

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

IMPLEMENTAÇÃO DE UM SISTEMA DE PROGRAMAÇÃO
DE MÁQUINAS COM COMANDO NUMÉRICO (EXAPT)
EM MINI-COMPUTADOR NACIONAL

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE
SANTA CATARINA PARA A OBTENÇÃO DO GRAU DE MES
TRE EM ENGENHARIA

JOVELINO FALQUETO

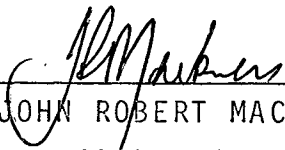
FLORIANÓPOLIS
SANTA CATARINA - BRASIL
NOVEMBRO - 1981

IMPLEMENTAÇÃO DE UM SISTEMA DE PROGRAMAÇÃO
DE MÁQUINAS COM COMANDO NUMÉRICO (EXAPT)
EM MINI-COMPUTADOR NACIONAL

JOVELINO FALQUETO

ESTA DISSERTAÇÃO FOI JULGADA PARA OBTENÇÃO DO TÍTULO DE
"MESTRE EM ENGENHARIA"

ESPECIALIDADE ENGENHARIA DE PRODUÇÃO E SISTEMAS, E APROVADA EM
SUA FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO



PROF. JOHN ROBERT MACKNESS, Ph.D.
COORDENADOR

APRESENTADA PERANTE A BANCA EXAMINADORA COMPOSTA DOS PROFESSORES:



JOHN ROBERT MACKNESS, Ph.D.
PRESIDENTE



AUREO CAMPOS FERREIRA, Ph.D.
ORIENTADOR



BERND EMIL HIRSCH, Dr. - Ing.
CO-ORIENTADOR



0.249.306-4

UFSC-BU

À Yuta e à Letícia

A G R A D E C I M E N T O S

Agradeço:

- aos Professores Aúreo Campos Ferreira, Bernd Emil Hirsch, John Robert Mackness e Caspar Erich Stemmer pelo empenho abnegado, devotado à execução da implementação de um sistema APT em mini-computador nacional;

- ao Bernd Richard Hoffmann, cujo conhecimento profundo do Sistema EXAPT de Programação CN em particular e de mini-computadores em geral, possibilitaram atingir os objetivos visados;

- às entidades: Associação EXAPT, de Aachen, RFA, Conselho Nacional de Desenvolvimento Científico e Tecnológico, Fundo de Incentivo à Pesquisa Técnico-Científica, do Banco do Brasil, à Gesellschaft für Mathematik und Datenverarbeitung, à Secretaria Especial de Informática, e à Universidade Federal de Santa Catarina, que tiveram participação direta na obtenção dos meios necessários à elaboração deste trabalho.

S U M Á R I O

	Pag.
CAPÍTULO I	
1. INTRODUÇÃO	1
1.1. O Problema e sua Relevância	1
1.2. Visão Global do Trabalho	6
1.3. Objetivos do Trabalho	6
1.4. Programação Manual e Programação com auxílio de Computador	7
1.4.1. Introdução	7
1.4.2. Programação Manual	10
1.4.3. Programação com Auxílio de Computador	14
1.4.4. Utilização da Programação com Auxílio de Computador	17
CAPÍTULO II	
2. AS LINGUAGENS DE PROGRAMAÇÃO E O SISTEMA DE PROGRAMAÇÃO EXAPT	19
2.1. Família APT	19
2.1.1. Processador e Pós-Processador	22
2.2. Outras Linguagens	25
2.3. O Sistema EXAPT de Programação	26
2.3.1. Linguagem de Programação	26
2.3.2. O Processador EXAPT	34
2.3.3. Os Pós-Processadores EXAPT	47
2.3.4. Programas Auxiliares	47

CAPÍTULO III

3.	IMPLEMENTAÇÃO DO PROCESSADOR EXAPT-BÁSICO E DO SISTEMA DAFES	54
3.1.	Introdução	54
3.2.	O Computador Suporte	56
3.2.1.	O "Hardware"	56
3.2.2.	O "Software"	58
3.3.	Sequência da Implementação	59
3.3.1.	Organização do Sistema EXAPT no Computador	59
3.3.2.	Características dos Módulos a Implementar	63
3.3.3.	Passos da Implementação	65
3.4.	Utilização do Processador EXAPT-BÁSICO	74
3.5.	Utilização do Sistema DAFES	74

CAPÍTULO IV

4.	MANIPULAÇÃO DE UM SISTEMA CN IMPLEMENTADO EM MINI-COMPUTADOR	77
4.1.	Introdução	77
4.2.	Pós-Processadores	78
4.2.1.	Conceituação	78
4.2.2.	Variáveis Influentes nos Pós-Processadores	80
4.2.3.	Entrada/Saída	86
4.2.4.	Construção do Pós-Processador	96
4.2.5.	Pós-Processador Geral (GNPP)	101
4.3.	Programas Auxiliares	106
4.3.1.	Programas de Simulação Gráfica	106
4.3.2.	Manipulação de Fitas Perfuradas	108

CAPÍTULO V

5.	RUMOS DO CN NA DIREÇÃO DO CAD/CAM	111
5.1.	Gráficos por Computador	115

CAPÍTULO VI

6.	CONCLUSÕES, RECOMENDAÇÕES E SUGESTÕES PARA FUTUROS TRABALHOS	118
6.1.	Conclusões	118
6.2.	Contribuições	119

6.3.	Sugestões para Futuros Trabalhos	120
ANEXO 1	- Organização Estrutural dos Módulos do Processador EXAPT e do Sistema DAFES	127
ANEXO 2	- Mudanças no Processador EXAPT e no sistema DAFES devidas ao Compilador e Sistema Operacional do Computador SISCO MB-8000	136
ANEXO 3	- Programas e Arquivos Auxiliares da Implementação	143
ANEXO 4	- Método de Utilização do Processador EXAPT	158
ANEXO 5	- Programa PLOT e Exemplos de Peça Fresada e Usinada, Programadas no Sistema EXAPT Implementado no SISCO MB-8000	163
ANEXO 6	- A História do Comando Numérico e o Método de Fabricação CN	178

LISTA DE FIGURAS

	Pag.
FIGURA 1 - Comportamento do custo de "Hardware" e de "Software" no tempo	3
FIGURA 2 - Desenvolvimento dos custos de processamento de dados	3
FIGURA 3 - Programação manual	11
FIGURA 4 - Programação com auxílio de computador	15
FIGURA 5 - Programação manual & Programação por Computador: comparativo entre custos e número de blocos	18
FIGURA 6 - Elementos construtivos das linguagens de programação da família APT	21
FIGURA 7 - Processamento de um programa de peça em linguagem tipo APT, obtendo-se meios de controle para diversas máquinas	23
FIGURA 8 - Os 3 Modos de utilização da linguagem EXAPT	27
FIGURA 9 - Determinação automática da Seqüência, Ferramentas, Dados de Corte e Trajeto das Ferramentas	29
FIGURA 10 - Entradas e saídas dos processadores do sistema EXAPT de programação CN	30
FIGURA 11 - Programa exemplo em EXAPT-BÁSICO, sem existência de arquivo de ferramentas	32
FIGURA 12 - Programa exemplo em EXAPT com extensão para tecnologia automática de furação e fresagem (arquivo de ferramentas e de materiais previamente construídos)	33

FIGURA 13 - Processador EXAPT: Estágios principais do processamento dos comandos (módulo EINGAB)	35
FIGURA 14 - Processador EXAPT: estrutura modular-módulos	37
FIGURA 15 - Arquivos administrados pelo Sistema EXAPT de programação CN	44
FIGURA 16 - Método de trabalho do programa de manipulação de fitas e arquivos JANUS	50
FIGURA 17 - Modo de trabalho dos módulos de gerenciamento de banco de dados MAPEX e DAFES	52
FIGURA 18 - Implementação de sistemas de "software" em computador	55
FIGURA 19 - Configuração existente na UFSC para o computador suporte	57
FIGURA 20 - Tipos de organização do Processador EXAPT em grandes computadores	60
FIGURA 21 - Organização do Processador EXAPT em mini-computadores	61
FIGURA 22 - Processador EXAPT: estrutura do sistema modular	62
FIGURA 23 - Estrutura do módulo de manuseio de dados MODIOS (MODIOS - Modular input/output system)	63
FIGURA 24 - Inter-relacionamento entre os grupos de subrotinas constituintes do módulo EINGAB	66
FIGURA 25 - Fluxograma adotado na implementação SISCO MB-8000	67
FIGURA 26 - Função do programa MTW processado no computador auxiliar PDP-11/34	71

FIGURA 27 - Função do programa RTAP3 no computador de teste NOVA	72
FIGURA 28 - Diagrama das partes componentes dos p̄os-processadores	79
FIGURA 29 - P̄os-processador: variáveis de maior influência	81
FIGURA 30 - Ferramentas de comprimentos muito diferentes ocasionam problemas que o p̄os-processador pode resolver	85
FIGURA 31 - Representação de segmento de CLDATA com 3 registros de 5, 2 e 4 palavras	87
FIGURA 32 - Exemplo de transformação de comando EXAPT em registro CLDATA	90
FIGURA 33 - Princípio de construção da saída de um p̄os-processador	93
FIGURA 34 - Funções desenvolvidas pelo p̄os-processador nas suas diversas fases de processamento	97
FIGURA 35 - Trabalho de processamento na 1ª fase	99
FIGURA 36 - Organização em lista de armazenamento virtual intermediário (subrotina DAI0)	100
FIGURA 37 - Trabalho de processamento na 2ª e 3ª fases	102
FIGURA 38 - Estrutura do programa GNPP para furação e fresagem	103
FIGURA 39 - Construção de um arquivo, usando dois outros já existentes, através do programa auxiliar MERGE	110

