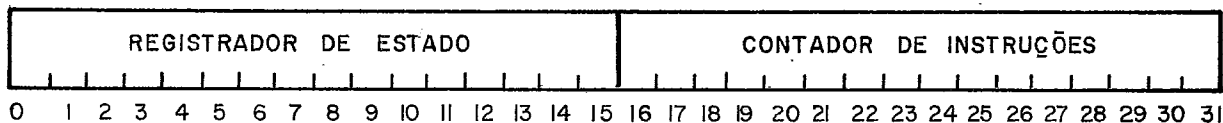
Ocupa 2 palavras de 16 "bits", ou seja, 32 "bits" ,
que tem o seguinte significado:

bit

- 1 - W - Modo de "espera".
- 2 - PE - Proteção de memória permitida
- 5 - V - Indicador de "over-flow"
- 6 - 7 - CC - Código de condição
- 8 - 15 - SMSK - Máscara de Interrupção do sistema
- 16 - 31 - IC - Contador de instruções.

A PEP deste computador corresponde à concatenação dos registradores de "estado" ("status register" ou "STR") e do "contador de instruções" ("instruction counter" ou IC = R7).

PEP



Os níveis de interrupção definidos são os seguintes:

Nível	Causa
0	"hardware check"
1	chamada de supervisor
2	não utilizada
3	não utilizada
4	externa 0
5	externa 1
6	externa 2
7	externa 3

A interrupção do tipo "hardware check" pode ser

Falha de Alimentação

Erro de Paridade na Memória

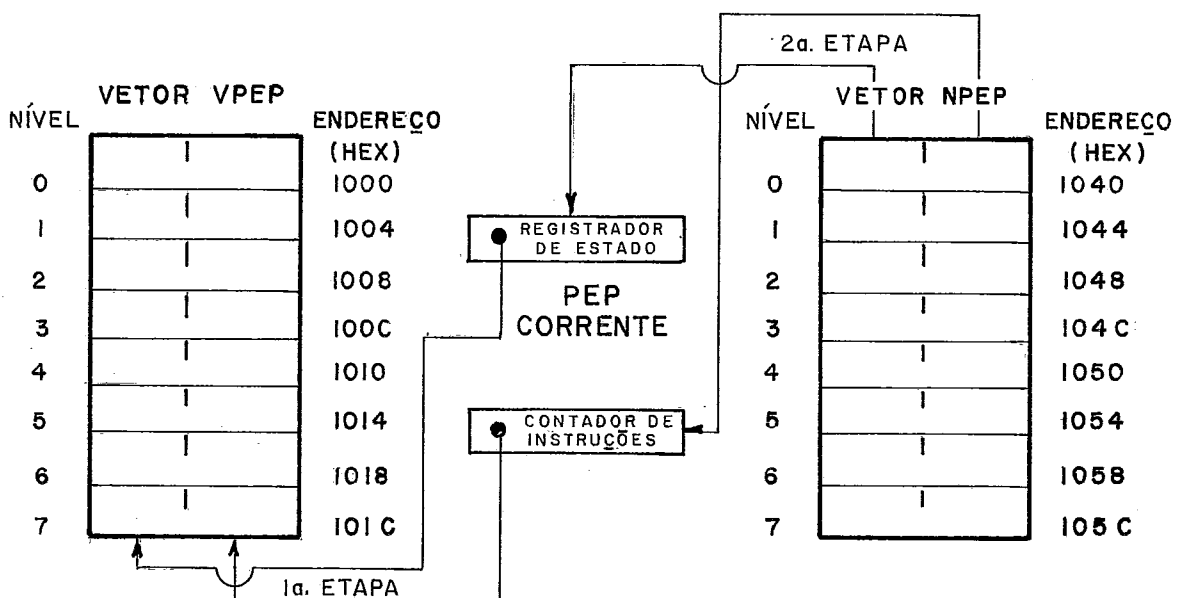
Erro em Unidade do Sistema

A interrupção de "chamada de supervisor" é causada pela execução de uma instrução específica.

As interrupções "externas" provêm de dispositivos periféricos, relógio, painel, etc.

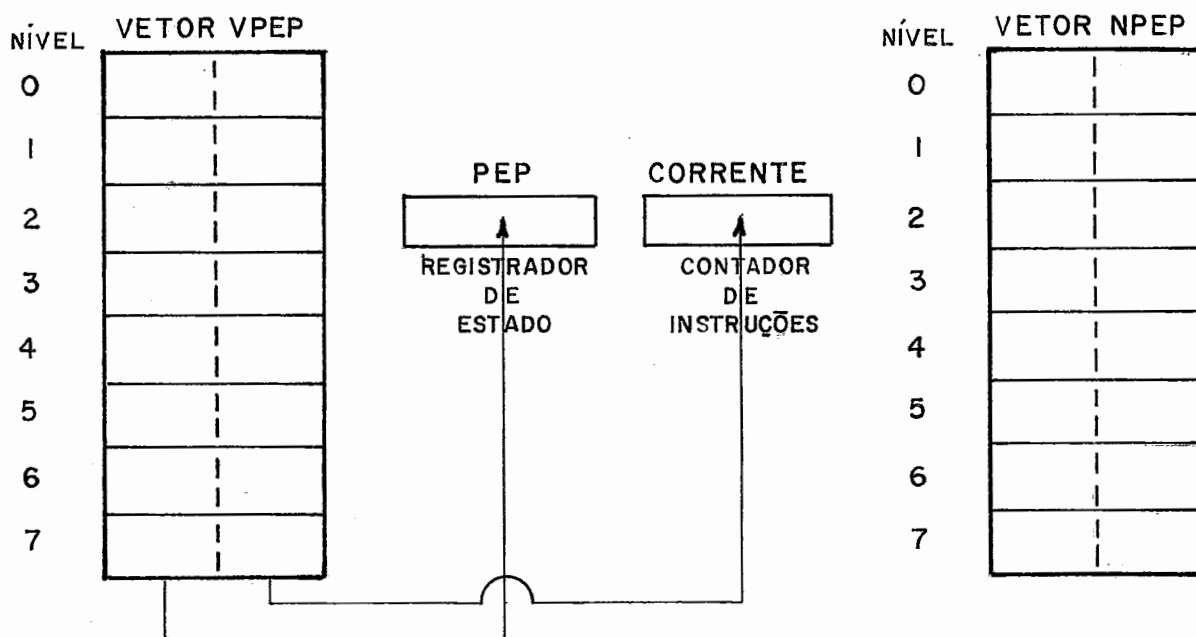
Ao ocorrer uma interrupção que não esteja mascarada e seja de maior prioridade, os registradores de "estado" e "contador de instruções" são armazenados na posição correspondente a este nível do Vetor de VPEP, se o processador está em estado de espera ou normal (1ª etapa). A seguir, a posição da NPEP correspondente ao nível é carregada, dos "bits" 0 a 15 no registrador de estado, e de 16 a 31, no contador de instruções (2ª etapa). O processador continua com os novos conteúdos destes registradores.

O diagrama abaixo mostra o que se passa:



No FACOM U-200 os vetores de interrupção estão divididos em 2 partes, ou seja: Vetor VPEP e Vetor NPEP. No espaço de memória estes vetores estão situados na área de endereços reservada para registradores de trabalho, registradores dos dispositivos de E/S e Vetores de interrupção. O Vetor VPEP se inicia no endereço 1000_{16} e o Vetor NPEP, no endereço 1040_{16} . No endereço 1070_{16} existe a PEP para carga inicial do sistema.

Ao se retornar ao programa interrompido os registradores serão restaurados, como em todos os demais computadores descritos a partir da VPEP, ficando as NPEP intactas, o que se representa graficamente abaixo:



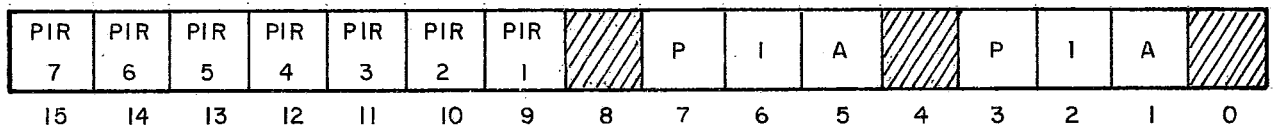
O "contador de instruções" conterá o endereço da próxima instrução a executar e o "estado" é o anterior à interrupção, restaurando-se as condições necessárias à continuação do processamento.

2.3 - O Sistema de Interrupções do Computador PDP-11/45

Um computador que merece ter descrito seu sistema de interrupção é o PDP-11/45 [11, 12]. Neste, o mecanismo de prioridades para interrupções é o mesmo de acesso à Via. A Unidade de Arbitração de Prioridades da Via, que se localiza no PC, serve para ambos os propósitos: os dados ou pedidos de interrupção só podem ser transmitidos depois que o dispositivo se tornou "mestre" da Via.

Todavia, há 2 fontes de interrupção: por "hardware" e por "software", neste último caso usando o registrador de "Pedido de Interrupção de Programa" ("Program Interrupt Request" ou "PIR"), o qual ocupa um endereço de memória 777772.

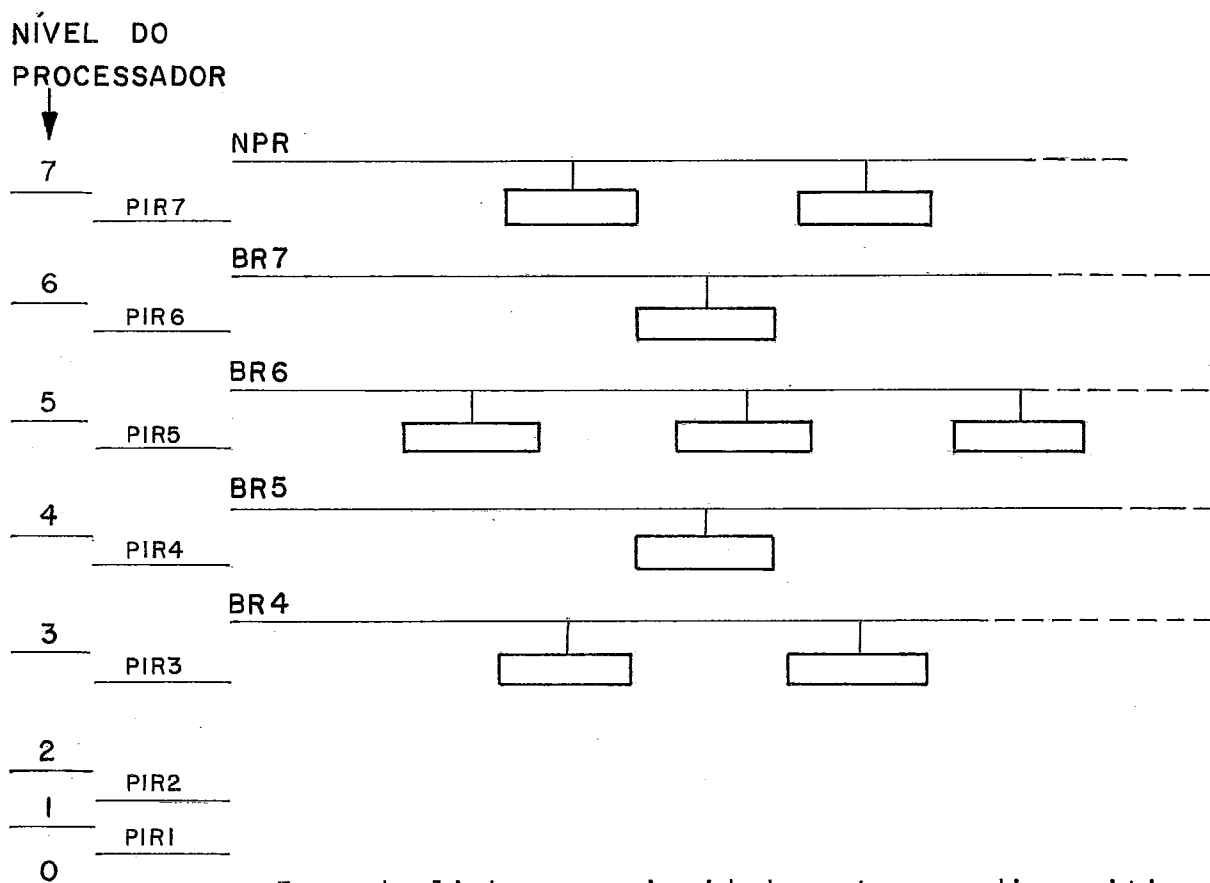
Este registrador tem o formato abaixo:



O "byte" de mais alta ordem contém "bits" que correspondem a pedidos de interrupção por "software" em níveis (PIR 7 a PIR1). Estes "bits" são ativados pelo programador e funcionam como um pedido de interrupção, da mesma forma que os dos dispositivos externos. Diferentemente do pedido de interrupção por "hardware", quando um pedido é atendido, o desvio é feito para um único endereço, que se encontra armazenado na posição 240 e é carregado no "contador de instruções" e o "estado" é transferido a partir da posição 242 para o "registrador de estado do processador". Os dois campos, denominados "PIA" ("Program Interrupt Active"), conterão, gerado pelo "hardware", o nível de prioridade que deverá ser atribuído ao processador e um índice que permitirá ter acesso a uma Tabela de Vetores de Interrupção para estes 7 níveis de prioridade de interrupção por "software". Ambos serão usados pela rotina de interrupção, para alterar o nível do processador no EP e para desviar indiretamente, usando uma entrada da Tabela.

Fica também a cargo desta rotina de interrupção, manipular uma fila de interrupções pendentes, por "software", para um mesmo nível e "limpar" o "bit" correspondente, para os pedidos atendidos.

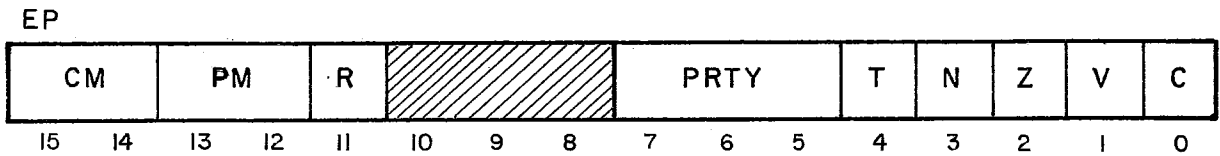
As linhas de pedido para a Via estão dispostas em 5 níveis de prioridade (NPR, BR7 a BR4) e sua relação com os níveis de interrupção por "software" são apresentados no gráfico a seguir:



Em cada linha, a prioridade entre os dispositivos é determinada pela sua posição: o de maior prioridade está mais próximo do PC.

