

UFPb/CCT

SISTEMAS I

TEXTO: SISTEMATIZAÇÃO DE ROTINAS ADMINISTRATIVAS

AUTORA: MARIA SILVIA OLIVEIRA

ORIENTADOR: MARCOS BLAUTH



Biblioteca Setorial do CDSA. Abril de 2021.

Sumé - PB

## I. INTRODUÇÃO

Este trabalho tem como objetivo apresentar a Metodologia de Sistematização de Rotinas. Através de um procedimento bem conhecido por todos vocês e muito simples; Apresenta também a representação deste procedimento conforme a convenção ASME<sup>1</sup> e em uma outra representação<sup>2</sup> que, com o passar dos tempos, se torna mais e mais difundida. Finalmente, apresentaremos algumas questões para serem respondidas.

(1) American Society of Mechanical Engineers.

(2) Chamaremos a esta "Convenção Derivada do HIPO".

## II. DESCRIÇÃO DOS PROCEDIMENTOS DA ROTINA DE MATRÍCULA

Para fazer a matrícula, o aluno deve apresentar a carteira de estudante ao funcionário da Coordenadoria, este verifica se o envelope deste aluno está junto aos demais; neste caso o funcionário da Coordenadoria verifica se há algum aviso de débito, caso contrário, o funcionário da Coordenadoria obtém informações tais como: Nome, Período, Curso (atual e anterior, se houver), situação (transferido, reiniciante ou calouro), e providencia o envelope.

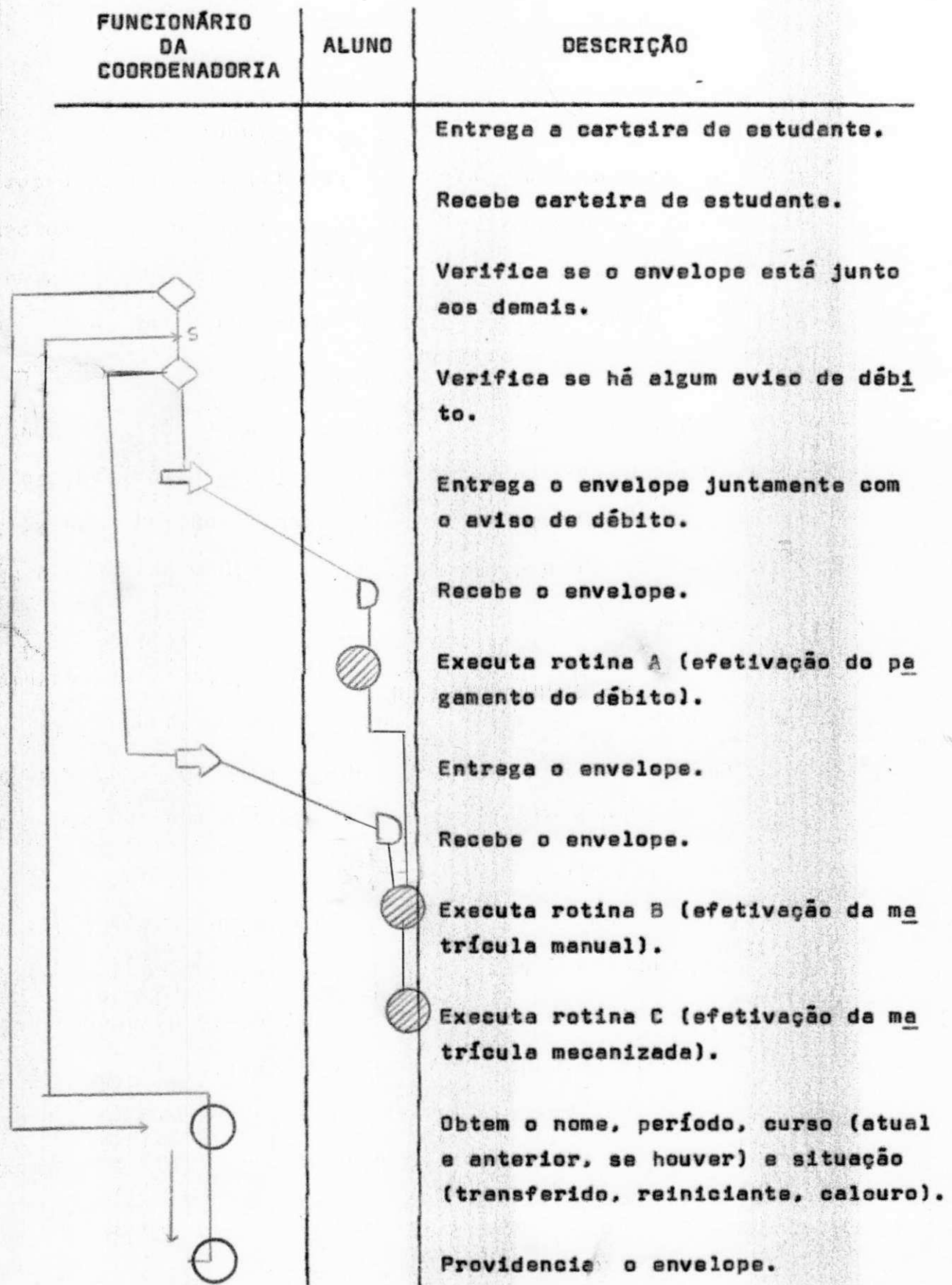
Se houver algum aviso de débito, o funcionário da Coordenadoria entrega o envelope juntamente com o aviso de débito ao aluno, e este efetua a quitação com o funcionário da contabilidade, o qual rubrica o aviso de débito e o devolve para o aluno.

Desde que o débito foi pago, o aluno preenche o PSA (que está no envelope), em 2 vias, efetua a quitação do RETAM (que se encontra no envelope), e os entrega ao funcionário da Coordenadoria, para que este assine as 2 vias do PSA, arquive a 1ª via e entregue a 2ª via para o aluno.

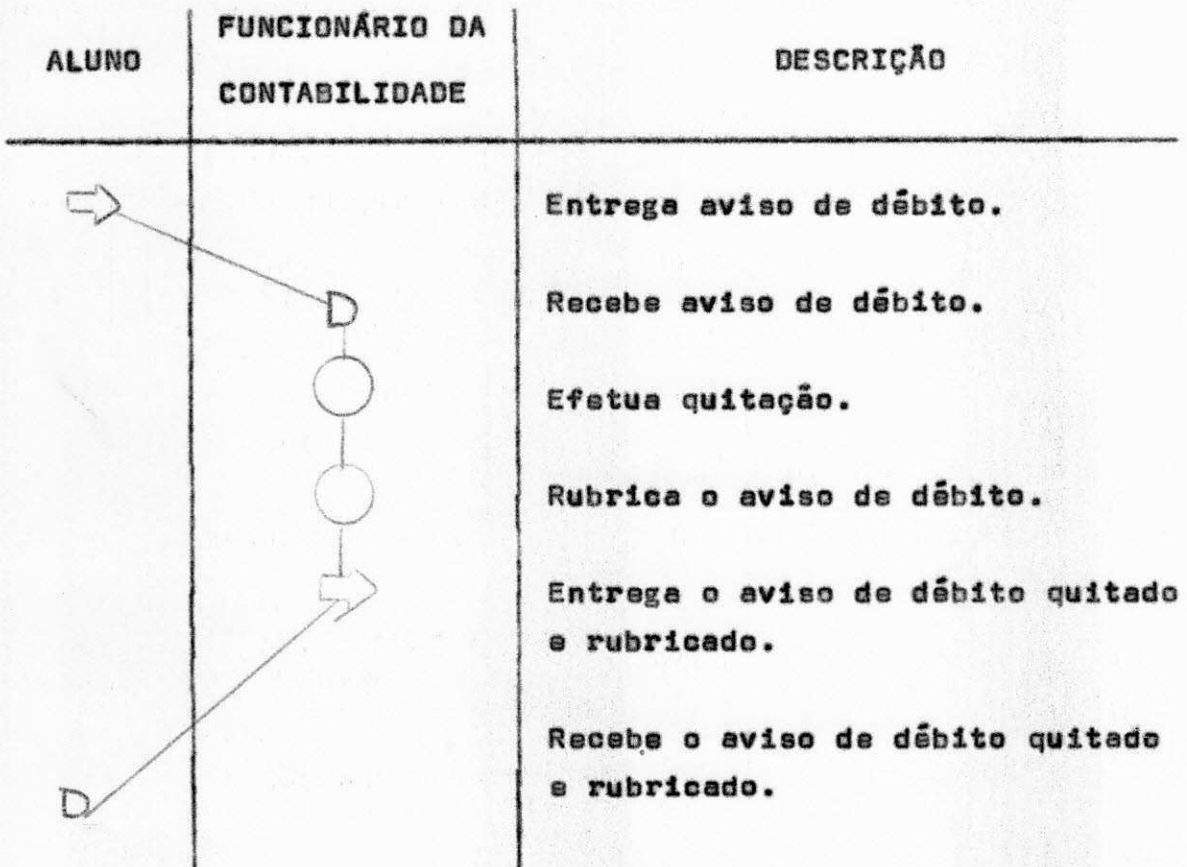
O funcionário da Coordenadoria envia a 1ª via do PSA para o perfurador perfurar os dados, que serão entregues ao operador, que obtém a RDM (em 2 vias), rodando o programa com os dados. Envia as 2 vias ao funcionário da Coordenadoria, que arquiva a 2ª via e entrega a 1ª ao aluno.