FIGURA 40 - Avanço da automatização do tratamento da informação na produção de peças usinadas	112
FIGURA 41 - Participação dos diversos departamentos de uma empresa de um banco de dados central	115
FIGURA 42 - As fases do projeto (o que é produzido), de planejamento (como e onde é produzido) e de controle (quando é produzido) da produção, se encaminham aceleradamente para a automatização	122
FIGURA 43 - Fases do processamento do módulo EINGAB	129
FIGURA 44 - Carregamento do módulo EINGAB	130
FIGURA 45 - Arquivos produzidos pela execução dos módulos do processador EXAPT	161
FIGURA 46 - Esquema da estrutura dos módulos componentes do Programa PLOT	164
FIGURA 47 - Exemplo de programa de peça com posicionamento e seu traçado no "plotter" com o programa PLOT	167
FIGURA 48 - Foto das peças HASTE ISO-40 e CRUZ, usinadas com programas obtidos da implementação no SISCO MB-8000	168
FIGURA 49 - Saída do pós-processador CSEPEL (peça HASTE ISO-40)	173
FIGURA 50 - Máquinas CN instaladas nos USA e na RFA entre os anos de 1954 e 1978	181
FIGURA 51 - Processamento integrado de informações na produção	187
FIGURA 52 - Componentes principais de um sistema CN	189

LISTA DE TABELAS

	Pag.
TABELA 1 - Elementos estruturais dos módulos EXAPT	38
TABELA 2 - Modos de implementação da linguagem EXAPT em diferentes computadores	64
TABELA 3 - Paralelo entre as informações do ar- quivo CLDATA e as da fita perfurada	92
TABELA 4 - Formatos de saída de algumas funções para o pós-processador NBH-65 da UFSC	95
TABELA 5 - Funções e módulos do programa auxiliar PLOT	165
TABELA 6 - Programação por computador. Sistemas instalados no Brasil	183

R E S U M O

O objetivo principal deste trabalho é a transferência, para um mini-computador nacional, de um sistema da família APT utilizado na programação de Máquinas-Ferramentas com Comando Numérico (MFCN).

Inicialmente, um estudo resumido da evolução do Comando Numérico é apresentado. Em seguida são analisadas as características principais da programação manual, da programação por computador e das linguagens da família APT e simbólica.

Tomando como exemplo uma das linguagens da família APT, é feito um estudo da construção do processador modular EXAPT e de seus pós-processadores.

Na seqüência descreve-se a metodologia de implementação do processador EXAPT, os problemas e soluções adotadas, bem como o método de uso do sistema.

Finalizando são apresentadas as conclusões, as contribuições e as sugestões para trabalhos futuros, além de dois exemplos de uso do sistema, programando uma peça para torno e outra para centro de usinagem.

A B S T R A C T

The primary objective of this dissertation is to document the transfer to a Brazilian manufactured mini computer of one system of APT family of languages for programming numerically controlled machine tools.

Initially a brief summary of the evolution of NC is presented. This is followed by an analysis of the main characteristics of manual programming and computer programming of machine tools using languages of the APT family.

Using as an example one of the APT family languages, a study of the construction of the EXAPT modular processor and post-processors is then made. This is followed by the presentation of the methodology adapted for implementing the EXAPT processor together with the problems and recommended solutions.

Finally in the conclusions, recommendations for future development of this work are made and two examples of the system with a lathe and a machining centre are described.

C A P Í T U L O I

1. INTRODUÇÃO

1.1. O Problema e sua Relevância

Dentro do processo produtivo, as MFCN desempenham um papel muito importante, principalmente ao se considerar que cada vez mais se deve produzir com máxima eficiência. Até o advento do CN, a preocupação maior foi sempre descobrir maneiras de reduzir o tempo de produção, aumentando as velocidades de corte, ao fabricar uma peça por usinagem.

Com a inclusão de MFCN na produção, houve uma redução considerável nos tempos mortos (posicionamentos, preparação, troca de ferramentas, troca de peças, etc.) diminuindo ainda mais o tempo total de fabricação. Levando em conta que as MFCN têm sua eficiência otimizada no caso de pequenos e médios lotes, o tempo de produção destes lotes é normalmente pequeno, exigindo maior rapidez e flexibilidade na preparação dos programas, o que dificilmente pode ser conseguido com programação manual.

Dessa forma a programação manual começou a ser ineficiente, especialmente no caso de uma quantidade cada vez maior e diversificada de tipos de máquinas e controles. Além disso há casos em que a complexidade das peças tornaria a programação manual muito trabalhosa e por vezes extremamente complexa.

Considerando todos estes fatores, e uma série de outros a serem apresentados neste trabalho, a Programação com Auxílio de Computador (PAC) ou simplesmente Programação por Computador tornou-se de suma importância no mundo inteiro.

No Brasil a utilização de MFCN é relativamente recente pois as primeiras instalações têm, quando muito, 15 anos; [29] no entanto, o interesse no campo é grande e as perspectivas atuais de crescimento são muito promissoras.

Pode-se citar como pontos importantes deste desenvolvimento:

- Fabricação no Brasil de MFCN.
- Fabricação no Brasil de Controles CN.
- Número de empresas usuárias de MFCN em rápido crescimento.
- Alto número de empresas já decididas a entrar no ramo ou pelo menos interessadas na utilização deste tipo de maquinário.
- Institutos de pesquisa e ensino trabalhando no tema.

Quanto à programação, no Brasil são usadas as seguintes possibilidades:

- a) programação manual
- b) programação em grandes computadores
- c) programação em pequenos ou mini-computadores importados, com linguagens particularizadas.

Poucas são as empresas que, trabalhando com CN ou mesmo já decididas a fazê-lo, têm a seu dispor um grande centro de processamento de dados.

Também os sistemas instalados em micro-computadores não satisfazem sempre às necessidades dos usuários, de vez que, geralmente, tratam-se de sistemas particularizados para determinados processos de fabricação.

Além disto observa-se que é importante analisar o comportamento do setor de informática, no qual o CN se apóia.

Pode-se começar esta análise considerando que ao lado de um barateamento e de uma capacidade crescentes com o passar dos anos, do "hardware" fabricado, nota-se um encarecimento do custo de "software", conforme figuras 1 e 2 [24].

Uma das causas do aumento do custo do "software" é o crescente papel do fator mão de obra, o que, conjunta-

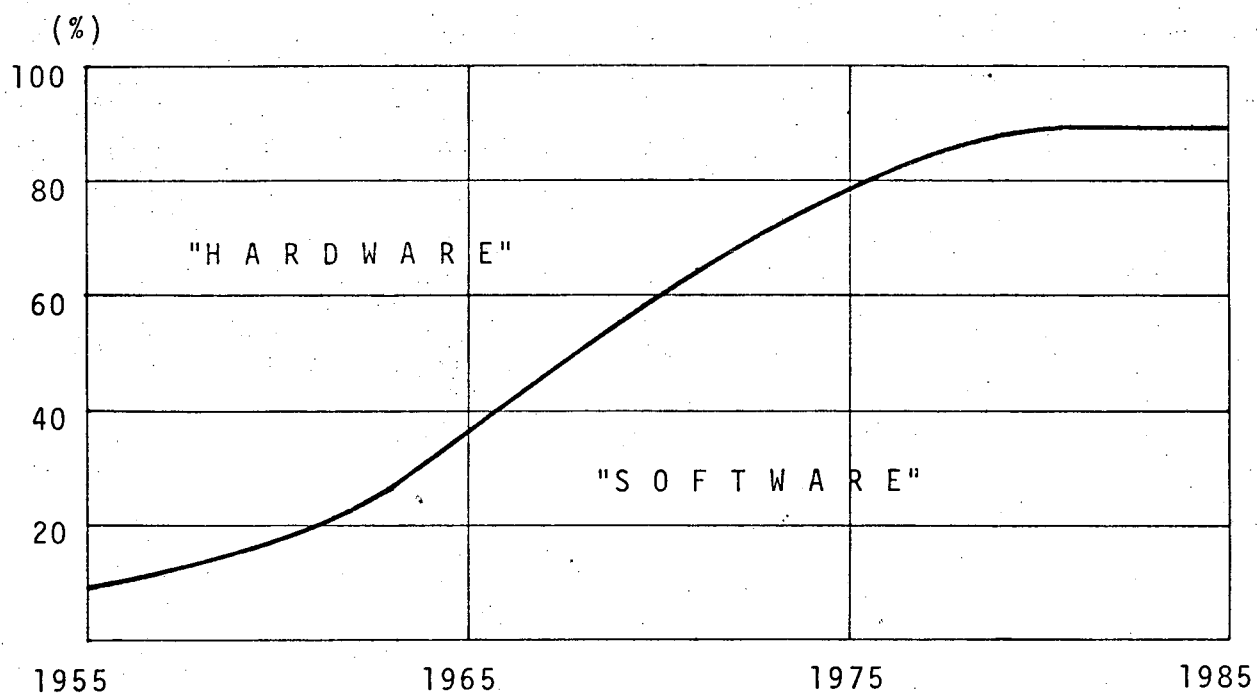


FIGURA 1 - Comportamento do custo de "Hardware" e de "Software" no tempo.