Se há uma interrupção por "hardware" em um nível, usando a Via (linhas BR), e há uma por "software" no mesmo nível (PIR), esta última será atendida em primeiro lugar. De qualquer maneira, por "hardware" ou "software", só serão atendidos os pedidos de prioridade mais alta que a do processador central. Esta prioridade se encontra no registrador EP (Estado do Processador),

que ocupa o endereço 777776. Seu formato é apresentado a seguir:

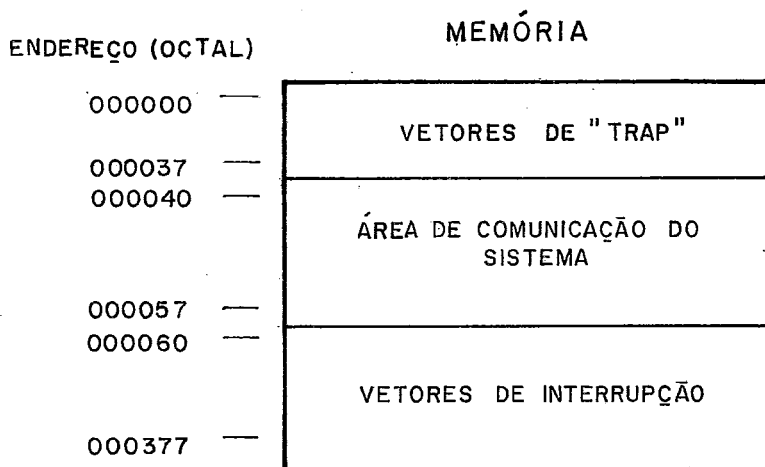


onde são os seguintes os "bits" usados:

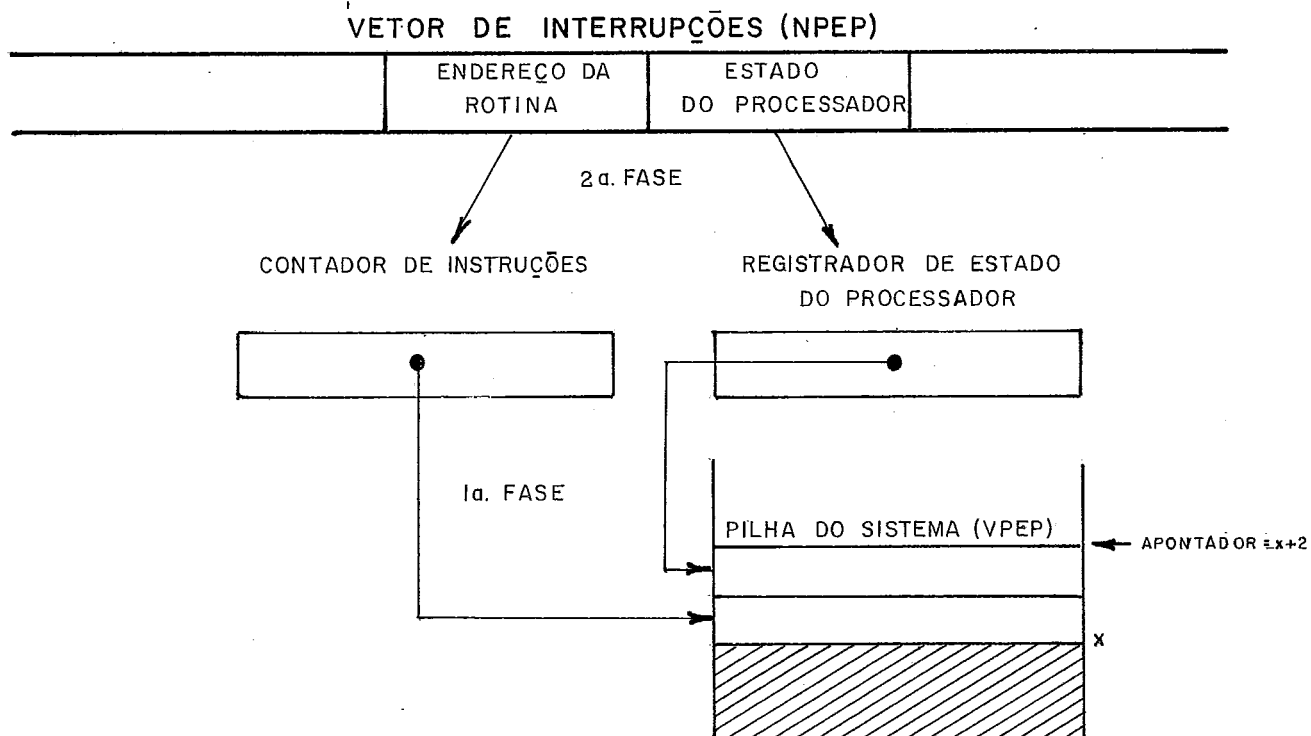
0	- C	- "Carry"
1	- V	- "Over-flow"
2	- Z	- "Zero"
3	- N	- "Negative"
4	- T	- "Trap"
5 - 7	- PRTY	- "Priority" (do processador Central)
8 - 10	- ———	- "Not used"
11	- R	- "General Register Set"
12 - 13	- PM	- "Previous Mode"
14 - 15	- CM	- "Current Mode".

O processador central pode operar, através do campo "PRTY" do registrador de estado, em 1 entre 8 níveis (de 0 a 7). Desta forma, a mudança de prioridade do PC só não mascara a linha "NPR" ("Non Processor Request"). Nesta estão conectados todos os dispositivos do tipo "data-break" ou "DMA". A diferença entre as linhas "NPR" e "BR" é que a primeira dá acesso à Via entre os ciclos de uma instrução, assim que o processador libera a Via, para uma transferência de dados, enquanto as restantes ("BR") só garantem o acesso ao final da execução de cada instrução, para pedidos de interrupção. Portanto, não são feitos pedidos de interrupção usando a linha "NPR", mas apenas as "BR".

No PDP-11/45 [11, 12] existe uma clara distinção entre interrupção e "trap". Este último tipo é também uma interrupção, porém interna. Os vetores de "trap" e interrupção são distintos e se encontram em áreas reservadas de memória, como no gráfico a seguir:



Cada rotina de interrupção tem 2 palavras reservadas no VI, que contém a NPEP, ou seja, correspondem ao novo "contador de instruções" e novo "registrador de estado do processador" e a mudança de contexto é feita em 2 fases: na primeira, a PEP corrente é colocada como VPEP, na pilha do sistema; na segunda, a NPEP é carregada nos registradores. A mudança de contexto é mostrada no gráfico abaixo:



A mudança de prioridade do processador funciona como uma "máscara". Os níveis iguais ou inferiores são inibidos, tanto

para pedidos nas linhas tipo "BR", como pedidos do tipo "PIR". A flexibilidade, todavia, é um pouco menor do que o de uma "máscara", que possibilita "inibir/permitir" qualquer nível, independentemente de prioridade.

Apesar da distinção que é feita entre "trap" e interrupção no PDP-11/45, o procedimento básico para mudança de contexto, é o mesmo. A única distinção é que para as interrupções, não há endereços predeterminados para os periféricos, como há para os "traps". Cada dispositivo, ao ganhar o controle da Via e pedir interrupção, quando é atendido pelo processador, envia ao processador o endereço da sua NPEP, que se encontra numa posição fixa dentro da área reservada para o Vetor de Interrupção. Há apenas 3 exceções, que são os endereços (octais) seguintes:

- 240 - "PIR" ("Programing Interrupt Request")
- 244 - "FPE" ("Floating Point Exceptions")
- 250 - "MMV" ("Memory Management Violations/Traps")

Já na área reservada para o Vetor de "traps", os endereços (octais) são fixos:

- 00 - Reservado;
- 04 - "Odd Address", "Fatal Stack Violation (Red)", "Warning Stack Violation (Yellow)", "time-out", "Parity Error";
- 10 - "Illegal and Reserved Instructions";
- 14 - "BPT" (Instrução "Break Point Trap") e "bit" T do registrador EP;
- 20 - "IOT" (Instrução "I/O Trap");
- 24 - "Power Fail";
- 30 - "EMT" (Instrução "Emulator Traps");
- 34 - "TRAP" (Instrução "Trap").

Todos os "traps" são atendidos com prioridade sobre as demais interrupções, tanto por "software", como por "hardware". Como "traps" neste caso incluem-se "FPE" e "MMV". A relação de prioridade entre as interrupções "PIR" e "BR" já foram descritas. O processador não mascara os "traps", excluindo-se "FPE".

Portanto, também a distinção entre interrupções por "software" usando a "PIR", e por "hardware" é puramente arbitrária já que há diversas instruções que podem gerar "traps" durante a execução de programas e também o "bit" T do EP. A própria distinção entre os Vetores de "Trap" e Interrupções também não foi totalmente obedecida ao incluir na área do Vetor de Interrupções a NPEP da "PIR", "FPE" e "MMV", que, tipicamente, são interrupções do tipo "interno", ou seja, "Traps".

Deve-se observar 2 detalhes na mudança de contexto no PDP-11/45. A primeira diz respeito à transferência da PEP corrente para registradores temporários antes de colocá-los na pilha. A segunda é o fato de que a pilha onde será salva a PEP corrente como VPEP, é apontada pelo registrador apontador da pilha correspondente ao "modo" indicado pelo novo EP, nos "bits" 15 e 14, enquanto o "modo" velho é colocado nos "bits" 13 e 12.

A "latência" por "hardware" nas linhas "BR" é de 4,95 μ s com memória de núcleos, 2,85 μ s com "MOS" e 2,25 μ s com bipolar. Na linha "NPR" é de 3,5 μ s no pior caso.

2.4 - O Sistema de Interrupções dos Computadores NOVA

Os computadores Nova, Nova 800, Nova 1.200 e Supernova [18] são caracterizados pelo uso do mesmo conjunto de instruções. Sua arquitetura é baseada na existência de 2 Vias (Memória e Entrada/Saída). Sua palavra é de 16 bits numerados de 0 a 15, da esquerda para a direita e o bloco de memória pode ter até 32K palavras, em módulos de 1, 2, 4 ou 8K. O ciclo de memória de ferrite é 800 ns. Algumas poucas posições de memória são reservadas para funções especiais:

<u>Endereço</u> (octal)	<u>Função</u>
0 - 1	Locais para Interrupção de Programa
20 - 27	Local de auto-incremento
30 - 37	Local de auto-decremento

Hã 4 acumuladores dos quais 2 servem como indexadores e todas as operações aritméticas e lógicas são executadas em um deles. As instruções são todas de 1 palavra.

Os dispositivos mais lentos são conectados diretamente na Via. Os rápidos todavia são conectados em um Canal Multiplexor.

Existe a opção nos modelos Supernova e Nova 800 para canais que operam em 2 velocidades.

São as seguintes as velocidades em palavras/segundo para algumas operações, nos diversos modelos:

<u>Função</u>	<u>Série 800</u>		<u>Série</u>	<u>Supernova</u>		
	<u>Alta</u>	<u>Normal</u>	<u>1.200</u>	<u>Alta</u>	<u>Normal</u>	<u>Nova</u>
.Entrada	1.250.000	500.000	833.333	1.250.000	434.700	285.500
.Saída	1.000.000	500.000	555.555	1.000.000	357.100	227.500
.Incremento na memória	833.333	454.545	416.666	833.333	357.100	227.500
.Adição na memória				833.333	357.100	187.500

A alta velocidade no canal é obtida pela interrupção da execução de uma instrução em pontos determinados. Em velocidade normal a transferência só é efetuada ao final da execução de uma instrução completa. Os canais são normalmente usados para discos e fitas magnéticas e outros dispositivos que requeiram uma atenção especial, pois o canal é capaz de transferir informações de vários dispositivos simultaneamente.

O programador não atua sobre o canal e sim sobre os dispositivos. Quando um dispositivo precisa transferir dados, ele solicita serviço ao canal, de acordo com a sua prioridade.

No máximo pode-se ter 62 dispositivos externos.

A limitação de interrupções é feita da seguinte maneira:

- a) Todas as interrupções podem ser "inibidas" ou "per-

mitidas" em nível do Processador Central, por um "flag", que é acessível pelo programador.

- b) Nos dispositivos as interrupções podem ser "inibidas" ou "permitidas" por um "flag" existente em cada dispositivo. Há, todavia, um outro "flag", que indica se o dispositivo está pedindo interrupção (pendente) e que pode ser "limpo" pelo programador.

As instruções existentes para limitar e alterar a prioridade de interrupções são do tipo:

- a) Inibe/Permite - atua sobre o "flag" do processador e atinge todas as interrupções.
- b) Inibe/permite Seletivo - usa uma "máscara", que pode estar colocada em qualquer dos 4 acumuladores e que vai afetar o "flag" de cada um dos dispositivos. A cada "bit" do acumulador correspondem 1 ou mais dispositivos. A máscara de 1s (uns) corresponde a "inibe" e a máscara de 0s, (zeros), a "permite" para todos os dispositivos.

Todas as interrupções podem ser "inibidas" e o programador executar todas as operações de entrada/saída programada.

Se as interrupções são desejáveis, elas devem ser "permitidas", restando ainda o recurso de "inibir/permitir" seletivamente com o uso de uma "máscara".

O atendimento de uma interrupção é feito por "hardware" da seguinte forma:

- a) O processador "inibe" todas interrupções usando o "flag" "Interrupt On" do processador;
- b) Salva o "contador de instruções" no endereço 0 (zero) da memória;
- c) Desvia para o endereço contido na posição de endereço 1 (um) da memória, o que corresponde a um desvio

vio indireto usando esta posição. Este endereço pode ser indireto por sua vez.

A interrupção só será atendida nas seguintes condições:

- a) Terminou a execução de uma instrução ou transferência de dados de um canal;
- b) As interrupções estão "permitidas" ("Interrupt On"=1);
- c) Nenhum dispositivo está esperando transferência de dados pelo canal. As transferências de dado tem prioridade sobre interrupções.

A interrupção que é atendida, é sempre a do dispositivo de mais alta prioridade, isto é, o mais próximo do PC na Via. Isto porque, ao ser iniciada a rotina de serviço de interrupções, ela deverá determinar qual o dispositivo que pediu para interromper o PC.

Além disso, deve salvar os conteúdos dos acumuladores e indicadores que porventura for utilizar. A identificação do dispositivo pode ser feita por instruções do tipo "IO skips" ou "Interrupt Knowledge". Esta última coloca em um dos acumuladores o código do dispositivo de maior prioridade que solicitou interrupção durante a execução da última instrução. A partir deste ponto, pode-se desviar para a rotina específica do dispositivo.

Ao retornar de uma interrupção, deve-se restaurar acumuladores e indicadores, bem como restaurar o esquema de prioridades do programa interrompido, antes de permitir novamente as interrupções (pelo uso do "flag" "Interrupt On"). Ao ativar este "flag", a instrução seguinte sempre será executada sem interrupção, para permitir a execução de um desvio, utilizando o conteúdo do endereço 0 (zero) da memória. A instrução "permite" é, portanto, do tipo "não interrompível".

Desejando-se atender a interrupções de maior prioridade durante o atendimento de uma interrupção. Neste caso, antes

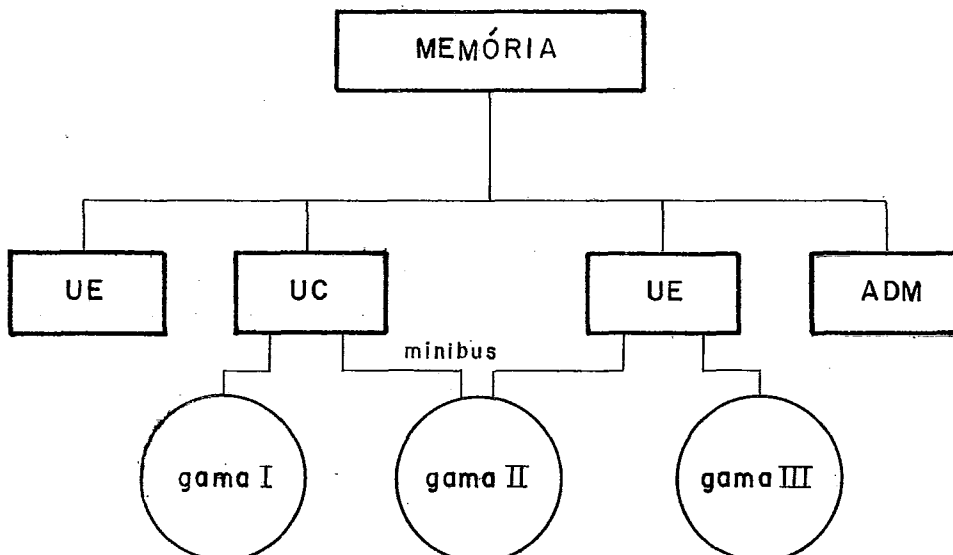
de "permitir", deve-se fazer uso da "máscara" e "inibir/permitir" seletivamente.

Opcionalmente pode-se ter um dispositivo para atender a "Falha de Alimentação/Recuperação". A sua interrupção é atendida como os demais dispositivos externos. Para ser atendida esta interrupção, todavia, as interrupções devem estar "permitidas". Não se pode inibir todavia, o pedido de interrupção em nível de dispositivo neste caso pois não há "flag" para tal.

2.5 - O Sistema de Interrupções do Computador MITRA-15

O Computador MITRA-15 da CII [19, 20] é apresentado em 2 modelos (20 e 30) e se destina a aplicações no cálculo científico, controle de processos e processamento em tempo real. A arquitetura do MITRA-15 é construída em torno de uma memória de ferrite que pode ter de 4 a 32K palavras, com 4 acessos diretos dos quais podem ser conectados de 1 a 4 Unidades de Tratamento microprogramadas (UT) ou Acesso Direto à Memória (ADM).

As UTs mudam de função com a mudança de seus micro programas e podem servir com Unidades Centrais (UC) e Unidades de transferência (Entrada/Saída) (UE) ou mesmo Unidades Especiais, adaptadas a um processamento específico.



Os periféricos se dividem em 3 gamas, numeradas de I a III, conforme sejam controlados pela UC, pela UC ou UE, ou pela UE. São conectados à UC e UEs por pequenas vias ou "minibus".

O modelo 20 é construído em torno de uma única Unidade de Tratamento, do tipo UC e portanto executando as instruções da máquina e as funções de entrada/saída dos periféricos da gama I. O modelo 30 possui, além da UC, UEs que permitem acoplar periféricos das gamas II e III e executar um repertório de instruções maior.

O modelo 20 possui um repertório de 77 instruções e o modelo 30, 86 instruções.

As palavras são de 16 bits + 1 de paridade + 1 de proteção. O bit de proteção pode ser modificado por uma instrução especial, dinamicamente. Entre os indicadores, há um de proteção de memória que, se estiver ligado, causa uma interrupção se o programa tenta ter acesso a uma palavra protegida.

Cada Unidade de Tratamento (UT) possui um bloco de registros rápidos, cinco indicadores, uma memória morta de microprogramas, um operador e de um sistema de interrupções e suspensões.

A memória rápida é de circuito integrado bipolar (MSI) em conjuntos de 8 registros de 16 bits. Pode ter de 2 a 8 conjuntos, ou seja, até um total de 64 registradores por UT, com tempo de acesso de 60 ns por palavra.