### III. DIAGRAMAÇÃO ASME

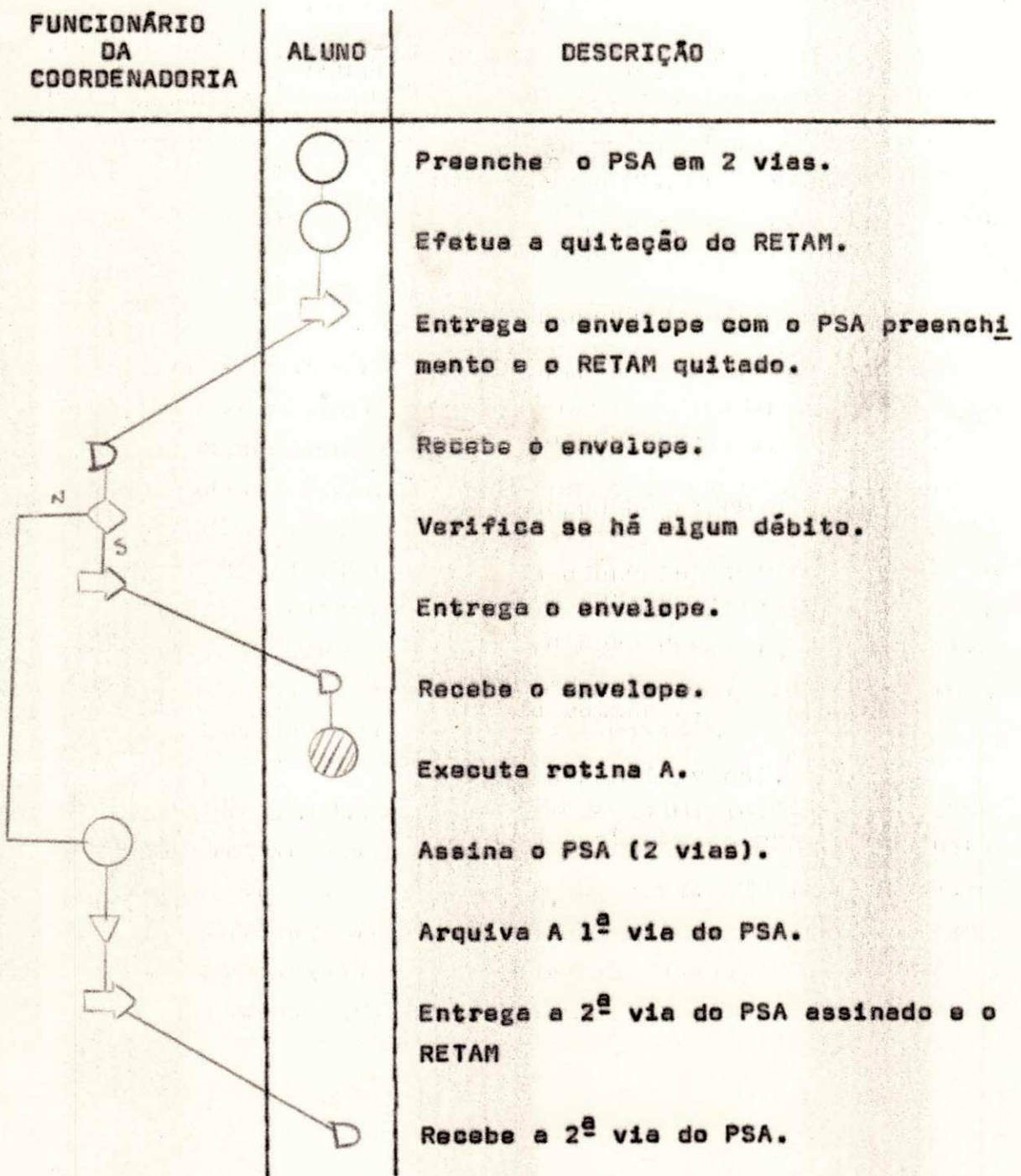
III. 1 - ROTINA PRINCIPAL



**III. 2 - ROTINA SECUNDÁRIA A (EFETIVAÇÃO DO PAGAMENTO DO DÉBITO).**

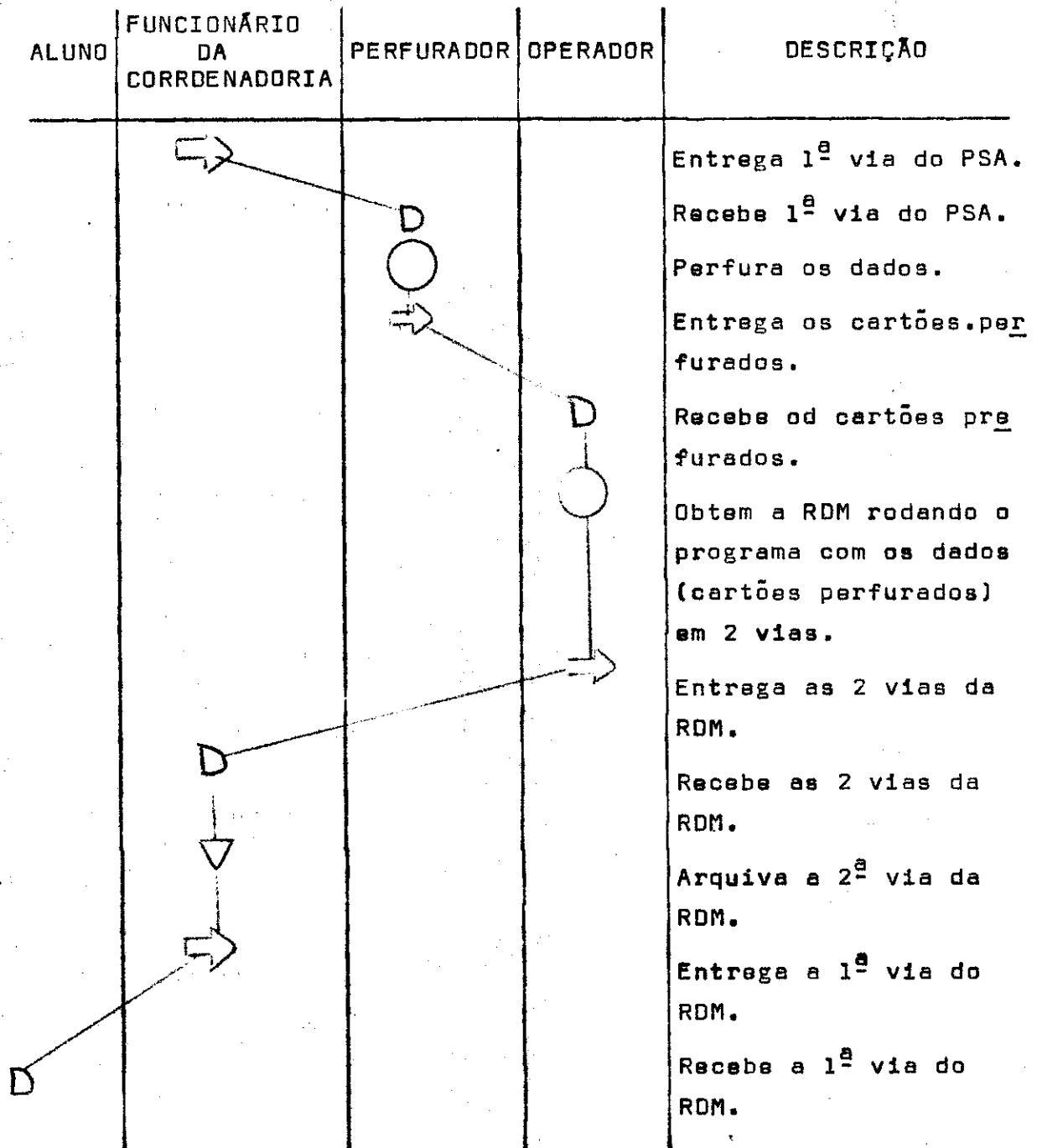


III. 3 - ROTINA SECUNDÁRIA B (EFETIVAÇÃO DA MATRÍCULA MANUAL)



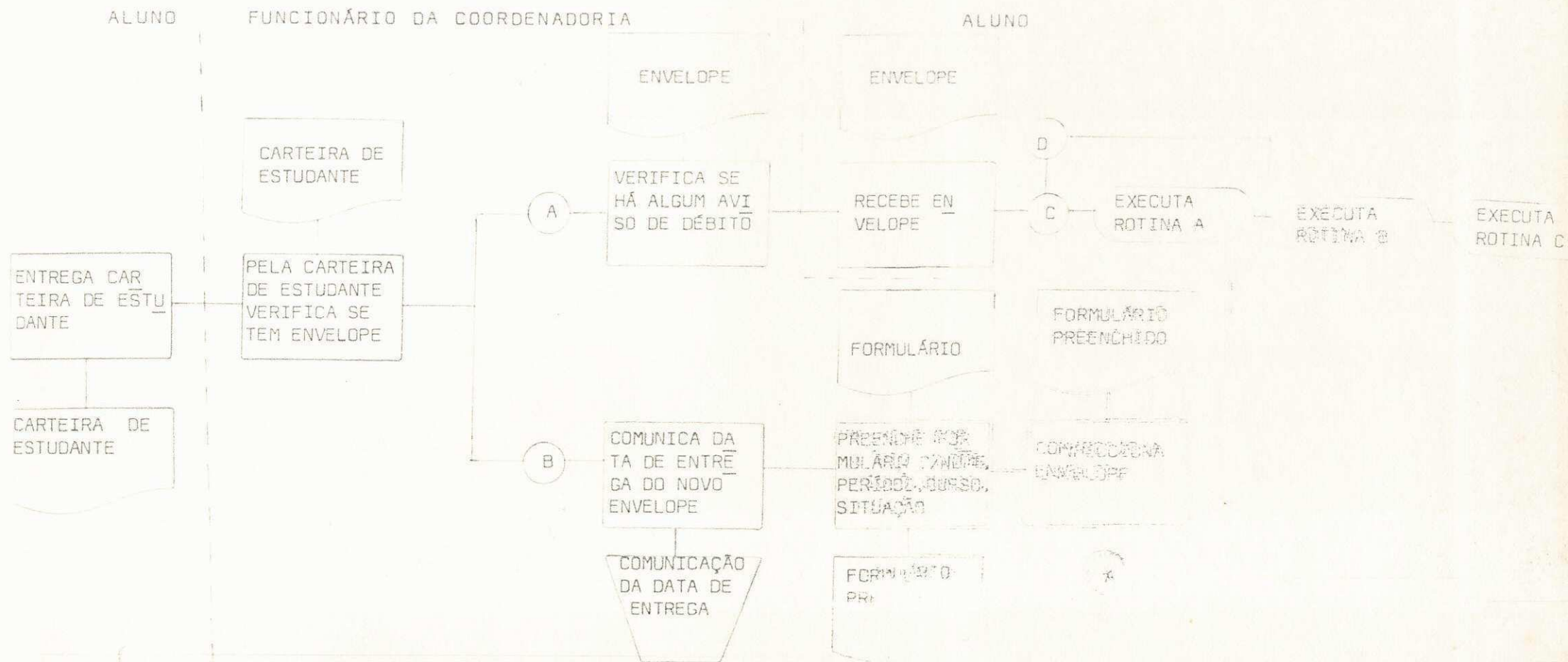


III. 4 - ROTINA SECUNDÁRIA C (EFETIVAÇÃO DE MATRÍCULA MECANIZADA)



IV. DIAGRAMAÇÃO DERIVADA DO HIPO

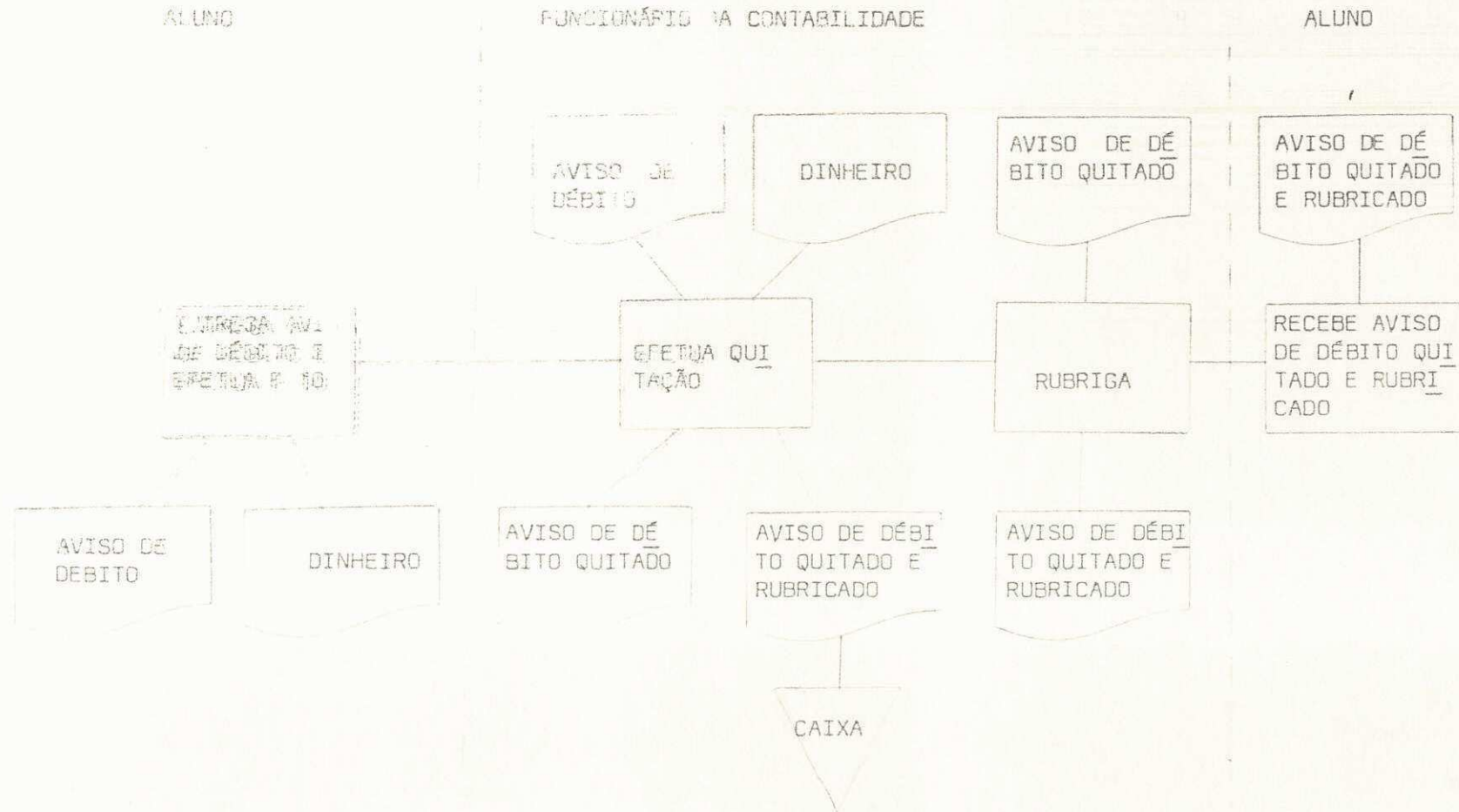
IV. 1 - Rotina Principal



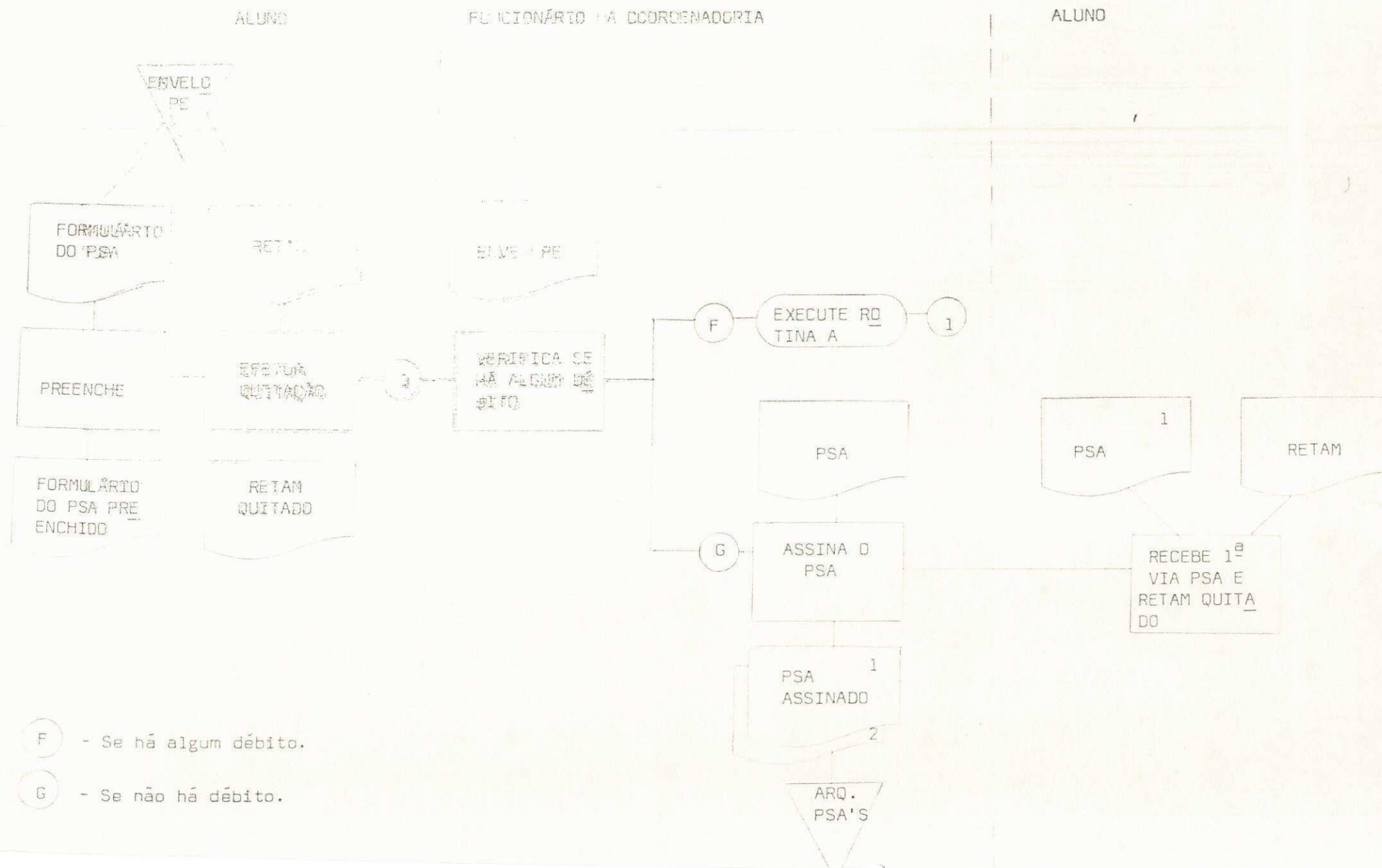
- (A) - Se o envelope está junto aos demais.
- (B) - Se o envelope não está junto aos demais.
- (C) - Se há algum aviso de débito.
- (C) - Se não há aviso de débito.

ROTINA A - EFETIVAÇÃO DO PAGAMENTO DE DÉBITO.  
 ROTINA B - EFETIVAÇÃO DA MATRÍCULA MANUAL  
 ROTINA C - EFETIVAÇÃO DA MATRÍCULA MECANIZADA

IV. 2 - ROTINA SECUNDARIA A (EFETIVAÇÃO DO PAGAMENTO DO DÉBITO)



TV. 3 - ROTINA SECUNDÁRIA B (EFETIVAÇÃO DA MATRÍCULA MANUAL)



IV. 4 - ROTINA SECUNDÁRIA C (ELETIVACÃO DE MATRÍCULA MECANIZADA)

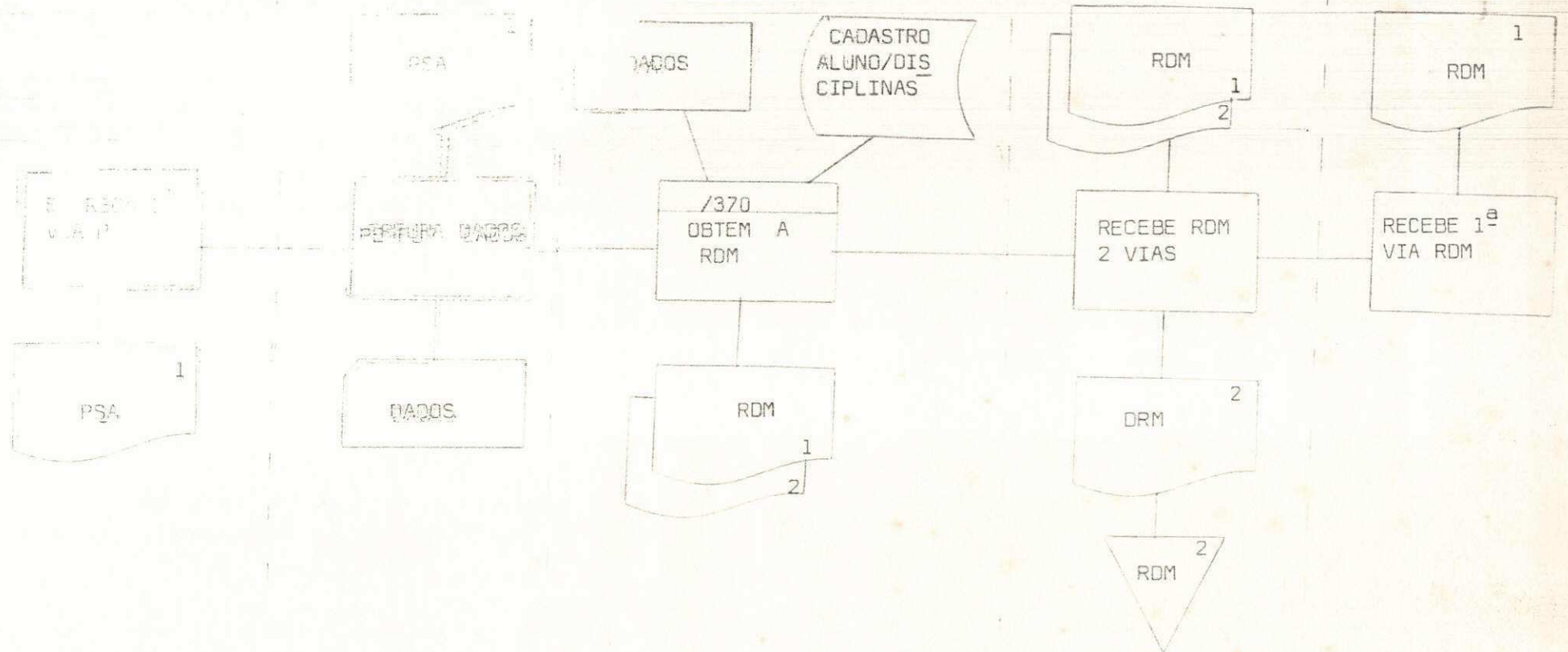
FUNCIONÁRIO DA COORDENA  
DORIA

PERFURADOR

OPERADOR

FUNCIONÁRIO DA COORDE  
NADORIA

ALUNO



## V. PERGUNTAS

- V.1 - Não olhe para os fluxos. A partir somente da descrição tente desenhar os fluxos ASME e o outro. Perguntas:
- a - A descrição está boa? Sugirá melhorias ou identifi que as frases que não estão claras.
  - b - Qual a convenção mais adequada?
  - c - Como se identifica (SEPARA) uma operação de outra ?  
O que é uma operação?
- V.2 - Qual o critério usado para a divisão deste procedimento em rotinas? Cite outros critérios e/ou exemplos que pos sam ser usados com a mesma finalidade.
- V. 3 - Toda operação tem uma entrada e uma saída! Essa afirmati va é verdadeira ou falsa? Justifique!
- V. 4 - Quais as diferenças básicas entre a representação ASME e a outra representação apresentada? comentar vanta gens e desvantagens de uma em relação a outra.

UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO  
CURSO DE PROCESSAMENTO DE DADOS  
DISCIPLINA; ANÁLISE E PROJETO DE SISTEMAS II

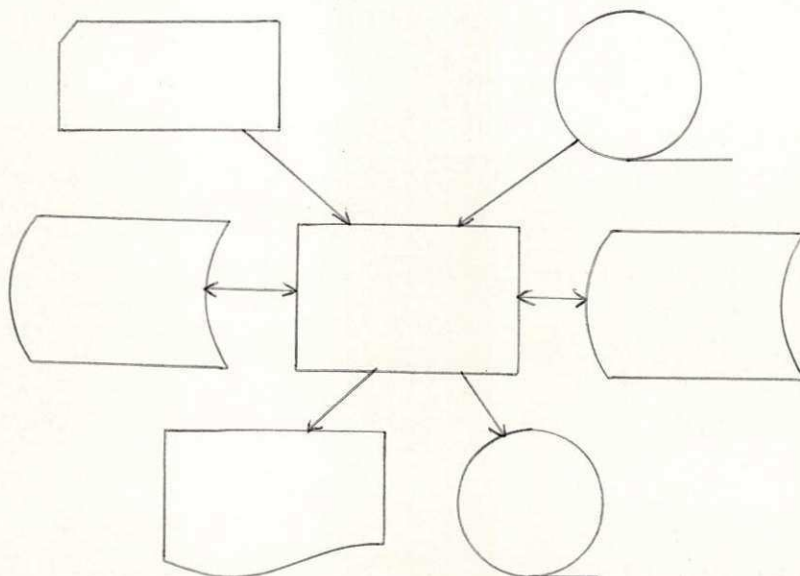
DIAGRAMAÇÃO DE SISTEMAS

ELABORAÇÃO: MARIA SILVIA DE OLIVEIRA  
ORIENTAÇÃO: MARCOS BLAUTH

## DIAGRAMAÇÃO DE SISTEMAS

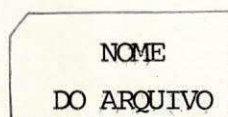
Qualquer sistema de Processamento de Dados pode ser caracterizado por uma hierarquia que existe, genericamente, entre entrada, processamento e saída.

As operações (Processamento) devem ser representadas por um retângulo no centro do desenho, com as entradas geralmente acima ou à esquerda deste, e as saídas abaixo ou à direita. A representação de arquivo de entrada ou saída segue a mesma convenção.

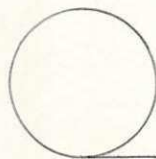


- Como representar um Programa Genérico.

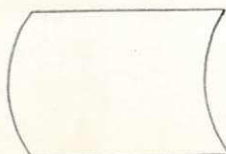
Notação:



= Arquivo em Cartão

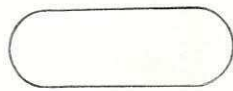


= Arquivo em Fita.

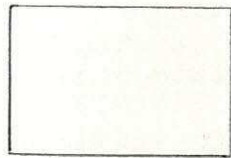


= Arquivo em Disco.





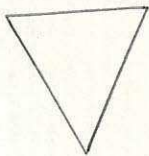
= Início ou Fim.



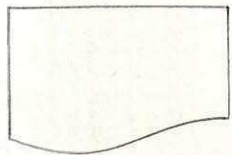
= Processamento.



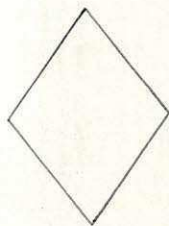
= Conexão (mesma página)



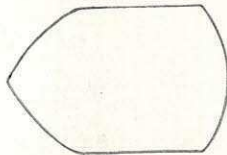
= Merge.



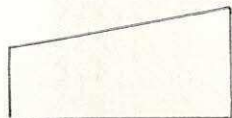
= Relatório.



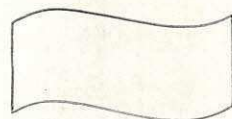
= SORT



= DISPLAY



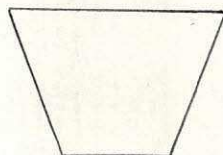
= Console



= Fita de Papel

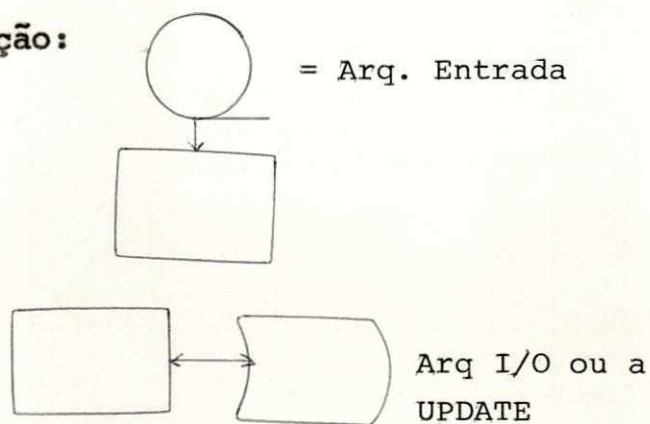


= Conexão (outra página)



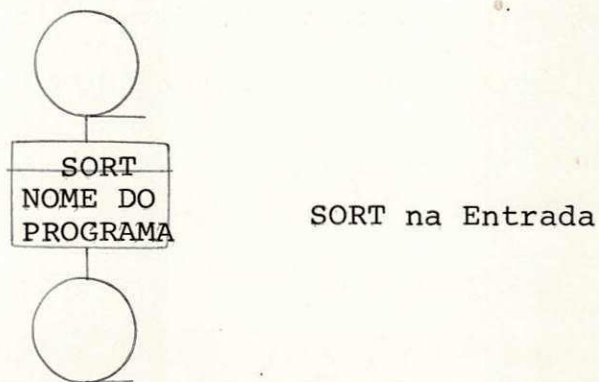
= Operação Manual ou Oral.

Formas de Utilização:



Como representar o SORT (INTERNO)

O SORT pode ser na entrada e/ou na saída



O SORT na entrada é aquele que classifica os registros a medida em que estes são lidos. O Programa propriamente dito (Procedimentos Codificados pelo Programador) é ativado na "OUTPUT-PROCEDURE" ao SORT<sup>1</sup>.



(1) - A este respeito leia o capítulo 12 do livro Programação Cobol, de Alex Bastos.

O SORT na saída é aquele que classifica os registros para que sejam impressos ou gravados classificados numa certa ordem, após os procedimentos do programa, os quais, neste caso, estão dentro da "INPUT-PROCEDURE" do SORT<sup>2</sup>.

(2) - A este respeito, leia o capítulo 12 do livro Programação Cobol, de Alex Bastos.

UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO  
CURSO DE PROCESSAMENTO DE DADOS  
DISCIPLINA: ANÁLISE E PROJETO DE SISTEMAS II

AUDITORIA  
DE  
SISTEMAS

AUTORA: MARIA SILVIA DE OLIVEIRA  
ORIENTADOR: MARCOS BLAUTH

## ÍNDICE

### I. INTRODUÇÃO

### II. CONCEITOS DE AUDITORIA

- a) Antonio de Loureiro Gil (1)
- b) Jorge Costa Pondé (2)
- c) Conclusão

### III. MEDIDAS DE DESEMPENHO

- a) Eficácia
- b) Eficiência
- c) Confiabilidade

### IV. MÉTODOS DE AUDITORIA

- a) Test-Desk
- b) Questionário
- c) Usando Computador

### V. REAVALIAÇÃO DOS MÉTODOS DE AUDITORIA

### VI. CONCLUSÃO E SUGESTÕES

### VII. BIBLIOGRAFIA

## I. INTRODUÇÃO

Este trabalho é produto de uma pesquisa sobre auditoria, pesquisa esta que foi realizada durante o período de estágio, por se tratar de um assunto pouco difundido, e de grande necessidade, pois a UFPb ainda não utiliza auditoria formal.

Sendo auditoria um assunto que dá margem à divergências, principalmente quanto ao seu enquadramento no ciclo de vida, apresentaremos, no item II, conceitos divergentes de 2 autores, para que possamos nos posicionar a este respeito.

À medida que o sistema se torna mais complexo, fica muito difícil analisar algumas medidas, por isto citaremos, no item III, algumas medidas de desempenho.

Como existe vários métodos de auditoria, analizaremos, no item IV, alguns destes métodos, especificando quais são os que precisam algum conhecimento sobre processamento eletrônico de dados, e quais se processam independentemente do computador.

Com o decorrer dos tempos, o esquema de auditoria pré-determinado poderá se tornar obsoleto, necessitando, assim, uma reavaliação deste; sendo assim, descreveremos, no item V, o ciclo desta reavaliação.

Finalmente, apresentaremos, no item VI algumas su gestões.

## II. CONCEITOS DE AUDITORIA

Como há divergências entre alguns autores a respeito do conceito, e em particular, o enquadramento no ciclo de vida, citaremos 2 conceitos que se divergem.

### a) ANTONIO DE LOUREIRO GIL (1)

Diz que a auditoria deverá ser realizada após o sistema estar em funcionamento, sendo reavaliada periódica e regularmente.

Alguns trechos do autor, que confirmam o que citamos acima:

..... Avaliar o grau de controle interno, tanto do CPD, quanto dos sistemas quanto em OPERAÇÃO a nível do CPD.

..... Verifica diretamente as informações CADASTRAIS.

..... Um sistema e seu controle estão OPERANDO de acordo com a documentação do sistema que foi apresentada.

..... Integração do auditor com os programas de computador em OPERAÇÃO.

(1) - Artigo Auditoria/Informática: Uma conjunção e novos rumos, escrito pelo citado autor. Boletins informativo da CAPRE, Vol 4, nº 4, Vol 5, nº 1. Pág. 40-42 out/nov 76, jan/mar 77.

b) JORGE COSTA PONDE (2)

Acha que a auditoria deve acompanhar todo o ciclo de vida do sistema. Nas palavras do autor:

"Exame analítico e pericial que acompanha todo o ciclo de vida de um sistema de PD desde a definição do projeto funcional até seus últimos estágios, expressos na obtenção das apurações ou do serviço para o cliente".

c) CONCLUSÃO

Analizando os 2 conceitos citados, chegamos à conclusão que um enquadra a auditoria após as operações e outro acha que a auditoria deve acompanhar todo o ciclo de vida.

Vale salientar que a auditoria tem mais sentido após o sistema estar pronto pois:

- a) É a partir disto que as falhas vão surgindo;
- b) Além disso, na elaboração do sistema há uma intensa e direta integração entre o usuário e o analista, não necessitando portanto de auditoria.

(2) - Boletim Informativo da CAPRE, Vol. 4, nº 2, Pág. 14-21, abr/jun. 1976.



### III. MEDIDAS DE DESEMPENHO

Para dar uma idéia citaremos variáveis que podem ser medidas, a fim de analisar o desempenho do sistema:

#### a) Eficácia

O importante é que o sistema funcione, conforme a necessidade do usuário, não levando em conta a otimização dos custos.