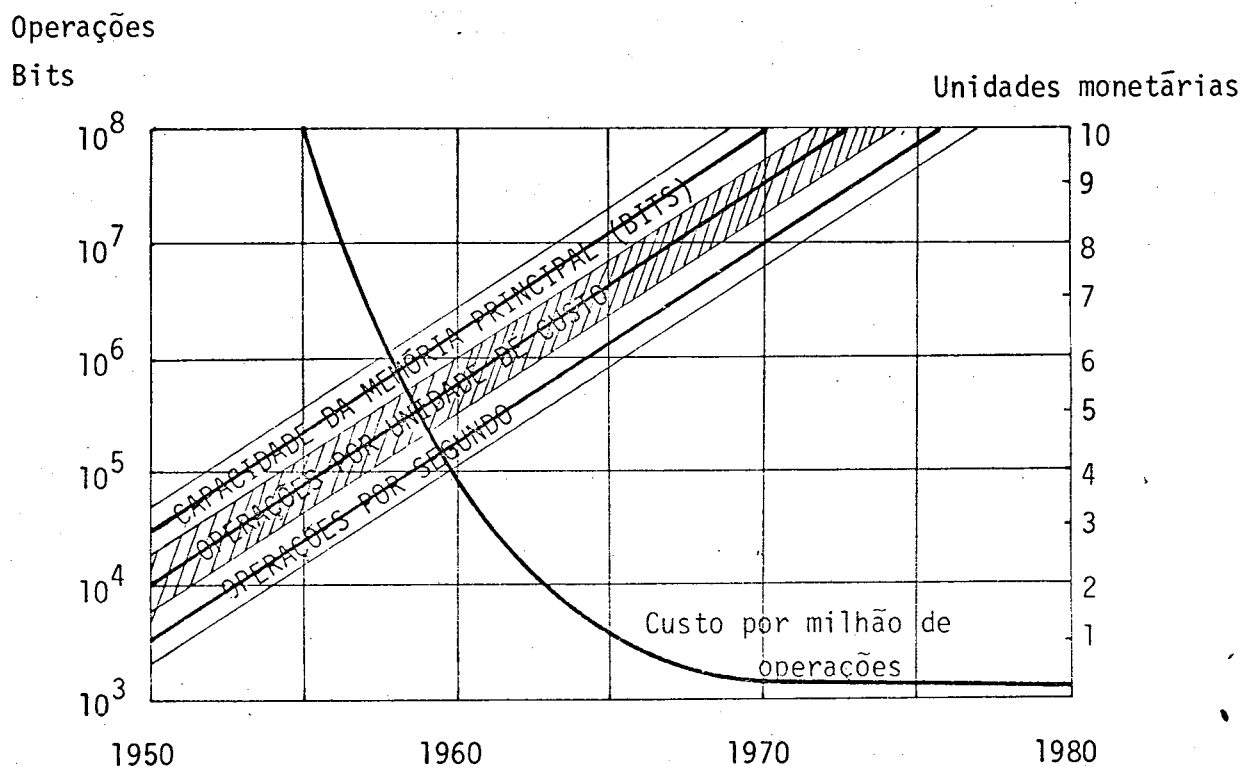


FIGURA 2 - Desenvolvimento dos custos de processamento de dados.

mente com o barateamento do "hardware", fortalecem a posição da programação por computador, que ganha em flexibilidade e qualidade em relação à manual.

Assim um ponto importante para o desenvolvimento do CN no Brasil é a disponibilidade de um sistema de programação de máquinas CN por computador que seja universal, independente de controle, de computador, de processo de fabricação e implantável em mini-computador nacional.

Empresas que, por motivos os mais diversos, não têm possibilidade de utilização de um grande centro de computação mas estão interessadas em manter um mini-computador dedicado a este trabalho, poderão utilizar a programação por computador.

A Universidade Federal de Santa Catarina (UFSC), embora disponha de um grande centro de computação, resolveu adotar um "mini" para aplicações com CN e seria importante registrar aqui as razões que a levaram a isto.

A UFSC tem atuado em situação de ponta na área de CN desde 1969, quando implantou o 1º curso de CN no seu programa de Pós-Graduação em Engenharia Mecânica. O Convênio de Cooperação Técnica que mantém com a Universidade Técnica do Reno-Vestfália (RWTH), de Aachen, RFA, traz extraordinárias oportunidades para fortalecer esta situação de pioneirismo, dado que a RWTH tem sido o berço irradiador do desenvolvimento da programação por computador na Alemanha.

Na escolha de um sistema de programação CN, a UFSC optou por implantar, como exemplo da família APT, o sistema EXAPT, porque:

1. Está normalizado internacionalmente pela ISO, entre outras linguagens derivadas do APT;
2. É aplicável a todos os processos de fabricação por usinagem;
3. Permite o processamento de informações geométricas de forma compatível com o APT, bem como, as informações tecnológicas (ve-

- locidade de corte, avanço, ciclos de usinagem, etc.);
4. É portátil, isto é, pode ser implementado em mini-computadores com compilador FORTRAN IV disponível e capacidade suficiente de memória;
 5. Permite adaptações às experiências e técnicas já desenvolvidas e dominadas por cada usuário, em seu trabalho específico.
 6. A linguagem é continuamente aperfeiçoada e adaptada aos progressos da tecnologia pela Sociedade EXAPT, agremiação que foi fundada e que é controlada pelos próprios usuários do sistema.

Quanto ao computador suporte do sistema, enquanto os computadores importados podem ser encontrados em uma faixa bastante ampla de capacidade de armazenamento, os nacionais, até o momento atendendo a demanda dos "minis" e "midis" são os seguintes:

COBRA, EDISA, LABO, SID, SISCO [17]

Entre estes computadores que já vêm sendo usados em controle de processos, o computador SISCO MB-8000, por ser o único, até o momento do início deste trabalho, trabalhando com o compilador FORTRAN IV, e por suportar os periféricos necessários à programação CN, foi escolhido para implementação do sistema EXAPT.

Por tudo que foi exposto é importante se dispor no Brasil de um sistema de PAC, universalmente utilizado e para o qual se possa contribuir com adaptações e soluções próprias, para que a produção nacional de peças por CN possa se tornar mais eficiente, e assim possa competir em condição de igualdade no mercado internacional, além de trazer consigo a transferência de uma tecnologia que, embora mundialmente difundida e largamente utilizada, ainda é bastante incipiente no Brasil.

1.2. Visão Global do Trabalho

O trabalho tem no capítulo introdutório uma descrição do problema e sua relevância enfocando ainda os passos genéricos seguidos na programação manual e na programação por computador.

O capítulo II trata inicialmente das linguagens de programação, salientando aspectos importantes da família APT e da família de linguagens simbólicas. Na segunda parte se tem uma descrição bastante detalhada do sistema EXAPT - objeto da implementação - com visualização da construção modular do processador, dos pós-processadores e programas auxiliares.

O capítulo III discorre sobre a implementação desenvolvida no computador SISCO MB-8000, caracterizando-se o "hardware" e o "software" mínimos necessários. A seguir são apresentados os passos adotados na implementação e o modo de utilização do processador e do sistema DAFES de gerenciamento de banco de dados.

O capítulo IV enfoca o problema de criação de "software", sobretudo pós-processadores e programas auxiliares, para viabilizar a utilização eficiente de um sistema CN de PAC.

O capítulo V apresenta uma visão abrangente dos rumos que a fabricação e o projeto estão tomando, à medida que o computador vai assumindo cada vez mais a realização de tarefas nestas áreas.

No capítulo VI são apresentadas as conclusões e as contribuições do trabalho, bem como sugestões para trabalhos futuros.

1.3. Objetivos do Trabalho

Os objetivos básicos do presente trabalho podem ser resumidos em:

- Implementação, em mini-computador nacional, de um sistema da família APT, utilizado na programação de MFCN.

- Criação de condições de manuseio do sistema implementado.

O trabalho de implementação se subdivide em:

- a) Implementação propriamente dita, isto é, instalar e fazer funcionar o referido sistema em mini-computador nacional.

- b) Elaborar documentação detalhada sobre o funcionamento do sistema para consulta posterior.

- c) Relatar a sistemática de implementação a dotada no trabalho, seus problemas e soluções.

O trabalho de criação de condições de manuseio compreende:

- a) Documentar a construção de processadores tipo APT.

- b) Documentar as finalidades e a metodologia de construção de pós-processadores.

- c) Construir programas auxiliares ao sistema.

1.4. Programação Manual e Programação com Auxílio de Computador

1.4.1. Introdução

Uma peça pode, teoricamente, ser construída com a utilização de uma MFCN, de quatro modos:

1. Entrada Manual de Dados: Acionando diretamente as chaves apropriadas do controle. Estas chaves representam funções executáveis pela MFCN. A máquina executará as funções solicitadas e usará a peça.
2. Entrada através de um portador de dados: Prever todas as operações necessárias à execução da peça e registrá-las em um portador de dados adaptado ao controle. Subme-

ter em seguida este portador ao controle que o interpretará e solicitará da máquina, a execução das funções nele codificadas, usinando a peça.

3. DNC (Direct Numerical Control: Controle Numérico Direto): as informações de controle ficam armazenadas em memórias auxiliares de computadores, que são ligados diretamente ao gabinete da MFCN. A transmissão dos dados é direta, isto é, a MFCN trabalha "on line" com o computador.
4. CNC (Computerized Numerical Control: Controle Numérico Computadorizado): A entrada de dados é feita em uma das três maneiras anteriores, uma única vez por peça e fica armazenada na memória do CNC, repetindo-se o programa memorizado tantas vezes quantas peças forem usinadas.

No primeiro método, a seqüência de operações necessárias à execução da peça não está registrada em um meio que seja diretamente inteligível (interpretável) pelo controle. Estas operações estão momentaneamente na memória do operador, ou copiadas de algum meio independente do controle e a ele transmitidas através de comandos diretos (as chaves que identificam as funções da máquina).

Nos demais métodos as operações necessárias são codificadas em símbolos que serão posteriormente decodificados pelo controle, para comandar a máquina. Estes símbolos codificados são registrados em um portador de dados. A seqüência de operações necessárias é previamente estudada e construída por alguém.