Para a UC especificamente, o primeiro bloco é o bloco 0 (zero) e contém o contexto do programa em execução. Os outros blocos são afetados pelas transferências com os periféricos.

O bloco 0 da UC, contém os registradores de contexto de programa:

P	Contador de Instruções	}	Contexto do Programa
L	Base local		
G	Base geral		
A	Acumulador		
E	Extensão de A		
X	Índice		

Os indicadores são:

- C - Relatório ou Teste de Operação
- O - Desbordamento ou Teste de Operação
- MS - Mestre/Escravo
- MA - Máscara de Interrupções
- PR - Proteção de Memória.

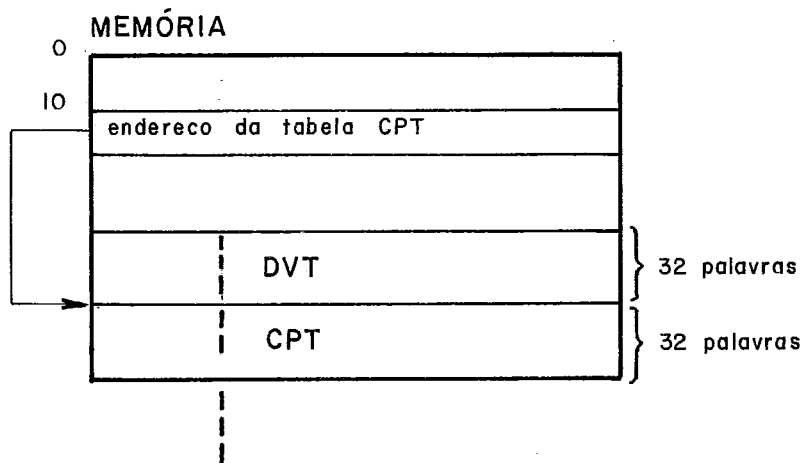
A memória de microprogramas de uma UT contém parte do sistema operacional que permite fazer entrada/saída e multiprogramação.

O MITRA-15 possui 32 níveis de interrupção, sendo 4 internas e 28 externas. As interrupções externas podem suportar até 4 dispositivos por nível o que nos dará a possibilidade de atender até 112 interrupções. Os dispositivos possuem, normalmente, 1 controlador único para cada um. Além da interrupção normal, o sistema pode atender a interrupções rápidas a suspensões e desgovernos (incidentes detetados ao fim de uma micro-instrução). A interrupção rápida consome $3\mu s$ para mudar o contexto enquanto a normal gasta $30\mu s$, ou seja, dez vezes mais.

O sistema de interrupções é ativado quando:

- a) um sinal de interrupção aparece;
- b) uma micro-instrução especial, que permite interrupção surge ("points interruptibles");
- c) as interrupções estão permitidas;
- d) o nível do programa correntemente em execução é inferior ao nível da interrupção incidente.

A cada um dos 32 níveis corresponde um endereço de memória, que contém um apontador para o contexto, que será carregado nos registradores e indicadores correspondentes antes de se iniciar a rotina de serviço. Estas 32 posições de memória constituem uma tabela (CPT), cujo endereço inicial é apontado pelo conteúdo do endereço $(A)_{16}$ ou $(10)_{10}$.



Assim que aparece um pedido de interrupção ocorre o seguinte:

- ela é registrada em um "flip-flop" (um por sinal);
- seu nível é codificado por "hardware" e comparado com o nível da tarefa em curso, cujo código se encontra num dos registros da memória rápida (registro 8);
- se a interrupção é considerada, seu nível (0 a 31) é levado por "hardware" para a micro-máquina, no momento da execução da micro-instrução de teste de interrupção;

Ocorrem, então, por micro-programa, as operações seguintes:

- leitura da posição de endereço $(10)_{10}$ da memória que contém o endereço início da tabela CPT;
- adição, ao endereço início de CPT, do nível do programa em execução, contido no registro 8 (da memória rápida), obtendo-se o endereço da posição da tabela que contém o ponteiro para o contexto velho;
- salvamento do contexto corrente nas posições do contexto velho;

- g) mudança do nível de interrupção corrente (registro 8) para o nível da interrupção nova (que vai ser atendida);
- h) nova leitura da posição de endereço $(10)_{10}$ da memória;
- i) adição do endereço inicial da tabela CPT com o novo valor do registro 8 obtendo-se o endereço da posição da tabela que aponta para o novo contexto;
- j) carga dos registradores e indicadores com o conteúdo do contexto novo.

Ao voltar a UC ao processamento, ela se encontrará executando a rotina de serviço correspondente à interrupção.

Ao terminar a execução da rotina de serviço deve-se retornar ao contexto interrompido, o que é feito por uma instrução especial "DIT" de desativação da interrupção, a qual causa:

- a) desativa a interrupção que provocou o atendimento;
- b) chamada de outra tarefa de mesmo nível, se houver um pedido de interrupção de nível inferior, a UT fica à espera de algum evento externo, no nível mais baixo, ou seja, nível \emptyset (zero), ou ainda, o nível do programa principal.

Qualquer dos níveis de interrupção externo pode, à escolha do programador, ser considerada de interrupção rápida. Esta difere da normal, pois o contexto da interrupção rápida se encontra já registrada em outro bloco diferente do bloco \emptyset . A mudança de contexto não requer, portanto, nenhuma transferência de dados. O endereço do bloco de registros da interrupção rápida é dado pelo conteúdo do registro 12. O contexto da tarefa interrompida não é perdido, pois é mantido intacto no bloco \emptyset (zero). O retorno da interrupção rápida é feito por uma instrução especial "DITR", de desativação de interrupção rápida que desativa o pedido, retorna o nível a \emptyset (zero) e restaura somente os indicadores, mas não provoca nenhuma alteração no bloco \emptyset (zero).

A Suspensão é uma parada do microprograma em execução e que inicia a execução de outro microprograma especial. Pode ser provocada pela micro-máquina ou por um periférico. Há 4 níveis de suspensão com 8 sinais por nível, o que dá um total de 32 possibilidades. O estado da micro-máquina é colocado numa pilha e restaurado ao final da suspensão.

O Desgoverno tem origem numa falha em nível de execução de uma micro-instrução. O microprograma de atendimento de desgoverno salva nos "bytes" de 4 a 9 da memória os conteúdos dos registradores L, P e indicadores, posiciona um dos "bits" da palavra de endereço 2 da memória para indicar qual o tipo de desgoverno e chama a seção 0 (zero) do supervisor. Os motivos de um desgoverno e são os seguintes:

- a) endereço de memória inexistente;
- b) proteção de memória;
- c) erro de paridade na memória;
- d) violação do modo de funcionamento (instrução privilegiada);
- e) instrução inexistente;
- f) "watch-dog";

Em todos os casos de desgoverno pode-se observar que:

- a) a instrução em curso não é concluída;
- b) a pilha da micro-máquina não é alterada;
- c) um microprograma especial provoca a chamada do supervisor.

2.6 - O Sistema de Interrupção dos Computadores ARGUS 700

Os computadores da família do ARGUS-700, da Ferranti Ltd., são os modelos T1, T2 E e S [21]. Os periféricos são os mesmos dos demais membros da família ARGUS, porém acrescentados de

mais recursos quanto à memória de massa que inclui discos desde o tipo "cassete" até unidades de cabeça fixa e grande capacidade.

O menor modelo é o T1, que possui um conjunto de instruções básico e pode ter até 24K palavras de memória. O modelo T2 tem o conjunto de instruções completo e também até 24K palavras de memória. Opcionalmente os modelos T podem ter um canal 700T/E e, neste caso, a memória fica limitada a 16K. Os modelos E e S usam, respectivamente, o conjunto completo de instruções e o conjunto expandido e podem ter, respectivamente, até 64K e 256K palavras de memória. O número de canais utilizados nos modelos E e S podem chegar a 6, dos tipos, respectivamente, 700T/E e 700T/E ou 700S. Todos os modelos são compatíveis e podemos, por expansão, passara de um modelo T1 a um modelo S. O modelo S pode ter, opcionalmente, um Processador de Palavra Dupla/Ponto Flutuante, Endereçamento no Espaço Virtual e Proteção de Memória em blocos de 256 palavras.

Além disso o acesso à memória do modelo S pode ser feita de forma simultânea e entrelaçada a 4 blocos diferentes tanto pelo processador, como canais.

Todos os modelos, a partir do T2 até S, tem proteção de memória em blocos de 4K.

As interrupções nos ARGUS-700 podem ser grupadas em 3 tipos:

Tipo 1 - Alimentação/Inicialização;

Tipo 2 - Interrupções Internas;

Tipo 3 - Interrupções Externas;

Verifica-se, todavia, que as interrupções dos Canais são consideradas Internas e que as Externas não incluem dispositivos comuns de Entrada/Saída. São 16 Sinais de interrupção, distribuídos em 6 níveis de interrupção como mostrado no quadro a seguir:

Número	Nível	Fonte	Tipo	Nível da Rotina	Observação
1	1	Alimentação/Inicialização	1	0	Não pode ser inibida <hr/> Dependendo da configuração do sistema. <hr/> Pelo painel Alimentação
0	2	Falha Alimentação/Parada	2	1	
2	2	Paridade/Violações	2		
3	2	Canal 0	2		
4	2	Canal 1	2		
5	2	Canal 2	2		
6	2	Canal 3	2		
7	2	Canal 4	2		
8	2	Canal 5	2		
9	2	"Spare"	2		
10	2	Interrupção pelo Operador	2		
11	2	Relógio	2		
12	3	Externa 0	3	2	Prioridade 1
13	4	Externa 1	3	3	Prioridade 2
14	5	Externa 2	3	4	Prioridade 3
15	6	Externa 3	3	5	Prioridade 4

As interrupções do tipo 1 ocorrem nos seguintes casos:

- a) Pressionando a chave "Inicialização" no Painel;
- b) Executando uma instrução especial "INT";
- c) Quando o computador é alimentado ao ser ligado ou realimentado após uma falha de alimentação;

As interrupções deste tipo interrompem qualquer outro tipo de interrupção e não são inibidas por qualquer meio.

As interrupções de tipo 2 pertencem a 2 classes: as que não são interrompidas e as que são. Na primeira classe se incluem a Falha de Alimentação/Parada e Paridade/Violações. Na segunda classe as interrupções provocadas pelos Canais (0 a 5), Operador, Relógio.

As condições de Violação que causam interrupção de nível 2 são as seguintes:

- a) Código de Instrução Inválido;
- b) Erro de Paridade na Memória;
- c) Endereço de Memória Inválido;
- d) Comando "TRP".

Todas as interrupções pedidas pelos dispositivos periféricos são atendidas pelos canais, que prosseguem o pedido do processador. O sistema de interrupções conhece, portanto, somente os canais e não os dispositivos.

Uma interrupção de um canal é gerada por uma dentre 3 formas:

- a) detecção de um erro no canal;
- b) microprogramas de rotinas de erro;
- c) "flags" dos dispositivos.

As interrupções dos canais são atendidas normalmente por programas curtos do modo "I" ("Interrupt Mode"). Geralmente, se mais processamento é necessário para atender a um dispositivo, desvia-se para outro programa em modo "O" ("Organiser Mode") ou "T" ("Task Mode").

As rotinas de interrupção do tipo 2 sã serão interrompidas pelas de tipo 1. Todas as demais, de tipo 2 ou 3 não serão atendidas e ficarão pendentes no registrador "IDR" ("Interrupt Demand Register"). As rotinas de tipo 2 devem ser pequenas já que não serão interrompidas pela maioria das outras interrupções. Uma interrupção tipo 2 será causada, também, por programa, usando uma instrução "INT".

As interrupções de tipo 3 são as chamadas de Externas, não no sentido de que provenham dos dispositivos de entrada/saída, mas sim com finalidade especial. Exemplos são o relógio tipo "watch dog", interrupções de outros processadores, etc. O tipo 3 permite interrupções superiores dentro do próprio tipo.

Assim, a rotina tipo 3 será interrompida por um outro pedido tipo 3 de nível superior, tipo 2 e tipo 1. Estas interrupções também serão geradas pelo uso da instrução "INT".

Algumas posições de memória são reservadas para uso do sistema de interrupções. Estas posições contêm apontadores para os contextos novo e velho. São os seguintes os endereços:

	Endereço	
	Contexto novo	Contexto velho
Tipo 1	00	--
Tipo 2	01	--
Tipo 3 - nível 1	02	02
nível 2	03	03
nível 3	04	04
nível 4	05	05
Modo "0"	06	06
Modo "PT/NT"	07	07
Modo "0"-indicadores de chamada("flags")	10	--

Nos computadores ARGUS-700, o "contexto" é um conjunto de 16 registradores rápidos.

Ao final da execução de cada instrução, o processador examina os pedidos de interrupção e atende a de mais alta prioridade, observadas as condições:

- a) se estiver em modos "0" ("Organiser"), "PT" ("Privileged Task") ou "NT" ("Normal Task"), as interrupções não estão inibidas (excluindo as de números 0, 1 e 2);
- b) estando no modo "I" ("Interrupt"), mas tipo 3, uma

prioridade mais alta ocorre.

Ao entrar na rotina de serviço, por "hardware", o contexto corrente, nos 16 registros rápidos, é salvo nas posições indicadas pelo apontador do contexto velho, do modo e nível correntes para que possa ser novamente usado quando o processador voltar a executar o programa interrompido (a menos que o programa corrente seja do tipo 1 ou tipo 2 e modo "I", caso em que o contexto corrente não é salvo). Os registros rápidos ficam disponíveis para serem usados pela rotina de interrupção.

O nível do pedido de interrupção é usado pelo processador para determinar o endereço do apontador do contexto novo. O apontador do contexto novo contém o endereço da primeira posição da área de memória, a partir da qual estão colocadas palavras contendo os valores que serão carregados nos 16 registros rápidos, para uso da rotina de interrupção. Uma exceção é o modelo ARGUS-700-S onde o contexto novo das interrupções de nível 2 e modo "0" (serve, também, para o velho), se encontram já colocados em registradores, ao invés de memória. A carga do contexto novo nos registradores e o retorno do processador à atividade provoca a execução da rotina de serviço.

Se a interrupção é do tipo 2, uma instrução "RID", ("Read Interrupt Demand Register") deve ser executada. Com seu uso será determinado o limpo e "flag" da interrupção de tipo 2 de mais alta prioridade que estiver ligado e que será usado para indicar para que rotina de interrupção será feito o desvio.

Ao concluir a rotina de interrupção, deve ser executada uma instrução "RIM" ("Release Interrupt Mode") ou "ORI" ("O-Mode Call and Release Interrupt Mode") fazendo com que o processador saia do modo "I". Neste caso, o processador reage de uma entre 3 maneiras:

- a) se há um pedido, será percebido e atendido automaticamente;
- b) retorno ao programa interrompido (modos "0", "PT" ou

"NT") após a execução de uma "RIM". "ORI" força a entrada no modo "0".

- c) mudança para o modo "0" se um "flag" (na palavra de endereço 10) indicar que uma "OMC" ("0-Mode Call") foi executada antes de executar-se uma "RIM". O "flag" pode ter sido ativado neste ou mesmo outro programa tanto pela instrução "ORI" como "OMC".

Portanto, todos os pedidos de interrupção serão atendidos antes que programas interrompidos dos modos "0", "NT" e "PT" sejam reiniciados ou antes que uma instrução "OMC" seja executada.

Existe um registrador de "máscara" para as interrupções, o "IMR" ou "Interrupt Mask Register" que é alterado por uma instrução especial, a partir da máscara colocada no Acumulador , "LIM" ou "Load Interrupt Mask".

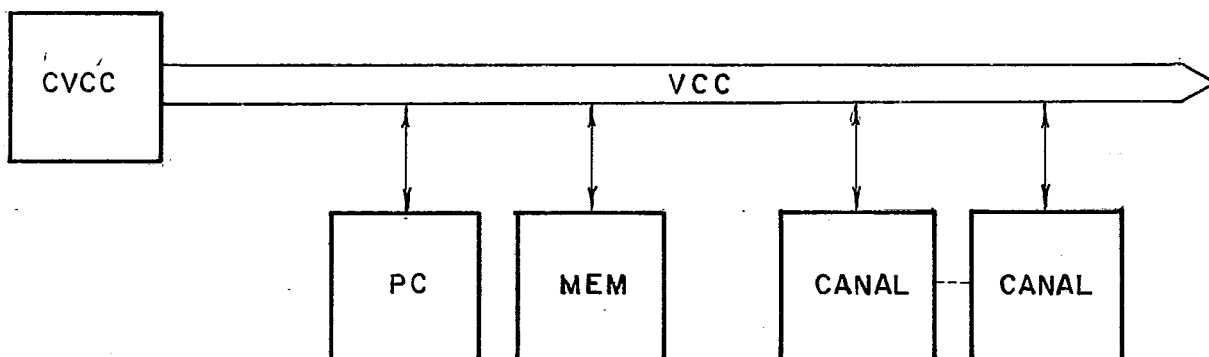
Existem, também, instruções especiais para "Inibir/Permitir" todas as interrupções, que são a "SIL", ou "Set Interrupt Lockout", e a "CIL", ou "Clear Interrupt Lockout", que atuam sobre o "flag" "Interrupt Lockout" o processador.