Por Exemplo:

Não há a preocupação se todos os relatórios são realmente necessários, se o número de vias estão sobrando, etc.

Algumas Medidas de Eficácia.

- Os documentos de entrada são fáceis de preencher?
- Os relatórios são convenientes? (LAY-OUT, sumarização, etc).
- A documentação
  - Completa?
  - Atualizada?
  - Padronizada?
  - Possui clareza nos procedimentos?
  - Legibilidade?
  - Está organizada?
- Os prazos são cumpridos?
- Funções são executadas a contento?

## b) Eficiência

Além do sistema estar funcionando, o CPD procura otimizar os custos, tais como: recursos humanos, relatórios, uso de equipamento, etc.

Algumas medidas de eficiência:

- Os recursos humanos estão sendo bem utilizados?
- Todos os relatórios são realmente necessários ?
- O equipamento está bem utilizado? (otimização de uso)

## c) Confiabilidade

É uma das medidas de desempenho de grande importância.

Algumas medidas de confiabilidade:

- Os controles são eficientes?

Sobre eficiência do dígito verificador, depende do método escolhido.

Por exemplo: O MÓD. 11 é mais confiável do que o MÓD.

10.

- É efetivada perfuração com verificação?

Se for com verificação é mais confiável do que sem verificação.

- Possui BACK-UP?

Se o sistema possui BACK-UP sistemáticos dos principais arquivos, ele é mais confiável, pois se os arquivos deste

sistema forem danificados existe uma outra cópia de reserva.

- Recuperação do arquivo

Existem as cópias avô e pai? se existem o sistema é mais confiável.

- Totais de controle

É associado a cada grupo de movimentos um mestre de lote que contenha totais de controles? Estes totais de controle são convenientes?

A confiabilidade do sistema depende da escolha destes totais.

Citaremos a seguir alguns totais de controle:

- Número de registros de um arquivo;
- Soma do saldo de cada um dos correntistas de uma agência bancária.

#### IV. MÉTODOS DE AUDITORIA

Citaremos a seguir métodos de medir as "MEDIDAS".

##### a) Test-Desk

É um método de auditoria, em que o auditor identifica a natureza, o formato e o dispositivo, em que os dados estão armazenados, através da documentação do sistema, a fim de testar os procedimentos contidos nos programas componentes deste sistema.

Como podemos verificar o auditor deve entender sobre processamento eletrônico de dados.

Verificamos também que a documentação dos sistemas é de grande importância para a aplicação desta técnica, pois através dela que nós compreendemos os procedimentos para podermos testá-los.

Citaremos a seguir a documentação necessária para a aplicação desta técnica:

- Manual do sistema de informação.

Identifica natureza dos sistemas, os programas componentes, os arquivos-cadastros utilizados, os subsistemas componentes e as necessidades e restrições em termos globais dos sistemas;

- Manual do programa.

Descreve o detalhamento de entrada, processamento e

saída e identifica os recursos necessários e as peculiaridades de cada programa;

- Manual de operação.

Descreve de maneira clara e sucinta a tarefa do operador;

- Manual do usuário.

Descreve os procedimentos de interligação do Departamento Usuário com o CPD, tanto quanto de recebimento da informação do CPD.

As técnicas de aplicação do TEST-DESK podem ser apresentadas, segundo a seguinte sistemática:

- Identificação do sistema a ser auditado;
- Compreensão do sistema, pela análise de sua documentação;
- Determinação das rotinas dos programas do sistema que serão avaliados;
- Preparo dos dados para o teste das rotinas dos programas considerados passíveis de verificação;
- Transcrição dos dados de teste para uma forma e um meio físico aceitos pelo computador eletrônico;
- Utilização dos programas de computador componentes do sistema de informação para processamento dos dados testes;
- Análise das informações obtidas a partir dos dados de teste e avaliação da eficiência e do funcionamento das rotinas dos programas determinados quando na análise da documentação do sistema;
- Avaliação do grau de controle interno do sistema.

## b) Questionários

Para aplicar este método, o auditor não precisa entender sobre Processamento Eletrônico de Dados; Apresentaremos a seguir 3 tipos de questionários que podem ser aplicados aos seguintes órgãos:

- a) Departamento Usuário
- b) Centro de Processamento de Dados
- c) Setor de Sistema de Informação

### Objetivos dos Questionários:

O questionário que será aplicado ao Departamento usuário tem como finalidade avaliar o grau de organização, estruturação e atribuições de cada Departamento que trabalha tarefa de um determinado sistema de informação. (Ex: Departamento de Pessoal, Departamento de Orçamento, etc.).

O questionário que será aplicado ao Centro de Processamento de Dados tem como finalidade verificar a organização do Centro de Processamento de Dados à semelhança dos procedimentos auditoriais aplicados aos Departamentos Usuários.

O questionário que será aplicado ao Setor de Sistema de informação tem como finalidade avaliar o controle interno de um dado sistema de informação desde o nascimento de um dado contábil, quer seja em um Departamento Usuário, quer seja no Centro de Processamento de Dados, até a elaboração e utilização da informação final.

Algumas perguntas referentes ao questionário de avaliação do DEPARTAMENTO USUÁRIO seriam:

- 1) Existe um Manual de Procedimentos atualizado em uso pelo Departamento Usuário?
- 2) Existe um Organograma e em uso pelo Departamento Usuário?
- 3) São as instalações físicas adequadas ao bom funcionamento do Departamento Usuário?
- 4) Estão os componentes do Departamento Usuário devidamente treinados para seus cargos?
- 5) São as tarefas do Departamento Usuário homogêneas e consistentes com os objetivos a serem atingidos por esses Departamentos?

Analisando as perguntas acima apresentadas, podemos verificar que a finalidade destas perguntas é "medir" o controle interno gerencial.

As perguntas referentes ao questionário de Avaliação do Centro de Processamento de Dados.

Este Departamento também estaria sujeito às perguntas apresentadas ao Departamento Usuário; entretanto o computador eletrônico exige um conjunto de perguntas mais amplo para o questionário do Centro de Processamento de Dados.

Citaremos a seguir algumas perguntas referentes ao questionário de avaliação do Centro de Processamento de Dados:

- 1) O Centro de Processamento de Dados é independente de todas as Unidades Operacionais das quais ele recebe as informações a serem processadas?
- 2) O acesso à sala de operação do computador é limitado às pessoas que lá permanecem por necessidade profissional?
- 3) As cópias de programas devidamente aprovadas e os documentos de suporte mantidos em biblioteca própria são entregues aos interessados apenas após autorização por escrito?
- 4) Os empregados das unidades de programação e análise são proibidos de operar os computadores, em serviços regulares?
- 5) As seguintes unidades de trabalho são segregadas tanto física como organizacionalmente?
  - i. Operação do computador
  - ii. Unidades de controle
  - iii. Biblioteca de programas e arquivos de informações
  - iv. Unidades de análise e programação
- 6) As técnicas de programação são padronizadas quanto a:
  - i. Testes de mesa?
  - ii. Uso de chaves de programa?



iii. Rotinas de início?

iv. Endereçamento?

7) O hardware do computador é revisado por técnicos qualificados em uma base regular?

8) Existe uma política estabelecida com referência a visitantes, limpeza, permissão ou não de fumar, etc., na sala dos computadores?

Algumas perguntas do Questionário para avaliação de um Sistema de Informação como um todo, no caso de um específico Sistema de Contas a Pagar, seriam:

1) Há um sistema adequado mantido para:

i. Requisição para compras ou contratação de serviços?

ii. Pedidos de compra?

iii. Aprovação para compras ou serviços?

iv. Aprovação para pagamento?

2) São mantidas contas individuais (razão auxiliar), para controle das contas a pagar a fornecedores e credores, de natureza diversa, e de despesas a pagar (impostos, encargos, previdências, etc.)?

3) As pessoas que escrituram e mantêm os registros analíticos de controle são independentes de:

- i. Funções de caixa?
  - ii. Funções de manuseio de comprovantes?
  - iii. Funções de lançamento no razão geral?
- 4) As contas individuais são periodicamente balanceadas e reconciliadas com a conta de controle do razão?
- 5) As faturas de fornecedores ou de prestadores de serviços somente são registradas mediante verificação prévia de sua correção?
- 6) Os extratos são obtidos de contas periódicas dos credores (ou confirmação de saldos)?
- 7) São devidamente investigados e prontamente regularizados:
- i. Saldos devedores em contas a pagar?
  - ii. Contas com prazos de vencimento expirados?
  - iii. Saldos pendentes de pagamento há longa data?
  - iv. Saldos de valores insignificantes?
- 8) O sistema de controles oferece margem de segurança para que todos os passivos da companhia (contas ou despesas a pagar) sejam registradas no exercício de competência?
- 9) As contas em moedas estrangeiras são periodicamente ajustadas às novas taxas de câmbio em vigor?

- 10) São as alterações dos programas do Sistema de Contas a Pagar revisadas e aprovadas por pessoas com autoridade além dos programadores que fizeram esses programas?
- 11) São essas alterações documentadas quanto a:
  - i. Razões da mudança?
  - ii. Efeitos da mudança?
  - iii. Ajustes necessários nos dados já acumulados?
- 12) Os documentos são de entrada do sistema de contas a pagar no Centro de Processamento de Dados, pré-numerados tipograficamente?
- 13) Os dados são processados em lote controlados numericamente?
- 14) As instruções aos perfuradores de cartão ou fitas são escritas de forma clara e concisa?
- 15) Todas as instruções aos operadores são fornecidas por escrito de forma clara e em linguagem que não deixe margem a dúvidas?
- 16) Existem check-points no processamento do contas a pagar e os operadores são devidamente instruídos quanto a seu uso?
- 17) Os arquivos-mestres do contas a pagar contêm etiquetas externas, header-labels e trailer-labels?

18) As etiquetas externas contêm:

- i. Número do carretel ou do disco?
- ii. Número de carretéis ou de discos do arquivo?
- iii. Número de série?
- iv. Número de identificação do programa?
- v. Data de elaboração do arquivo?
- vi. Data da gravação?
- vii. Densidade de gravação?

19) Os header-labels contêm:

- i. Número de identificação do programa?
- ii. Número do carretel ou do disco?
- iii. Data da elaboração do arquivo?
- iv. Data em que o arquivo foi considerado obsoleto?

20) Os trailer-labels contêm:

- i. Contagem de registro?
- ii. Hash-total?
- iii. Totais de valores?

c) Usando Computador

Para aplicar este método, o auditor deve entender so bre processamento eletrônico de dados, pois ele vai utilizar programas de computador para realizar a auditoria do sistema, que testa os dados dos arquivos do sistema.

A seguir citaremos alguns condicionantes para elaborar um programa para auditoria:

- LAY-OUT do registro a ser analisado;
- Fator de bloco do arquivo;
- Identificação dos campos do registro que serão analisados;
- Tipo de função a ser desempenhada;
- Tipos da saída desejadas.

Citaremos a seguir 2 tipos básicos de programas para auditoria:

- a) Programa de Análise de Arquivo
- b) Programa de Confronto de Arquivo

A seguir descreveremos as tarefas que os programas acima citados desempenham:

a) Programa de Análise de Arquivo executa as seguintes tarefas:

- Testa a validade de campos
    - . Numéricos
    - . Alfanuméricos
    - . Datas (01 < dia < 28, 29, 30, 31;  
01 < mês < 12;  
ano conforme condições conhecidas)
    - . Valores e quantidades entre limites conhecidos
  - Faz consistência cruzada
- Regra Geral: Dado que D1 verifica C1, então D2 deve verificar C2

D1, D2 = Conjunto de Dados

C1, C2 = Condições a Verificar

Ferramenta Usual; Tabela de decisão, otimização pe  
la negação, se possível ou ou  
tros meios.

b) Programa de Confronto de Arquivos:

Este programa analisa dois arquivos, A e B, um contra o outro, apresentando:

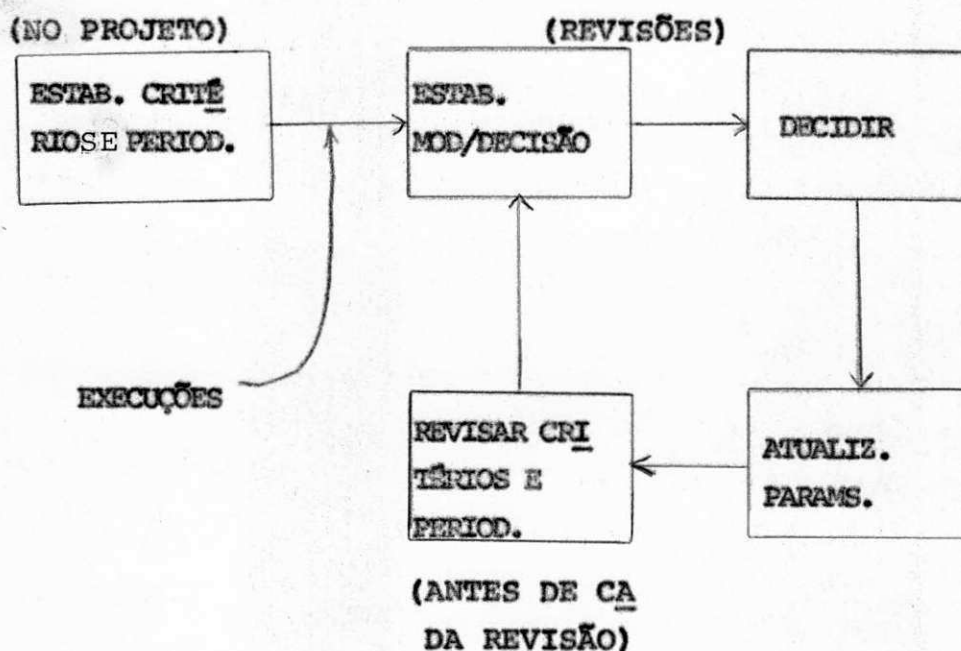
- Os registros do arquivo A que não estão no arquivo B;
- Os registros do arquivo B que não estão no arquivo A;
- Os registros que estão em ambos.

Para melhor caracterização, poderíamos considerar o arquivo A, como o cadastro-mestre que contém a descrição e quantidade do que há no estoque do almoxarifado, e o arquivo B contém os pedidos de materiais de cada Departamento, o PROGRAMA DE CONFRONTO verificaria se o que está sendo pedido (arquivo B) existe no almoxarifado (arquivo A), dependendo disto o pedido seria atendido ou não.

Uma outra aplicação deste programa seria na atualização de um arquivo, para fazer inclusões, alterações ou deletar algum registro. Com este programa descobriríamos se há inclusões de registros, registros estes que já existam no arquivo mestre, e alteração de registro inexistente.

## V. REAVALIAÇÃO DOS MÉTODOS DE AUDITORIA

Após um certo tempo, o esquema de auditoria pré-determinado poderá se tornar obsoleto, necessitando portanto de uma reavaliação; apresentaremos o esquema desta reavaliação a seguir:



## VI. CONCLUSÃO E SUGESTÕES

Auditoria é um conceito amplo, que pode abranger desde a simplicidade de contatos informais e periódicos entre o analista (ou auditor) e o usuário até complexos e extensos questionários e/ou modelos de decisão.

Quanto à utilização de auditoria sistemática, tem sido principalmente em empresas de grande porte e, em particular, as multinacionais. Na UFPb não existe ainda nenhum plano formal e pré-determinado para auditoria de qualquer sistema de informação. Fica, portanto, a sugestão de que os técnicos do NSC/UFPb estudem as vantagens e os custos de um plano de auditoria, o qual poderá (ou deveria?) ser aplicado, senão em todos os sistemas pelo menos no sistema de contabilidade pois, como sabemos, casos já ocorreram de diferenças de saldo constatadas em várias contas, quando da implantação do referido sistema em ocasiões anteriores. Essas diferenças poderiam, em nossa opinião, ter sido evitadas se uma equipe de auditores tivesse acompanhado convenientemente a referida implantação.

## VII. BIBLIOGRAFIA

1. CAPRE; Boletim Informativo, Rio de Janeiro, RJ - Vol. 4 , Nº 4, Pág. 40-45 OUT/NOV 76, JAN/MAR 77.
2. CAPRE; Boletim Informativo, Rio de Janeiro, RJ - Vol 4, Nº 2, Pág. 14-21 ABR/JUN 1976.



3. CAPRE: Boletim Informativo, Rio de Janeiro, RJ - Vol. 2,  
Nº 4, Pág. 16-22, SET/DEZ 1974.

4. Texto escrito por Marcos Blauth sobre Auditoria de Sistemas.

jps/

UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO  
CURSO DE PROCESSAMENTO DE DADOS

P L A N O D E E S T Á G I O

ESTAGIÁRIA: Maria Silvia de Oliveira

ORIENTADORES: Marcos Blauth

Likiso Hattori

Gentil José de Lucena Filho

## INTRODUÇÃO

Neste Plano de Estágio apresentaremos as atividades que serão realizadas durante o Estágio Supervisionado, que será realizado no Departamento de Sistemas e Computação, do Centro de Ciências e Tecnologia, da Universidade Federal da Paraíba, a partir do dia 19 de fevereiro do corrente ano, com a duração de 480 horas.

O Estágio será dividido em 3 etapas:

Na primeira etapa, sob a orientação de Marcos Blauth, desenvolveremos um trabalho sobre Análise e Projeto de Sistemas. Nesta etapa elaboraremos um texto sobre Sistematização de Rotinas Administrativas, um outro texto sobre Diagramação de Sistemas, e finalmente elaboramos um texto sobre Auditoria.

Na segunda etapa, sob a orientação de Likiso Hattori, desenvolveremos um trabalho sobre Software. Nesta etapa elaboraremos uma lista de exercícios sobre Bases Numéricas, assim como uma Apostila sobre Arquitetura do Sistema IBM/370, ministraremos aulas sobre Bases Numéricas, Organização do Sistema IBM/370, Memória, Endereçamento de Memória do IBM/370, Registradores, Apresentação dos Dados no IBM/370, Formatos de Instruções, Funções de Controle do Sistema, Program State Word - PSW, Estado de Programa e Job Control Language - JCL.

Na terceira etapa, sob a orientação do Prof. Gentil José de Lucena Filho, desenvolveremos um trabalho sobre Automação de Processamento de Texto, usando o SCRIPT. Nesta etapa faremos um Treinamento em edição e formatação automática de texto, assim como, a preparação de macros, a fim de facilitar o uso do formatador em larga escala.

## ATIVIDADES DO ESTÁGIO

O Estágio será dividido em três etapas:

Na primeira etapa, sob a orientação de Marcos Blauth, desenvolveremos um trabalho sobre Análise e Projeto de Sistemas. Nesta etapa executaremos as seguintes tarefas:

Tarefa:

Elaborar um texto sob Sistematização de Rotinas Administrativas. Neste texto faremos a descrição dos procedimentos da Rotina de Matrícula, depois representaremos estes procedimentos conforme a convenção ASME, e numa outra convenção derivado do HIPO.

Aplicação:

Este texto será utilizado como material didático para a disciplina Análise e Projeto de Sistemas I.

Recursos Didáticos:

Manual de Matrícula

**Justificativas:**

Este trabalho será realizado a fim de melhorar e atualizar a Rotina de Matrícula, assim como, padronizar a Representação de Rotinas Administrativas.

**Tarefa:**

Elaborar um texto sobre Diagramação de Sistemas. Neste texto, mostraremos como se representa as entradas, saída e as Operações de um Programa; mostraremos também como se representa um SORT INTERNO na entrada e na saída.

**Aplicação:**

Este texto será utilizado como material didático para a Disciplina Análise e Projeto de Sistemas II.

**Recursos Didáticos:**

Livro de Programação COBOL, de Alex Bastos.

**Justificativa:**

Padronizar a Diagramação de Sistemas.

**Tarefa:**

Elaborar um texto sobre Auditoria de Sistemas. Neste

texto apresentaremos dois conceitos divergentes sobre Auditoria, apresentaremos, também, umas variáveis que possam ser medidas a fim de analisar o desempenho do Sistema, quais sejam: Eficiência, Eficiência e Confiabilidade, como podemos utilizar vários métodos para medir estas "medidas", apresentaremos 3 métodos, quais sejam: Test-Desk, Questionários e Usando o Computador; apresentaremos, também, o ciclo de Reavaliação do Sistema, pois com o decorrer dos tempos o esquema de Auditoria se torna obsoleto, necessitando portanto de uma reavaliação. Finalmente, apresentaremos algumas sugestões.

#### Aplicação:

Este texto será utilizado como material didático para a disciplina Análise e Projeto de Sistemas II.

#### Recursos Didáticos:

- CAPRE, Boletim Informativo, Rio de Janeiro, Rj -Vol 4, Nº 4, Pág. 40-45 out/nov 76, jan/mar 77.
- CAPRE, Boletim Informativo, Rio de Janeiro, Rj - Vol 4, Nº 2, Pág. 14-21 abr/jun 1976.
- CAPRE, Boletim Informativo, Rio de Janeiro, Rj - Vol 2, Nº 4, Pág. 16-22, set/dez 1974.
- Texto escrito por Marcos Blauth sobre Auditoria de Sistemas.

**Justificativa:**

Por se tratar de um assunto pouco difundido, e de grande necessidade, pois a UFPb ainda não utiliza auditoria formal, e como sabemos, casos já ocorreram de diferenças de saldo constatadas em várias contas, quando da implantação do Sistema de Contabilidade em ocasiões anteriores. Essa diferença poderiam, em nossa opinião, ter sido evitada se uma equipe de Auditores tivesse acompanhado convenientemente a referida implantação.

**Tarefa:**

Resolver um exercício destinado a disciplina Análise e Projeto de Sistemas II.

**Aplicação:**

O exercício será aplicado aos alunos da disciplina Análise e Projeto de Sistemas II

**Recursos Didáticos:**

Conhecimentos anteriores sobre o assunto.

**Justificativa:**

Servir de modelo.



Na segunda etapa, sob a orientação de Likiso Hattori, desenvolveremos um trabalho sobre Software. Nesta etapa executaremos as seguintes Tarefas:

Tarefa:

Ministrar uma aula sobre Bases Numéricas.