A construção de um número finito de passos seqüenciais lógicos, executáveis por uma MFCN, visando a usinagem de uma peça chama-se PROGRAMAÇÃO DE CONTROLE NUMÉRICO.

A programação tem por objetivo a elaboração da seqüência dos comandos e, se necessário, do portador de dados, com os passos a serem executados pela MFCN. Esta elaboração pode

ser feita por dois métodos básicos:

- PROGRAMAÇÃO MANUAL
- PROGRAMAÇÃO POR COMPUTADOR

Toda máquina pode executar um determinado número de operações bem definidas, tais como:

- Ligar ou desligar o óleo refrigerante.
- Avançar a ferramenta de um ponto a outro.
- Iniciar, acelerar, desacelerar ou parar a rotação da árvore.

Estas operações serão executadas sob o comando do controle, que, por sua vez, as recebe do operador (via chaves do gabinete), ou do portador de dados, ou ainda diretamente do computador.

Se o método escolhido é o portador de dados, este contém as instruções do programa, que devem ser as que a MFCN "entende" e executa.

Com o desenho da peça, o programador pode construir uma série seqüencial de passos constituídos unicamente de funções executáveis pela MFCN, que, uma vez executados, resultam na obtenção da peça acabada. A seqüência de passos chama-se Programa da Peça. Para construí-lo, o programador, a partir do desenho da peça e de informações sobre a MFCN no qual ela será executada, calcula a trajetória da ferramenta, estabelece os instantes de atuação do fluido de corte, determina as rotações e os avanços que serão utilizados, etc.

Como se nota, o programador trabalha exclusivamente com as funções diretamente executáveis pela sua MFCN, isto é, utiliza linguagem de sua máquina (fazendo um paralelo com o termo usado nas linguagens de computação).

A este método de obtenção do portador de dados chama-se PROGRAMAÇÃO MANUAL.

Por outro lado o programador poderá também dispor de uma Linguagem de Programação, isto é, ele não se referenciará diretamente às funções executáveis pela máquina, mas utili-

zará um sistema de símbolos gerais que posteriormente serão processados por um computador, resultando no portador de dados que comandará a MFCN durante a usinagem.

A este segundo método chama-se **PROGRAMAÇÃO POR COMPUTADOR**.

1.4.2. A Programação Manual

A programação manual é usada principalmente para furação, torneamento e fresagem simples e outras usinagens ou processos de fabricação, de programação não muito complexa. Peças ou operações de máquina complexas, que geralmente envolvem o uso de centenas de informações completas (blocos de programa), são mais economicamente programadas pelo método automático, isto é, por intermédio do computador.

a) Passos Seguidos na Programação Manual

Quando se leva a efeito a programação de uma peça pelo método da programação manual, os seguintes passos normalmente são seguidos (figura 3):

1. verificação das fases do trabalho
2. estudo do desenho da peça
3. elementos de contorno
4. distância da ferramenta à peça
5. cálculos matemáticos
6. obtenção das informações tecnológicas
7. verificação de colisões
8. manuscrito do programa
9. perfuração e listagem do programa

Deve-se acrescentar ainda a estes passos as seguintes funções exercidas pelo programador manual:

- Preparação da ficha de processo dando a sequência das operações.

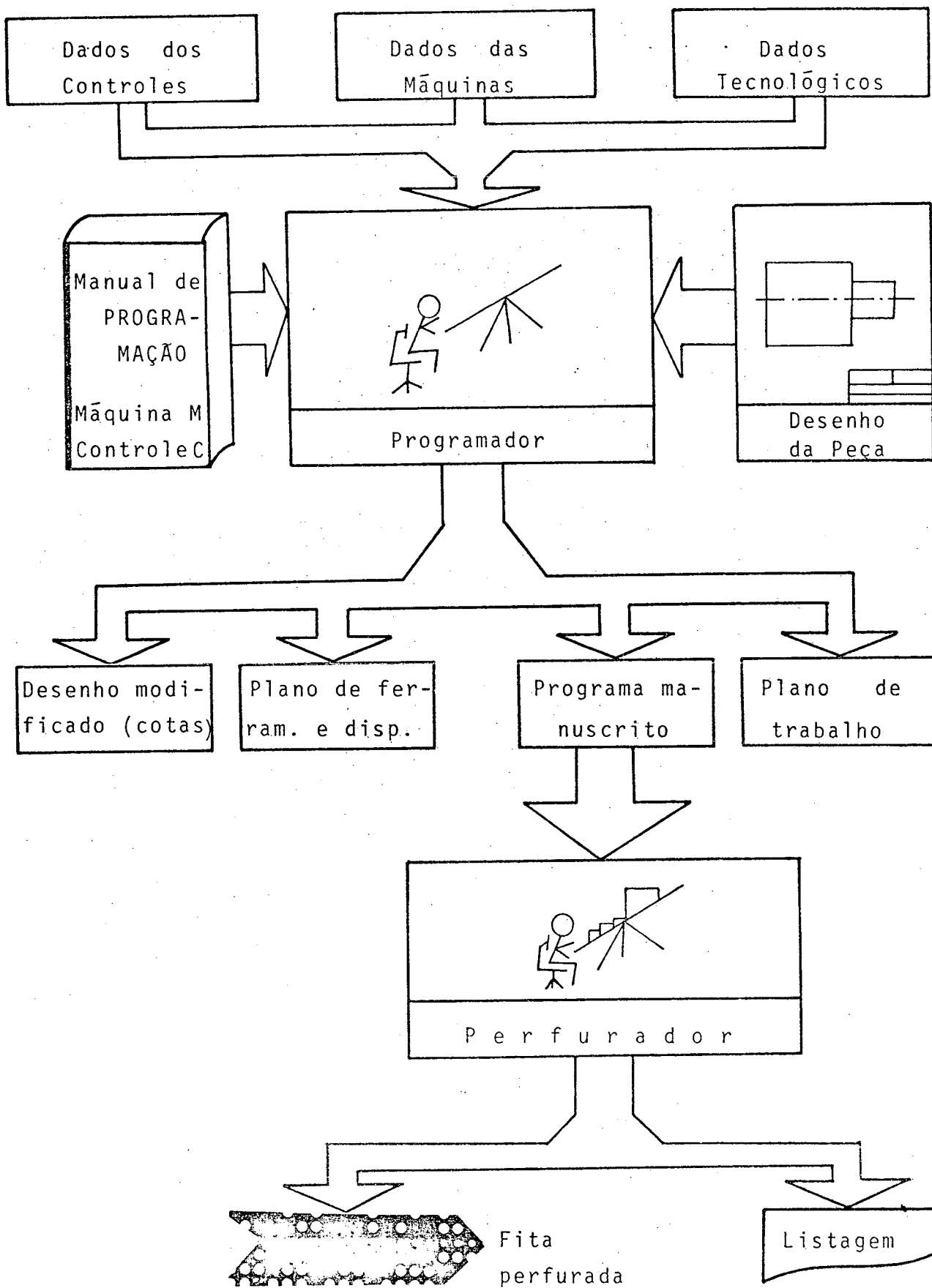


FIGURA 3 - Programação manual.

- Preparar instruções relativas às ferramentas.
- Preparar informações relativas ao dispositivo de fixação.

b) Preparação do Programa

Para escrever o programa da peça, usa-se uma ficha de processo, baseada no desenho da peça.

Esta ficha contém usualmente uma tabela fornecendo as coordenadas de furos, pontos de mudança de direção no contorno, etc., que, no total, caracterizam o caminho a ser seguido pela ferramenta.

Para fazer o programa, são necessários também dados sobre: ferramenta, dispositivos de fixação, e dados de programação, isto é, dados sobre as funções executáveis pela máquina.

O desenho da peça deve ser redimensionado, de modo a conter apenas dimensões absolutas ou incrementais, dependendo do tipo de controle, e referenciadas ao sistema de coordenadas da MFCN.

Deve-se escolher um ponto como origem da peça e um ponto para fixação na máquina. Além disso, desenhos separados para cada posição de fixação da peça devem ser feitos e neles deverão constar os dispositivos de fixação.

A seqüência de operações de máquina é separada em passos individuais, cada qual programado em um bloco de informação.

O contorno da peça é dividido em segmentos lineares, circulares (e, mais raramente, parabólicos).

O programador deve calcular os parâmetros característicos destes segmentos lineares, circulares, tais como: extremos de segmentos retilíneos; extremos, raio e centro de segmentos circulares. Os pontos intermediários destes segmentos são automaticamente calculados pelo controle e é o que se denomina de

interpolação linear ou circular, respectivamente.

c) Ficha de Processo

Contêm todos os dados necessários para programação da peça. Esta ficha inclui geralmente:

- Desenho da peça.
- Dados da ferramenta e dos dispositivos de fixação.
- Tabela com a seqüência de operações.

d) Manuscrito (Programa)

Na programação manual, o programador calcula os movimentos individuais da máquina ferramenta baseado na ficha de processo. Registra estas informações em um manuscrito que se chama Programa da Peça, que contém então: dados dimensionais, velocidades, avanços, ferramentas, refrigerante, etc., bloco por bloco.

Conforme se nota pela análise da figura 3, a programação manual sobrecarrega o programador com uma série de informações rotineiras e cansativas, já que ele terá que resolver problemas tais como:

1. Escolher uma MFCN baseando-se no desenho peça, o que implica num perfeito conhecimento das possibilidades da máquina e das ferramentas.
2. Os desenhos não são geralmente cotados em relação à fixação da peça na máquina, isto é, as cotas fornecidas do desenho não podem ser utilizadas diretamente mas devem ser recalculadas para um sistema de coordenadas de referência.
3. Como geralmente as peças brutas e acabadas diferem grandemente em sua forma e como o

programador deve, em cada instante, conhecer o contorno da peça, direção de avanço, aresta de corte, para evitar colisões e fazer a usinagem de modo favorável, esta é feita em várias etapas. Esta decomposição deve ser feita levando-se em conta o custo final do produto acabado, (a peça) mas infelizmente os desenhos nada informam sobre estas etapas.