CAPÍTULO 3

O COMPUTADOR G-10

3.1 - Descrição Geral

O computador G-10 é um computador digital de propósito geral [22]. Sua arquitetura, com uma Via Comum, o aproxima, sob este aspecto, dos computadores FACOM U-200 e PDP-11. Sua estrutura de blocos é apresentada no gráfico abaixo:



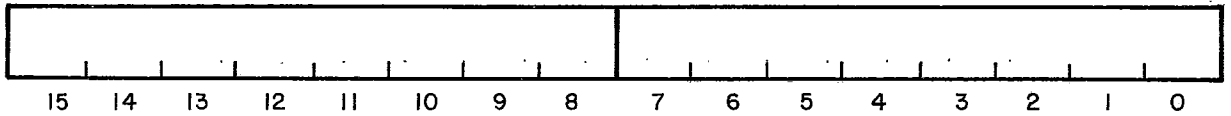
A Via Comum de Comunicação (VCC) possui um Controlador (CVCC) ao qual cabe arbitrar as prioridades entre os pedidos feitos pelos blocos que podem se tornar "ativos".

Conectados a Via Comum há um único processador central e um único bloco de memória, que terá de 4K a 32K palavras de 16 "bits" (mais 2 de paridade, um para cada "byte"). O acesso é em nível de palavra, ou seja, os endereços são de palavra.

Em cada palavra ou endereço, há 2 "bytes", denominados esquerdo (E) e direito (D).



Os "bits" de cada palavra são numerados de 0 a 15, do de mais baixa para o de mais alta ordem.



Os dispositivos são conectados aos canais através de Vias de Entrada/Saída (VES).

Os canais são de 2 tipos: Programado (chamado de Concentrador) e tipo "DMA" Seletor (chamado canal Seletor).

Qualquer dos 2 tipos de canal, concentrador (CCES) ou Seletor (CSES), terá até 16 dispositivos ligados, por meio de seus controladores.

Como o número máximo de canais, é 4 pode-se ter um máximo de 64 controladores. Considera-se aqui o número de controladores (chamados Interfaces) e não de dispositivos, pois um controlador em alguns casos suportará vários dispositivos, como por exemplo os discos.

A Via de Entrada/Saída de um canal Concentrador é chamada, abreviadamente, de VES Concentradora ou VESC, e a de um canal Seletor, VES Seletora ou VESS.

Pode-se ter 4 canais exclusivamente do tipo Seletor ou exclusivamente do tipo Concentrador, ou uma combinação dos 2 tipos, porém sempre num máximo de 4.

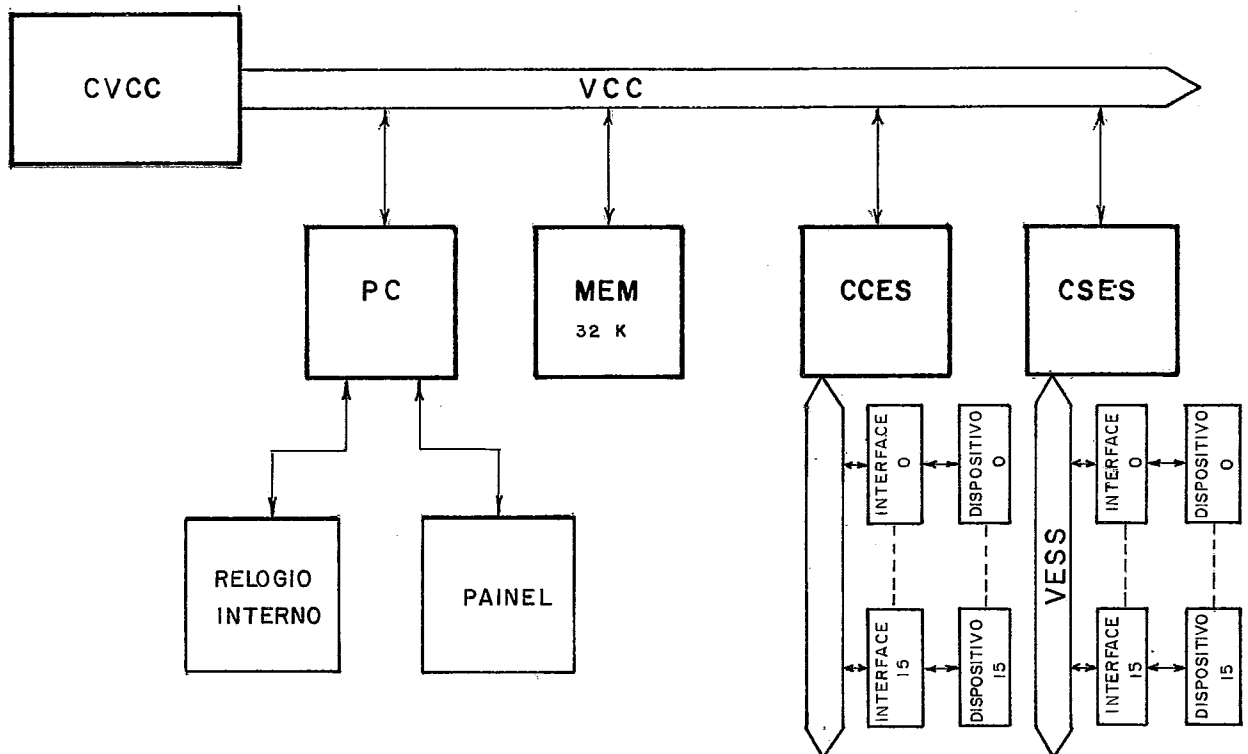
A configuração da primeira versão do G-10 prevê a existência de um canal Seletor (CSES) e um canal Concentrador (CCES), além do processador (PC) e de um bloco de memória com 32k palavras.

Possui, ainda, uma Via de Interrupções, independente da Via Comum.

Além disso, possui um Relógio de Tempo Real (denomi-

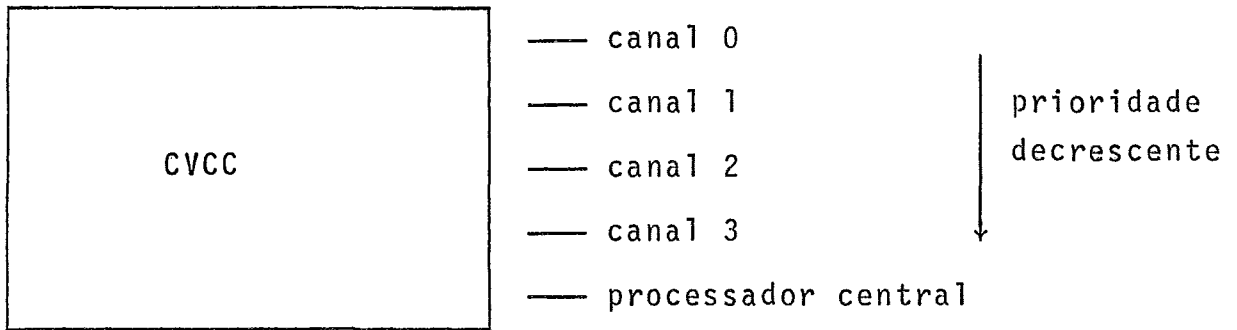
nado Relógio Interno) e um Painel de Controle (denominado Painel Frontal), ambos conectados diretamente ao PC.

A sua configuração inicial é a que é apresentada graficamente a seguir, sem figurar a Via de Interrupções:



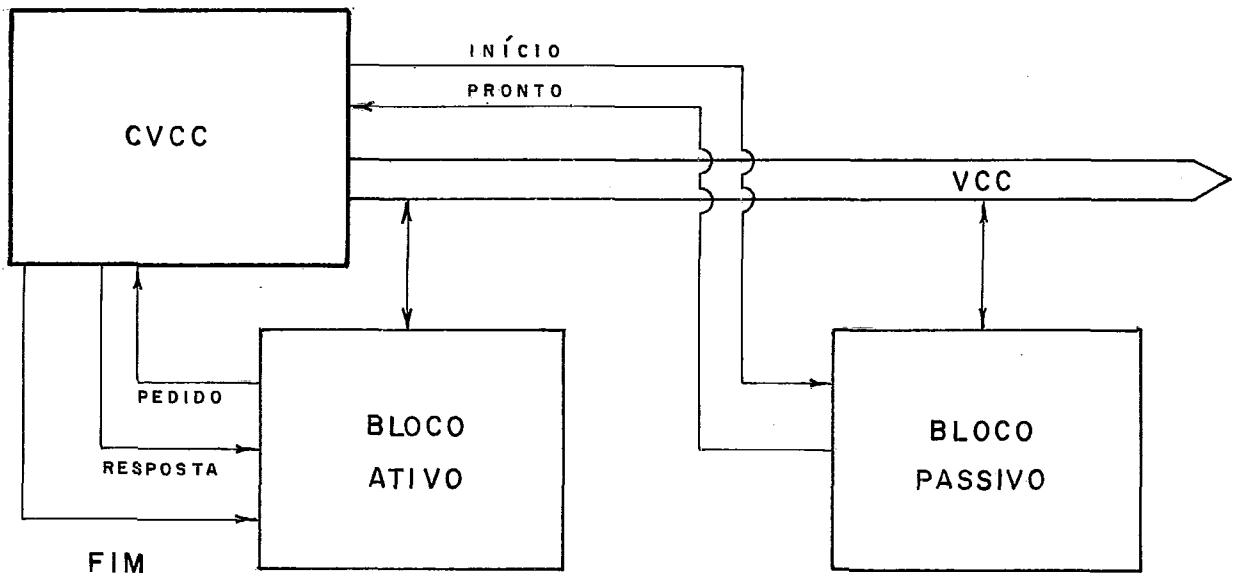
3.2 - Via Comum de Comunicação e Via de Interrupções

A Via Comum funciona de uma maneira assíncrona. Sua velocidade de transferência é de até 4 Mega-palavras/seg. A comunicação entre os blocos se faz pelo estabelecimento de uma relação do tipo "mestre/escravo" ou "ativo/passivo". Os blocos que se tornam ativos na Via são: PC e CSES. Os blocos que se tornam passivos são: memória, CSES e CCES. A prioridade é determinada por "hardware" de acordo com a ordem de entrada das linhas de pedido no CVCC e foi adotada a seguinte:



Quando algum canal se torna bloco "passivo", ele possui um endereço, por "hardware", pelo qual é identificado. Da mesma maneira, o bloco de memória também possui um endereço.

A comunicação entre dois blocos se estabelece através da VCC, através de um "diálogo" que utiliza os seguintes sinais: "pedido", "resposta", "início", "pronto" e "fim".



O diálogo se processa da seguinte maneira:

- a) Os blocos ativos enviam ao CVCC o "pedido";
- b) O CVCC resolve entre os "pedidos" qual o de maior prioridade e envia o sinal "resposta" ao bloco que

foi selecionado;

- c) O bloco que se tornou "ativo" na Via, coloca nas linhas de endereço de bloco (na VCC) o endereço do bloco "passivo", bem como os sinais de dados e endereço, se houver;
- d) O CVCC envia ao bloco "passivo", identificado pelo endereço, o sinal "início";
- e) O bloco "passivo" toma na Via os sinais de dados, endereço e direção e os transfere para os seus registradores e, se for o caso, os interpreta, por exemplo, como uma instrução de 2 palavras. Executa, então, as ações determinadas pela instrução ou, se for a memória, faz a transferência de dados;
- f) Ao final, envia o sinal "pronto" para o CVCC;
- g) O CVCC envia ao bloco "ativo" o sinal "fim". Antes que o sinal "fim" se esgote, o bloco "ativo" faz transferência de dados do bloco passivo para seus registradores, se for o caso. Termina o "ciclo" da Via. Volta-se ao início.

Os sinais "início" e "fim" são sincronizados com um relógio central que também gera pulsos com a frequência de 50ns para o PC (20 Mhz).

Quando o bloco "passivo" é a memória, pode ocorrer que esta ainda se encontre restaurando uma posição anteriormente lida (leitura destrutiva). O ciclo de memória é de 850ns e o tempo de leitura é de 300ns. Serão gastos os 550ns restantes na reestruturação da posição lida. Se isto ocorrer o CVCC é informado, através de uma linha, e passa então para outra fase de decisão, atendendo ao bloco de maior prioridade seguinte (excluindo o bloco prioritário da fase anterior). Desta maneira, um bloco "ativo" que queira estabelecer um diálogo com a memória não terá o controle da Via, se aquela ainda está "ocupada".

Tendo o "ciclo" de memória anterior sido utilizado

para escrever na memória, o diálogo terá terminado em 150ns, tempo para transferir endereço e dado. Assim, estará "ocupada" até o final "ciclo" por mais 700ns, sendo escolhida bloco "passivo". A Via passará a uma nova fase de decisão, como ocorre quando é leitura.

Um canal qualquer se torna bloco "passivo" para o processador central. A memória, todavia, se torna bloco "passivo" do PC e do CESS (Seletor).

Os sinais de VCC são grupados como se segue:

a) informação:

- . dados - 16 bits
- . endereço - 16 bits
- . endereço do bloco - 3 bits

b) controle:

- . pedido - 5 bits (1 bit para cada bloco ativo)
- . resposta - 5 bits (1 bit para cada bloco ativo)
- . direção - 1 bit
- . pronto - 5 bits (1 bit para cada bloco passivo).

c) sincronismo:

- . início - 5 bits (1 bit para cada bloco passivo)
- . fim - 5 bits (1 bit para cada bloco ativo).

Os sinais da via de Interrupções podem ser grupados:

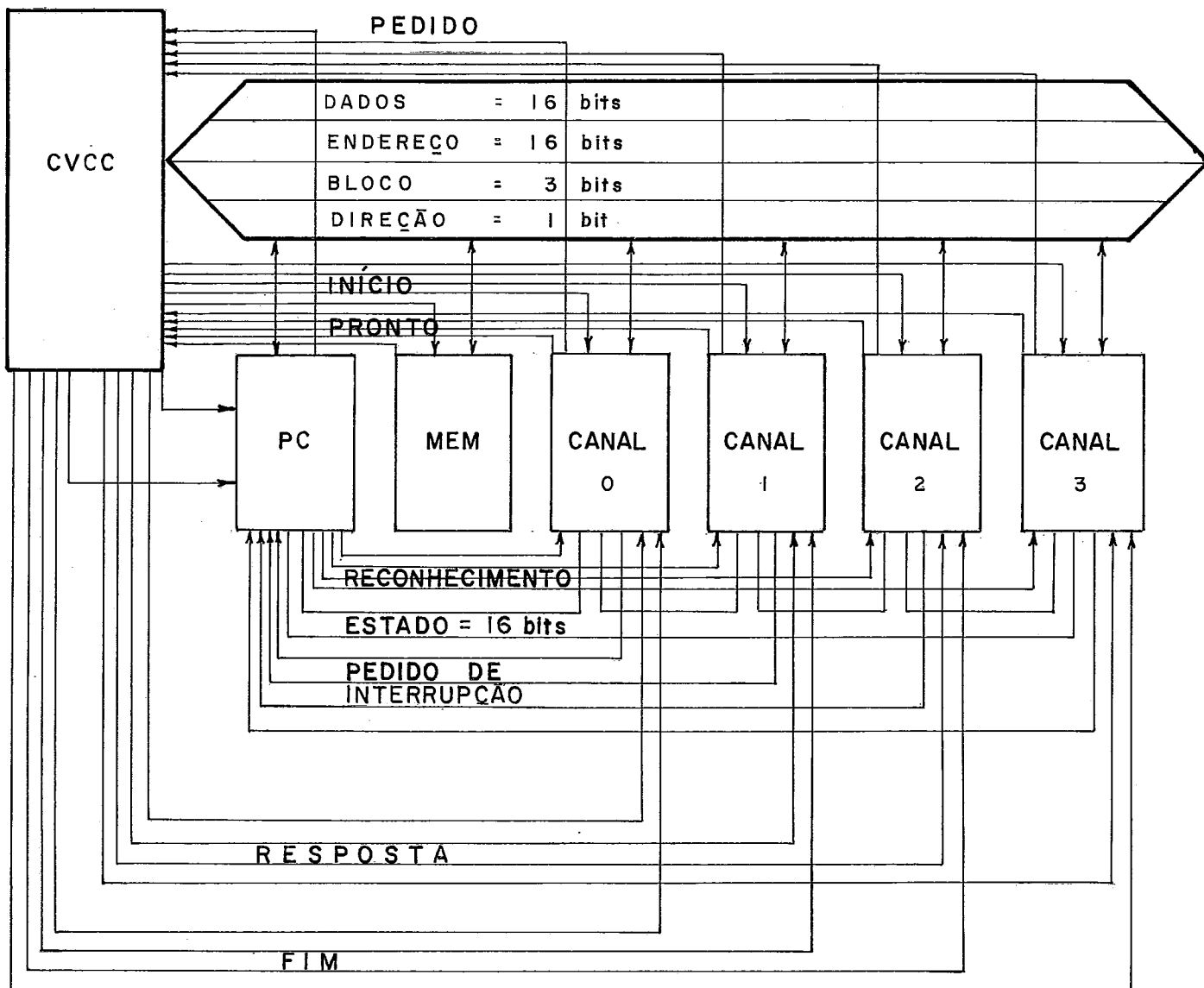
a) interrupção:

- . pedido - 4 bits (1 bit para cada canal)
- . reconhecimento - 4 bits (1 bit para cada canal)

b) estado:

- . estado - 16 bits

Tem-se, portanto, a estrutura do gráfico abaixo:



A Via de Interrupção é independente da Via Comum e é também assíncrona.