Aplicação:

Esta aula será ministrada aos alunos da disciplina Software Básico.

Recursos Didáticos:

Conhecimentos anteriores sobre o assunto.

Justificativa:

Por se tratar de um assunto de suma importância.

Tarefa:

Elaborar uma Lista de Exercícios sobre Bases Numéricas.

**Aplicação:**

Será distribuídas com os alunos da disciplina Software Básico.

**Justificativa:**

Necessidade de exercícios para que os alunos possam aplicar aquilo que aprendeu.

**Recursos Didáticos:**

Conhecimentos anteriores sobre o assunto.

**Tarefa:**

Elaborar uma Apostilha sobre Arquitetura do Sistema IBM/370.

**Aplicação:**

Esta Apostila será utilizada como material didático para a disciplina Software Básico.

**Recursos Didáticos:**

- Apostila sobre Arquitetura do Sistema IBM/370, elaborada por Likiso Hattori.

- Assembler Language Programming: The IBM System/360-  
George Struble.

Justificativa:

A Apostila que estava sendo utilizada, tinha sido elaborada para alunos a nível de Pós-Graduação, necessitando, portanto refazê-la, usando um linguajar a nível de Graduação.

Tarefas:

Ministrar aulas sobre os seguintes assuntos:

- Organização do Sistema IBM/370
- Memória do IBM/370
- Endereçamento de Memória do IBM/370
- Registradores
- Representação dos Dados no IBM/370
- Formatos de Instruções
- Funções de Controle do Sistema
- Program State Word - PSW
- Estado do Programa
- Job Control Language - JCL

Aplicação:

As aulas serão ministradas aos alunos da disciplina  
Software Básico.

## Recursos Didáticos:

Apostila sobre Arquitetura do Sistema IBM/370.

Na terceira etapa, sob a orientação de Gentil José de Lucena Filho, desenvolveremos um trabalho sobre Automação de Processamento de Texto. Nesta etapa executaremos as seguintes tarefas:

### Tarefas:

a) Treinamento em edição e formatação automática de texto. |

Tarefas típicas em edição automática de texto são:

- entrada e saída de documento
- substituição de strings
- inserção e deleção de strings

Tarefas típicas em formatação automática de texto são:

- criação de parágrafos
- indentação
- manipulação de figuras
- manipulação de roda-pés
- múltiplas colunas
- numeração de páginas
- centralização de cabeçalhos
- espaçamento
- etc.

b) Elaboração de Macros, a fim de facilitar o uso do formatador em larga escala.

**Aplicação:**

Preparação de documentos tais como:

- manuais de referência
- notas de aula
- documentação de pacotes de software
- anais de conferência
- artigos técnicos

**Recursos Didáticos:**

- Text Formatting With Waterloo SCRIPT:  
A Comprehensive Primer  
Burton L. Leathers

**Justificativa:**

A grande quantidade de documentos manipulada na Universidade e empresas em geral.

UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO  
CURSO DE FORMAÇÃO DE TECNÓLOGOS EM PROCESSAMENTO DE DADOS

R E L A T Ó R I O   F I N A L

ESTAGIÁRIA: Maria Silvia de Oliveira  
ORIENTADORES: Marcos Blauth  
Likiso Hattori  
Gentil José de Lucena Filho

## ÍNDICE

1. Apresentação
2. Introdução
3. Apresentação das Etapas de Trabalho
  - 3.1 - Sistemas
    - Sistematização de Rotinas Administrativas
    - Diagramação de Sistemas
    - Auditoria de Sistemas
  - 3.2 - Software Básico
    - Arquitetura do Sistema IBM/370
    - Job Control Language - JCL
    - Bases Numéricas
  - 3.3 - Automação de Processamento de Texto
    - Edição de Texto
    - Formatação Automática de Texto
4. Conclusão
5. Considerações Gerais
6. Apêndice
  - Cópia dos Trabalhos Desenvolvidos

## 1. APRESENTAÇÃO

Este Relatório, destinado à Coordenação do Curso de Formação de Tecnólogo em Processamento de Dados do CCT/UFPb, trata das atividades desempenhadas durante o Estágio Supervisionado que foi realizado no Departamento de Sistemas e Computação, da Universidade Federal da Paraíba, a partir do dia 1º de fevereiro do corrente ano, com a carga horário de 8 horas diárias, totalizando 600 horas.

O Estágio foi feito pela aluna MARIA SILVIA DE OLIVEIRA, do curso supra citado, e, orientada pelos Professores MARCOS BLAUTH, LIKISO HATTORI e GENTIL JOSÉ DE LUCENA FILHO.

## 2. INTRODUÇÃO

Durante o período de Estágio realizado no Departamento de Sistemas e Computação tive oportunidade de desempenhar diversas atividades referentes a Sistemas, Software e Automação de Processamento de Texto.

Na parte de Sistemas, estive sob a orientação do Professor MARCOS BLAUTH, sobre este assunto desenvolvemos os seguintes tópicos:



- Sistematização de Rotinas Administrativas;
- Diagramação de Sistemas;
- Auditoria.

Sobre Sistematização de Rotinas Administrativas, tivemos oportunidade de elaborar um texto no qual descrevemos a Rotina de Matrícula, e representamos esta Rotina conforme a convenção ASME e a convenção derivada do HIPO.

Sobre Diagramação de Sistemas, elaboramos um texto que mostra como se representa as entradas, saídas e as operações de um programa; mostra também como se representa um SORT INTERNO na entrada, e na saída.

Sobre Auditoria, elaboramos um texto que apresenta:

- CONCEITO DE AUDITORIA DE:
  - a) Antonio de Loureiro Gil
  - b) Jorge Costa Pondé
  - c) Conclusão
  
- MEDIDAS DE DESEMPENHO
  - a) Eficácia
  - b) Eficiência
  - c) Confiabilidade

- MÉTODOS DE AUDITORIA

- a) Test-Desk
- b) Questionários
- c) Usando Computador

- REAVALIAÇÃO DOS MÉTODOS DE AUDITORIA

Na parte de Software, estive sob a orientação do Professor LIKISO HATTORI, referente a este assunto desenvolvemos os seguintes tópicos:

- Arquitetura do Sistema IBM/370;
- Bases Numéricas;
- Job Control Language - JCL.

Referente a Arquitetura do Sistema IBM/370, tive mos oportunidade de elaborar um texto que engloba os seguintes ítems:

- Organização do Sistema
- Memória
- Registradores
- Dados
- Formatos e Instruções
- Funções de Controle do Sistema
  - . Program Status Word (PSW)
  - . Registradores de Controle
- Desenho e Endereçamento da Memória

- . Fator de Performance
- Funções da Unidade de Processamento
- Funções de Controle e Desvio
- Organização de Canal e Entrada/Saída
- Memória Virtual

Utilizando os conhecimentos adquiridos referentes a Arquitetura do Sistema IBM/370, ministramos aulas sobre:

- Organização do Sistema IBM/370
- Memória
- Registradores
- Dados
- Formatos e Instruções
- Funções de Controle do Sistema
  - . Program Status Word (PSW)
  - . Registradores de Controle
  - . Estado de Programa

Com referência a Bases Numéricas tivemos oportunidade de ministrar uma aula, e elaborar uma lista de exercícios destinadas aos alunos de Software Básico.

Referente a Job Control Language - JLC, estudamos alguns parametros; e utilizando os conhecimentos adquiridos ministramos aulas.

Na parte de Automação de Processamento de Texto, tivemos oportunidade de fazer um treinamento em Edição de Texto, e Formatação Automática de Texto.

Em edição de texto estudamos:

- Substituição do Strings
- Inserção e Deleção de Strings

Em Formatação Automática de Texto estudamos:

- Marginação
- Criação de Parágrafos
- Indentation
- Manipulação de Figuras
- Manipulação de Roda-Pés
- Múltiplas Colunas
- Numeração de Páginas
- Centralização de Cabeçalhos
- Espacejamento

### 3. APRESENTAÇÃO DAS ETAPAS DE TRABALHO

#### 3.1 - SISTEMAS

- Sistematização de Rotinas Administrativas

Sobre este assunto elaboramos um texto no qual fizemos uma descrição dos Procedimentos da Rotina de Matrícula, descrevendo todos os passos necessários para se fazer uma Matrí

cula, deixando claro as entradas, operações e saídas. Depois representamos esta descrição conforme a convenção ASME. Na convenção ASME representamos as ações de conformidade com o que segue:

⇒ - Transporte, Envio

▷ - Demora, Recebimento

◁ - Decisão

○ - Operações

⊗ - Ligação com outra Rotina

Para representar uma Rotina conforme a convenção ASME, devemos identificar os agentes. Feito isto colocaremos em colunas as ações referentes a cada agente, fazendo a descrição desta ação na mesma linha, conforme anexo I.

Já na convenção derivada do HIPO os símbolos se assemelham aos de Processamento de Dados e deixa bem destacadas as entradas, processamento e saídas. Nesta convenção a descrição será feita dentro dos símbolos.

Os símbolos que utilizamos para representar a Rotina de Matrícula foram:



- Operação.



- Documento.



- Comunicação Oral.



- Indica numa Decisão a Mudança do Sentido do Fluxo.



- Ligação com outra Rotina.



- Arquivamento.



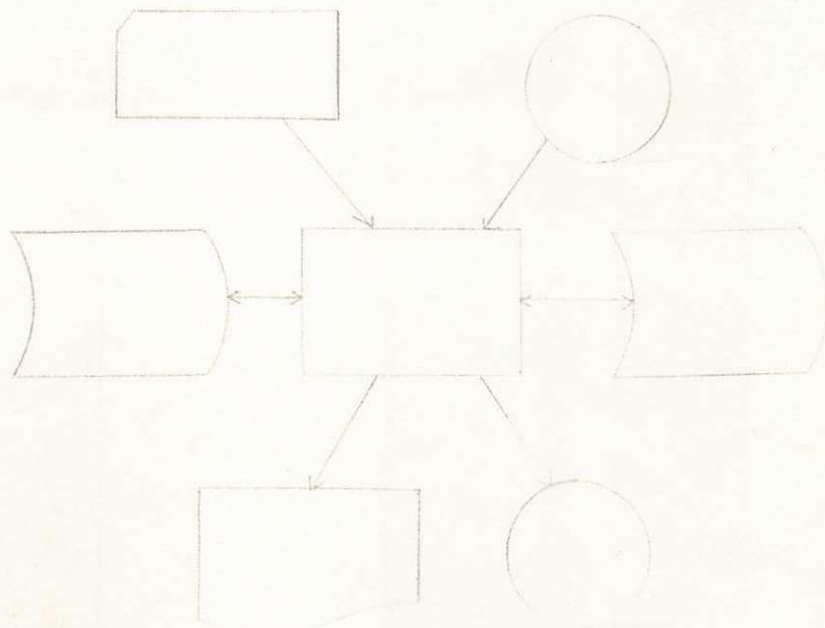
- Ligação na mesma Página.

## - Diagramação de Sistemas

Sobre este assunto, elaboramos um texto onde procuramos citar algumas regras, a fim de padronizar os diagramas de um Sistema.

O primeiro passo neste sentido foi mostrar como deveria ser representado as entradas, operações (processamentos) e saídas.

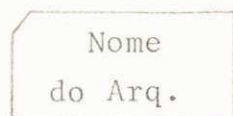
Uma das regras apresentada no texto é que um retângulo no centro do desenho, com as entradas geralmente acima ou à direita conforme a figura abaixo:



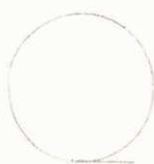
As operações ficam numa horizontal, ficando abaixo destas as saídas, e acima as entradas, conforme o modelo anexo II.



Mostramos também a notação que deve ser usada para representar um programa genérico, que é:



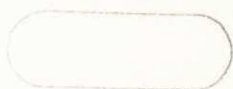
= Arquivo em cartão.



= Arquivo em fita.



= Arquivo em dispositivo de acesso direto.



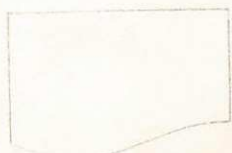
= Início ou fim.



= Processamento.



= Conexão (mesma página).



= Relatório



= SORT



= DISPLAY



= Terminal



= Fita de papel



= Conexão (outra página)



= Operação manual ou oral.

Mostraremos também as formas de utilização.



Finalmente, mostramos como se representa um SORT na entrada, e na saída.



Enfatizamos que o SORT na entrada classifica os registros a medida em que estes são lidos, ou seja, que o programa propriamente dito (procedimentos codificados pelo programador) é ativado na "OUTPUT-PROCEDURE" do SORT; assim como, o SORT na saída classifica os registros para que sejam impressos ou gravados classificados numa certa ordem, após os procedimentos do programa, os quais, neste caso, estão dentro da INPUT-PROCEDURE do SORT.



Para dar maior clareza daremos um exemplo de SORT na entrada.

Como nós sabemos há 3 fases. Uma antes de dar o SORT por exemplo:

INPUT-PROCEDURE

1. Lê um registro do arquivo

-

-

-

-

Release

Vá para 1

Exit da INPUT-PROCEDURE

SORT

SORT ARQUIVO ASCENDING KEY CHAVE

OUTPUT-PROCEDURE

1. Return um registro do SORT

Faz balanced-line

-

Programa

-

Propria-

-

mente di

-

to.

-

Grave um registro do arquivo

Vá para 1.

- Auditoria de Sistemas

Com referência ao tópico Auditoria de Sistemas tivemos oportunidade de fazer uma pesquisa e elaborar um texto que apresenta:

- CONCEITOS DE AUDITORIA: neste ítem, mostramos alguns trechos do artigo escrito por Antonio de Loureiro Gil; ele deixa claro que o enquadramento de Auditoria no ciclo de vida deve ser após o sistema estar implantado; mostramos também que na opinião de Jorge Costa Pondé a Auditoria deveria acompanhar todo o ciclo de vida. Num sub-ítem conclusão, dizemos que a Auditoria tem mais sentido após o sistema estar pronto pois:

a) É a partir disto que as falhas não surgindo;

b) Além disso, na elaboração do sistema há um intensa e direta integração entre o Usuário e o Analista, não necessitando portanto de Auditoria.

- MEDIDAS DE DESEMPENHO: neste ítem, para dar uma idéia, citamos algumas variáveis que podem ser medidas, a fim de analisar o desempenho do sistema. As variáveis mais significativas que levamos em consideração foram:

- Eficácia
- Eficiência
- Confiabilidade

No tocante a Eficácia, sabemos que o essencial é que o sistema funcione, conforme a necessidade do usuário, não levando em conta a otimização dos custos por exemplo:

Não há a preocupação se todos os relatórios são realmente necessários, se o número de vias estão sobrando, etc.

Neste item, citamos algumas medidas de Eficácia.

- Os documentos
  - . fáceis de preencher?
- Os relatórios
  - . são convenientes? (lay-out, sumarização, etc)
- A documentação
  - . completa?
  - . atualizada?
  - . padronizada?
  - . possui clareza nos procedimentos?
  - . esta organizada?
- Os prazos são cumpridos?
- Funções são executadas a contento?

Com referência a Eficiência sabemos que além do sistema estar funcionando, o CPD procura otimizar os custos, tais como: recursos humanos, relatórios, uso de equipamento, etc.

Citamos algumas medidas de Eficiência para dar uma idéia de que podemos medir:

- Os recursos humanos estão sendo bem utilizados?
- Todos os relatórios são realmente necessários?
- O equipamento está bem utilizado? (otimização de uso)

Uma outra medida de desempenho que falamos também foi Confiabilidade. Citamos alguns indicadores da Confiabilidade a seguir:

- Os controles são eficientes?  
Verificamos que a Eficiência do dígito verificar, depende do método escolhido, por exemplo o mód. 11 é mais confiável do que o mód. 10.
- É efetivada perfuração com verificação?
- O sistema possui back-up?
- Recuperação do arquivo  
Existem as cópias avô e pai?
- Totais de controle  
É associado a cada grupo de movimentos um mes



tre de lote que contenha totais de controle?  
Estes totais de controle são convenientes?

- MÉTODOS DE AUDITORIA; há vários métodos para avaliar um sistema, ou seja medir as "medidas" de desempenho. Mostramos três métodos os quais descreveremos a seguir:

#### TEST-DESK

Para aplicar este método precisamos:

- Identificar o sistema a ser auditado;
- Compreender o sistema, pela análise de sua documentação;
- Determinar as rotinas dos programas do sistema que serão avaliados;
- Preparar os dados de teste para uma forma e um meio físico aceitos pelo computador eletrônico;
- Utilizar os programas de computador componentes do Sistema de Informação para processamento dos dados testes;
- Analisar as informações obtidas a partir dos dados de testes e avaliação da Eficiência e do funcionamento das rotinas dos programas determinados quando na análise da documentação do sistema;

- Avaliar o grau de controle interno do sistema.

### QUESTIONÁRIO

Para saber se um sistema está funcionando adequadamente, o Auditor precisa avaliar o grau de Organização do Centro de Processamento de Dados, assim como do Departamento Usuário, além de analisar a estruturação e atribuições de cada Departamento que trabalha tarefa de um determinado sistema de informação. (Ex. Departamento de Pessoal, Departamento de Orçamento, etc.), o Auditor deve avaliar também o controle interno de um dado sistema de informação desde o nascimento de um dado contábil, que seja em um Departamento Usuário, que seja no CPD, até a elaboração e utilização da informação final. Em outras palavras o Auditor teria que aplicar questionários aos seguintes órgãos:

- Departamento Usuário
- CPD
- Setor de Sistema de Informação

### USANDO COMPUTADOR

Enfatizamos que este Método de Auditoria consiste no uso de programas que testam os dados dos arquivos do sistema.

Para dar uma idéia de como seriam estes programas

mas, mostramos dois tipos de programas típicos para Auditoria:

- Programa de Análise de Arquivo
- Programa de Confronto de Arquivo

Mostramos que o Programa de Análise de Arquivo, basicamente tem as seguintes funções:

- Testar a validade de campos
- Fazer consistência cruzada

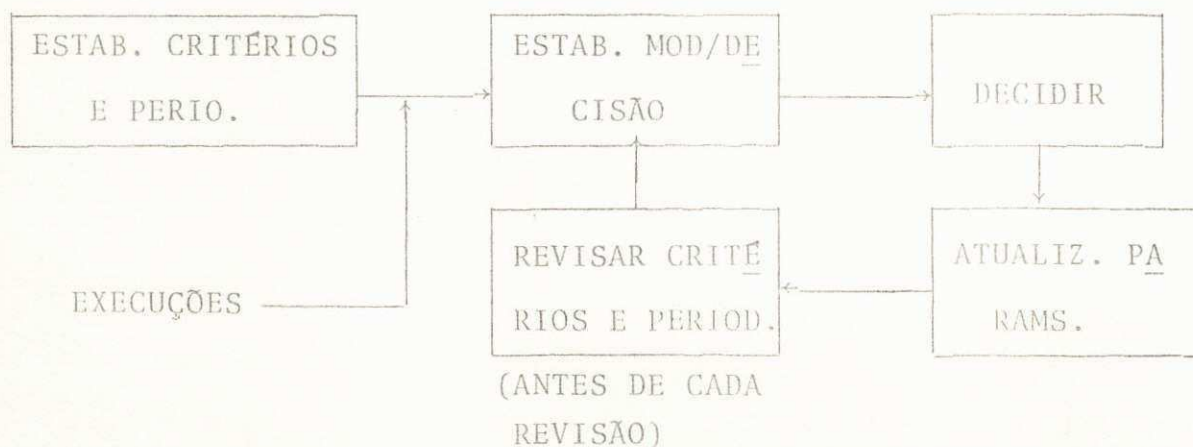
Assim como, no Programa de Confronto a função básica é analisar dois arquivos, A e B, um contra o outro, apresentando:

- Os registros do arquivo A que não estão no arquivo B;
- Os registros do arquivo B que não estão no arquivo A;
- Os registros que estão em ambos.

Mostramos que uma das aplicações deste programa seria na atualização de um arquivo. Com este programa temos condições de descobrir se há inclusões de registros que já existem no arquivo mestre, e alteração para registros que não existem neste mesmo arquivo.

- REAVALIAÇÃO DOS MÉTODOS DE AUDITORIA; enfatizamos que com o decorrer do tempo, o esquema de Auditoria pré-determinado poderá se tornar obsoleto, necessitando portanto de uma reavaliação; apresentamos neste ítem o esquema desta reavaliação:

(NO PROJETO)



### 3.2 - SOFTWARE BÁSICO

#### - Arquitetura do Sistema IBM/370

Com referência a este tópico, tivemos oportunidade de elaborar um texto que apresenta:

- ORGANIZAÇÃO DO SISTEMA; neste ítem, mostramos as unidades funcionais básicas do sistema: Memória, Unidade Central de Processamento - CPU, Canais de Entrada/Saída, Unidades de Controle e Dispositivos de Entrada/Saída, assim como as fun

ções fundamentais de cada unidade; e finalmente no sub-ítem utilização dos Canais de Entrada/Saída, enfatizamos que o Canal Byte Multiplexer é utilizado pelos dispositivos lentos, enquanto que o Canal Seletor é utilizado por dispositivos rápidos, e o Block Multiplexer é utilizado em Sistemas que operam em time-sharing.

- MEMÓRIA; mostramos as unidades de memória: Byte, Halfword, Fullword, Double Word e Extended Word, citando a quantidade em Bytes e o tamanho em Bits.

- REGISTRADORES; enfocamos três tipos de Registradores, enfatizando que os Registradores Gerais são controlados pelo programador, e são usados para efetuar Operações Aritméticas e Lógicas, e também como Registrador Base enquanto que os Registradores de Ponto Flutuante são controlados pelo sistema e são utilizados para Operações de Ponto Flutuante, já os Registradores de Controle auxiliam a PSW no controle de operação do computador. Falamos com mais detalhes num ítem intitulado com o nome destes Registradores.

- DADOS; neste ítem, enfatizamos que:

O INTEIRO BINÁRIO é usualmente representado numa fullword de 32 bits, e que um número negativo é representado no que é chamado de notação de complemento de 2.