4. Todas as instruções devem ser levadas para o formato de entrada do controlador.
5. Se forem ainda introduzidos princípios de otimização de tempos e de carga das máquinas, o programador ficará mais sobrecarregado ainda.

Em resumo, na programação manual o programador se preocupa enormemente com a máquina em que será executada a peça e com seu controle, já que é neles que será feita a usinagem. Além disso, a programação manual contém muito trabalho rotineiro, repetitivo, com grande preocupação quanto a detalhes. Tal fato requer muita concentração do programador, o que torna o trabalho excessivamente cansativo e conseqüentemente, suscetível a erros.

1.4.3. Programação com Auxílio de Computador

Existe outra filosofia de programação que procura deixar a cargo do programador o fornecimento das informações que se relacionam diretamente com cada peça, e utiliza computadores para trabalhar informações comuns para todas as peças, como o tratamento matemático ou tecnológico.

Por exemplo, se deixam armazenadas em um computador as informações sobre a MFCN (Figura 4), que não variam com o tempo. Um "software", quando necessário é solicitado pelo programador, insere-as automaticamente no programa. Manualmente se trabalham as informações que variam a cada novo problema, diretamente ligadas à peça, e não as da MFCN, os cálculos de trajetória,

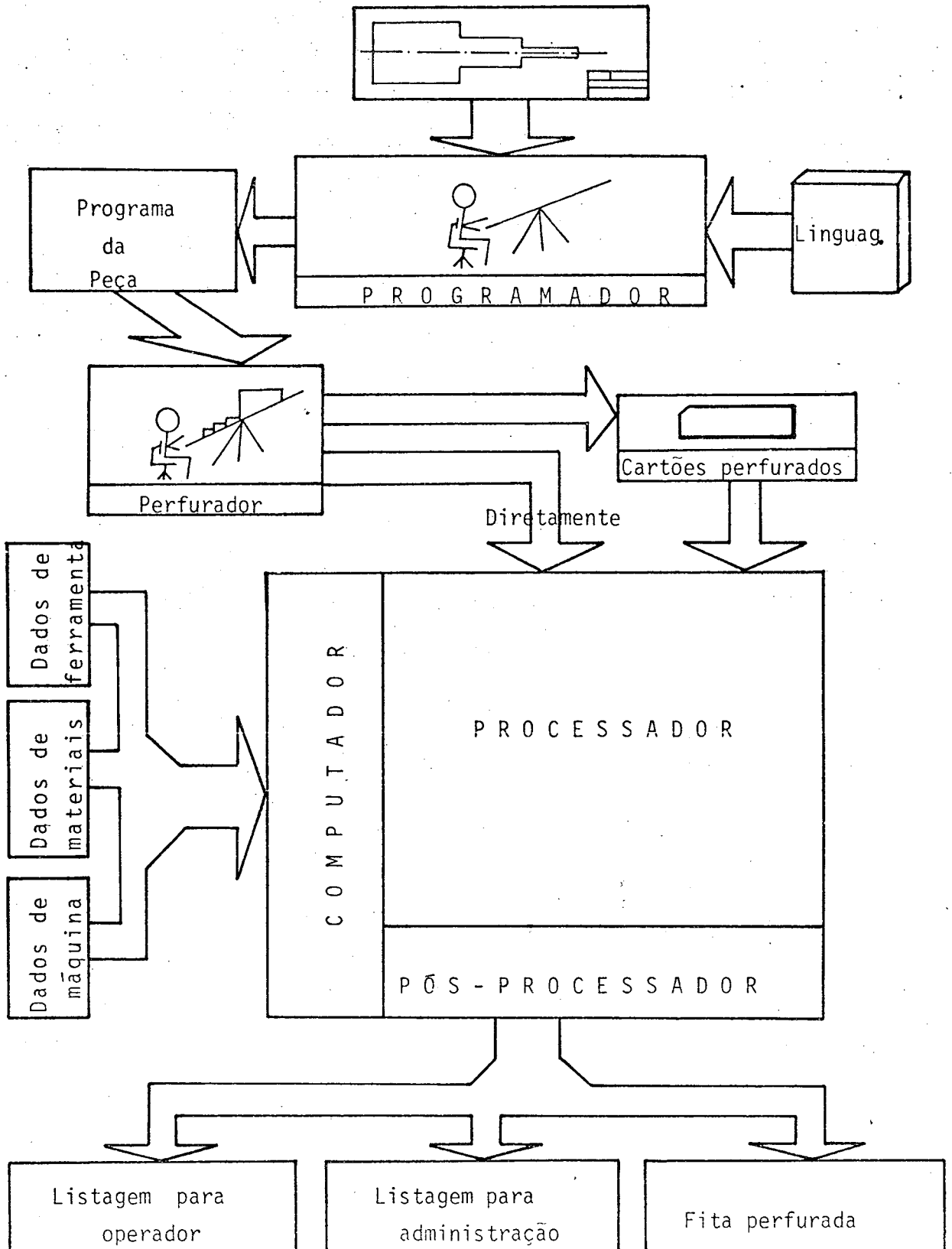


FIGURA 4 - Programação com auxílio de computador.

os valores de corte, avanço, etc., o que descarrega em muito o programador, e principalmente, tornando o sistema mais rápido, confortável, barato e seguro.

Na programação por computador (PAC) se transfere para este, o trabalho repetitivo e cansativo do cálculo de informações de contorno, escolha de velocidades, de avanços, de testes de colisão etc., liberando o programador para trabalhos mais criativos.

Na PAC o programador trabalhará agora não mais diretamente com as funções da MFCN, mas disporá de um instrumento intermediário, a linguagem de programação.

Por facilidade de trabalho, esta linguagem será tanto mais fácil e segura para o programador quanto mais semelhante à humana. O computador terá o trabalho de posteriormente transformar esta linguagem "humana" em outra mais compatível com sua estrutura construtiva, isto é, mudar de "humana" para "computacional".

O computador interpretará os comandos, executará as instruções dadas e produzirá o portador de dados (exemplo: fita perfurada).

Em resumo, a PAC supõe dois elementos importantes:

1. Uma linguagem de programação para codificação dos dados de entrada relativos à peça, à máquina/controle, em uma forma padrão, reconhecível por computador.
2. Programas de computador para interpretar a linguagem usada, e realizar os cálculos necessários para programar o portador de dados com as instruções para a MFCN.

A linguagem de programação descreve:

- geometria da peça
- caminho das ferramentas
- funções da MFCN

As linguagens de programação podem ser, quanto ao universo abrangido:

Universais (independentes de computador ou de MFCN).

Particulares (orientadas especificamente para determinada máquina, ou controle ou computador ou processo de produção).

Quanto à sua estruturação, as linguagens de programação podem ser:

- simbólicas
- mnemônicas

As linguagens simbólicas usam um ou dois sinais alfanuméricos para endereçamento e definições de funções. Por outro lado as mnemônicas utilizam palavras e símbolos de fácil memorização pelo programador.

As linguagens simbólicas têm a vantagem de requerer menor capacidade de memória do computador suporte e a desvantagem de serem orientadas para determinado computador, MFCN, ou processo de fabricação.

De modo geral, pode-se dizer que a programação com auxílio de computador é mais vantajosa para programas que tenham mais que uma centena de blocos (Figura 5, [27]). Além disso, a experiência tem mostrado que, em média, o tempo de geração e testes de um programa de peça em um sistema PAC é aproximadamente metade, e os custos ficam entre metade e três quartos, comparando-se com a programação manual [2].

1.4.4. Utilização da Programação com Auxílio de Computador

A utilização ou não de PAC depende de vários fatores podendo-se recomendar seu uso em situações tais como:

- Quando a complexidade da MFCN ou da peça requer translações, rotações ou grande número de ferramentas diferentes durante a usina-

gem.

- Quando se usinam peças diferentes, montadas seqüencialmente em uma mesa, a programação de todo o conjunto pode exigir mudanças de sistemas de coordenadas, o que se resolve facilmente com PAC.
- Quando o cálculo do contorno de uma peça exige a determinação matemática de muitos pontos isolados e de difícil obtenção.
- Quando se trabalha com muitas máquinas ou combinações máquinas-controles diferentes. Neste caso a programação manual é extremamente trabalhosa e insegura por exigir dos programadores enorme atenção para atender as peculiaridades de cada MFCN.