Um canal só não pode enviar um sinal de pedido de interrupção enquanto fizer parte de um par "ativo/passivo" de utilização da Via Comum. Após o sinal "início", o mecanismo de interrupção dos canais é "inibido" só voltando a ficar "permitido" após o sinal "pronto". Os canais que não estiverem dialogando na Via Comum, enviam seus "pedidos" de interrupção.

Ao final de cada instrução o processador testa as interrupções pendentes e reconhece a de maior prioridade. Após receber o sinal de "reconhecimento", o canal coloca na Via de Estado a sua "palavra de estado" (chamada de "vetor de interrupções"). O "estado" dos dispositivos só é conhecido pelo uso de uma instrução específica do tipo "testa estado do dispositivo". A "palavra de estado" do canal será conhecida pelo processador central sempre que ocorrer uma interrupção, mas poderá, também, ser conhecida através de uma instrução do tipo "testa estado do canal".

As etapas podem ser resumidas como:

- a) Os canais não "inibidos" enviam ao PC os seus "pedidos" de interrupção;
- b) No fim da execução da instrução corrente, se as interrupções não estão "inibidas", o PC "reconhece" o "pedido" de maior prioridade e envia o sinal.
- c) O canal seleccionado coloca na Via de Estado a sua "palavra de estado".

As etapas seguintes são executadas por micro-programas e serão descritas sob o título "Sistema de Interrupções".

3.3 - Processador Central

O processador central é constituído por uma Unidade de controle (UC) e um Fluxo de Dados (FD).

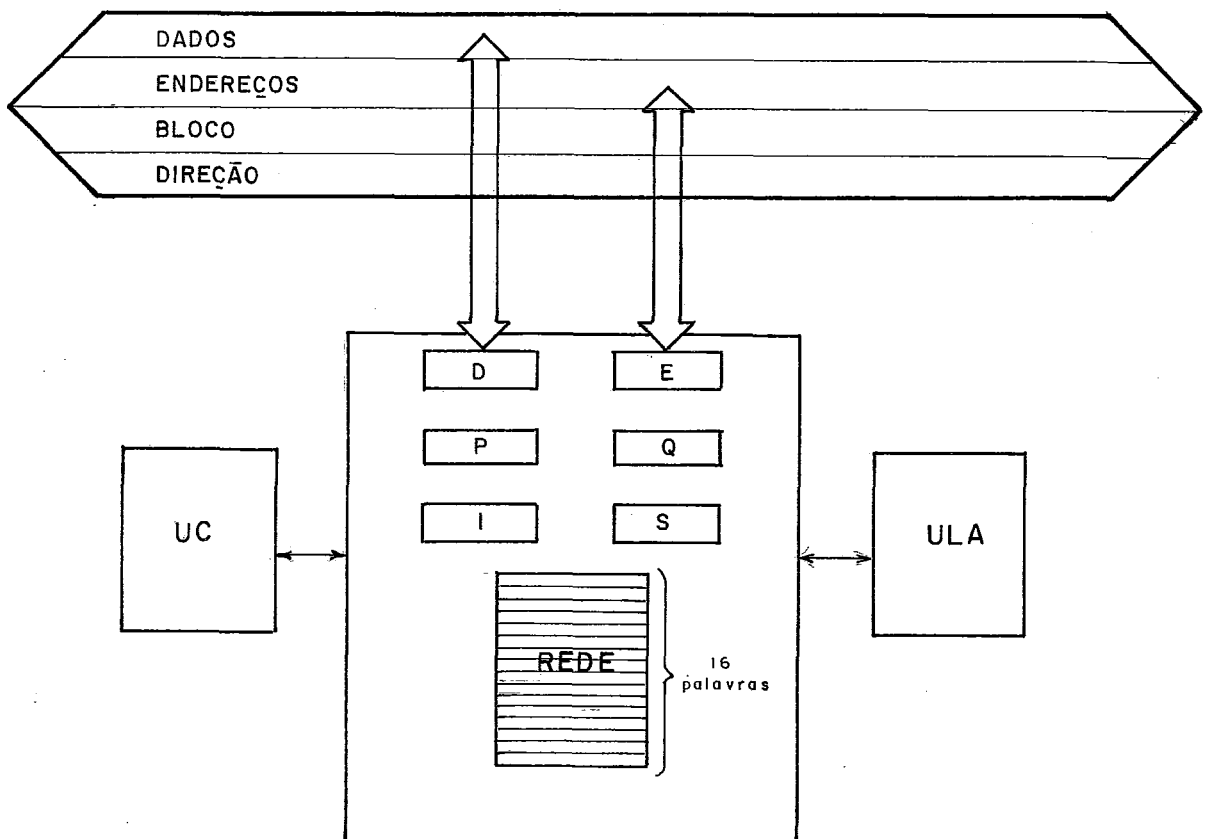
O fluxo de dados incorpora a Unidade de Lógica e Aritmética (ULA) e os registradores, todos de 16 bits. Os principais registradores são:

- a) Dados (D) - ligado às linhas de dados da VCC;
- b) Endereços (E) - ligado às linhas de endereço da VCC;
- c) Instruções (I) - armazena a instrução a executar no PC;
- d) Estado (S) - indica o estado do PC;

- e) P - utilizado pela ULA;
 f) Q - utilizado pela ULA;
 g) Rede - conjunto de 16 registradores com a finalidade:

- . rascunho: 4 registradores usados pela UC.
- . limite de dados (LD) - 1 registrador
- . limite de programa (LP) - 1 registrador
- . base de dados (BD) - 1 registrador
- . base de programa (BP) - 1 registrador
- . propósito geral (R) - 8 registradores denominados R \emptyset a R7 dos quais R \emptyset é o contador de instruções (CI) e R1 aponta para a pilha.

Esquemáticamente, representamos o PC como no gráfico abaixo:



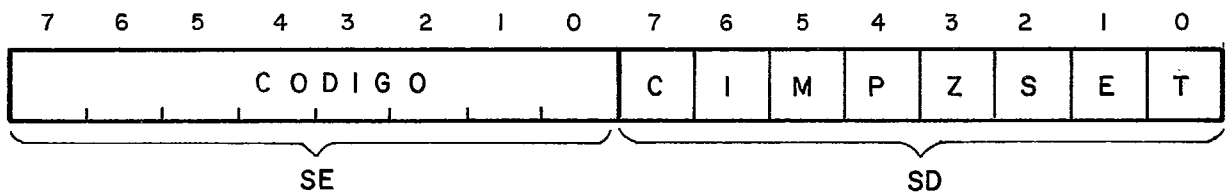
Destes registradores, sã̄o sã̄o visíveis ao programador os seguintes registradores:

- estado (S)
- limite de programa (LP)
- limite de dados (LD)
- base de dados (BD)
- base de programas (BP)
- registradores de propósito geral (R0 a R7), que incluem o contador de instruções (CI ou R0) e o apontador de pilha (R1).

A UC é microprogramada e seus microprogramas se encontram numa memória do tipo "ler somente" ("ROM").

Merece descrição especial o registrador de "estado" (S):

REGISTRADOR S



É dividido em 2 "bytes": esquerdo (SE) e direito (SD). O "byte" esquerdo (SE) é usado em 2 situações:

- a) pela instrução "chamada de supervisor" que colocará aí o código de 8 "bits" contido na instrução ("byte" direito) e que indica o motivo da interrupção.
- b) pelo sistema de interrupções que aí colocará o "byte" esquerdo da "palavra de estado" do canal que interrompeu.

O "byte" direito (SD) será atualizado, automaticamente ou por programa, e tem o seguinte significado:

bit

0 - T - Transbordamento ("overflow")

bit

- 1 - E - Extensão ("carry")
- 2 - S - Sinal
- 3 - Z - Zero
- 4 - P - Par/Impar
- 5 - M - Modo ("Supervisor" ou "Usuário")
- 6 - I - Interrupções ("Inibidas" ou "Permitidas")
- 7 - C - Condição de salto.

Os "bits" T, E, S, Z e P são atualizados por resultados de operações aritméticas, automaticamente. O "bit" M é atualizado automaticamente por interrupções, inclusive "chamada de Supervisor", passando neste caso para modo "Supervisor". O "bit" I é modificado por programa, exclusivamente; o "bit" C será atualizado pela instrução "testa estado", segundo a lógica "E" ou "OU" nos "bits" do registrador de estado. Há, também, instrução para "modificar estado" que, como a anterior, usa uma máscara de 8 "bits", devendo-se especificar se o teste ou modificação será feito sobre os "bytes" SD ou SE. O registrador de "estado" (S) pode, ainda, ser salvo por um "salva estado" e restaurado por um "restaura estado". Ambas as instruções utilizam para a transferência um registrador da rede.

Também merece referência o fato de os registradores de propósito geral que são usados como "contador de instruções" (R0) e "ponteiro de pilha" (R1) terem sido escolhidos ao inverso dos computadores PDP-11 [11,12]. Considerando somente o "contador de instruções", a inversão se verifica também com relação ao computador FACOM U-200 [17]. Podemos constatar nestes computadores, que são usados, para "contador de instruções", o R7 e, para "ponteiro de pilha" o R6. A vantagem de se usar o R0 com CI é a de podermos indexar usando este registrador. O uso do registrador R0 não permite distinguir a "ausência de índice" do "índice 0" (zero). Desta forma, sempre que não for apontado um outro registrador (R1 a R7), o CI será tomado como índice, o que permite endereçamento relativo ao CI em instruções curtas. Todavia, esta facilidade não foi utilizada. As instruções curtas não usam o registro de índice. Ape -

nas as instruções longas, onde o índice 0(zero) com significado de "ausência de indexação", foi adotado.

3.4 - Estrutura de Entrada/Saída

As operações de Entrada/Saída no computador G-10 são de dois tipos: "Programada" e "DMA".

As operações de entrada/saída "Programada" são usadas para o Canal Concentrador em todos os tipos de operação e, para o Canal Seletor, nas operações que não envolvam transferência de dados, uma vez que a transferência de dados se faz diretamente com a memória (Tipo "DMA"). O Canal Seletor transfere até 1 Mega-palavras/seg.

Para as operações do tipo "Programado" são usadas as seguintes instruções:

- 1) Função: Execução de operações do tipo: armazena no registrador do controlador para execução, "inibe/permite" interrupção do canal, "altera/testa" estado do dispositivo e "testa" estado do canal.
- 2) Entrada: Transfere um dado do registrador do controlador para um registrador do PC ou posição de memória.
- 3) Saída: Transfere um dado de um registrador do PC ou posição de memória para o registrador do controlador.

Para a operação do tipo "DMA" existe a instrução "Transfere Bloco", que provoca a transferência de um bloco inteiro de dados, sob controle do canal.

Usa-se para cada um dos canais, as instruções seguintes:

- Concentrador: Função, Entrada, Saída.
- Seletor : Função, Transfere Bloco.

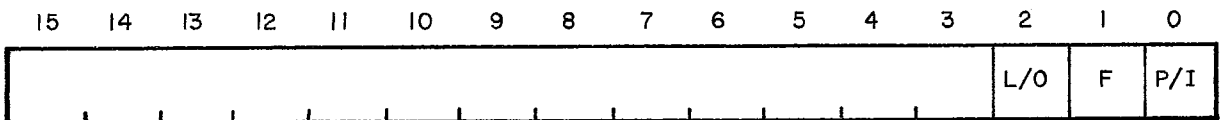
A instrução "Função" com operação do tipo "permite/ /inibe interrupção do canal" atua sobre uma "máscara" de interrupção interna do PC. O controle do estado do canal deve ser feito por "software".

A mesma instrução "Função" com operação do tipo "altera estado do dispositivo" atua sobre o registrador de estado do controlador de tal forma a:

- "inibir/permitir" pedidos de interrupção do dispositivo;
- "inibir/permitir" pedidos de interrupção de dispositivos de níveis interiores.

O formato geral dos registradores de "estado" dos controladores ("interfaces") é o seguinte:

REGISTRO DE ESTADO DE INTERFACE



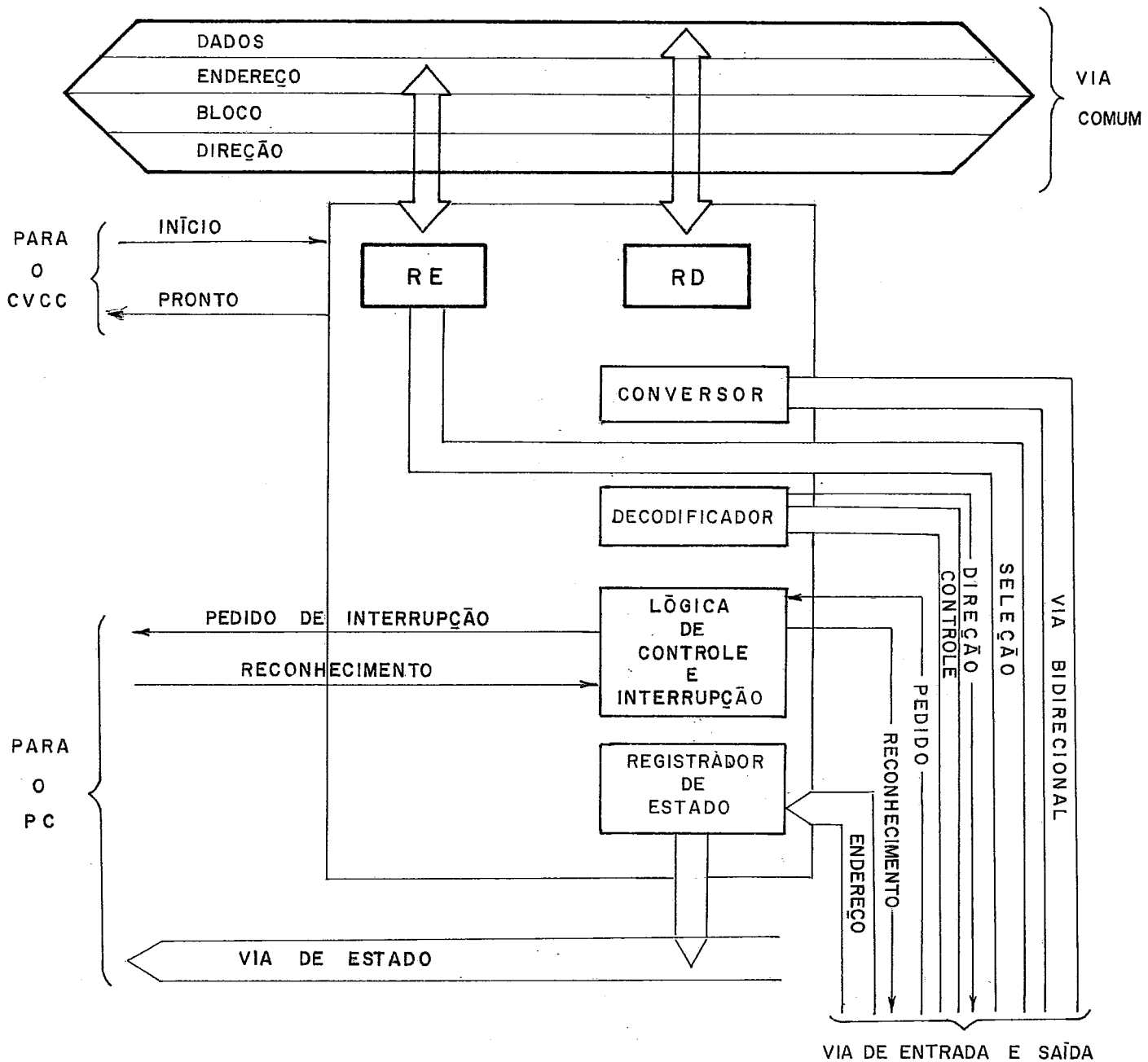
O uso dos "bits" é o seguinte:

bits

- 0 - P/I - "permite/inibe" interrupção no dispositivo.
- 1 - F - "permite/inibe" interrupção nos níveis inferiores.
- 2 - L/O - "livre/ocupado".
- 3 - 15 - de acordo com cada tipo de dispositivo.

O Canal Concentrador possui três registradores: "dados" (RD), "endereço" (RE) e "/estado". Os registradores de "dados"

e "endereços" estão conectados as linhas correspondentes da Via Co mum e o de "estado", a Via de Estado.