Números de PONTO FLUTUANTE negativos não são representados em forma de complemento; para indicar que o número é negativo, basta colocar 1 no BIT de Sinal.

No DECIMAL ZONADO a primeira metade do byte representa a zona do dígito; e a segunda metade contém a representação de um dos dígitos de 0 (zeros) a 9.

No DECIMAL COMPACTADO o sinal do número é representado nos 4 bits de mais baixa ordem do byte de mais baixa ordem, nos 4 bits restantes do byte fica o último dígito; os outros bytes contém dois dígitos decimais cada.

O STRING DE CARACTERES pode ter comprimento de até 256 bytes, sendo que um caracter é representado num byte de oito bits.

- FORMATOS DE INSTRUÇÕES; neste ítem enfocamos os tipos de Formatos de Instruções RR, RX, RS, SI, SS; e mostramos que em todos esses tipos os bits de 0-7 contém o código de operação. Enfatizamos que:

Nas Instruções RX, o primeiro operando é um registrador, com endereço R1 dado nos bits 8-11, e o segundo operando está na memória, seu endereço é especificado por um Base B2 (Registrador cujo endereço é dado nos bits 16-19), Registrador de Índice X2 (um Registrador cujo endereço é dado nos bits 12-15) e o Deslocamento dado nos bits 20-31 da instrução.

Nas Instruções RS, o primeiro operando é um Registrador, com endereço R1 dado nos bits 8-11 da instrução, e o segundo está na memória, com endereço especificado por um Base B2 (Registrador cujo endereço é dado nos bits 16-19 da instrução) e um Deslocamento D2 dado nos bits 20-31 da instrução, e o terceiro operando é um Registrador, cujo endereço R3 é dado nos bits 12-15 da instrução. Há algumas instruções RS, contudo, que não tem o terceiro operando; neste caso, R3 (parte da instrução) é ignorada.

Nas instruções SI, o primeiro operando está na memória com endereço especificado pelo Registrador Base B1 e Deslocamento D1, e o segundo operando é a quantidade de oito bits nos bits 8-15 da mesma instrução.

Nas Instruções SS, os dois operandos estão na memória; o endereço do primeiro é dado pelo Registrador Base B1 nos bits 16-19 e Deslocamento D1 nos bits 20-31; e o endereço do segundo operando é dado pelo Registrador Base B2 nos bits 32-35 e Deslocamento D2 nos bits 36-47. Instruções do tipo SS são flexíveis em que os operandos pode ser de vários tamanhos, atuais codificados nos bits de 8-15 da Instrução. Em algumas instruções os dois operandos podem ser de comprimentos diferentes, neste caso o campo Length é dividido em L1, um campo de quatro bits nos bits de posições 8-11 dando o comprimento do primeiro operando, e L2, um campo de quatro bits nos bits de posições 12-15 dando o comprimento do segundo operando.

- FUNÇÕES DE CONTROLE DO SISTEMA; neste ítem, enfocamos a filosofia de multiprogramação, ou seja o computador é projetado para que diversos programas operem concorrentemente, compartilhando recursos de memória principal e Unidade Central de Processamento; e enfocamos também a independência dos dispositivos de entrada e saída, ou seja as entradas e saídas são feitas independentemente da execução do programa.

Mostramos que o sistema é controlado pela Program Status Word - PSW, e pelos Registradores de Controle.

- PROGRAM STATUS WORD - PSW; dizemos que a PSW denota o estado do computador indicando as seguintes informações: interrupções que podem ocorrer, estado de programa, bits de condição e mask, código de Interrupção, e endereço da instrução corrente.

- REGISTRADORES DE CONTROLE; mostramos que os 16 Registradores de Controle auxiliam a PSW ao controle de operação do computador, e citamos o conteúdo das Registradores.

- DESENHO E ENDEREÇAMENTO DA MEMÓRIA; enfatizamos que em alguns modelos a Memória Principal pode ser composta de Memória Principal Grande e um Buffer de armazenamento rápido, conhecida como "cache".



Descrevemos a seguir um tópico relacionado com a função e uso de memória.

#### FATOR DE PERFORMANCE

Mostramos que a performance depende da capacidade, da dimensão de acesso, do tempo de ciclo e do nível de "interleaving", assim como, da paridade e verificação de erro, da proteção da memória, geração de endereço, e de endereçamento.

- FUNÇõES DA UNIDADE DE PROCESSAMENTO; como sabemos a potência do computador depende do repertório de instruções que podem ser usadas para escrever programas. Por isso mostramos um conjunto de instruções, citando em que aplicação estas instruções seriam mais úteis.

- FUNÇõES DE CONTROLE E DESVIO; mostramos que a ordem normal de execução pode ser alterada por: uma instrução de desvio, uma instrução que especifica uma execução indireta, e por uma interrupção.

- ORGANIZAÇão DE CANAL E ENTRADA/SAÍDA; mostramos que os sistemas operacionais que permitem multiprogramação e/ou time-sharing devem ter independência dos dispositivos de IO, operações de IO da execução assincrona, proteção de memória para programas não operativos, e capacidade de adaptar diversos tipos de dispositivos.

Enfatizamos que um dado computador do Sistema IBM/370, pode ter um ou mais canais. Enfocamos os seguintes tipos de canais:

#### CANAL SELETOR

Mostramos que o canal seletor opera no modo burst, no qual o canal é dedicado completamente ao dispositivo usado no momento, e nenhum outro dispositivo pode usar o canal até a conclusão do programa canal, e enfatizamos que os canais seletores são geralmente usados por dispositivos de alta velocidade.

#### CANAL MULTIPLEXER

Mostramos que um canal multiplexer pode ter vários subcanais, cada um capaz de executar um programa canal correntemente com outros e que os dispositivos ligados ao canal multiplexer são usualmente de baixa velocidade. Enfatizamos que um canal multiplexer têm elementos que podem ser usados por apenas uma operação ao mesmo tempo, mas no modo multiplex, as operações são quebradas em pequenos intervalos assim eles podem ser servidos in turn, e que um canal multiplexer pode também operar no modo burst.

- MEMÓRIA VIRTUAL; mostramos que no sistema IBM/370, pode-se ter diversas configurações de espaço virtual, dependendo dos tamanhos de segmentos e páginas desejadas, desde que há dois tamanhos de segmentos (64 k e 1 MBytes) e dois ta

manhos de páginas (2k e 4 kBytes), pode-se ter quatro configurações diferentes; por conseguinte, o deslocamento dentro de uma página é afetado pelo tamanho da página.

Mostramos, também que os endereços lógicos são transformados em endereço real por meio de duas tabelas de convenção, uma tabela de segmento, e uma tabela de páginas residentes na memória, que reflete o atributo corrente de memória real.

- Job Control Language - JCL

Sobre este assunto estudamos os seguintes parâmetros:

- CLASS
- NSGLEVEL
- PRTY
- TIME
- PGM
- EXEC
- PROC
- \*
- DATA
- DD
- DUMMY
- COPIES
- DCB
- BLKSIZE
- LRECL
- RECFM
- DISP
- DSNAME
- LABEL
- OUTLIM
- SPACE
- SYSOUT
- UNIT
- VOLUME

Aplicando os conhecimentos adquiridos ministramos algumas aulas.

- Bases Numéricas

Com referência a este assunto ministramos uma aula, mostrando como se deve proceder para fazer transformações de uma base para outra, e como se efetuam as operações nas bases binária, octal e hexadecimal.

3.3 - AUTOMAÇÃO DE PROCESSAMENTO DE TEXTO

- Edição de Texto

A edição de texto através do computador depende fundamentalmente da existência de um editor de texto. Um exemplo de tal editor é EDIT que funciona sob o comando do CMS (Conversational Monitor System), no Sistema Operacional VM/370.

As operações típicas de edição são:

- SUBSTITUIÇÃO DE STRINGS; entre outros comandos, para se substituir string podemos utilizar o comando CHANGE que tem o seguinte formato:

CHANGE/String1/String2 ex: c/abc/def

- INSERÇÃO DE STRINGS; para inserir strings, primeiro posicionamos o apontador de linha, na linha após a qual queremos inserir, e para isto podemos utilizar os seguintes comandos:

TOP - Posiciona o apontador para a primeira linha do arquivo.

BOT - Posiciona o apontador para a última linha do arquivo.

UP n - Posiciona o apontador n linhas acima da linha corrente.

DOWN n - Posiciona o apontador n linhas abaixo da linha corrente.

LOCATE/STRING - Busca a primeira ocorrência do 'String', começando a partir da próxima linha, e continuando até o fim do arquivo.

- DELEÇÃO DE STRINGS; para deletar a corrente e as linhas seguintes podemos utilizar o comando DELETE, que tem o seguinte formato:

DELETE n            ex: DE 2

Na inexistência de um editor automático de texto em nossas instalações, procedemos a esta fase, isto é, à preparação dos arquivos de entrada para posterior formatação, manualmente. Isto significa que, todas as operações de edição, tais como: Substituição, Inserção e Deleção de Strings, tudo foi feito manualmente, sendo o arquivo de input no caso, um deck de cartões.

- Formatação Automática de Texto

- MARGINAÇÃO; para definir a margem superior, utilizamos o comando:

.TM n

Onde n é o número de linhas, cujo valor padrão é 6.

Para ajustar a margem para esquerda, utilizamos o comando:

.AD n

Onde n é o número de linhas que será ajustada, cujo valor padrão é 0 (zero).

A margem inferior é definida pelo comando:

.BM n

Onde n é o número de linha da margem inferior, cujo valor padrão é 6.

O espaço entre o texto e o rodapé é definido pelo comando:

.FM n

Onde n é o número de linhas que vai existir entre o texto e o rodapé, cujo valor padrão é 1.

- CRIAÇÃO DE PARÁGRAFOS; para criar um parágrafo é necessário:

- . deixar um espaço em branco ou mais no início do cartão; ou
- . utilizar o comando .BK (break); ou
- . usar o comando .PR (parágrafo) que tem o seguinte formato:

.PR parágrafo.

- INDENTATION; para fazer uma indentation numa linha para direita, faremos:

. IN n ou .IN+n

Onde n é o número de espaços a ser deslocado. Se quisermos voltar a escrever como antes devemos fazer:

. IN-n ou .UN n.

- MANIPULAÇÃO DE FIGURAS; se estamos escrevendo em uma página, e existe uma figura que não cabe nesta página há dois modos de resolver o problema:

Um modo é utilizar o comando .CP(condition page), contudo a utilização deste comando pode ocasionar desperdício



de espaço, pois se a figura não couber nesta página, passará para outra página sem preencher completamente a primeira.

Um outro modo de resolver este problema é utilizar o comando .FK (floating keep), neste caso, o formatador armazena a figura para a próxima página e continua escrevendo na mesma página o texto que vem após a figura, economizando, assim, espaço.

- MANIPULAÇÃO DE RODA-PÉS; quando queremos referenciar algo no roda-pé, podemos utilizar um separador entre o texto e o roda-pé, utilizando o comando .FN SET, que tem o seguinte formato:

```
.FN SET n/String/String/String/
```

Nos programas que rodamos utilizamos como separador alguns hífen, ou seja, ----- utilizando do comando:

```
.FN SET 1//-----//
```

E para definir o roda-pé utilizamos:

```
.FN BEGIN
Trecho do -
-
Roda-pé -
-
.FN END
```

- MULTIPLAS COLUNAS; para formar multiplas colunas utilizamos o seguinte comando:

```
.TB <n1 n2....>  
  << 'string'|char/> n <L|R|C|'char' >...>  
  < SET < char>>
```

- NUMERAÇÃO DE PÁGINAS; para numerar páginas utilizamos o comando:

```
.PA <% + 1 | n | + n | -n>
```

- CENTRALIZAÇÃO DE CABEÇALHOS; para centralizar cabeçalhos podemos lançar mão dos seguintes comandos:

```
.TT /String1/String2/String3/
```

Se fizermos:

```
.TT //String//
```

O String ficará centralizado.

```
.CE; cabeçalho
```

- ESPACEJAMENTO; o espaçamento padrão é simples, se quisermos espaçamento duplo devemos fazer:

```
.DS
```

#### 4. CONCLUSÃO

Com este estágio tivemos condições de ampliar os conhecimentos em sistemas, pois desenvolvemos a Sistematização de uma Rotina Administrativa, Diagramação de Sistemas e Auditoria.

Na parte de Software, tivemos oportunidade de estudar assuntos que durante o curso não tinham sido enfocados, por exemplo, Arquitetura do Sistema IBM/370.

Na parte do Processamento de Texto, tivemos chance de aprender algo que não foi ensinado durante o curso.

#### 5. CONSIDERAÇÕES GERAIS

Procuramos desenvolver algo sobre sistemas, por que achamos que com apenas duas disciplinas, não é suficiente para aprender e aplicar todos os conhecimentos necessários ao analista; em nossa opinião, é de suma importância aplicamos aquilo que aprendemos, se não acabamos esquecendo ou deixamos de verificar detalhes que com a aplicação vem à tona.

Também, achamos que o programa de Língua Portuguesa, deveria ser mais focado para a vida prática, isto é, deveria ensinar como fazer um relatório, ou seja enfatizar o que contém uma apresentação, um prefácio, etc, pois notamos que

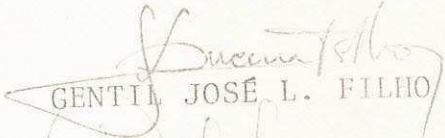
muitos alunos encontram muitas dificuldades em fazer um relatório.

MARIA SILVIA DE OLIVEIRA

= Estagiária =














ORIENTADORES:

MARCOS BLAUTH

  
GENTIL JOSÉ L. FILHO

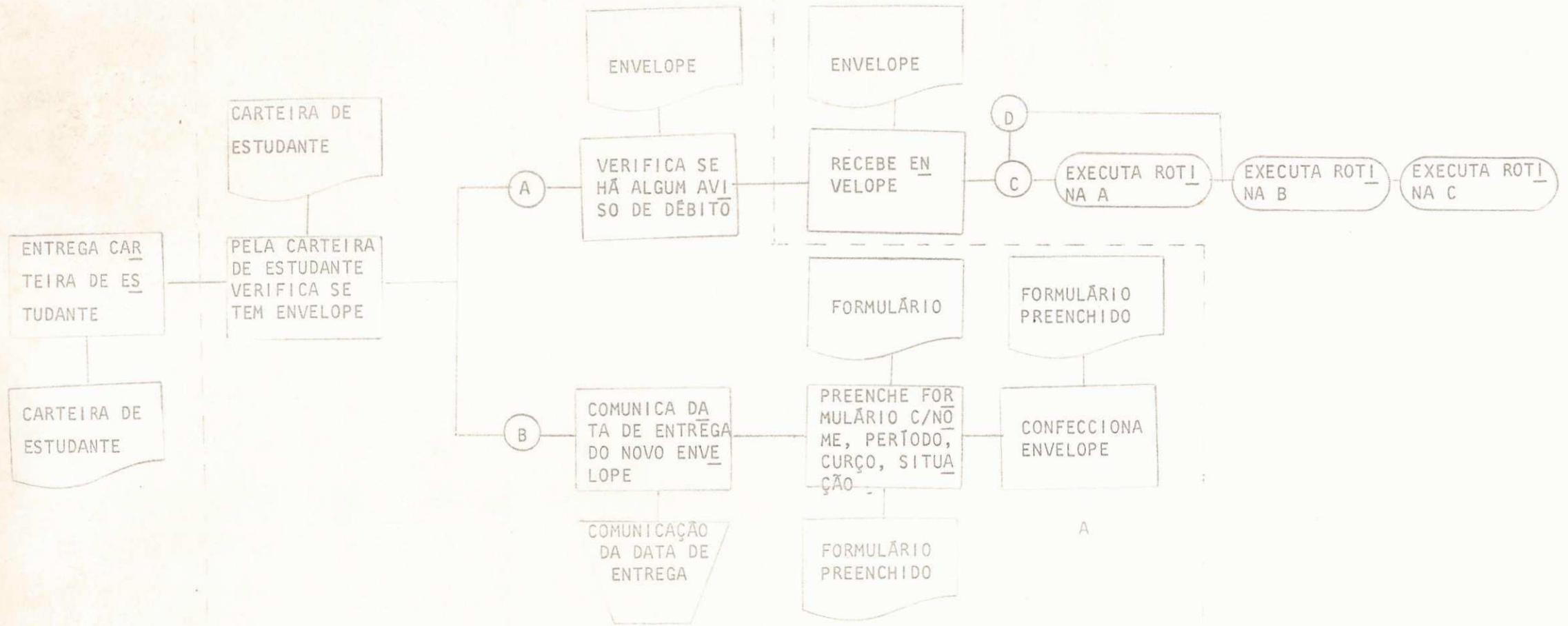
  
LIKISO HATTORI

ANEXO I - Diagramação ASME

FUNCIONÁRIO DA COORDENAÇÃO	ALUNO	DESCRIÇÃO
		Entrega a carteira de estudante.
		Recebe carteira de estudante.
		Verifica se o envelope está junto aos demais.
		Verifica se há algum aviso de débito.
		Entrega o envelope juntamente com o aviso de débito.
		Recebe o envelope.
		Executa rotina A (efetivação do pagamento do débito).
		Entrega o envelope.
		Recebe o envelope.
		Executa rotina B (efetivação da matrícula manual).
		Executa rotina C (efetivação da matrícula mecanizada).
		Obtem o nome, período, curso (atual e anterior, se houver) e situação (transferido, reiniciante, calouro).
		Providência o envelope.

ALUNO      FUNCIONÁRIO DA COORDENADORIA

ALUNO



UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO

ARQUITETURA  
DO  
IEM/370

ELABORAÇÃO: MARIA SILVIA DE OLIVEIRA

ORIENTAÇÃO: LIKISO HATTORI

ABRIL/79

## ÍNDICE

- 01 - Introdução
- 02 - Organização do Sistema
- 03 - Memória
- 04 - Registradores
- 05 - Dados
- 06 - Formatos e Instruções
- 07 - Funções de Controle do Sistema
  - 7.1 - Program Status Word (PSW)
  - 7.2 - Registradores de Controle
  - 7.3 - Estado de Programa (Program State)
  - 7.4 - Interrupção (Interrupt)
  - 7.5 - Código de Condição (Condition Code)
  - 7.6 - Monitoramento
  - 7.7 - Timers do Sistema (System Timers)
- 08 - Desenho e Endereçamento da Memória
  - 8.1 - Fator de Performance
  - 8.2 - Paridade e ECC
  - 8.3 - Proteção de Memória
  - 8.4 - Geração de Endereço
  - 8.5 - Atributos de Memória Permanente



09 - Funções da Unidade de Processamento

9.1 - Conceito de Conjunto de Instruções

9.2 - Formatos e Códigos de Instruções

9.3 - Aritmética de Ponto Fixo

9.4 - Aritmética Decimal

9.5 - Aritmética de Ponto Flutuante

9.6 - Operações Lógicas

10 - Funções de Controle e Desvio

10.1 - Desvio Condicional

10.2 - Execução Indireta e Ligação de Subrotina

10.3 - Instruções de Controle

11 - Organização de Canal e Entrada/Saída

11.1 - Organização de IO

11.2 - Operação de IO

11.3 - Organização do Canal de IO

11.4 - Projeto e Operação dos Canais

11.5 - Endereçamento de IO, Instruções e Comandos

11.6 - Interrupção de IO

12 - Memória Virtual

13 - Bibliografia

A família do sistema/370 é um crescimento lógico e uma extensão da família do sistema/360. A filosofia do projeto é para que o usuário de ambos os sistemas tenham aparentemente a mesma estrutura, pois sendo assim, o usuário se adapta com mais facilidade ao outro tipo de computador.