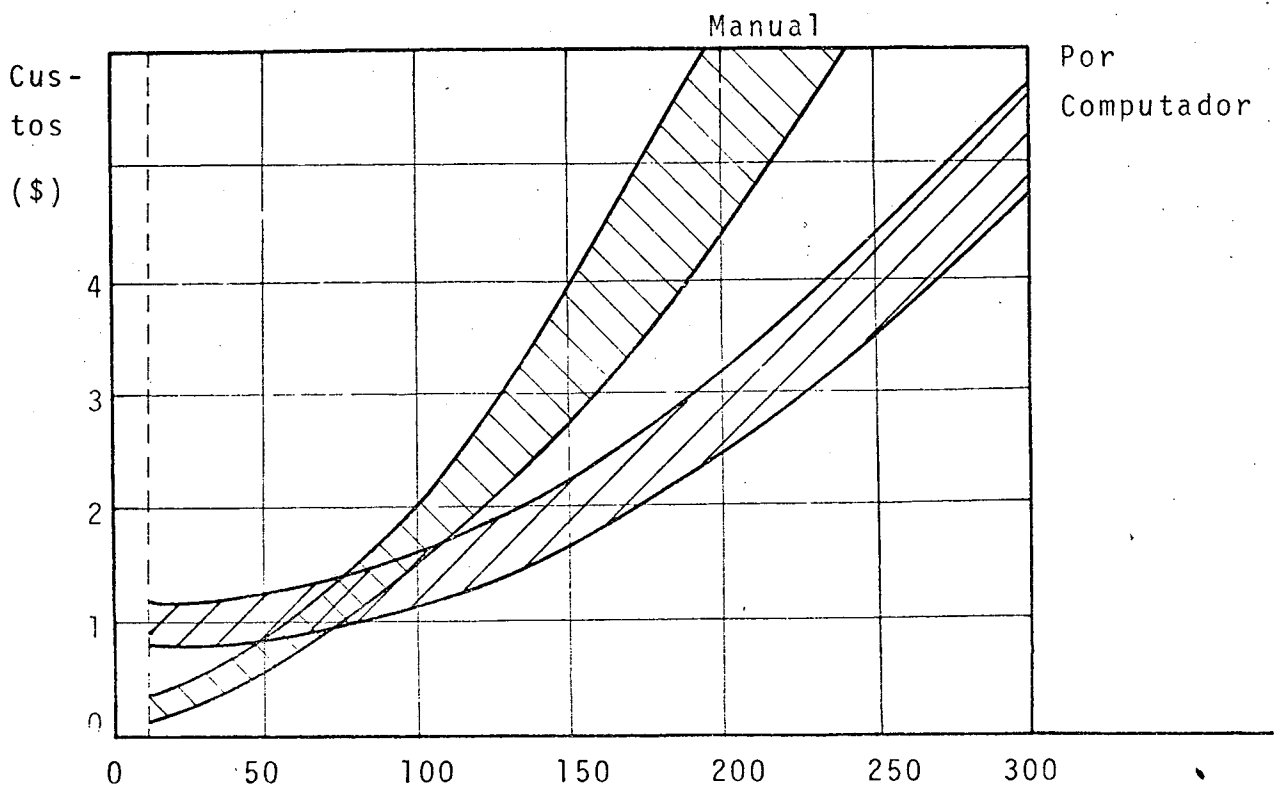


FIGURA 5 - Programação manual & Programação por computador: comparativo entre custos e número de blocos.

C A P Í T U L O I I

2. AS LINGUAGENS DE PROGRAMAÇÃO E O SISTEMA DE PROGRAMAÇÃO CN EXAPT

2.1. Família APT

Logo após a fabricação da primeira MFCN, foi criado pelo MIT (Massachusetts Institute of Technology - USA) para resolver problemas geométricos de programação de peças complexas de indústria aeronáutica, o sistema APT (Automatically Programmed Tools). Baseando-se nele, várias linguagens de programação foram posteriormente desenvolvidas e vieram a constituir a família APT.

Com a utilização, rapidamente crescente, dos sistemas de fabricação CN, diversas empresas produtoras e associações, lançaram no mercado sistemas computacionais para auxiliar a obtenção do portador de dados para a fabricação de peças.

As linguagens da família APT, têm todas um caráter universal comum, isto é, são independentes tanto do computador suporte quanto da MFCN.

Diferem pois conceitualmente das linguagens primordialmente orientadas para um determinado computador ou para uma determinada MFCN.

O largo uso das linguagens tipo APT hoje verificado deve-se principalmente a três fatores:

- Versatilidade
- Padronização
- Desenvolvimento.

De fato, as linguagens desta família são aplicáveis a qualquer sistema CN, independentemente de fabricante e a qualquer computador, na hipótese de que tenha memória disponível suficiente. Têm, por outro lado, grande poder descritivo, geralmen-

te, em termos de geometria da peça a ser usinada, bem como, em certos casos, em termos da tecnologia a ser empregada. Estas últimas podem pois trabalhar com alto grau de automatização, liberando o programador da determinação das informações tecnológicas com que se vê envolvido em outros casos.

Outras são mais potentes em termos da descrição geométrica de peças, podendo trabalhar com 3 ou mais eixos.

Por outro lado, o alto desenvolvimento alcançado pelos membros desta família, devido ao grande esforço de trabalho concentrado em cada uma delas, tornou-as geralmente de maior aplicabilidade que suas outras congêneres. Para dar uma idéia do grau de trabalho concentrado no desenvolvimento destas linguagens, basta dizer que sã na família APT existem mais de 500 homens/ano investidos [26].

Deve-se destacar ainda o fato de terem estas linguagens palavras em língua inglesa, indicadoras da função que se deseja, o que facilita sobremaneira seu uso universal. Eventualmente, (família APT), estas palavras em inglês, podem ser substituídas por outras traduzidas para a língua desejada. Os elementos com que se constroem os programas de peças, nas linguagens da família APT, são mostrados na figura 6.

Embora já se tenha dito no primeiro capítulo que o "hardware" varia muito rapidamente, tendendo para um barateamento e maior capacidade das máquinas, o que às vezes força as empresas usuárias de um sistema de programação CN a mudar de computador, estas linguagens são implementáveis nos novos sistemas com facilidade, por serem escritas na linguagem de alto nível mais largamente utilizada para tais finalidades: FORTRAN.

Esta característica é sumamente importante por simplificar os problemas em caso de mudança de "hardware", já que se tornaria muito difícil retrainar todo o pessoal envolvido em CN, se uma mudança fosse decidida.

Como exemplo, pode-se citar a VFW - Vereinigte Flugtechnische Werke GMBH, que após ter utilizado a linguagem APT por 11 anos, introduziu também em sua empresa a linguagem EXAPT, para tratar principalmente de problemas tecnológicos de seus tor-

ELEMENTOS DA LINGUAGEM	LETRAS	ABCDEFGHIJKLM NOPQRSTUVWXYZ
	DÍGITOS	1 2 3 4 5 6 7 8 9 0
	SÍMBOLOS ESPECIAIS	., / + - () * = \$
PALAVRAS	PALAVRAS PRINCIPAIS	MACHIN DRILL WORK GOTO
	VOCÁBULOS	DIAMET DEPTH
	MODIFICADORES	MAX 421 NCBOHR
	SÍMBOLOS (LIVRE ESCOLHA)	10 45 68.7 - 90.0
INDICAÇÕES	NÚMEROS	
	INDICAÇÕES DE ORDEM PROGRAMO-TÉCNICAS	MACHIN/NCBOHR
	INDICAÇÕES DE DEFINIÇÃO	MAX 421 = DRILL / DIAMET, 10, DEPTH, 45
PROGRAMAS	INDICAÇÃO DE EXECUÇÃO	WORK / MAX 421 GOTO / 68.7, - 90.0,0
	IDENTIFICAÇÃO DA PEÇA	PARTNO / BOHRBEISPIEL
	ESPECIFICAÇÃO DA MÁQUINA	⋮ MACHIN / NCBOHR
	DEFINIÇÃO DO TRABALHO	MAX 421 = DRILL / DIAMET, 10, DEPTH, 45
	COMANDO DE TRABALHO	WORK / MAX 421
	COMANDO DE NOVA POSIÇÃO	GOTO / 68.7, - 90.0,0
FIM DE PROGRAMA	⋮ FINI	

FIGURA 6 - Elementos construtivos das linguagens de programação CN da família APT.

nos e centros de usinagem. O fato não provocou qualquer problema de programação ou de organização, dada a igualdade quase total de vocabulário e de sintaxe de ambas. Além do mais, nesta empresa, o programador faz seu programa abstraindo do processador. Chamar o processador adequado é problema exclusivo do computador.

2.1.1. Processador e Pós-Processador

As linguagens da família APT trabalham em duas fases bem distintas. A primeira é constituída de um conjunto de programas cujo papel é fazer os cálculos aritméticos e geométricos necessários para a definição matemática da peça, determinar o percurso da ferramenta e produzir uma solução geral, absolutamente independente da máquina em que vai ser usinada a peça programada. Esta solução geral é um conjunto de informações aproveitável pelo universo das MFCN.

O conjunto de programas que produz esta solução é conhecido por PROCESSADOR.

Os dados obtidos na primeira parte, no processador, são em seguida trabalhados por um segundo conjunto de programas chamado pós-processador, que tem como objetivo particularizar a solução genérica da primeira parte para uma determinada MFCN, com seu controle.

Este sistema traz as seguintes vantagens:

- O trabalho genérico, que constitui a maior parte do tempo de processamento, pode ser feito uma única vez e seu resultado ser trabalhado posteriormente por diferentes pós-processadores e então a mesma peça poderá ser usinada em diferentes máquinas, como mostra a figura 7.

- Não se torna necessário dispor de um sistema de processamento completo para cada máquina.

- O trabalho de construção, manutenção e desenvolvimento dos sistemas de processamento, fica sobremaneira simplificado, dado que a maior parte do processamento e a fase mais trabalhosa é constante para todas as máquinas. Somente a parte de especificação para uma determinada máquina deve ser construída separadamente para cada caso. Mas mesmo assim o trabalho fica reduzido em aproximadamente 10 vezes, que é a ordem de grandeza que se separa o processador do pós-processador.