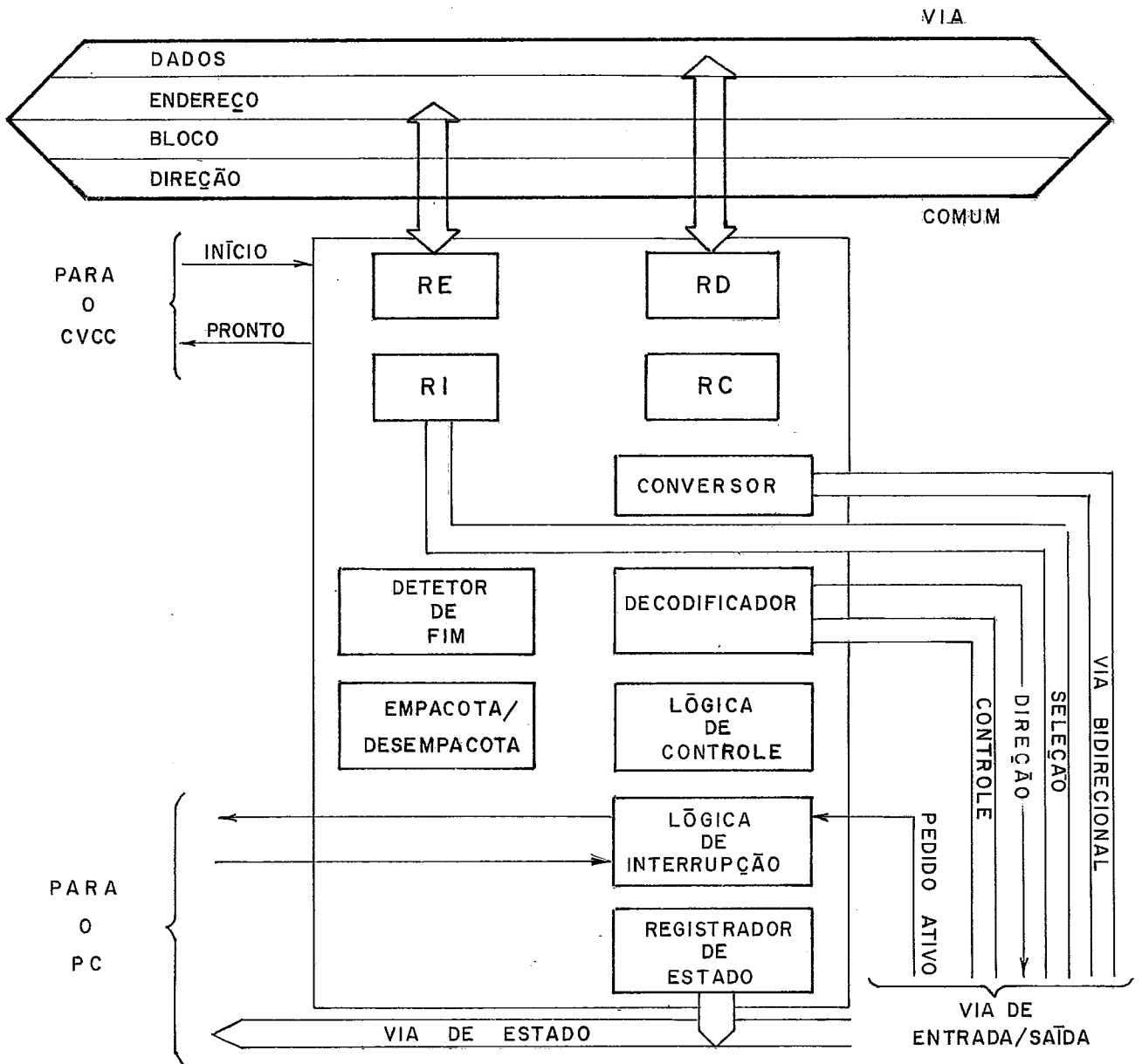
O Canal Concentrador possui as seguintes unidades lógicas: conversor, decodificador e lógica de controle e interrupção.

O decodificador interpreta a instrução que se encontra no RE envia sinais de controle para o conversor fazer a transferência entre a Via Bidirecional e RD, indicando o sentido, e con

troles para os dispositivos. Se a instrução é para o próprio canal, envia sinais para a lógica de controle e interrupção.

O canal estará em 1 entre 2 estados: "Execução" e "Interrupção". O estado de "Execução" é aquele compreendido entre os sinais "início" e "pronto" do diálogo com a Via Comum. Durante o estado de "Execução" os pedidos de interrupção ficam inibidos.

O Canal Seletor possui um número maior de registradores que o Concentrador: "dados" (RD), "endereço" (RE), "instrução" (RI), "contador de palavras" (RC) e "estado".



Possui, também, um maior número de unidades: conversor, decodificador, lógica de controle, lógica de interrupção, detetor de fim e lógica para "empacotar/desempacotar".

As diferenças para o Concentrador são:

- a) registradores: de "instrução" e "contador de palavras", além dos de "dados", "endereço" e "estado" (comuns);
- b) unidades: detetor de fim, lógica para "empacotar/desempacotar", separação das lógicas de controle e de interrupção, além do conversor e decodificador que são comuns. O controle é microprogramado;
- c) via de entrada/saída: não possui as linhas de reconhecimento de pedido de interrupção e de endereço, pois o único dispositivo que pode encaminhar um pedido ativo é o que foi selecionado.

O próprio canal seletor gera a interrupção de "fim de transferência de bloco", sob comando do detetor de fim. As demais são geradas pelo dispositivo selecionado.

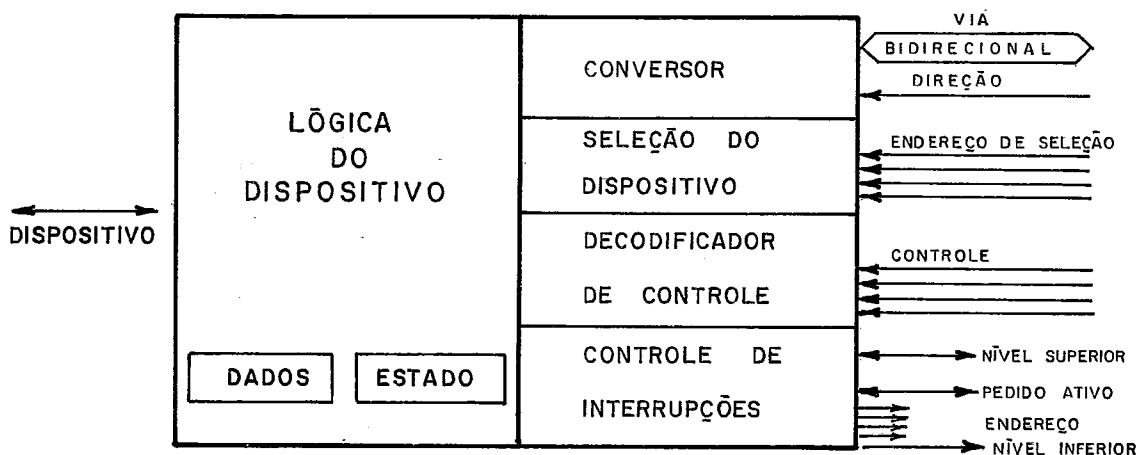
O endereço inicial do bloco é colocado no RE e o contador de palavras no RC. Após cada transferência de uma palavra, o RE é incrementado e o RC, decrementado. Quando o RC chega a 0 (zero) o detetor de fim sinaliza ao controle.

Os dados serão "empacotados/desempacotados", em uma palavra, como for especificado na instrução "transfere bloco".

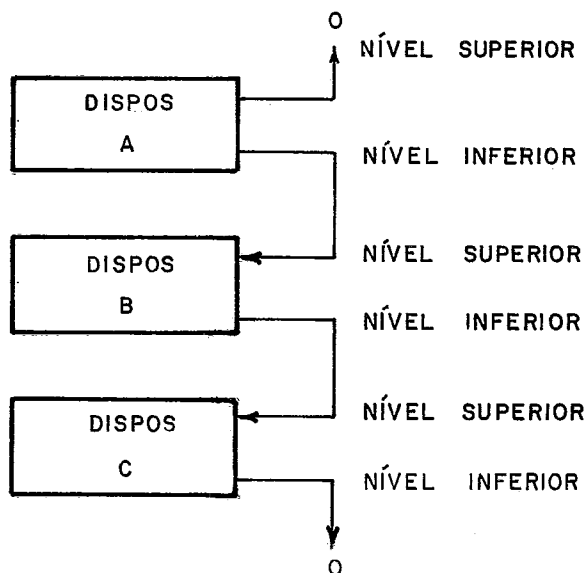
Os dispositivos são conectados as Vias de Entrada/Saída através dos seus controladores, chamados "Interfaces Padrão", ou simplesmente "Interfaces".

As "interfaces" possuem uma parte padrão, com a qual se ligam à Via de Entrada/Saída, e uma parte específica com lógica de controle para cada dispositivo. Os sinais da parte padrão correspondem aos sinais da Via de Entrada/Saída. Há dois registrado-

res: "dados" e "estado".



Aos sinais da Via de Entrada/Saída devem se acrescentar os sinais "nível superior" e "nível inferior". Estes sinais permitem inibir os pedidos de interrupção de níveis inferiores a partir de um dado dispositivo. A ligação destes sinais é mostrada abaixo:



Serão ligados ao Canal Concentrador os seguintes dispositivos:

- 1) Tele-impressora tipo ASR-33 com modificações;
- 2) Leitora de Cartões;
- 3) Perfuradora de Fita de Papel;
- 4) Leitora de Fita de Papel.

Serão ligados ao Canal Seletor os seguintes dispositivos:

- 1) Discos de cabeça móvel;
- 2) Impressora de baixa velocidade.

O motivo de se conectar a impressora de baixa velocidade, ao Canal Seletor é que a mesma possui um "buffer" cíclico que é carregado por blocos de caracteres. A descarga do "buffer" é automática. A impressora possui a particularidade de enviar "pedidos" de interrupção assíncronos também pelo Canal Concentrador, para não prender o Canal Seletor.

Ao Canal Seletor serão opcionalmente conectados outros dispositivos de alta velocidade, como fitas magnéticas.

Ao Canal Concentrador serão opcionalmente conectados outros dispositivos como por exemplo conversores A-D e D-A de baixa velocidade.

Os canais seletores buscam o endereço do bloco e contador de palavras em posições fixas de memória:

ENDEREÇO	MEMÓRIA
71	Nº DE PALAVRAS - CANAL 0
72	ENDEREÇO DO BLOCO - CANAL 0
73	Nº DE PALAVRAS - CANAL 1
74	ENDEREÇO DO BLOCO - CANAL 1
75	Nº DE PALAVRAS - CANAL 2
76	ENDEREÇO DO BLOCO - CANAL 2
77	Nº DE PALAVRAS - CANAL 3
78	ENDEREÇO DO BLOCO - CANAL 3
;	;

3.5 - Sistema de Interrupções

O Sistema de Interrupções possui 2 estruturas de prioridades em estágios diferentes:

- a) processador central
- b) canais

Até que um pedido de interrupção de um dispositivo seja atendido, ele terá concorrido com outros pedidos nestes 2 estágios.

As "interfaces" conectadas na Via de Entrada/Saída de um canal tem sua prioridade determinada exclusivamente pela posição física: quanto mais próximo do canal, maior a prioridade.

A prioridade devido à posição é obtida por meio dos sinais "nível superior" e "nível inferior". Sempre que o sinal "nível superior" é colocado em uma interface ela ativa o sinal "nível inferior", formando-se uma cadeia. Todo sinal "nível inferior" de uma interface se torna "nível superior" da interface seguinte. O sinal "nível superior" colocado em uma interface inibe portanto, os "pedidos ativos" daquela interface e de todas as posteriores, por propagação em cadeia.

O sinal "nível inferior" pode se tornar ativo de três maneiras:

- a) em virtude da existência de um "nível superior" na mesma "interface";
- b) em virtude de existir um "pedido ativo" na "interface";
- c) em virtude de estarem "inibidos" os "níveis inferiores" de interrupção, no registrador de "estado" da "interface".

Sempre, portanto, que existe um "pedido ativo" em uma "interface", todas as de níveis inferiores terão seus pedidos "ini-

bidos" automaticamente.

Desta forma, o "pedido ativo" e o "endereço" do dispositivo que chegam ao canal são sempre, relativos à "interface" de maior prioridade. O sinal de "pedido" de interrupção que o canal envia ao PC é o prosseguimento deste sinal.

Este tipo de estrutura permite que se tenha, na Via de Entrada/Saída, uma única linha de "pedido ativo" e uma única de "endereço" da "interface".

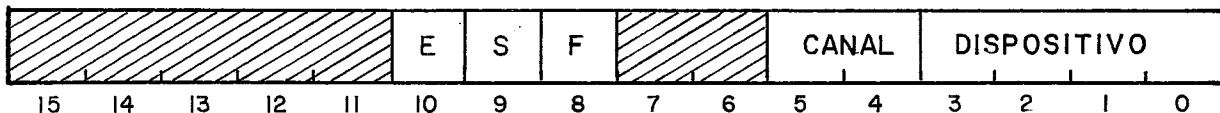
Os "endereços" das interfaces não são fixos, podendo ser registrados mecanicamente. Não há nenhuma relação entre o "endereço" da "interface", que é um número arbitrário, e a "prioridade" para interrupção. A única restrição é que não pode haver duas "interfaces" com o mesmo "endereço" em uma mesma Via de E/S.

Já os "pedidos" de interrupção que os canais enviam ao PC necessitam de uma linha para cada canal. Todos os sinais chegam ao PC, o contrário do que ocorre entre as "interfaces" e os canais.

"Reconhecido" um "pedido" de interrupção de um canal, o sinal enviado pelo PC é prosseguido pelo canal até a "interface" do dispositivo. A "interface" envia o "estado" do dispositivo, do qual serão extraídas informações para gerar a "palavra de estado" do canal.

A "palavra de estado" do canal tem o formato abaixo:

PALAVRA DE ESTADO DO CANAL



onde os campos são:

bits

- 0 - 3 - DISPOSITIVO - endereço do dispositivo
- 4 - 5 - CANAL - endereço do canal

<u>bits</u>		
6 - 7	-	não usado
8	-	F - fim de tratamento
9	-	S - erro por superposição de ins_ trução
10	-	E - erro no dispositivo
11 - 15	-	não usado

O canal coloca na Via de Estado a sua "palavra de es_ tado", para conhecimento do PC.

A partir deste ponto, a interrupção é atendida pelo PC, através de microprogramas.

A estrutura de prioridade de interrupções do estágio do processador central abrange não apenas os sinais que provêm dos dispositivos externos, mas também outros, provenientes do próprio processador central.

São as seguintes as interrupções, em ordens da mais alta para a mais baixa prioridade:

- 1) Violação de Proteção de Memória
- 2) Instrução Inválida
- 3) Chamada de Supervisor
- 4) Paridade na Memória
- 5) Falha de Alimentação - nível 0
- 6) Painel de Controle - nível 1
- 7) Relógio Interno - nível 2
- 8) Canal 0 - nível 3
- 9) Canal 1 - nível 4
- 10) Canal 2 - nível 5
- 11) Canal 3 - nível 6

As interrupções numeradas de 1 a 7 são chamadas de "internas". Destas, as de números 2 e 3 são chamadas de "internas por software" e as demais (1,4,5,6,7), "internas por hardware".

As interrupções de 8 a 11 são chamadas "externas". As interrupções de 1 a 5 não podem ser "inibidas". As interrupções 1, 2 e 4 causam a parada do processador "assincronamente", isto é, assim que é detectado o erro, o microprograma não continua. A interrupção 3 ocorre com a execução da instrução "Chamada de Supervisor".

As interrupções de 1 a 3 são mutuamente exclusivas, isto é, a ocorrência de uma delas acarreta a não ocorrência de uma das outras. Todavia, se ocorrer a interrupção 4 simultaneamente com uma das acima (de 1 a 3), será atendida após a execução da primeira instrução da rotina de serviço de interrupção concorrente (de 1 a 3).

O sistema de "níveis" de interrupção engloba da interrupção 5 até a 11. São 7 níveis, numerados de 0 a 6. Inclui apenas as interrupções "síncronas", isto é, aquelas que são atendidas após cada "ciclo de instrução".

O nível 0 (zero) não é "inibível", porém os níveis de 1 a 6 são. O "bit" I do registrador de "estado" (S), quando ativado, inibe os níveis 1 a 6.

Apesar de haver apenas 7 níveis explícitos, na verdade há um total de 11, pois a diferença se encontra apenas na maneira como são geradas as interrupções: de 1 a 4 ocorrem durante a execução de microprogramas enquanto que as restantes, de 5 a 11 ("níveis" 0 a 6), ocorrem em virtude de sinais enviados por algum "dispositivo" externo ao processador.