Apresentaremos, no item 02, Organização do Sistema, onde representaremos as unidades básicas de um sistema de computador, e citaremos as principais funções destas unidades; no item 03, Memória, citaremos as unidades de memória; no item 04, Registradores, citaremos os tipos de Registradores e as funções referentes a cada Registrador; no item 05, Dados, mostraremos os diversos tipos de dados, e como estes dados seriam armazenados na memória; no item 06, Formatos de Instruções, comentaremos os cinco tipos de instruções; no item 07, Funções de Controle do Sistema, enfocaremos a filosofia de multiprogramação, e independência dos dispositivos de entrada e saída; no item 7.1, Program Status Word, mostraremos as informações contidas neste word; no item 7.2, Registradores de Controle, citaremos as informações contidas neste Registrador e as funções destes; no item 7.3, Estado de Programa, descreveremos os estados de programa, ou seja, estado de supervisor/problema, wait/processamento normal, etc; no item 7.4, Interrupção, citaremos os tipos de interrupções, descrevendo cada um, citaremos também, as locações das PSWs novas e velhas; no item 7.5, falaremos sobre Código de Condição, citaremos os codi

gos e o resultado destes códigos, referente a operação com ponto fixo, e com ponto flutuante, o local onde é armazenado e qual a sua finalidade; no item 7.6, Monitoramento, descreveremos seu funcionamento; no item 7.7, Timers do Sistema, descreveremos as funções do interval timer e o timer-of-day cleck; no item 08, Desenho e Endereçamento da Memória, citaremos dois tipos de memória, e o que seria armazenado na memória; no item 8.1, Fator de performance, analisaremos de que depende a performance do sistema, de que a memória é constituída e sua capacidade; no item 8.2, Paridade e ECC, citaremos as funções de cada um; no item 8.3, Proteção de Memória, citaremos como é feita esta proteção e o que previne; no item 8.4, Geração de Endereço, mostraremos como se calcula o endereço efetivo; no item 8.5, Atributos de Memória Permanente, citaremos as áreas fixas de memória ocupadas pelas funções de controle; no item 09, Funções da Unidade de Processamento, diremos de que depende a potência do computador; no item 9.1, Conceito de Conjunto de Instruções, faremos a classificação e citaremos em que aplicação estas instruções seriam mais úteis; no item 9.2, Formatos e Códigos de Instruções, descreveremos, também, o funcionamento de cada tipo de instrução; no item 9.3, Aritmética de Ponto Fixo, citaremos onde são executadas estas operações, e mostraremos um exemplo de instrução; no item 9.4, Aritmética Decimal, citaremos algumas instruções importantes, e mostraremos alguns exemplos de instruções; no item 9.5, Aritmética de Ponto Flutuante, especificaremos os tipos de instruções que operam com ponto flutuante e alguns exemplos; no item 9.6, Operações Lógicas, citaremos tipos de instruções que executam operações lógicas e alguns exemplos; no item 10, Funções de Controle e Desvio, diremos o que ocasiona uma alteração na ordem normal de execução; no item 10.1, Desvio Condicional,

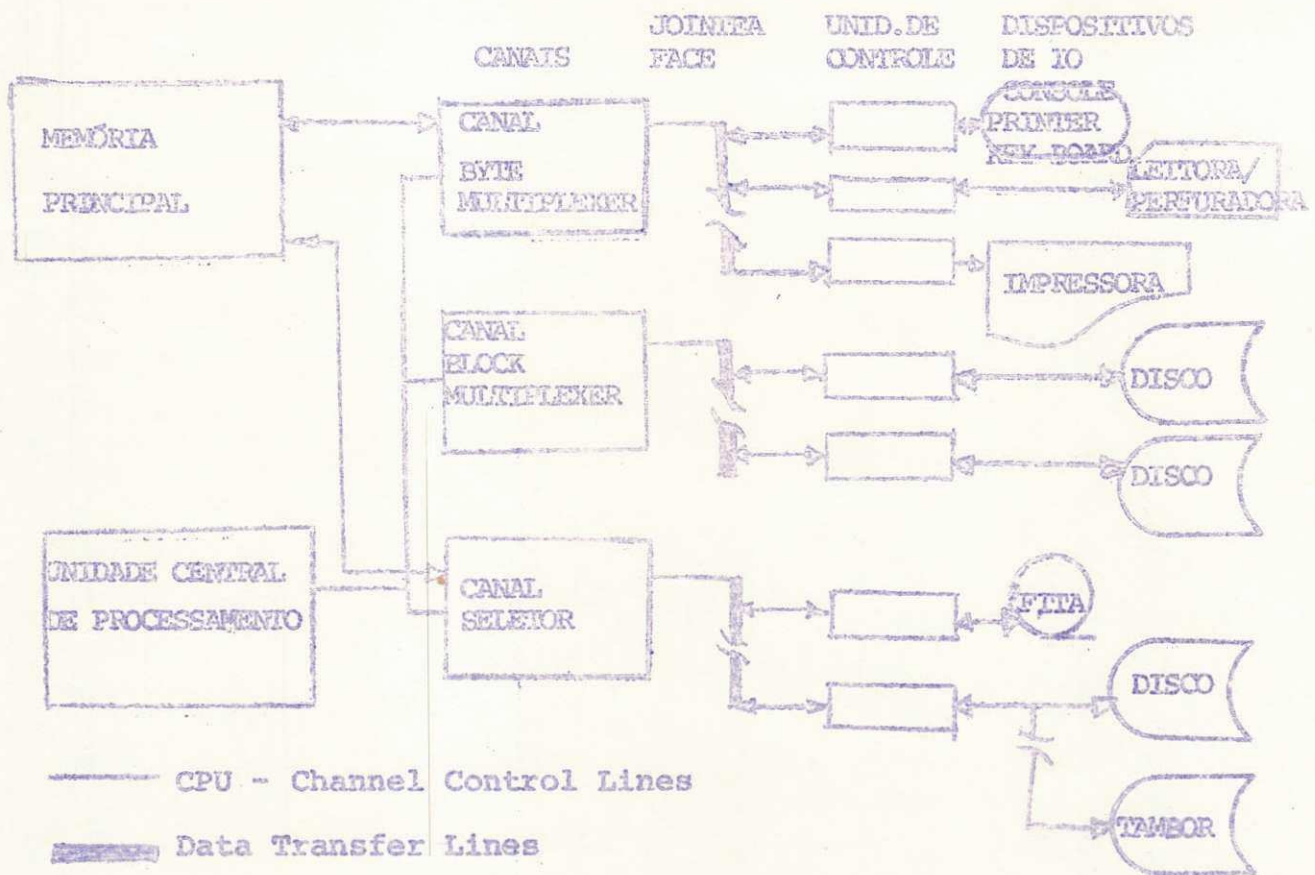
mostraremos instruções que permitem um desvio condicional; no ítem 10.2, Execução Indireta e Ligação de Subrotina, citaremos instruções que tem estas funções; no ítem 10.3, Instruções de Controle, enfocaremos as funções destas instruções, e citaremos alguns exemplos; no ítem 11, Organização de Canal e Entrada/Saída, representaremos através de uma figura, a organização de canal e entrada/saída; no ítem 11.1, Organização de IO, representaremos através de uma figura a organização de IO, o que inicializa uma operação de IO, e o que gera o comando; no ítem 11.2, Operação de IO, explicaremos como se executa estas operações; no ítem 11.3, Organização do Canal de IO, enumeraremos as principais funções dos canais de IO, no ítem 11.4, Projeto e Operações dos Canais, citaremos os modos de operação, e mostraremos as diferenças entre os modos; no ítem 11.5, Endereçamento de IO, Instruções e Comandos, mostraremos o formato das instruções, assim como, o formato dos comandos de canal; no ítem 12, Memória Virtual, enfatizaremos como funciona a transformação dos endereços lógicos em endereço real; no ítem 13, Bibliografia, citaremos os livros que utilizamos para elaborar esta Apostila.

## 2 - ORGANIZAÇÃO DO SISTEMA

Neste ítem, mostraremos as unidades funcionais básicas do sistema, as funções fundamentais de cada unidade, e a utilização dos canais de Entrada/Saída.

Esse sistema é composto por cinco unidades funcionais básicas:

- a) Memória
- b) Unidade Central de Processamento - CPU
- c) Canais de IO
- d) Unidades de Controle
- e) Dispositivos de IO



FUNÇÕES BÁSICAS DAS UNIDADES:

- a) MEMÓRIA - Armazenar instruções e dados.
- b) CPU - Controlar e executar as instruções básicas do computador, por exemplo: Instruções de IO.
- c) CANAIS DE IO - Dirige e controla o fluxo de informações entre a memória e os dispositivos de IO, permitindo que a CPU processe concorrentemente com operações de IO. O Canal também pode funcionar independentemente da CPU.
- d) UNIDADES DE CONTROLE - Obter instruções da memória, decodificá-las, calcular os endereços dos operandos, etc.
- e) DISPOSITIVOS DE IO - Transferir ou armazenar informações do meio externo a memória.

UTILIZAÇÃO DOS CANAIS DE IO:

CANAL BYTE MULTIPLEXER - É utilizado pelos dispositivos lentos ; tais como: Leitora, Impressora, Console, etc.

CANAL SELETOR - É utilizado por dispositivos rápidos; tais como:  
Fita, Disco.

CANAL BROCK MULTIPLEXER - Opera, concorrentemente, compartilhando recursos de memória principal, CPU e dispositivos de IO, é utilizado por dispositivos rápidos; por exemplo: Disco.

### 3 - MEMÓRIA

O tamanho máximo de memória é até  $2^{24}$  bytes (cerca de 16 milhões), sua unidade básica de endereçamento de memória é o byte de 8 bits, que pode representar 1 caracter, 2 dígitos decimais ou 8 dígitos binários. Os bytes podem ser processados separadamente ou agrupados em forma de palavras ou campos de tamanho variável.

UNIDADE DE MEMÓRIA	BYTES	TAMANHO EM BITS
BYTE	1	8
HALFWORD	2	16
FULLWORD	4	32
DOUBLE WORD	8	64
*EXTENDED WORD	16	128

(\* ) Disponível em Alguns Modelos.

No item 8.4, mostraremos como calcular o endereço efetivo.

#### 4 - REGISTRADORES (Memória Local)

O sistema tem basicamente 36 registradores, e 64 bits de PSW\*.

Os 36 registradores estão distribuídos da seguinte forma:

- a) 16 Registradores Gerais;
- b) 4 Registradores de Ponto Flutuante;
- c) 16 Registradores de Controle.

##### a) REGISTRADORES GERAIS

São controlados pelo programador, cada um consistindo de 32 bits, usados para operações aritméticas e lógicas, e como registrador base.

##### b) REGISTRADORES DE PONTO FLUTUANTE

São controlados pelo sistema; cada um consistindo de 64 bits, que são utilizados para operações de Ponto Flutuante.

(\*) Program Status Word



c) REGISTRADORES DE CONTROLE

Auxiliam a PSW no controle de operação do computador. Falaremos sobre esses registradores com mais detalhes no ítem 7.2.

5 - DADOS

O 370 provê instruções para processar diversos tipos de dados, tais como:

- a) Inteiro Binário
- b) Ponto Flutuante
- c) Inteiro Decimal
- d) String de Character

a) REPRESENTAÇÃO DE INTEIRO BINÁRIO

Um inteiro binário é usualmente representado numa full word de 32 bits. Com o propósito de especificar os bits na word, nós numeramos as posições dos bits de 0 a 31, da esquerda para direita, como mostra na figura 1. Em inteiros binários, o bit da posição contém o sinal do número. Se o bit de sinal é um 0, o número é positivo; se é um 1, o número é negativo. Por exemplo, na figura 1, mostra a representação de inteiro binário do número decimal + 27.





Um número de ponto flutuante é representado pela sua mantissa vezes uma potência de 16 com seu sinal ligado ao resultado. O expoente da potência de 16 pela qual a mantissa é multiplicada é codificado na característica. A característica, nas posições dos bits de 1 a 7 da representação pode conter números variado de 0 (zero) (decimal) a 127. A característica é codificado no que é chamada notação de excesso 64, significando que a característica é 64 vezes maior do que o expoente. Então uma característica de 66 representa um expoente de  $66 - 64 = 2$ , e a magnitude no número de ponto flutuante é a função vezes  $16^2$ . Similarmente, a característica de 61 representa um expoente de  $61 - 64 = -3$ , e a magnitude do número de ponto flutuante com uma característica de 61 é a mantissa vezes  $16^{-3}$ . A notação do excesso 64 é usado para que um domínio de magnitudes, variando de  $16^{-64}$  a  $16^{63}$ , possa ser representado.

Outra maneira de consideração a característica é dizer que o expoente é codificado em notação de complemento de 2, exceto que um 0 (zero) na posição bit 1, que é a posição do sinal da característica, representa um sinal menos e um 1 representa um sinal mais.

Alguns exemplos do sistema IBM/370 números de ponto flutuante ajudaria a clarear a representação. Primeiro são vários exemplos de números de ponto flutuante de precisão simples mostrado em binário. O padrão



Exemplo:

```
$JOB . FLOATING-POINT
```

```
REAL*8 A
```

```
READ A
```

```
'
```

```
'
```

```
'
```

Como está especificado na declaração REAL\*8 A, o número A estará armazenado numa palavra de 8 bytes ou seja ocupará uma locação de 64 bits.

Supondo que  $A = 242,25$

Como primeiro passo devemos transformar a parte inteira e fracionária deste número que está na base decimal para a base binária.

$A = 11110010.01$

Depois normalizaremos este número, levando em consideração que o sistema IBM/370 manipula com números hexadecimais; isto quer dizer que a parte inteira deve conter um número de dígitos que seja múltiplo de quatro; se isto não acontecer, complete com zeros a esquerda. Portanto o número normalizado será:







Um exemplo de um número Decimal Zonado é mostrado na Figura 2. Os três bytes representam o número + 387, com a zona branca nos primeiros 2 bytes e a zona + no último byte.



Figura - 2: Representação do Número + 387

REPRESENTAÇÃO DO DECIMAL COMPACTADO

Um dígito decimal pode ser representado em 4 bits; entretanto, dois dígitos podem ser compactados em um byte. O sistema IBM/370 pode fazer aritmética em números expressos nesta forma, que é chamado Decimal Compactado. O sinal do número é representado nos 4 bits de mais baixa ordem do byte de mais baixa ordem: Usualmente 1100 para mais, 1101 para menos. O byte de mais baixa ordem também contém o último dígito significativo do número; os outros bytes contêm dois dígitos decimais cada. A aritmética pode ser executada em números cujos tamanhos podem variar de 1 a 16 bytes; números podem entretanto incluir de 1 a 31 dígitos decimais na Figura 3 mostraremos um campo Decimal Compactado, cujo comprimento é dois bytes, contendo o número - 196

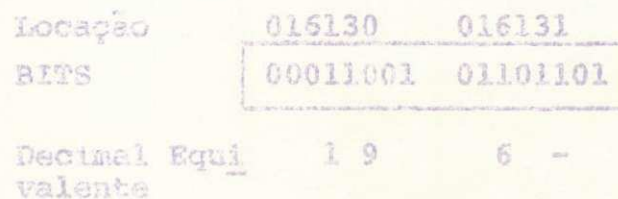


Figura - 3: Representação do Número - 196.

O número é sempre endereçado do seu byte de mais alta ordem - locação 016130 na Figura 3, por exemplo:

#### d) REPRESENTAÇÃO DE STRING DE CARACTERES

No sistema IBM/370 um caracter é representado num byte de oito bits; as instruções permitem processamento de Strings de caracteres com comprimento até de 256 bytes.

A representação do String SOFTWARE será:

11100010	11010110	11000110	11100011	11100110	11000001	11011001	11000101
S	O	F	T	W	A	R	E

#### 6 - FORMATOS DE INSTRUÇÕES

Há 5 tipos de Formatos de Instruções. Os formatos diferem na característica de endereçamento - seja operandos que estão em registradores, na instrução mesmo ou em memória principal endereçada por Registrador Base e Deslocamento ou por Registrador Base, Registrador de Índice e Deslocamento. O porquê da variação em espaço solicitado por especificação de endereços do operando da memória, endereços de registradores, e dados na informação, algumas instruções ocupam dois bytes, outras quatro bytes, e ainda outras ocupam seis bytes.

Em todas as instruções o primeiro byte é o código de operação (Operation Code). Este byte indica o comprimento da instrução e o tipo de operação a ser executada.

### INSTRUÇÕES RR

Os operandos dessas instruções estão em registradores, dando ao tipo o nome RR, entendendo por Registrador a Registrador. Quatro bits contêm cada endereço do Registrador, assim o endereço do primeiro operando está nos bits 8-11 da instrução, e o endereço do segundo operando nos bits 12-15. A instrução é portanto do comprimento de dois bytes, como mostra a Figura 5.



Figura - 5: Formato da Instrução RR

Como exemplo, a instrução codificada em hexadecimal como 1837 é uma instrução tipo RR. O código de operação, 18, significa load de um Registrador. Os endereços dos operandos são 3 e 7; na execução da instrução o conteúdo do Registrador 7 será carregado no Registrador 3, substituindo o conteúdo prévio do Registrador 3. O conteúdo do Registrador 7 continua inalterado.

Uma instrução do tipo RX tem a forma como mostra a Figura 6.

O primeiro operando é um registrador, com endereço R1 dado em bits 8-11 da instrução. O segundo operando é armazenado na memória; seu endereço é especificado por um Base B2 (Registrador cujo endereço é dado em bits 16-19), Registrador de Índice X2 (um Registrador cujo endereço é dado nos bits 12-15) e em Deslocamento dado em bits 20-31 da instrução.



Figura - 6: Formato de uma Instrução RX

Por exemplo, a instrução codificada em hexadecimal como 5B 35 C0 24 é uma instrução tipo RX. O código de operação é 5B significando subtração (inteiro binário). O primeiro operando está no Registrador 3. O campo X2 é 5, o campo B2 é C, o Deslocamento é 024; juntos eles definem o endereço do segundo operando. Suponha que no Registrador 5 contenha 00000050 e Registrador C (ou 12) contenha 00007404. Então o endereço do segundo operando é gerado como mostra abaixo:

Conteúdo do Registrador 5	00000050 (X2)
+Conteúdo do Registrador C	00007404 (B2)
+Deslocamento	<u>024 (D2)</u>
	007478

O endereço efetivo é 007478 em hexadecimal. Esta instrução será executada como segue: O conteúdo da locação 007478 atualmente a fullword na locação 007478 a 00747B será subtraída do conteúdo do Registrador 3. O resultado ficará no Registrador 3, o conteúdo de todas locações de armazenamento ficam inalteradas.

INSTRUÇÃO RS

Uma instrução do tipo RS tem a forma mostrada na Figura 7. O primeiro operando é um Registrador, com endereço R1 dado em bits 8-11 da instrução. O segundo operando é armazenado na memória, com endereços especificados por um base B2 (Registrador cujo endereço é dado nos bits 16-19 da instrução) e um Deslocamento D2 dado nos bits 20-31 da instrução. A instrução tipo RS são somente instruções as quais tem três operandos; o terceiro operando é um Registrador, cujo endereço R3 é dado nos bits 12-15 da instrução.



Figura - 7: Formato de uma Instrução RS

Por exemplo, a instrução codificada em hexadecimal como 9868C024 é uma instrução do tipo RS. O código de operação é 98, significando load multiple. O primeiro operando é o Registrador 6, o terceiro operando é o Registrador 8. O segundo operando é definido pelo base C e o Deslocamento 024. Se o Registrador C contém 00007404, o endereço do segundo operando é definido:

Conteúdo do Registrador C	00007404
+Deslocamento	<u>024</u>
	007428

O endereço efetivo é 007428 em hexadecimal. A execução desta instrução será como segue. Registradores de 6 (primeiro operando) a 8 (terceiro operando) será carregado de locações consecutivas de memória, começando com a locação 007428. Registrador 6 será carregado portanto nas locações 007428 a 00742B. Registrador 7 será carregado nas locações 00742C a 00742F, e o Registrador 8 será carregado nas locações 007430 a 007433. Prévios conteúdos de Registradores 6, 7, e 8 são perdidos, entretanto os conteúdos de todas as locações de armazenamento ficam inalterados.



Figura - 8: Formato de uma Instrução SI.

\* Há algumas Instruções RS, contudo, que não tem o terceiro operando. Neste caso, o R3 (parte da Instrução) é ignorada.

Uma Instrução do tipo SI é da forma mostrado na Figura 8. O primeiro operando está na memória com endereço especificado pelo Registrador Base B1 e Deslocamento D1. O segundo operando é a quantidade de oito bits nos bits 8-15 da mesma instrução. É chamado I2, ou de operando imediato.

Por exemplo, a instrução codificada em hexadecimal como 92F3C231 é uma Instrução do tipo SI. O código de operação é 92, significando Move Immediato. O primeiro operando está na memória numa locação definida pelo Registrador Base C e Deslocamento 231. Se o Registrador C contém 00007404, o endereço do operando é determinado pelos:

Conteúdo do Registrador C	00007404
+Deslocamento	<u>231</u>
	007635

Então o endereço efetivo é 007635 em hexadecimal. O segundo operando é os oito bits codificado em hexadecimal como F3, do segundo byte da instrução.

Na execução desta instrução, o byte de dado imediato, o segundo operando F3, é armazenado na locação 00007635 da memória.



Figura - 9: Formato de uma Instrução SS.

### INSTRUÇÕES SS

Uma Instrução do tipo SS é da forma mostrada na Figura 9. É o único tipo de instrução que tem o comprimento de seis bytes. Os dois operandos estão na memória: o endereço do primeiro é dado pelo Registrador Base B1 e Deslocamento D1 nos bits 20-31; o endereço do segundo operando é dado pelo Registrador Base B2 (cujo endereço é dado nos bits 32-35 da instrução) e Deslocamento D2 de bits 36-47. Instruções do tipo SS são flexíveis em que os operandos pode ser de vários tamanhos, os tamanhos atuais codificados nos bits 8-15 da instrução. Números de 0 a 255 pode ser expressos no campo Length da instrução: Eles representam comprimentos de 1 a 256 bytes: L = 0 implica que o comprimento do operando é de 1 byte, L = 1 implica que o comprimento do operando é de 2 bytes, etc.

Por exemplo, a instrução codificada em hexadecimal como D202C106C735 é uma instrução do tipo SS. O código de operação D2 significa Move Character. O endereço do primeiro operando é definido pelo Registrador Base C e Deslocamento 106. Dado que o Registrador C contém 00007404, o endereço é:



Conteúdo do Registrador C 00007404

- 24 -

+Deslocamento	<u>106</u>
	00750A

O endereço efetivo é 00750A. Similarmente, o endereço do segundo operando é:

Conteúdo do Registrador C 00007404	
+Deslocamento	<u>735</u>
	007B39

O campo Length contém 02, portanto cada operando tem o comprimento de 3 bytes.

Na execução desta instrução, os três bytes começando da locação 007B39 (segundo operando) são movidos para três locações de armazenamento começando de 00750A (primeiro operando). Isto é, o byte de locação 007B39 é movido para locação 00750A, o byte de 007B3A é movido para 00750B, e o byte de 007B3B é movido para 00750C.

Em algumas instruções os dois operandos podem ser de diferentes comprimentos. Nestas instruções o campo Length é dividido em L1, um campo de quatro bits nos bits de posições 8-11 dando o comprimento do primeiro operando, e L2, um campo de quatro bits nos bits de posições 12-15 dando o comprimento do segundo operando. Os números destes campos podem ser de 0 a 15.

A instrução F224C248C<sup>2</sup>82 é uma Instrução SS deste tipo. O código de operação F2 significa pack. O campo Length do primeiro operando contém um 2, portanto o primeiro operando tem o comprimento de três bytes. O endereço do primeiro operando é:

Conteúdo do Registrador C	00007404
+Deslocamento	<u>248</u>
	00764C

O campo Length do segundo operando contém 4, portanto o segundo operando tem o comprimento de cinco bytes. O endereço do segundo operando é:

Conteúdo do Registrador C	00007404
+Deslocamento	<u>A82</u>
	007E86

A execução desta instrução ocupa cinco bytes das locações 007E86 - 007E8A (segundo operando) assumindo ser no formato Decimal Compactado. O número Decimal Compactado é armazenado em três bytes de locações 00764C - 00764E (primeiro operando).

## 7 - FUNÇÕES DE CONTROLE DO SISTEMA

O computador é projetado para que diversos programas operem concorrentemente, compartilhando recursos de memória

principal e unidade central de processamento, na filosofia das classificadas de multiprogramação.