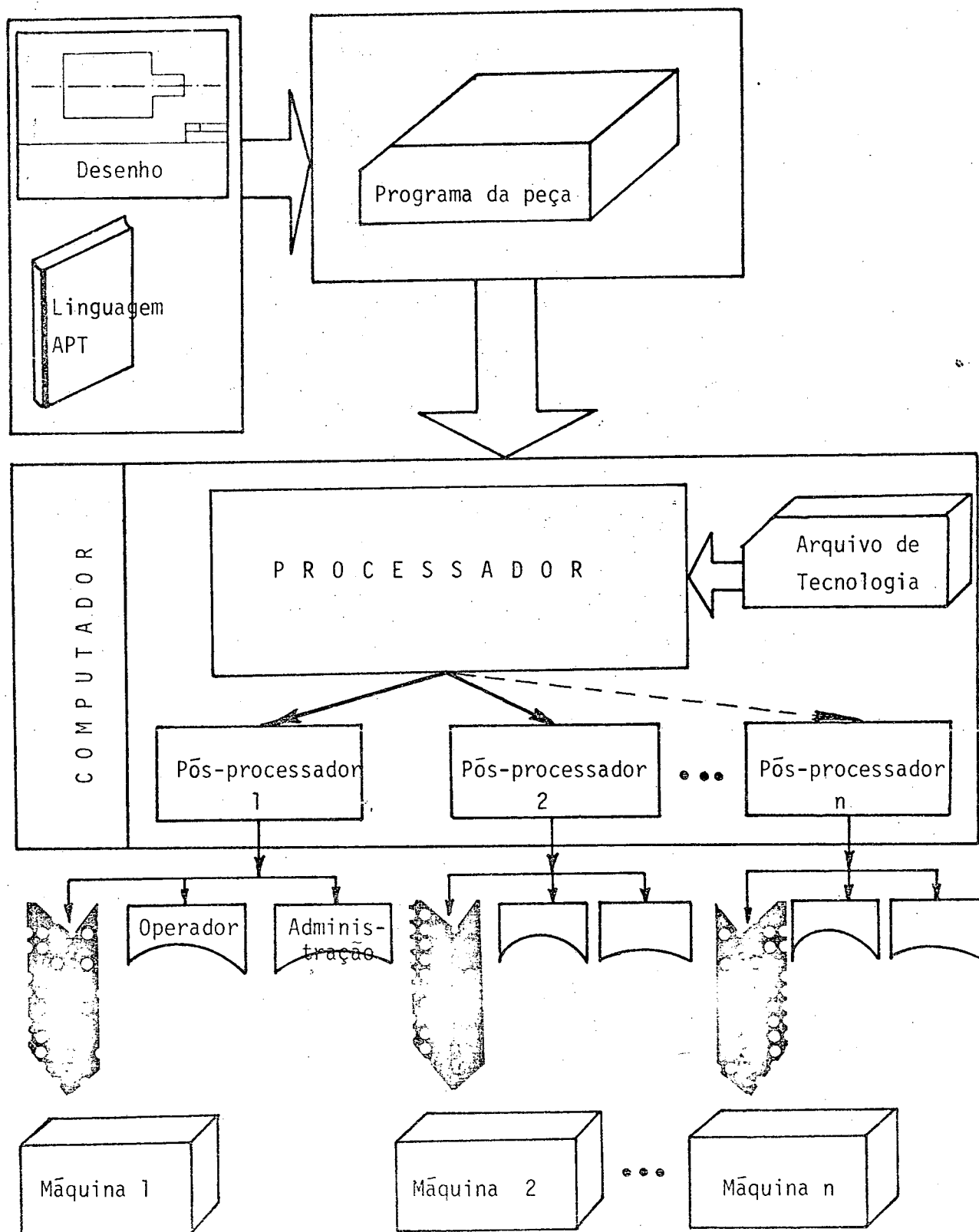


FIGURA 7 - Processamento de um programa de peça em linguagem tipo APT, obtendo-se meios de controle para diversas máquinas.

O desenvolvimento de sistemas CN com linguagens de programação similares e sobretudo com a saída do processador normalizada, vem ao encontro da necessidade de se unificar os programas adaptadores desta saída à máquina CN, o que torna o custo e o tempo de elaboração destes pós-processadores mais baixo. Ficou evidente a importância de uma padronização, quando, em 1967, a Alemanha, a França, a Inglaterra e os Estados Unidos submeteram respectivamente as linguagens EXAPT, IFAPT, NELAPT e APT como candidatas nacionais à normalização da ISO (International Organization for Standardization). Para não dificultar as pesquisas e o desenvolvimento na área, o grupo de trabalho sobre o assunto, criado pela ISO, decidiu normalizar não todo um pacote completo, mas sim a saída do processador e os tipos de registros de pós-processador. A questão da normalização será novamente abordada no capítulo IV.

É fácil entender a solicitação de padronização destas linguagens APT, quando se considera o grau de universalidade que as mesmas alcançaram e do nível de desenvolvimento atingido. Não é difícil conjecturar ou mesmo construir sistemas mais simples, e talvez mais eficientes, para um determinado segmento de sua generalidade, as vantagens assim conseguidas.

A capacidade de incorporar a tecnologia já desenvolvida pelo usuário é característica a ser também frisada nos sistemas APT, além de terem formato livre (ver item 2.2).

Além do mais, sendo desenvolvidas e mantidas tanto por institutos acadêmicos (Universidades de Aachen e Berlin na Alemanha, MIT nos E.U.A., National Engineering Laboratory na Inglaterra, etc.), quanto por empresas associadas, garante-se a independência e o desenvolvimento em rumos que interessam aos usuários e com isto se guarda a necessária equidistância de métodos, fabricantes de máquinas e computadores, etc., que organizações exclusivamente comerciais poderiam comprometer.

Finalmente, estas linguagens são constantemente trabalhadas e desenvolvidas, o que lhes garante um acompanhamento constante das inovações trazidas pelo desenvolvimento tecnológico.

2.2. Outras linguagens

A proliferação de sistemas de programação CN tem sido muito grande. Empresas fabricantes de máquinas como: centros de usinagem, tornos, máquinas de corte por chama, fresas e ainda institutos de pesquisa e escritórios de programação, lançam no mercado programas com as mais variadas habilitações.

Os sistemas não baseados no APT não têm distinção clara entre as duas fases de processamento, isto é, o direcionamento para uma determinada MFCN já é parte do próprio processamento como um todo. Cada instrução do programa da peça tem como objetivo direto a produção de blocos de informação para a usinagem, na fita perfurada, que controlará a MFCN.

Existem linguagens, ditas especiais, para resolução de problemas de um determinado tipo de usinagem, de computador, de controle, etc.

As linguagens especiais podem ser mais eficientes em termos computacionais (menos capacidade de memória exigida, maior rapidez de processamento, etc.) mas deve-se estar atento para o perigo de, no futuro, ao se ter máquinas de outros tipos não atendidas pela linguagem, se necessitar também de outro sistema de programação.

Tipo de Formato

Diz-se que uma linguagem é de formato livre quando as conexões existentes entre a estrutura e a ordenação dos dados de entrada no computador, têm formato livre.

Nas linguagens de formato fixo o programador está obrigado a colocar suas informações de entrada para o processador em campos previamente estabelecidos, do meio de entrada usado. Tais campos são em número não muito grande e portanto as linguagens de formato fixo são se adequam a tipos de problemas em que os dados de entrada se restringem a um pequeno número. Tendo que seguir o formato de entrada, tais linguagens obrigam o programador a um método fixo de trabalho, tirando-lhe a possibilidade de

ter seu estilo. Por serem pouco redundantes, estas linguagens apresentam poucas possibilidades de verificação. Geralmente estas linguagens têm estrutura rígida demais para permitirem extensões para outros campos de fabricação, que não aquele para o qual foi inicialmente projetada. Para cobrir todo o campo da fabricação necessitam sistemas diferentes em cada processo. Apesar do aumento em número, se teria uma diminuição em tamanho, diminuindo a capacidade de memória necessária para conter estes sistemas.

De outro lado uma linguagem de formato livre sempre significa menor esforço de programação, menor possibilidade de erro e correções mais simples. Compõem-se de um número limitado de palavras da linguagem usual e mnemônicas, dando várias possibilidades ao programador de definir uma mesma situação por diferentes métodos, o que lhe facilita o trabalho e lhe permite o uso de estilo próprio.

2.3. O Sistema EXAPT de Programação

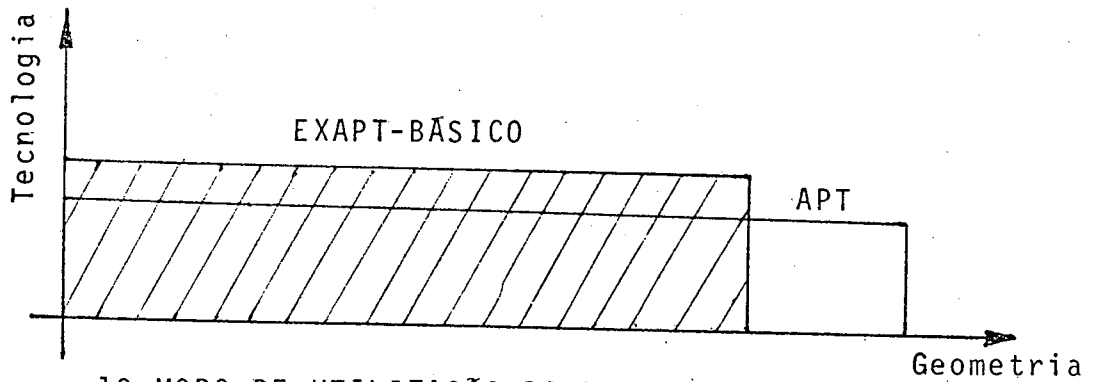
Sistema de Programação EXAPT é um conjunto de elementos, que, como nos demais sistemas da família APT, é composto de:

- Linguagem de Programação
- Processador
- Pós-Processadores
- Programas Auxiliares