Em virtude disto, todas as 11 interrupções possuem uma posição no Vetor de Interrupções, isto é, tem um endereço de rotina de serviço, apesar de diferirem na maneira com que são geradas. Em outras palavras, existe uma prioridade "implícita" para as interrupções "internas", geradas por microprogramas assim como existe uma "explícita" para aquelas geradas em dispositivos "externos", enviadas na forma de sinais e classificadas por níveis.

O Vetor de Interrupções contém todos os endereços de

rotinas para tratamento de interrupções de cada um dos dispositivos (64 interfaces) para as interrupções dos níveis 3 a 6 (canais), bem como das interrupções "internas".

No caso dos dispositivos, o endereço é formado pela concatenação dos campos "canal" e "dispositivo" da "palavra de estado" do canal. Nos demais casos é gerado por microprogramas.

O Vetor de Interrupções é mostrado esquematicamente no gráfico abaixo:

ENDEREÇO	MEMÓRIA
0	ENDEREÇO ROTINA INTERFACE (00)16
1	ENDEREÇO ROTINA INTERFACE (01)16
:	:
:	:
:	:
63	ENDEREÇO ROTINA INTERFACE (3F)16
64	ENDEREÇO ROTINA CHAMADA SUPERVISOR
65	ENDEREÇO ROTINA FALHA DE ALIMENTAÇÃO
66	ENDEREÇO ROTINA INSTRUÇÃO INVÁLIDA
67	ENDEREÇO ROTINA RELÓGIO INTERNO
68	ENDEREÇO ROTINA VIOLAÇÃO DE MEMÓRIA
69	ENDEREÇO ROTINA INTERRUPTÃO P/PAINEL
70	ENDEREÇO ROTINA ERRO DE PARIDADE
:	:
:	:
:	:

O computador G-10 não usa explicitamente a "máscara" de interrupção e o Vetor de Interrupções não contem as PEP,mas apenas os endereços das rotinas de serviço.

O "estado" do processador, o "contador de instruções" e a "base de programa" correntes serão armazenados na pilha, como VPEP.

Ao ocorrer uma interrupção a micro-rotina que trata de interrupções executa os seguintes passos:

- a) "Inibe os demais pedidos de interrupção;
- b) Coloca na pilha, apontada pelo registrador R1, o "contador de instruções" (R0), a "base de programa" (BP) e o registrador de "estado" (S); incrementando o R1;
- c) Coloca 0(zero) na "base de programa" (BP) e (FFFF)₁₆ no "limite de programa" (LP);
- d) Se a interrupção é "externa", coloca o "byte" esquerdo da "palavra de estado" do canal no "byte" esquerdo (SE) do registrador de "estado" (S). Se é "Chamada de Supervisor", coloca o código da chamada no "byte" esquerdo (SE) do registrador de "estado" (S). Coloca o "bit" M(modos) do registrador de "estado"(S) no modo supervisor;
- e) Carrega o "contador de instruções" (R0) com o endereço da rotina de serviço para tratamento da interrupção, a partir do Vetor de Interrupções;
- f) "Permite" as interrupções e executa a primeira instrução da rotina de serviço.

Após executar uma rotina de serviço, a última instrução deve ser um "retorno de interrupção". Esta instrução restaura os registradores de "estado", "contador de instruções" e "base de programa" do programa interrompido, bem como "permite" as interrupções que foram "inibidas" de uma forma seletiva. Então serão "permitidas" interrupções de nível inferior ao painel, ao relógio interno, ao canal 0, ao canal 1 ou ao canal 2, anteriormente "inibidas" de forma seletiva, através de instruções tipo "função" para o relógio ou os canais, com operação do tipo "inibe interrupção".

Tem-se, portanto, no G-10, a possibilidade de limitar interrupções de três maneiras:

- a) Todas as interrupções "externas" (canais) e algumas "internas", mas provocadas por dispositivos externos (Painel e Relógio Interno), ou seja, níveis 1 a 6,

através do "bit" I do registrador de "estado" do processador (S).

- b) Seletivamente, alguns níveis, que incluem o Relógio Interno e os Canais, ou seja, níveis 2 a 6, através de uma instrução "função" para os dispositivos enumerados acima, usando a operação "inibe/permite" interrupções, aplicada a uma "máscara" no mecanismo de interrupção do PC.
- c) Cada um dos dispositivos ou todos os dispositivos de nível inferior a um determinado dispositivo de uma Via de E/S, através de uma instrução "função" com as alternativas "inibe/permite" para um dado dispositivo, ou "inibe/permite níveis inferiores", aplicada a um dispositivo em particular.

Observa-se que as interrupções dos dispositivos são podem ser "inibidos" e não "desarmadas", explicitamente. Significa que as interrupções ficarão sempre pendentes se não forem atendidas. Todavia, a execução de uma nova operação no mesmo dispositivo limpa o "flag" de interrupção, para que não haja superposição de interrupções. Se não for atendida, uma interrupção fica pendente num dispositivo até que outra instrução de E/S seja executada para o mesmo dispositivo.

Uma interrupção pendente num dispositivo será perdida, neste caso, quando for executada outra operação de entrada/saída.

As instruções de E/S e outras, usadas para limitar as interrupções e alterar o sistema de prioridade, incluem instruções que fazem o efeito das do tipo:

- a) "inibe/permite" (alteração do "bit" I do registrador de "estado" do processador);
- b) "inibe/permite seletivo" ("função" para relógio e canais);

- c) "selecione prioridade seguinte" ("retorno de interrupção").

Não é utilizada explicitamente a "máscara" de interrupção, não havendo instruções como "altere registro de máscara".

Após atender uma interrupção e executar a primeira instrução da rotina de serviço, qualquer nível de interrupção (inclusive inferior) pode ocorrer e ser atendido.

Para evitar que sejam perdidos os conteúdos dos registradores do programa interrompido, a primeira instrução da rotina de serviço deve ser um "inibe", usando o "bit" I do estado. Por "software" salvam-se os registradores e "inibe-se" seletivamente os níveis iguais e inferiores, "permitindo-se" os superiores. O efeito é o de uma "máscara" implícita à custa de um maior "overhead" e pior "tempo de resposta" devido a um grande número de instruções. O tempo de resposta (latência) por "hardware" é de 6 μ s. Porém somente 3 registradores são salvos na pilha e todos os demais registradores deverão ser manipulados por "software". O sistema de interrupções não é, portanto, ótimo, embora possa atender a maioria das aplicações.

CAPÍTULO 4

UMA PROPOSTA PARA O SISTEMA DE INTERRUPÇÕES DO G-10

4.1 - Premissas Básicas

O sistema de interrupções proposto tem como objetivos:

- a) obter um melhor "tempo de resposta";
- b) obter um menor "overhead";
- c) diminuir os riscos de "saturação" do sistema;
- d) evitar o "over-run" nas operações de entrada/saída;
- e) obter maior flexibilidade na limitação de interrupções e alteração do sistema de prioridades;
- f) permitir uma maior utilização de rotinas reentrantes para os dispositivos do mesmo tipo;
- g) diminuir o número de posições fixas de memória, ou seja, aumentar a memória útil.
- h) obter maior flexibilidade na administração das rotinas de serviço dos dispositivos evitando desvios e retornos desnecessários.

Tomam-se ainda por base os seguintes fatos:

- a) O computador G-10 teve sua utilização definida como sendo um computador de propósito geral para aplicações em engenharia, técnico-científicas e com grandes possibilidades de utilização em controle de processos, comunicações e outras aplicações em tempo real;
- b) O custo cada vez menor do "hardware" em relação ao

"software" de um sistema de computação, especialmente das memórias rápidas ou "scratch-pad memories";

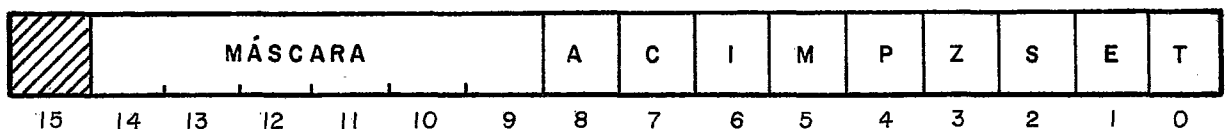
- c) As modificações propostas podem ser introduzidas sem alterar a arquitetura básica do sistema, ou seja, se rão feitas apenas em nível do processador central.

A solução apresentada se baseia num largo emprego de memórias rápidas para atingir os objetivos acima propostos.

4.2 - Máscara de Interrupções

Considera-se indispensável, para dar maior flexibilidade à limitação das interrupções e alteração do sistema de prioridades a introdução da "máscara de interrupção".

Esta será incluída no "byte" esquerdo do registrador de estado de processador (S), que terá o aspecto abaixo:



Apenas os níveis de interrupção que podem ser "inibidos" tem correspondência com os "bits" da máscara e são:

NÍVEL	"BIT"	NÍVEL	"BIT"
1	9	4	12
2	10	5	13
3	11	6	14

O "bit" 15 fica disponível para futuras extensões.

O "bit" 8 conterá o indicador de interrupção "ativa", indicando que a rotina de serviço foi iniciada e não concluída.

O antigo campo chamado "CÓDIGO" deste mesmo registrador será transferido para uma posição fixa de memória. Além de man

ter uma das funções anteriores, isto é, conter o código ("byte" direito) de uma "Chamada de Supervisor" conterá também, nos casos de outros tipos de interrupção os seguintes dados:

- a) Violação de Proteção de Memória - o endereço absoluto da instrução que causou a violação;
- b) Instrução Inválida - o endereço absoluto desta instrução;
- c) Paridade na Memória - o endereço absoluto da posição de memória em que ocorreu o erro.
- d) Canais - não mais o "byte" esquerdo, mas as informações relevantes da palavra de estado do canal.

4.3 - Rede de Registradores e Blocos de Contexto

Utilizando memórias rápidas serão constituídos 9 blocos para conter os contextos seguintes:

- a) programas do usuário ou supervisor;
- b) rotinas de interrupção.

Estes blocos de contexto serão distribuídos como se segue:

BLOCO	CONTEXTO
0	Programas (usuário ou supervisor)
1	Interrupção nível 6
2	Interrupção nível 5
3	Interrupção nível 4
4	Interrupção nível 3
5	Interrupção nível 2
6	Interrupção nível 1
7	Interrupção nível 0
8	Interrupções Internas

Cada "bloco de contexto" conterá os seguintes registradores:

- a) de trabalho (R1 a R8) - 8 registradores;
- b) de estado - (S);
- c) contador de instruções (CI);
- d) limite de dados (LD);
- e) limite de programa (LP);
- f) base de dados (BD);
- g) base de programa (BP);
- h) apontador da pilha de subrotinas (AS);
- i) apontador da pilha de dados (AD);

Chama-se no texto de registradores "essenciais" aos 8 últimos do conjunto.

Os registradores de rascunho em número de 4, utilizados pela micro-máquina ficarão em uma memória em separado.

Tem-se um conjunto de 9 blocos de 16 registradores, perfazendo um total de 144 palavras de 16 "bits".

Deve-se observar que o "contador de instruções" (CI) é independente dos registradores de trabalho, não se confundindo mais com o R0. Também o R1 deixa de ter sua antiga função de apontador da pilha, uma vez que a mesma não mais será utilizada pelo sistema de interrupções. Como pilha dos programas a serem manipulados por "software", de forma não automática, o programador pode escolher qualquer registrador, a seu exclusivo arbítrio, pois as operações de "push" e "pop" são executadas com base em qualquer registrador, já que o G-10 permite auto-incremento e auto-decremento dos registradores de trabalho.

Haverá uma pilha para uso das instruções de chamada e retorno de subrotinas, automática, isto é, manipulada pelo "hardware". Ver o parágrafo 4.14 para maior clareza.

As instruções "salva registradores" e "restaura registradores" que utilizavam a antiga pilha serão implementadas de outra forma pois as mudanças de contexto não mais obrigam a execução destas operações.

A existência de um registrador de trabalho como apontador de uma pilha geral não se justifica mais com a solução atual. O programador disporá efetivamente de 8 registradores de trabalho. São criadas entretanto 2 outras pilhas, com outras finalidades, ou seja, atendimento automático de subrotinas embutidas ("nesting") e salvamento seletivo de registradores de trabalho. Deve ficar claro que desviar para subrotinas durante o atendimento de interrupção é fácil dentro da estrutura atual. Assim, de qualquer "bloco de contexto" é possível o tratamento automático de subrotinas e salvamento seletivo de registradores de trabalho, mesmo embutidas.

4.4 - Mudança de Índice de Contexto

A mudança de contexto será baseada em um registrador de "índice de contexto". A "mudança do contexto" portanto pode ser referida também como "mudança de índice de contexto".

Este é um registrador de 4 "bits" que conterà o número do "bloco do contexto" corrente, ou seja, um valor de 0 a 8.

Se o "índice de contexto" for \emptyset (zero), significa que não há interrupções e neste caso o "bloco de contexto" \emptyset (zero) é o corrente.

Ocorrendo interrupções elas serão atendidas em ordem de prioridade e o valor do "índice de contexto" será gerado para que indique o número do "bloco de contexto" que, será ativado. A prioridade das interrupções é a mesma já anteriormente estabelecida e levando em consideração que as interrupções internas são mutuamente exclusivas e tem a mais alta prioridade.

A ativação de um "bloco de contexto" é feita por comutação, mantendo-se inalterados os conteúdos dos demais blocos.

A latencia por "hardware" para mudança do contexto esperada é de no máximo de 3 μ s, aproximadamente, com base nos exemplos similares.

Observe-se que o contexto é de 16 registradores e que contem todo o estado do programa.

As interrupções serão atendidas como anteriormente, ao final do ciclo de execução de cada instrução.

Com as mudanças de contexto, todos os registradores ficarão inalterados, permitindo retornar à instrução que seria executada a seguir no programa interrompido.

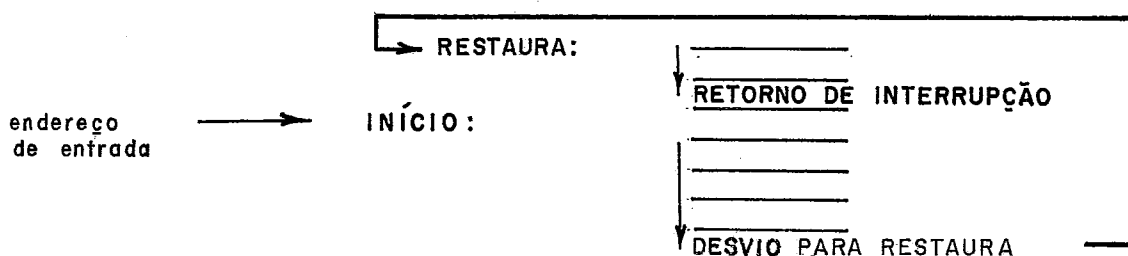
Todavia tamanha vantagem, com a redução do tempo de resposta e menor "overhead" naturalmente só é obtida com um certo tributo que será pago em se obedecer a algumas convenções do sistema.

4.5 - Convenções do Sistema

A convenção fundamental do sistema diz respeito ao "software", isto é, às rotinas de interrupção.

Baseia-se no fato de que, ao ser executada a instrução "retorno de interrupção" o estado do contexto tem que ser exatamente igual ao inicial, isto é, quando a rotina foi iniciada.

Todas as rotinas terão o aspecto geral do diagrama abaixo:



A instrução de "retorno de interrupção" não é coloca

da no final do programa e sim antes do ponto de entrada da primeira instrução.

Isto faz com que, antes de retornar ao programa interrompido o "contador de instruções" do bloco corrente seja incrementado e aponte para a instrução seguinte, que é o próprio ponto de entrada. Assim fica garantida a restauração do CI. O registrador de "estado" (S), se tiver sido alterado durante a rotina, será restaurado. Normalmente, nenhum dos outros registradores necessitará de restauração, o que não impede todavia de, por exemplo, deixar valores nos registradores de trabalho, para serem usados na execução seguinte da mesma rotina.

Uma segunda convenção é de "hardware" e diz respeito ao uso da "máscara" de interrupções, colocada no registrador de "estado" (S), portanto, afetando o "software".

Ao ocorrerem interrupções, as "máscaras" serão puras e simplesmente usadas para alterar o registrador de máscara de interrupção do sistema de interrupções. Fica, portanto, a cargo do programador escolher quais as "máscaras" no registrador S lhe darão um correto desempenho do sistema.

Embora ao atender uma interrupção o processador central envie aos dispositivos externos o sinal de "reconhecimento" e "limpe" o "flip-flop" correspondente aquele nível, o sistema de prioridades não atenderá nunca pedido de interrupção de nível igual ou inferior ao que se encontra no registrador de "índice de contexto", mesmo que outro pedido seja registrado no mesmo nível ou inferior.

Uma terceira convenção, decorrente da primeira é a de que o uso de sub-rotinas obriga sempre o retorno à mesma rotina de serviço da qual se desviou, para que sejam restaurados os registradores do "bloco de contexto", inclusive o contador de instruções, com a execução da instrução "retorno de interrupção" e desativado aquele nível de interrupção.