As entradas e saídas são feitas independentemente da execução do programa.

7.1 - Origram Status Word (PSW)

O PSW denota o estado do computador indicando as seguintes informações: interrupções que podem ocorrer, estado de programa, bits de condição e mask, código de interrupção, e endereço da instrução corrente.

SYSTEM MASK	KEY	MWF.	INTERRUPTION CODE	ILC	CC	PROGRAM MASK	INSTRUCTION ADDRESS
0	7 8 11 12	13 15 16	31	32 33 34 35 36	39 40	63	

- 0-7 System Mask
- 0 Channel 0 mask
- 1 Channel 0 mask
- 2
- 3
- 4
- 5
- 6 Channel 6 mask
- 7 External mask
- 8-11 Protection key
- 12 Not used
- 13 Machine-check mask (M)
- 14 Wait state (W)
- 15 Problem State (P)
- 16-31 Interruption Code
- 32-33 Instruction Length Code (ILC)
- 34-35 Condition Code (CC)
- 36-39 Program Mask
- 36 Fixed Point Overflow Mask
- 37 Decimal Overflow Mask
- 38 Exponent Underflow Mask
- 39 Significance Mask
- 40-63 Instruction Address

Os 16 Registradores de Controle auxiliam a PSW ao controle de operação do computador. Os Registradores não fazem parte da memória endereçável mas podem ser carregados e armazenados por instruções apropriadas que são executáveis em estado de supervisor. Somente 5 (0, 2, 8, 14 e 15) são normalmente usados, os demais são reservados para possíveis expansões. Conteúdo dos Registradores:

Word	Bits	Nome do Campo	Função	Valor depois do Reset do Sistema
0	0	Block Multiplex Mode	Block Multiplexing Control	0
0	24	Timer Mask	Extended External Masking	1
0	25	Interrupt Key Mask	Extended External Masking	1
0	26	External Signal Mask	Extended External Masking	1
2	0-31	Channel Masks	Extended IO Masking	1
8	16-31	Monitor Masks	Monitoring	0
14	0	Hard Stop Mode	Machine Check Handling	1
14	1	Synchronous MCEL Mask	Machine Check Handling	1
14	2	IO Extended Logout	Machine Check Handling	0
14	4	Recovery Report Mask	Machine Check Handling	0
14	5	Configuration Report Mask	Machine Check Handling	0
14	6	External Damage Report Mask	Machine Check Handling	1
14	7	Warning Mask	Machine Check Handling	0
14	8	Asynchronous MCEL Mask	Machine Check Handling	0
14	9	Asynchronous Fixed Log Mask	Machine Check Handling	0
15	8-31	MCEL Pointer	Machine Check Handling	512

\* Machine Check Extended Logout.

Quatro alternativas são possíveis: problema/ supervisor, wait/ativo, masked/interrupt e parado/operativo.

A CPU está no estado de supervisor quando o bit 15 da PSW é zero. Neste estado, todas as instruções do computador são válidas - incluindo instruções da IO, instruções de proteção, instruções de mudanças do estado que são classificadas de instruções privilegiadas. Quando o bit 15 da PSW é um, o CPU está no estado de problema, em que somente as instruções não privilegiadas podem ser executadas. O bit 15 da PSW é modificada pela execução da instrução privilegiada que carrega uma nova PSW.

O estado da wait é Registrado quando espera uma interrupção pelo carregamento de uma nova PSW com o bit 14 igual a 1. No estado de wait nenhuma instrução é processada, mas o timer continua rodando. Quando o bit 14 torna-se zero, o processamento das instruções continua normal.

Quando a CPU pode ser interrompida por uma interrupção que é possível a um bit de mask da PSW ou de um Registrador de controle é igualado a um. Quando o bit de mask é zero a interrupção não poderá ocorrer e é dito masked off ou impossível. O estado dos bits de mask são alterados pelo carregamento de uma nova PSW, ou Registrador de controle, ou pela execução de duas instruções privilegiadas - SET SYSTEM MASK - e ST PROGRAM MASK - que

altera os bits de mask da PSW.

- 29 -

A CPU pode estar em estado parado ou de operação:

Parado - não executa instruções e nem interrupções podem ser feitas.

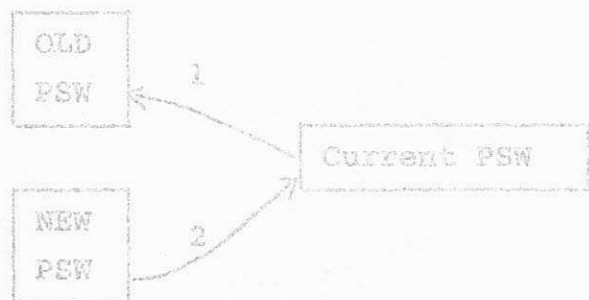
Operação - o sistema está esperando ou o ando, como indica o bit de estado de wait, a interrupção pode ser feita.

Os estados parado e de operação podem ser registrados somente pela intervenção manual no painel de controle do sistema ou por mal funcionamento da máquina.

#### 7.4 - Interrupção (Interrupt)

O sistema de interrupção permite ao sistema 370 responder a condição externa, requisitar monitoramento ou serviços do sistema, e para condições de erros sobre base dinâmica.

Cinco tipos de interrupções são possíveis entrada/saída, programa, chamado de supervisor, externa a machine check. Associada a cada tipo de interrupção têm duas locações fixas na memória principal - cada qual suportando uma palavra dupla. Na primeira, a CPU armazena a PSW corrente no instante de interrupção (PSW velha), e a segunda (PSW nova) a CPU torna automaticamente a PSW corrente.



O estado da CPU na hora da interrupção é guardada na PSW velha para futura inspeção, nos bits de 16 a 31 contém o código de interrupção que indica a causa ou fonte da mesma.

Locações das PSWs novas e velhas.

<u>Tipo de interrupção</u>	<u>locação da OLD PSW</u>	<u>locação da NEW PSW</u>
Externa	24	88
Chamada de supervisor	32	96
Programa	40	104
Machine Check	48	112
Entrada/Saída	56	120

Na interrupção de entrada/saída o endereço de canal de IO e a unidade de controle associada é guardada nos bits 16 a 31 da PSW velha e informações adicionais na CSW (Channel Status Word) quando uma interrupção é aceita pela CPU. Os canais de 0-5 são controlados pelos bits 0-5 da PSW e os adicionais 6-11 são controlados pelo bit 6 da PSW e pelo Registrador de controle 2.

Interrupções de programa podem ser causados por especificações impróprias das facilidades do computador, uso ilegal

de dados ou instruções, ou requisição de monitoramento do sistema.

Interrupção de chamada de supervisor é provocada pela mudança de estado de problema para o estado de supervisor e é iniciada pela instrução de chamada de supervisor (SVS).

A interrupção externa permite a CPU a responder ao timer, uma chave de interrupt sobre a console do operador, ou a uma das linhas de sinal para a CPU. É identificada pelos bits 24-31 da PSW velha, e o controle é feito pelo bit 7 da PSW e um bit apropriado do Registrador de Controle 0. Quando o bit 7 da PSW, bits 24, 25, e 26 do Registrador 0 são iguais a 1, então é possível chave de interrupção e sinais externos de interrupção.

Interrupção de machine check (MCI) resulta de mal funcionamento do sistema tanto do hardware como do software. Quando o erro é irreparável resulta na parada da CPU e o bit 13 da PSW é ajustado a 1. Se for de software então a interrupção é normal. O circuito do /370 provê (errors checking and correction CCC) para checagem e correção de erros em bit de erro na memória.

Os erros da machine check são controlados pelo bit 13 da PSW e pelo Registrador de Controle 14. Quando a interrupção é provocada pela CPU, um código (MCIC) de 8 bytes é armazenado na locação fixa 232 e a nove PSW é trazida da locação 112.

Inclui ainda falha de endereçamento de memória, códigos de região, e controle de Registrador geral e de controle, são armazenados numa posição fixa da memória. Quantidade mínima



de memória para processamento da machine check:

232	Machine check interruption code
240	Reservado
248	Falha de endereçamento de memória
252	Código da região
256	Fixed-logout área
352	Floating Point register save área
384	General register save área
448	Control register save área
512	

A PSW velha contém o endereço da próxima instrução que deve ser executada se não ocorrer interrupção. Os bits 32 a 33 da PSW velha contém "Instruction Length Code" (ILC) que indica o comprimento da última instrução interpretada pela CPU. A ILC só é válida para interrupções de programas e de supervisor.

#### 7.5 - Código de Condição (Condition Code)

A PSW contém 2 bits (CC) que é ajustado pela maioria das operações aritméticas e lógicas.

Condition code	Resultado
0	soma é zero
1	soma é menor do que zero
2	soma é maior do que zero
3	overflow

## Operações de ponto flutuante

Condition code	Resultado
0	operandos são iguais
1	primeiro operando é pequeno
2	o primeiro operando é maior
3	indefinido

O código de condição é armazenado como parte da PSW velha e a nova condição é armazenada com a nova PSW ou com instruções de mask de programa. No processamento normal o código de condição é usado pela instrução de desvio e para alterar a seqüência de execução das instruções.

## 7.6 - Monitoramento

Quando uma instrução de chamada de monitor é executada pela CPU, o bit de mask no Registrador de Controle 8 especificado pelo campo de classe de monitor na instrução é inspecionado. Se o bit é zero nenhuma interrupção de programa é iniciada.

Se o bit é uma interrupção de programa podem ser feitas assim:

1. O código de interrupção é colocado na PSW velha;
2. O número de classe do monitor é colocado na locação de memória 149 (locação 149 = 0);
3. O código de monitor (Base+Deslocamento) é colocado nas locações 157-159 (locação 156 = 0).

#### 7.7 - Timers do Sistema (System Timers)

Incluem o interval timer e o timer-of-day check. O timer de intervalo é mantido na locação 80 da memória e tem uma decisão de 3.33 milasegundos. Ele ocupa uma palavra de 32 bits, somente os bits de 0-23 são usados para um ciclo de 15.5 horas. Funciona com uma taxa de 68 ciclos por segundo e serve para determinar o tempo decorrido em intervalos de tempos pequenos.

O relógio de tempo do dia é um contador interno de uma palavra dupla que provê o tempo decorrido com precisão, independente da atividade do sistema. A precisão é da ordem de 1 microsegundo, e o ciclo é de aproximadamente 142 anos. O conteúdo só pode ser mudado por uma instrução privilegiada ou pela chave existente no painel.

## 8. DESENHO E ENDEREÇAMENTO DA MEMÓRIA

Memória principal provê o sistema com armazenamento endereçável de alta velocidade de acesso de dados. Dados e programas devem ser carregados na memória principal antes de serem processados. Em alguns modelos a memória principal pode ser composta de memória principal grande e um buffer de armazenamento rãpido, conhecida como "cache".

Relacionamos a seguir uma variedade de tópicos relacionados com a função e uso de memória.

### 8.1 - Fator de Performance

A performance depende da capacidade, da dimensão de acesso, do tempo de ciclo e do nível de "interleaving", depende ainda da paridade e verificação de erro, da proteção da memória, geração de endereço, e endereçamento.

Capacidade da memória - vai desde 94k até 3072 kbytes, dependendo do modelo. A construção varia desde núcleos de ferrite até a tecnologia de monolíticos. O primeiro tem organização de 3 dimensões, e o segundo organização 2 1/2 dimensão.

Dimensão do acesso - refere-se a quantidade de byte que pode ser transferido em cada acesso a memória pela CPU. Varia de 4 a 16 bytes.

Os demais itens devem ser descritos em conjunção com cada modelo.

### 8.2 - Paridade e ECC

Os dados são transferidos da memória para a CPU em múltiplos de bytes. A cada byte é necessário um único bit de paridade. A paridade velha é mantida um machine check é gerada se ocorrer erro de paridade.

ECC (Error Checking and Correction) refere-se a facilidade da memória que associa um código de ECC a um grupo de bytes (ECC block). ECC permite corrigir erros em bit isolado automaticamente.

### 8.3 - Proteção de Memória

A facilidade da proteção de memória, protege contra acessos não autorizados a blocos específicos da memória. A cada 2kbytes da memória existe uma chave de proteção. O campo de

chave de proteção ocupa os bits 8-11 na PSW e os bits 0-3 da palavra de endereço de canal.

Um programa em execução do estado de supervisor pode referenciar um bloco de memória primeiro inspecionando a chave de memória depois colocando a chave de proteção no PSW.

#### 8.4 - Geração de Endereço

O sistema IBM/370 usa um esquema de endereçamento base/deslocamento que permite endereçamento relativo dentro de um bloco de 4096 bytes de memória principal.

##### Regras Para Representação de Endereço

Um Operando em um Registrador é endereçado pelo número do Registrador. Este endereço do registrador é um número de 0 (zero) a 15, e é expresso em quatro bits. Nós mostraremos este número em exemplos com um dígito hexadecimal, 0 (zero) a F, correspondendo à representação binária 0000 a 1111.

O endereço de um Operando na memória pode ser expresso em instruções de duas maneiras:

Primeiro - é a soma do conteúdo de um Registrador e um número de 12 bits chamado deslocamento.

O Registrador usado é chamado Registrador Base.

Segundo - é semelhante a soma do conteúdo do Registrador Base e conteúdo de outro (ou o mesmo) Registrador, chamado Registrador de Índice e um Deslocamento de 12 bits.

A Figura 4 ilustra a formação de um endereço especificado pelo Registrador Base e Deslocamento, e pelo Registrador Base, Registrador de Índice e Deslocamento. A Figura 4-a mostra como um apontador para um Registrador usado como Base e um Deslocamento define o endereço de um operando. A Figura 4-b inclui também um Registrador de Índice.

Algumas instruções usam operandos endereçados por Registrador Base, Registrador de Índice e Deslocamento.

Depois da soma do conteúdo do Registrador Base e Deslocamento - ou da soma dos conteúdos dos Registradores Base e de Índice, e Deslocamento - ter sido efetuada, somente os 24 bits de mais baixa ordem são usados como um endereço. Para compactar nós mostraremos conteúdos dos Registradores, Deslocamento, e endereço atual em hexadecimal em alguns exemplos.

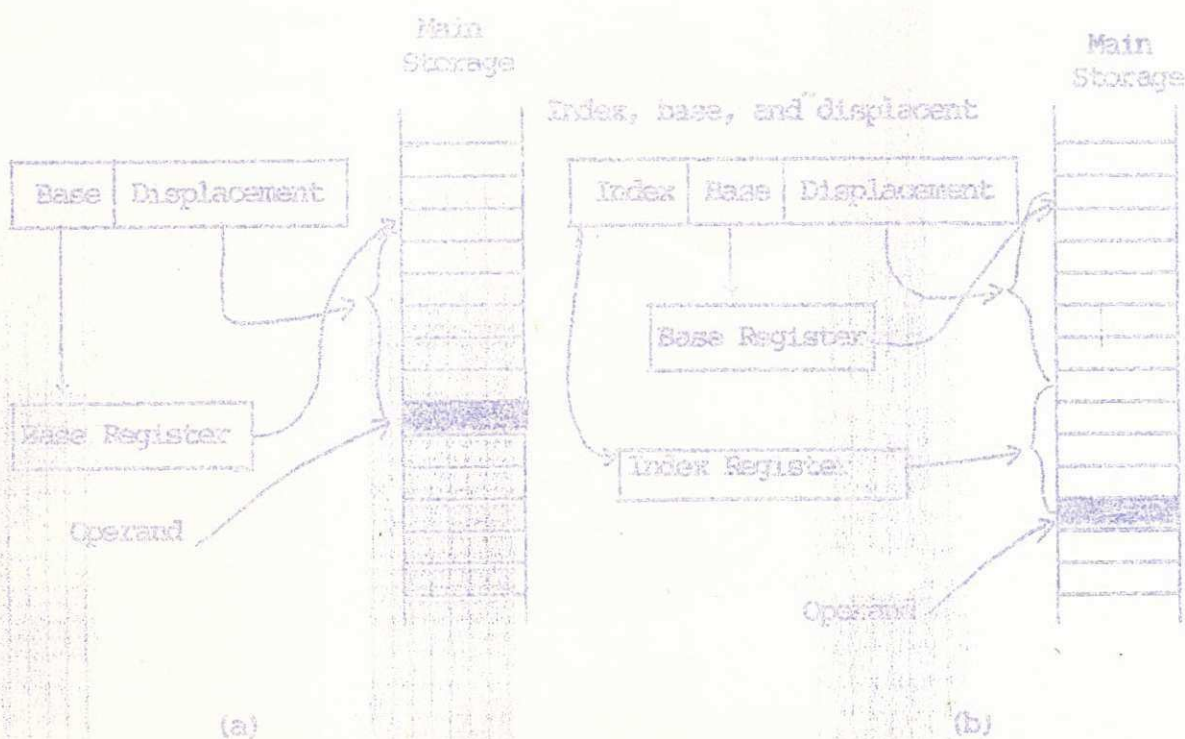


Figura - 4: Formação de um Endereço de um Operando (a) utilizando Registrador Base, e Deslocamento, e (b) utilizando Registradores de Índice, Base e Deslocamento.

a) Supondo que o endereçamento de uma instrução cujo operando está na memória seja especificado pelo Registrador Base 9 e Deslocamento 00E (hexadecimal). Supondo também que o Registrador 9 contém (hexadecimal)00006844. O atual endereço do operando - ao qual chamaremos endereço efetivo - é formado por:

Conteúdo do Registrador 9	00006844		
+Deslocamento	<table style="border-collapse: collapse; margin-left: auto; margin-right: auto;"> <tr> <td style="border-bottom: 1px solid black; padding: 0 10px;">00E</td> </tr> <tr> <td style="padding: 0 10px;">006852</td> </tr> </table>	00E	006852
00E			
006852			



O endereço efetivo é os 24 bits de mais baixa ordem da soma, 006852. O conteúdo do Registrador 9 não foi alterado; a adição é executada num Registrador não endereçável na Unidade Central de Processamento. O endereçamento do operando pode ser representado em forma de diagrama tal como mostra a Figura 5.

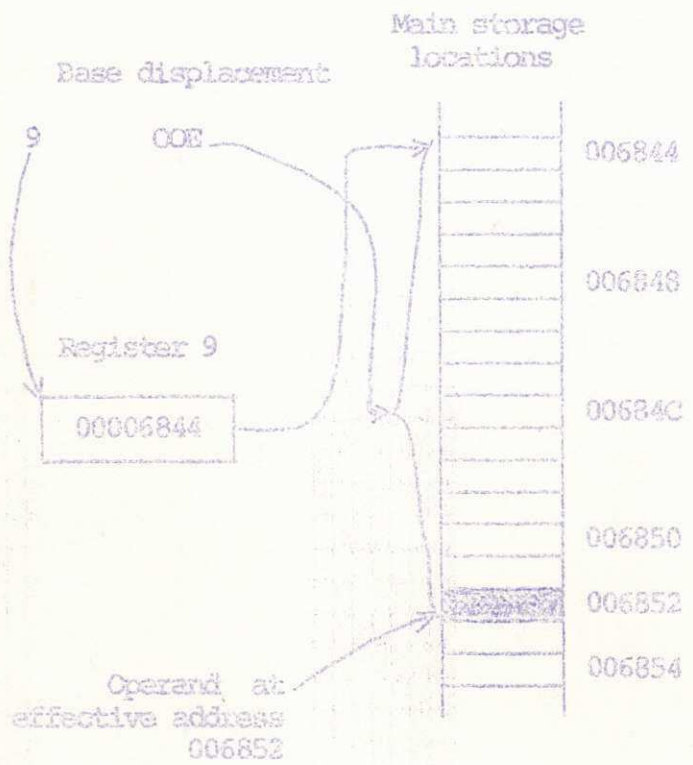


Figura - 5: O Endereçamento de um Operando.

b) Supondo que o Registrador B (decimal 11) contém 50011C48. Uma instrução especificada pelo Registrador Base B e Deslocamento 46A. O endereço efetivo é:

Conteúdo do Registrador B	50011C48
+Deslocamento	46A
	0120B2

O endereço efetivo é 0120B2.

c) Uma instrução que usa um Registrador Base, de Índice e Deslocamento pode ser chamado por Base B, de Índice 4 e Deslocamento 074. Se o Registrador 4 contém 00001240, o endereço efetivo é:

Conteúdo do Registrador B	50011C48
+Deslocamento do Registrador 4	00001240
+Deslocamento	<u>      46A</u>
	0132F2

O endereço efetivo é 0132F2.

d) Uma instrução que normalmente usa Registrador Base, de Índice e Deslocamento pode ser chamado de Base B, de Índice 0 (zero), e Deslocamento 074. O endereço efetivo é:

Conteúdo do Registrador B	50011C48
+(não o conteúdo do Registrador 0 (zero)	0
+Deslocamento	<u>      074</u>
	011C8C

O endereço efetivo é 011C8C.

### 8.5 - Atributos de Memória Permanente

As funções de controle do sistema ocupam áreas fixas de memória. Posições 0-127 independente do modelo é compatível com o /370 exceto para o bit ASCII (bit 12) da PSW. Área fixa

de "logout" vai de 232 a 511 para qualquer modelo, bytes 128-167 não são usados, e os bytes 168-231 depende dos modelos. A CPU extended logout ocupa 192 bytes a partir da posição 512 no modelo 145 e 1000 no 165.