4.6 - Organização de Memória

Do ponto de vista do sistema de interrupções, apenas duas palavras da memória primária, de núcleos, serão usadas. Uma conterá o indicador de tipo de interrupção interna. Esta palavra é a de endereço 0 (zero). A segunda palavra, endereço 1, conterá o código de interrupção. Desta forma, a partir do endereço 2, inclusive, estará a tabela de endereço de bloco e contadores de palavras para os canais, como mostrado no diagrama abaixo:

<u>ENDEREÇO</u>	<u>MEMÓRIA</u>
0	INDICADOR INTERRUPTÃO INTERNA
1	CÓDIGOS DE INTERRUPTÃO
2	Nº DE PALAVRAS - CANAL 0
3	'
4	'
5	'
6	'
7	'
8	Nº DE PALAVRAS - CANAL 3
9	ENDEREÇO DE BLOCO - CANAL 3
	: : : : : :

Como é necessário ao sistema operacional distinguir qual das interrupções internas ocorreu, já que para elas é reservado um único bloco de registradores de contexto, a palavra de endereço 0 (zero) conterá um código que indique o tipo de interrupção.

A palavra de endereço 0 (zero) será usada pela rotina que atende interrupções internas para selecionar uma das sub-rotinas que atenderão a cada tipo de erro.

A segunda palavra da memória conterá o endereço de memória onde ocorreu um erro nos casos de:

- a) Violação da Proteção de Memória;
- b) Instrução Inválida;
- c) Paridade na Memória.

Ainda nesta mesma palavra serão colocados:

- a) palavra de estado do canal, no caso de interrupções externas;
- b) código de Chamada de Supervisor.

4.7 - Sequência de Atendimento das Interrupções

A sequência de execução da micro-rotina que trata das interrupções fica alterada para os seguintes passos:

- a) Inibe os demais pedidos de interrupção;
- b) Calcula o valor do "índice de contexto";
- c) Coloca o "índice de contexto" no registrador de "índice de contexto";
- d) Coloca na palavra de endereço $\emptyset\emptyset$ da memória o código gerado pela interrupção se for interna. Coloca o "bit" A (ativado/desativado) no estado "ativado", o "bit" M (modo) no estado Supervisor e o "bit" I do registrador S no estado "inibe";
- e) Executa a mudança de "bloco de contexto" (comutação);
- f) Inicia a primeira instrução da rotina de serviço.

Ao iniciar-se a rotina de interrupção, o "bit" I do registrador S estará automaticamente inibido, cabendo ao programador desinibi-lo quando julgar conveniente, deixando atuar a "máscara" de interrupção.

Deve-se observar que no caso de interrupções externas (canais), o desvio será feito para uma rotina geral do Canal e não mais diretamente para uma rotina do dispositivo colocada esta em uma posição fixa de memória como anteriormente.

Cabe ao programador identificar o dispositivo, usan-

do a palavra de endereço de memória $\emptyset 1$. Então será feito por "software" o desvio para a rotina do dispositivo. Este método permite o uso de rotinas reentrantes, sem a necessidade de repetição de várias cópias da mesma rotina, para um mesmo tipo de dispositivo do qual existam vários, conectados em canais diferentes ou posições diferentes de um mesmo canal.

4.8 - Alteração de Prioridades e Limitação das Interrupções

A alteração de prioridades e limitação das interrupções fica portanto sendo feita com as seguintes alternativas:

- a) Todos os níveis de interrupções externas ou internas, causadas por dispositivos externos, excluindo-se Falha de Alimentação, através do uso do "bit" I do registrador de estado do processador (S);
- b) Alguns níveis de interrupção, de 1 a 6, através do uso da "máscara" de interrupções do registrador S;
- c) Cada um dos dispositivos, ou dispositivos inferiores a um determinado, através dos "bits" do registrador de estado da interface.

Embora a "máscara" com todos os "bits" inibindo interrupções equivalha em função ao "bit" I do registrador S inibindo, deve ser mantido o bit I, por comodidade para o programador, como o fazem vários outros computadores especialmente alguns sistemas de tempo real onde as rotinas de interrupção não são interrompidas em hipótese alguma e seu tempo de execução fica restrito entre os 100 e 200 μ s. As demais tarefas complementares ao atendimento do dispositivo que provoca a interrupção são concluídas, por programa de alta prioridade que é ativado por "flag", dentro do Sistema Operacional.

4.9 - Instruções Relativas às Interrupções

As instruções necessárias ao processamento das interrupções são do seguinte tipo:

- a) inibe/permite - altera o bit "I" do registrador S , que atua sobre as portas de controle, impedindo a chegada de sinais à rede de prioridade.
- b) altere registro de máscara - altera a "máscara" de interrupção contida no registrador S, a qual atua sobre a máscara do mecanismo de interrupção.
- c) selecione prioridade seguinte - que faz o "retorno da interrupção" e restaura o contexto do programa que foi interrompido e, portanto, o sistema de prioridades que vigorará a partir da próxima instrução.

4.10 - Interrupções Múltiplas e Restauração do Contexto Anterior

No caso de interrupções múltiplas o contexto anterior seria perdido.

Tal não acontece, pois ao se retornar de uma interrupção, o "índice de contexto" será decrementado, sucessivamente, testando-se o "bit" A do registrador S para saber se o nível correspondente está "ativado"/"desativado". Estando "ativado", prossegue a execução com o bloco de contexto indicado pelo "índice de contexto".

4.11 - Retorno da Interrupção

A instrução de retorno de interrupção deve antes de restaurar o contexto velho do programa interrompido:

- a) Inibir todas as interrupções;
- b) Decrementar o "índice de contexto";
- c) Se há interrupção pendente já registrada e não atendida ("bit" A do registrador S indicando "desativado"), será atendida passando-se para o item e) abaixo. Não havendo interrupção pendente, o "índice de contexto" continuará a ser decrementado, ou seja, retorna-se ao item b). Caso contrário passa-se ao item d).
- d) Encontrado uma interrupção pendente com o "bit" A indicando bloco "ativado" no registrador S, deve-se pros

- seguir o seu atendimento;
- e) Executar a mudança de "bloco de contexto" (comutação);
 - f) Sõ no caso de "índice de contexto" ser \emptyset (zero), permitir todas as interrupções. Executar a primeira instrução do programa interrompido no bloco apontado pelo índice de contexto. Se o "índice de contexto" não é \emptyset (zero) retornar ao item b) acima.

Este procedimento evitarã que, em situação de alta demanda de interrupções, algumas que tenham sido permitidas pelas "mãscaras" das rotinas, não tenham sido atendidas por serem de nível mais baixo e ocorra uma "saturação".

Os procedimentos dos itens b), c), d), e) e f) serão repetidos até que haja o retorno ao programa interrompido, ou seja ao bloco \emptyset (zero), quando as interrupções serão novamente permitidas.

4.12. Inicialização do Sistema

Ao ser inicializado, o sistema operacional, deverá carregar os "registradores de contexto". Para tal existirá uma instrução especial do tipo "carregue registradores de contexto", indicando o endereço inicial do conjunto de 8 ou 16 palavras a ser carregado e o "bloco de contexto" ao qual pertence o registrador, ou seja, o "índice de contexto". Este carregamento afetará opcionalmente 8 ou 16 registradores de uma sõ vez. No caso de afetar 8 registradores, estarão excluídos apenas os de trabalho. Como todavia os registradores de trabalho, R1 a R8, conterão, a critério do programador, valores com os quais os programas ou as rotinas de interrupção trabalharão ao se iniciarem, há a opção de afetar todos os 16 registradores. Esta mesma instrução será usada para mudar de tarefa, juntamente com a instrução "salve registradores de contexto".

4.13 - Mudança de Modo e Conjunto de Instruções

A inclusão do "bit" do modo na de codificação da instrução permite expandir o conjunto de instruções pois o supervisor usa parte do conjunto que é comum ao modo usuário. Conca - tenando-se este "bit" de modo ao código de operação pode-se,

para o conjunto não comum a ambos, reutilizar os mesmos códigos de operação. Ficarã eliminada a interrupção do tipo "instrução privilegiada", uma vez que os códigos terão significados diferentes em cada um dos modos. Como exemplo de instruções não utilizadas pelo sistema operacional, incluem-se as de ponto-flutuante. Como exemplo de instruções não utilizadas pelos programas do usuário estão todas as "privilegiadas".

4.14 - Chamada e Retorno Automáticos de Sub-Rotinas

Ao ser executada a instrução "chamada de sub-rotina" serão colocadas na pilha de sub-rotinas apontada pelo registrador AS, os 8 registradores essenciais, excluindo-se os de trabalho. O registrador AS do "bloco de contexto" correspondente será incrementado de 8. Ao ser executada a instrução "retorno de sub-rotina", os 8 registradores essenciais serão restaurados e portanto também o registrador AS.

Para salvar registradores de trabalho, o programador ao iniciar a sub-rotina, executará a instrução "salva registrados" que inclui a sequência de registradores que se iniciará sempre com R1 e terminará com o que for indicado na instrução, os quais serão colocados a partir da posição apontada. O registrador da pilha de dados, ou seja, o AD, do "bloco de contexto", será incrementado de um número igual ao dos registradores salvos.

Antes de retornar da sub-rotina, o programador deve executar a instrução "restaura registradores" que da mesma maneira carregará os registradores R1 até o que for indicado, com os conteúdos das posições indicadas pelo número correspondente de palavras até a posição apontada por AD. O registrador AD será decrementado convenientemente.

Mesmo que o programador não restaure o conteúdo dos registradores de trabalho e se retorne da sub-rotina, o registrador AD será também restaurado, da mesma maneira como o AS, a partir do conteúdo da pilha de sub-rotinas.

O uso da pilha automática de sub-rotinas se justifica ainda pela possibilidade que, em um determinado bloco, apontado pelo "índice de contexto", haja modificações nos registradores essenciais, especificamente, S, CI, LD, LP, BD e AD.

Não fica excluída a possibilidade de alteração dos registradores BP e AS, dependendo do conjunto de instruções que for implementado. Mesmo atualmente há instruções que permitem executar outra instrução tanto na área de programas como até mesmo na de da dos.

4.15. Mudança de Tarefas

Para facilitar o "scheduling" do sistema operacional e a mudança de tarefas, serão utilizadas as instruções "carregue re gistradores de contexto" e "salve registradores de contexto". Estas, opcionalmente, poderão manipular 8 ou 16 registradores. No ca so de estarem envolvidos 8, serão os essenciais (com a exclusão dos de trabalho, R1 a R8).

A carga será feita a partir do endereço indicado, na instrução, até o número de posições seguintes, para os registradores do "bloco de contexto" que for indicado na própria instrução, pelo seu "índice de contexto".

O salvamento, também da mesma forma, será de 8 ou 16 registradores e, no caso de 8, sã os essenciais para as posições sucessivas de memória apontadas pelo endereço contido na própria ins trução e vindos do "bloco de contexto" também indicado na mesma.

Se o "scheduling" se basear em pedido de operações de entrada/saída, a mudança de tarefas depende basicamente da estrutura do sistema operacional.

Se todavia o Relógio Interno for usado, para "Time-Slicing" ou operações de "Time-Sharing", a rotina de Serviço do nível de interrupção do relógio poderá entre outras providências, tais como ativar e desativar programas (nas tabelas do sistema ope racional), inclusive trocar o contexto do "bloco de contexto" \emptyset (zero), salvando o corrente e carregando um novo.

CAPÍTULO 5

CONCLUSÕES

Algumas conclusões podem ser tiradas para justificar o sistema de interrupções proposto.

Do Capítulo 1 deduzimos que embora em um computador digital de propósito geral o sistema de interrupção não seja essencial, de fato, para a maioria das aplicações dos computadores atuais é indispensável.

Verificamos também que um dos problemas básicos para atender interrupções é o da mudança do contexto.

No Capítulo 2 constatamos que há máquinas bem simples como os computadores NOVA, mas que sistemas mais completos como IBM/360, FACOM U-200 e PDP-11 utilizam o conceito de Palavra de Estado do programa e, os três primeiros a máscara de interrupções como um sistema flexível de alterar e limitar prioridades de interrupções. A mudança de contexto no Computador MITRA-15 atinge o mais alto grau de sofisticação. Neste computador encontramos também um largo emprego de memórias rápidas. Bastante complexo é o sistema de interrupções do ARGUS 700 que possui também um registro de "máscara" de interrupções e um variado número de "modos" de funcionamento do processador.

Tudo indica que, com a rápida evolução da tecnologia de circuitos, os custos decrescem cada vez mais encorajando a construção de sistemas de interrupção cada vez mais potentes e sofisticados. São procurados os melhores tempos de resposta e o menor "overhead", a um menor custo.

Para um computador como o G-10, que é considerado como estando no nível dos PDP-11 em muitas das suas características,

seria desejável dotá-lo, nas versões industrializáveis, de um sistema de interrupções que incorporasse o melhor até agora usado nos demais computadores da sua classe, mas já com vistas ao futuro próximo da tecnologia do "hardware".

Assim procurou-se adotar a "máscara" de interrupções como o melhor e mais flexível meio de alterar o sistema de prioridades de interrupções. Também para a mudança de contexto foi adotado um processo inspirado na interrupção rápida do computador MITRA 15, o qual permite um alto desempenho. No ARGUS 700, projetado para sistemas em tempo real, encontramos uma instrução de "retorno de interrupção", que em sistemas de alta demanda de interrupções externas, permite evitar a saturação. Também no ARGUS 700S encontramos 3 "blocos de contexto", respectivamente para o modo corrente, interrupções de nível 2 e modo "0", onde se manifesta a tendência ao uso das memórias rápidas.

Pode-se assim prever que a implementação do sistema proposto conjugando os melhores conceitos já usados em outras máquinas para conduzir a um sistema de interrupções com características de alto desempenho, adaptando-se todavia estes conceitos às tendências da tecnologia atual.

BIBLIOGRAFIA

- [1] Lorin, Harold - "Parallelism in Hardware and Software: Real and Apparent Concurrency", Prentice-Hall, Inc., Englewood Cliffs , New Jersey, 1972.
- [2] Beizer, Boris - "The Architecture and Engineering of Digital Computer Complexes", Plenum Press, New York, 1971.
- [3] Bell, Gordon and Newell, Allen - "Computer Structures: Readings and Examples", Mc Graw-Hill, Inc., 1971.
- [4] Langdon Jr., Glen George e Fregni, Edson - "Projeto de Computadores Digitais", Editora Edgard Blucher Ltda., 1974.
- [5] General Automation, Inc. - "The Value of Power", Adam Osborn & Associates, Inc., 1972.
- [6] Holland, Ed. - "Minicomputer I/O and Peripherals", IEEE Comp. Group News, vol. 3, pp. 10-14, July/Aug., 1970.
- [7] Rinder, Robert - "The Input/Output Architecture of Minicomputers", Datamation, vol. , pp.119-124, May, 1970.
- [8] Vachon, Bradstreet and Weiske Jr., William - "Interfacing: A balancing act of hardware and software", Electronic Design, 11, pp. 58-63, May 27, 1971.
- [9] Di Mauro, Jim - "Input-Output: The basic options", EE/Systems Engineering Today, pp. 98-88, May, 1973.
- [10] Kocher, Christopher P. - "Interrupt driven I/O", EE/Systems Engineering Today, pp. 100 e 101, May, 1973.
- [11] Digital Equipment Corporation - "PDP-11/05-10-35-40 - Processor Handbook", 1973.
- [12] Digital Equipment Corporation - "PDP-11/45 - Processor Handbook", 1973.

- [13] Hewlett Packard Company - "A Pocket Guide to HP Computers" ,
Cupertino, California.
- [14] Varian Data Machines - "Varian 620/f - Reference Handbook", Ir
vine, California.
- [15] Van Gelder, M.K. and England, A.W. - "A Primer on Priority In-
terrupt Systems", Control Engineering, vol. 16, pp. 101-105 ,
March, 1969.
- [16] Milstead, F.C.; Neely, G.L. and Hall, J.L. - "Get the facts
behind the Mini Specs", Electronic Design, , April 29,
1971.
- [17] Fujitsu Ltd. - "FACOM U-200 - Hardware Manual", Tokyo.
- [18] Data General Corporation - "How to Use the Nova Computers",
Southbors, Massachussets, Febr., 1974.
- [19] Dat, M. et Sagnes, C. - "Prise en compte des interruptions -
Etuds de: 90-10, MITRA-15 e 10010", Institute National des
Science Appliquees, Departement de Genie Electrique, France,
1974.
- [20] CII - Compagnie Internationale pour L'Informatique - "MITRA-15-
Manuel de Presentation", 1972.
- [21] Ferranti Limited - "ARGUS 700 - System Reference Manual" -
Wythenshawe Division, Manchester, England, October, 1973.
- [22] Projeto Guarany's - "Computador G-10 - Projeto da Lógica - RF-H-
-02", USP, 1974.