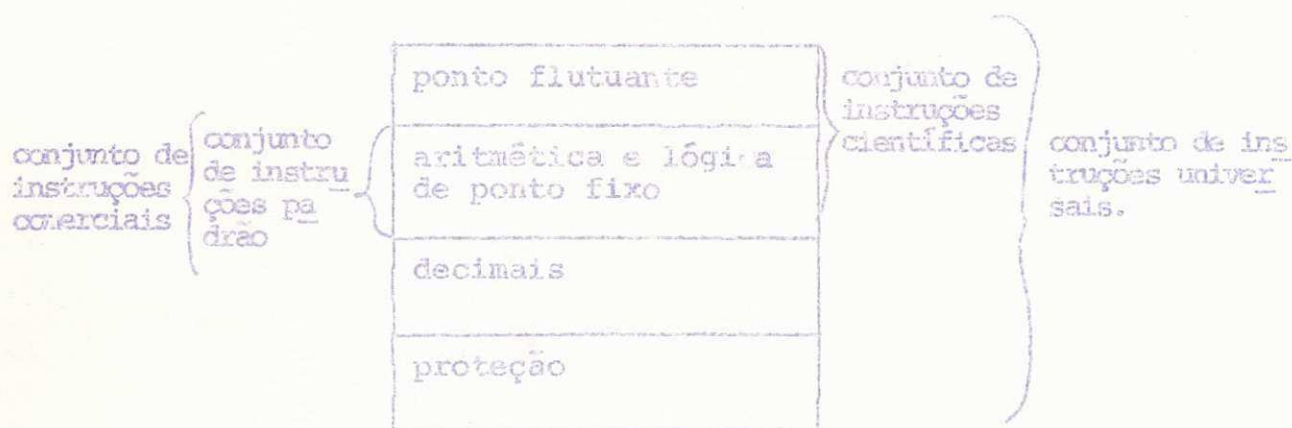
## 9 - FUNÇÕES DA UNIDADE DE PROCESSAMENTO

A potência do computador é reflexo do repertório de instruções que podem ser usadas para escrever programas.

### 9.1 - Conceito de Conjunto de Instruções

São normalmente classificadas em :

- Aritmética e lógica binária de ponto fixo
- Ponto Flutuante
- Decimal
- Proteção
- Entrada/Saída



Entrada/Saída serão descritas posteriormente.

### 9.2 - Formatos e Códigos de Instrução

Um Operando pode ser um número de Registrador, um endereço ou um valor indireto.

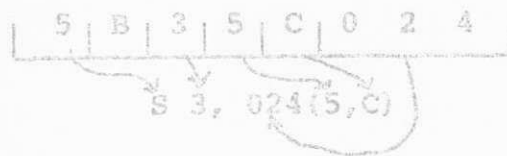
Tipo

RR	<table border="1"> <tr> <td>Op</td> <td>R1</td> <td>R2</td> </tr> <tr> <td>0</td> <td>7 8</td> <td>11 12 15</td> </tr> </table>	Op	R1	R2	0	7 8	11 12 15	Op R1, R2				
Op	R1	R2										
0	7 8	11 12 15										
RX	<table border="1"> <tr> <td>Op</td> <td>R1</td> <td>X2</td> <td>R2</td> <td>D2</td> </tr> <tr> <td>0</td> <td>7 8</td> <td>11 12 15 16 19 20 31</td> </tr> </table>	Op	R1	X2	R2	D2	0	7 8	11 12 15 16 19 20 31	Op R1, D2 (X2, B2)		
Op	R1	X2	R2	D2								
0	7 8	11 12 15 16 19 20 31										
RS	<table border="1"> <tr> <td>Op</td> <td>R1</td> <td>R3</td> <td>B2</td> <td>D2</td> </tr> <tr> <td>0</td> <td>7 8</td> <td>11 12 15 16 19 20 31</td> </tr> </table>	Op	R1	R3	B2	D2	0	7 8	11 12 15 16 19 20 31	Op R1, R3, D2 (B2)		
Op	R1	R3	B2	D2								
0	7 8	11 12 15 16 19 20 31										
SI	<table border="1"> <tr> <td>Op</td> <td>I2</td> <td>B1</td> <td>D1</td> </tr> <tr> <td>0</td> <td>7 8</td> <td>15 16 19 20 31</td> </tr> </table>	Op	I2	B1	D1	0	7 8	15 16 19 20 31	Op D1 (B1), I2			
Op	I2	B1	D1									
0	7 8	15 16 19 20 31										
SS	<table border="1"> <tr> <td>Op</td> <td>L1</td> <td>L2</td> <td>B1</td> <td>D1</td> <td>B2</td> <td>D2</td> </tr> <tr> <td>0</td> <td>7 8</td> <td>11 12 15 16 19 20 31 32 35 36 47</td> </tr> </table>	Op	L1	L2	B1	D1	B2	D2	0	7 8	11 12 15 16 19 20 31 32 35 36 47	Op D1 (L1, B1), D2 (L2, B2)
Op	L1	L2	B1	D1	B2	D2						
0	7 8	11 12 15 16 19 20 31 32 35 36 47										
SS	<table border="1"> <tr> <td>Op</td> <td>L</td> <td>B1</td> <td>D1</td> <td>B2</td> <td>D2</td> </tr> <tr> <td>0</td> <td>7 8</td> <td>15 16 19 20 31 32 35 36 47</td> </tr> </table>	Op	L	B1	D1	B2	D2	0	7 8	15 16 19 20 31 32 35 36 47	Op D1 (L, B1), D2 (B2)	
Op	L	B1	D1	B2	D2							
0	7 8	15 16 19 20 31 32 35 36 47										

### 9.3 - Aritmética de Ponto Fixo

A aritmética de ponto fixo é executada nos Registradores gerais e é definido como palavra ou meia palavra. As operações de divisão, multiplicação, "shifting" usa pares adjacentes de Registradores gerais e é endereçado pelo Registrador da esquerda. Quando uma meia palavra é colocada num Registrador ela é colocada mais a direita.

#### Exemplo de instrução



Todas as instruções são de tipo RR, RK ou RS

Temos duas instruções de suma importância:

CVB - Convert-to-binary

CVD - Convert-to-decimal

### 9.4 - Aritmética Decimal

Instruções de aritmética decimal opera sobre dados decimais compactados e utiliza operandos mantidos na memória (SS).

Instruções importantes:

PACK - Converte decimal zonado em decimal compactado

UNPK - Compactado para zonado

ED - Compactado em EBCDIC para saída

Exemplos de instruções:

F	3	3	4	4	0	2	4	5	0	3	6
---	---	---	---	---	---	---	---	---	---	---	---

PACK 24(3,4), 36(4,5)

F	A	3	3	4	0	2	4	7	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

AD 24(3,4), 0(3,7)

### 9.5 - Aritmética de Ponto Flutuante

As operações aritméticas de ponto flutuante são executadas nos Registradores de ponto flutuante e pode ser números curtos ou longos ou de precisão estendida. Temos instruções do tipo RR e RX, eis alguns exemplos:

AE	0,TSUB	Add normalized (short)
ADR	5,6	Add normalized (longo)
MXD	3,CHICO	Multiply (long/extended)

## 9.6 - Operações Lógicas

Instruções lógicas executam operações lógicas, movimento de dados, edição, deslocamento, teste e tradução. Os operandos de comprimento variável são executadas da esquerda para a direita e pode ter no máximo 256 bytes, exceto as instruções CLCL (Compare Logical Long) e MVCL (move long) que são limitadas somente pelo tamanho de memória.

Elas podem ser de formatos RR, SS, SI, RX e RS, por exemplo: MVC A2( ),JOIA move 8 bytes memória a memória.  
IC 0,='X'55' insere o caracter 55 no Registrador 0.

## 10 - FUNÇÃO DE CONTROLE E DESVIO

A ordem normal de execução pode ser alterado por: uma instrução de desvio, uma instrução que especifica uma execução indireta, e por uma interrupção.

### 10.1 - Desvio Condicional

O sistema IBM/370 inclui quatro instruções que permitem desvio condicional:

Branch-on-condition (BCR e BC)

Branch-on-count (BCTR e BCT)

Branch-on-index high (BXH)

Branch-on-index low or equal (BXLE)

## 10.2 - Execução Indireta e Ligação de Subrotina

A instrução BALR (branch-and-link) guarda o endereço de retorno num desvio para um outro endereço. A instrução EX permite executar uma instrução fora de sequência.

## 10.3 - Instruções de Controle

A maioria das instruções de controle são privilegiadas, e só podem ser usadas quando a CPU está no estado de supervisor. Coletivamente, controla a execução da CPU pela modificação da PSW e chaves de memória, carregando e armazenando Registradores de Controle, permitindo a CPU identificar canais a serem inspecionados, mantendo o relógio time-of-day, permitindo programas serem compartilhados, permitindo a um programa rodar em estado de problema.



Exemplos de instruções:

LPSW - Load PSW

STIDC - Store channel ID

ISK - Set storage Key

## 11 - ORGANIZAÇÃO DE CANAL E ENTRADA/SAÍDA

A eficiência da entrada e saída é um fator de suma importância na performance do sistema. Sistemas operacionais permitem multiprogramação e time-sharing deve ter independência dos dispositivos de IO, operações de IO de execução assíncrona, proteção de memória para programas não operativos, e capacidade de adaptar diversos tipos de dispositivos.

### 11.1 - Organização de IO

Operações de entrada/saída são executadas pelos canais de IO, unidades de controle, dispositivos de IO operando sob controle de um programa de controle de IO que é executado na CPU. A maioria das unidades funcionais são tratadas como entidades independentes por outra unidade do sistema. Um dispositivo de IO é ligado a um canal de IO através de uma unidade de contro

le que contém um buffer de dados e circuitos lógicos. As informações são passadas de uma unidade de controle da IO a um canal de IO. Uma unidade de controle de IO pode ser compartilhada ou não por vários dispositivos de IO.

### 11.2 - Operação de IO

A CPU inicia a operação de IO pela especificação do canal e o dispositivo a ser usado além da locação de memória para iniciar um programa de canal. Este programa especifica as operações de IO que devem ser realizadas e existe várias CCWs (Channel Command Word). CCW especifica uma operação, flags, um contador e uma locação de memória. Seis operações são definidas: read, write, read backward, control, sense, e transfer no canal.

### 11.3 - Organização do Canal de IO

Um canal de IO dirige e controla o fluxo de informações entre a memória e os dispositivos de IO permitindo a CPU processar concorrentemente com operações de IO. O canal pode funcionar independente ou compartilhar recursos da CPU.

Os canais podem executar as seguintes funções:

- aceita uma instrução de IO da CPU
- endereça um dispositivo de IO especificado por uma instrução de IO
- traz uma CCW de programa de canal na memória
- decodifica as CCWs
- testa se uma CCW é válida
- executa as funções da CCW
- coloca sinais de controle na interface de IO
- aceita sinais de controle da interface de IO
- transfere informações entre dispositivo de IO e memória
- verifica a paridade dos bytes transferidos
- conta o número de byte transferidos
- recebe e mantém informações do status de dispositivo de IO
- emite requisição de interrupção para a CPU
- sequencia e controla interrupções de IO dos dispositivos
- atualiza a locação 64 da memória com informações do status para aceitar interrupção
- armazena a informação de status na locação 64 quando requisitada pela CPU.

#### 11.4 - Projeto e Operação dos Canais

Três tipos se fazem presentes: multiplexador de byte, seletor e multiplexador de bloco. Cada um caracteriza o modo de operação, o modo do byte é utilizado por dispositivos lentos e o modo burst pelos equipamentos rápidos. Todos os três tipos de

canais podem ocupar o modo de operação burst.

O canal seletor possui apenas um subcanal enquanto o byte multiplexer e block multiplexer podem ter mais.

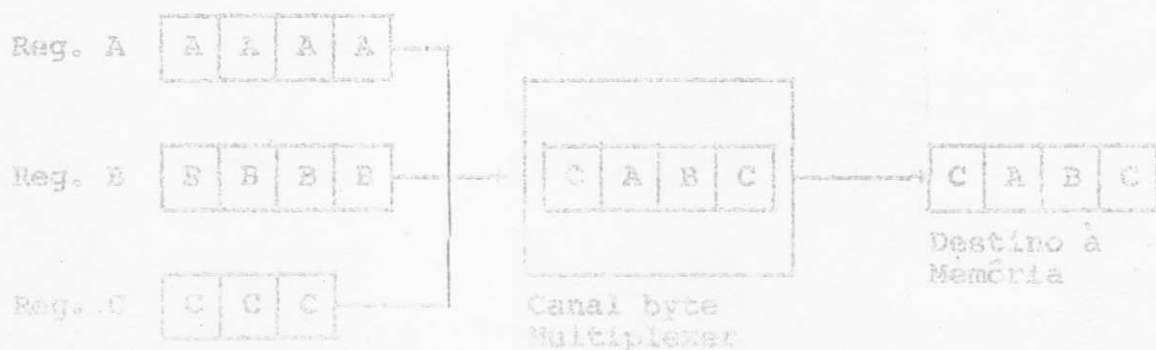
Cada canal tem circuito lógico e um Registrador interno para manter as seguintes informações de um dispositivo ativo:

- chave de proteção
- endereço de dados
- comando do código de operação
- CCWs flags
- contador de byte
- status de canal, e o endereço da próxima CCW

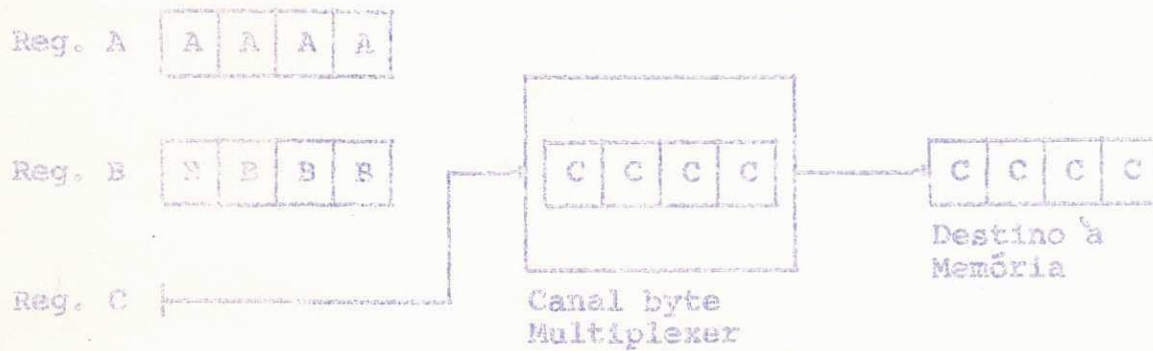
A cada 1024 bytes da memória provê uma UCW(unit control word) podendo ter até 1024 UCW.

No canal byte multiplexer utiliza o canal multiplexador IBM 2870, podendo suportar até 256.

#### Multiplexer Mode



### Burst Mode



O canal seletor pode estar integrado à CPU ou fora, neste caso utiliza o canal seletor IBM 2860, podendo ter até 8 unidades de controle e 256 dispositivos de IO.

O canal é caracterizado por:

- programa de canal são iniciadas antes que devam estar no modo burst
- blocos de dados são interleaved
- facilidades de canal são liberados depois de uma operação de transferência
- facilidades de canal são liberadas para uma certa operação no dado.

Como características dos canais o projeto inclui comando de correção de erros automaticamente, no modelo 165 pode ser extendido para acoplar mais 12 canais de IO. Provê ainda possibilidade de ligar 2 sistemas 370 através de um canal. A velocidade de transferência do canal multiplexador de bloco normalmen

te é de 1.5 Mbytes podendo ir até 3 Mbytes por segundo no modelo 165.

### 11.5 - Endereçamento de IO, Instruções e Comandos

Tem dois estados básicos: estático e operativo. Os dispositivos de IO que tem acesso via canal tem um endereço de 16 bits, sendo os 8 de mais alta ordem o endereço do canal e os outros 8 da unidade de controle. No canal multiplexador identifica o subcanal.

Instruções de IO são de formato SI e tem o seguinte formato:



Calculado o endereço efetivo teríamos:



Exemplos:

START IO SIO  
Teste IO TIO  
Teste Canal TCH

A palavra de endereço do canal (CAW) fica na loca  
ção 72 formato:

KEY	0000	Endereço do Comando
0 3 4	7 8	31

Comando de canal tem o seguinte formato:

Código de Comando	Endereço de Dados	Flags	0000	///	Contador
0 7 8	31 32	36 37	39 40	47 48	63

As instruções de IO são privilegiadas e só podem  
ser executadas no estado de supervisor.

### 11.6 - Interrupção de IO

Pode ser iniciada por um canal ou dispositivo de  
IO. A prioridade da requisição de interrupção é baseada no tipo  
de canal e seu endereço e a prioridade é indicada pelo código  
de interrupção, posição sobre "Cabo de Interface do IO e outros".







A tradução de endereço é controlada pelo bit translation mode na PSW e pelo conjunto de bits de controle dos Registradores de controle 0 a 1. Controles adicionais estão localizados nas tabelas de tradução.

Quando o DAT (Dynamic Address Translation) estiver instalado pode funcionar tanto no modo de conversão como sem conversão de endereço. O bit S da PSW que dizê, quando 0 (zero) o próprio endereço lógico é o endereço real.

O Registrador de controle 0 provê 4 bits para controle do tamanho de página e de segmento.

bit 6-9 do Reg. Page size

01 2 kbytes

10 4 kbytes

bit 11-12 do Reg. Segment size

00 64 kbytes

10 1 Mbytes

Os bits 0-25 do Registrador de Controle dá o comprimento da tabela de segmento e o endereço inicial.

LENGTH	SEGMENT-TABLE ADDRESS	
0 7 8	25 26 31	

A tabela de segmento tem as seguintes informações:

LGTH	0000	PAGE-TABLE ADDRESS	00	I
0	3 4	7 8	28 29 30	31

- LGTH - Complemento de tabela de páginas
- PAGE-TABLE ADDRESS - Endereço inicial de páginas
- I - Assinala se a tabela é válida ou não

A tabela de páginas possui as seguintes informações:

a) Página de 4 kbytes

PAGE ADDRESS	I	0	0	/
0	11 12	13	14	15

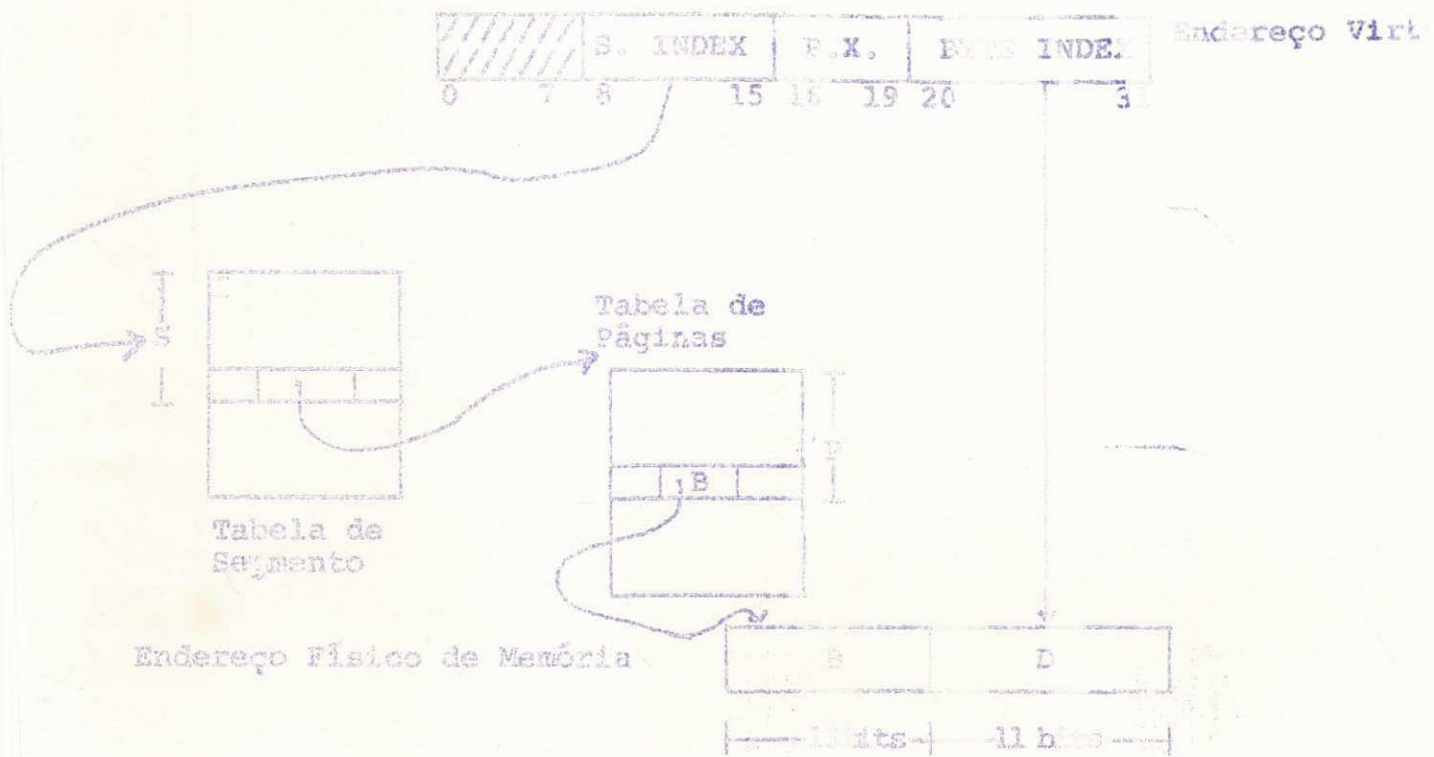
b) Página de 2 kbytes

PAGE ADDRESS	I	0	/
0	12 13	14	15

PAGE ADDRESS - Dá os bytes mais a esquerda do endereço real

- I - Indica se a conversão de endereço é possível
- 0 - impossível
- 1 - possível

### ENDEREÇO VIRTUAL E CONVERSÃO DE MEMÓRIA



13 - BIBLIOGRAFIA

- 13.1 - Harry Katzan, Jr. - Computer organization and the system/370  
Van Nostrand Reinhold Company - 1971
- 13.2 - David K. Hsiao - System Programming, Concepts Operating  
and Data Base Systems
- 13.3 - Stuart E. Madnick and John J. Donovan - Operating Systems  
McGraw-Hill Book Company, 1974
- 13.4 - Esorge Struble - Assembler Language Programming: The IBM  
system/370  
Addison-Wesley Publishing Company, 1971
- 13.5 - IBM GA22-7000-2 - Principles of Operation  
International Business Machines Corporation, 1972
- 13.6 - Herbert Heilerman - Digital Computer System Principles  
McGraw-Hill Book Company, 1973

jps/.