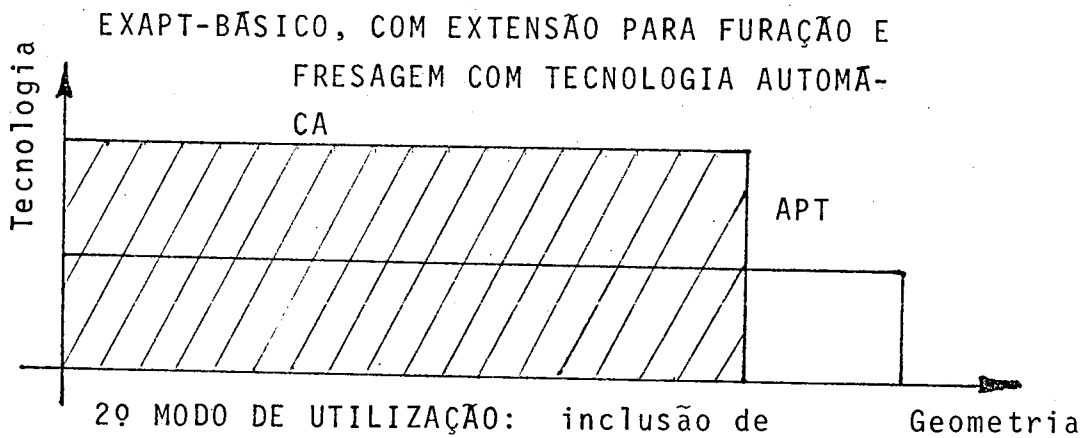
2.3.1. Linguagem de Programação

A linguagem de programação EXAPT tem como elementos construtivos básicos, os mesmos da figura 6, comuns a todas as linguagens tipo APT, e, é composta de um conjunto de regras que regem a interligação destes elementos, com o objetivo de definir a usinagem de peças.

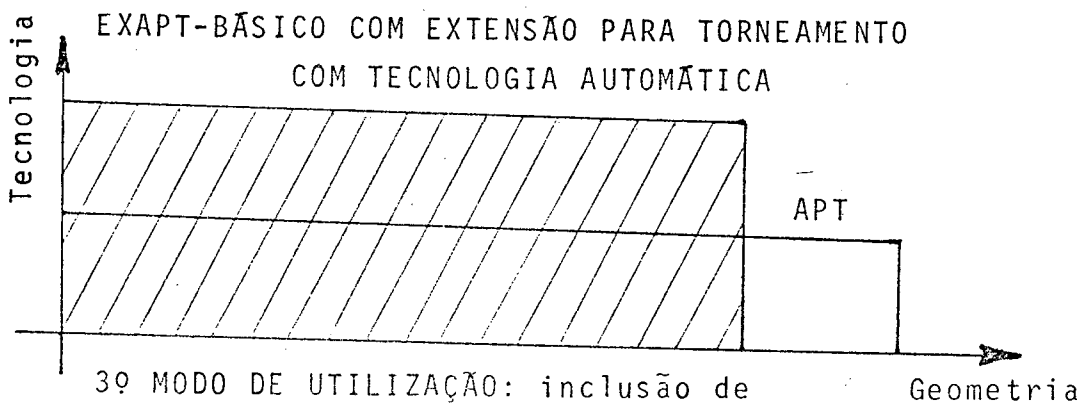
A utilização prática da linguagem pode ser feita de três modos, ilustrados na figura 8, que contém adicionalmente uma comparação qualitativa entre a capacidade de descrição



1º MODO DE UTILIZAÇÃO DA LINGUAGEM EXAPT:
EXAPT-BÁSICO



2º MODO DE UTILIZAÇÃO: inclusão de
termos descrevendo furação e fresagem
com tecnologia automática



3º MODO DE UTILIZAÇÃO: inclusão de
termos descrevendo torneamento com
tecnologia automática

FIGURA 8 - Os 3 Modos de utilização da linguagem EXAPT.

tecnológica e geométrica destes modos, relativamente à linguagem APT.

O EXAPT denominado BÁSICO na figura 8, tem menos capacidade de descrição geométrica que o APT e capacidade descritiva tecnológica maior.

Este modo de utilização, isto é, o EXAPT-BÁSICO, representa aproximadamente 70 a 80% da capacidade total da linguagem. Entretanto todos os problemas de usinagem em torneamento, furação, fresagem, corte por chama, estampagem, eletro-erosão, etc., solucionáveis com 2 1/2 eixos, são resolvidos com EXAPT-BÁSICO.

Na realidade, qualquer problema nos métodos de fabricação especificados, pode ser solucionado com EXAPT-BÁSICO. Entretanto deve-se enfatizar que, tendo construído arquivos de materiais, ferramentas, ciclos de trabalho e máquinas, uma empresa pode escrever mais racionalmente seus programas, deixando que o computador trabalhe automaticamente estas informações tecnológicas, que eram tratadas pelo programador, no EXAPT-BÁSICO. Para fazer este trabalho automaticamente, deve-se utilizar alguns termos mais poderosos. Estes podem ser utilizados uma vez construídos os arquivos próprios e implementado um ou dois módulos para onde o processamento é desviado quando ocorre a incidência de algum destes termos (ver item 2.3.2).

Na figura 9 vê-se como é definido, automaticamente, um ciclo de trabalho que chama 4 ferramentas do arquivo respectivo, define sua sequência de trabalho, seus dados de corte e seu trajeto.

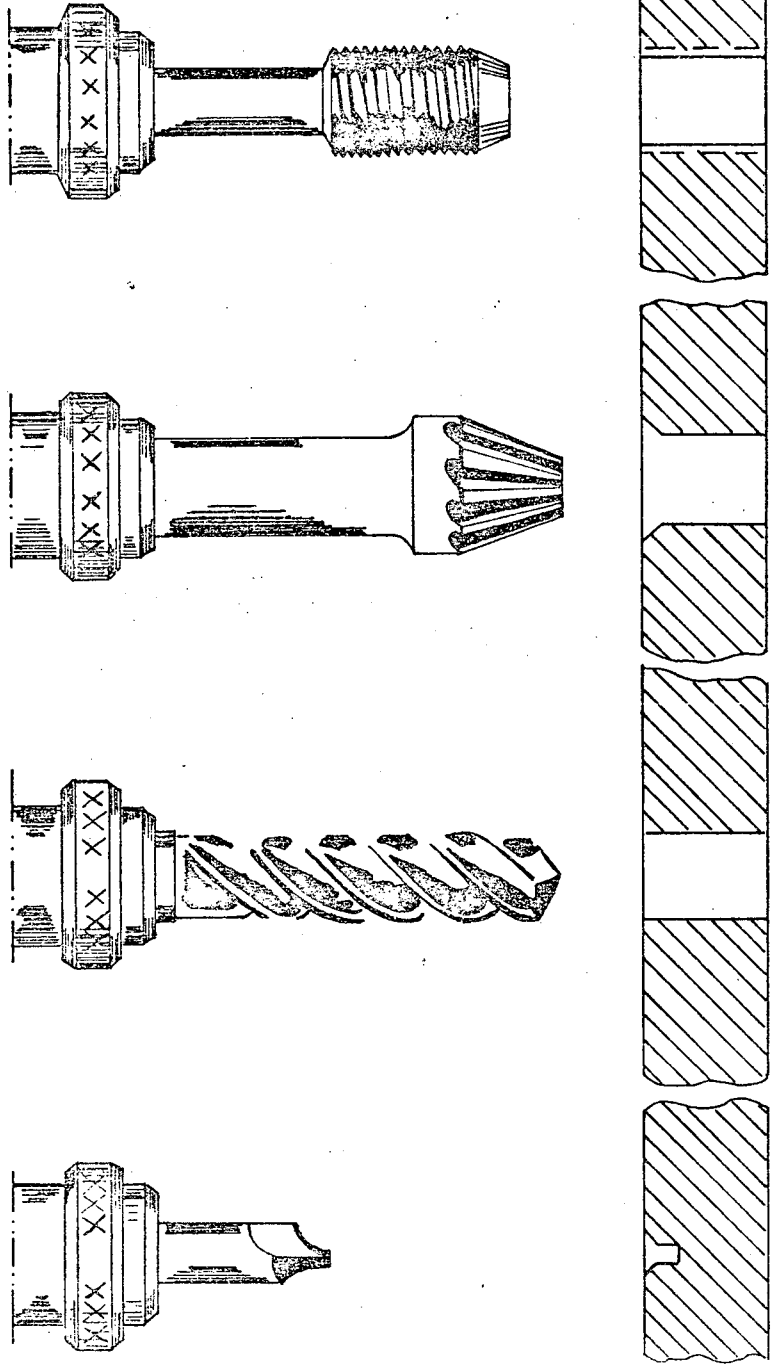
O EXAPT-BÁSICO constitui, para as empresas que ainda não têm arquivos de tecnologia, uma linguagem de formato livre, universal, com algum tratamento tecnológico automático.

Os "inputs" para o EXAPT-BÁSICO (figura 10), são o programa da peça e o arquivo de ferramentas, mas se este arquivo não estiver ainda disponível, ele pode ser formado progressivamente com o uso concomitante da linguagem, e em algum ponto do desenvolvimento, pode-se adotar as extensões para tratamento de torneamento ou furação e fresagem com tecnologia automática.

Definição: F1 = TAP / DIAMET, 16, DEPTH, 18, TAT, 1, BEVEL, TOLPO

WORK / F1

GOTO / P2



4711	4728	4825	1367	Ferramenta
1500	600	140	220	Rotações
0,2	0,25	0,33	2,0	Avanço

FIGURA 9 - Determinação automática da Sequência, Ferramentas, Dados de Corte e Trajeto das Ferramentas.

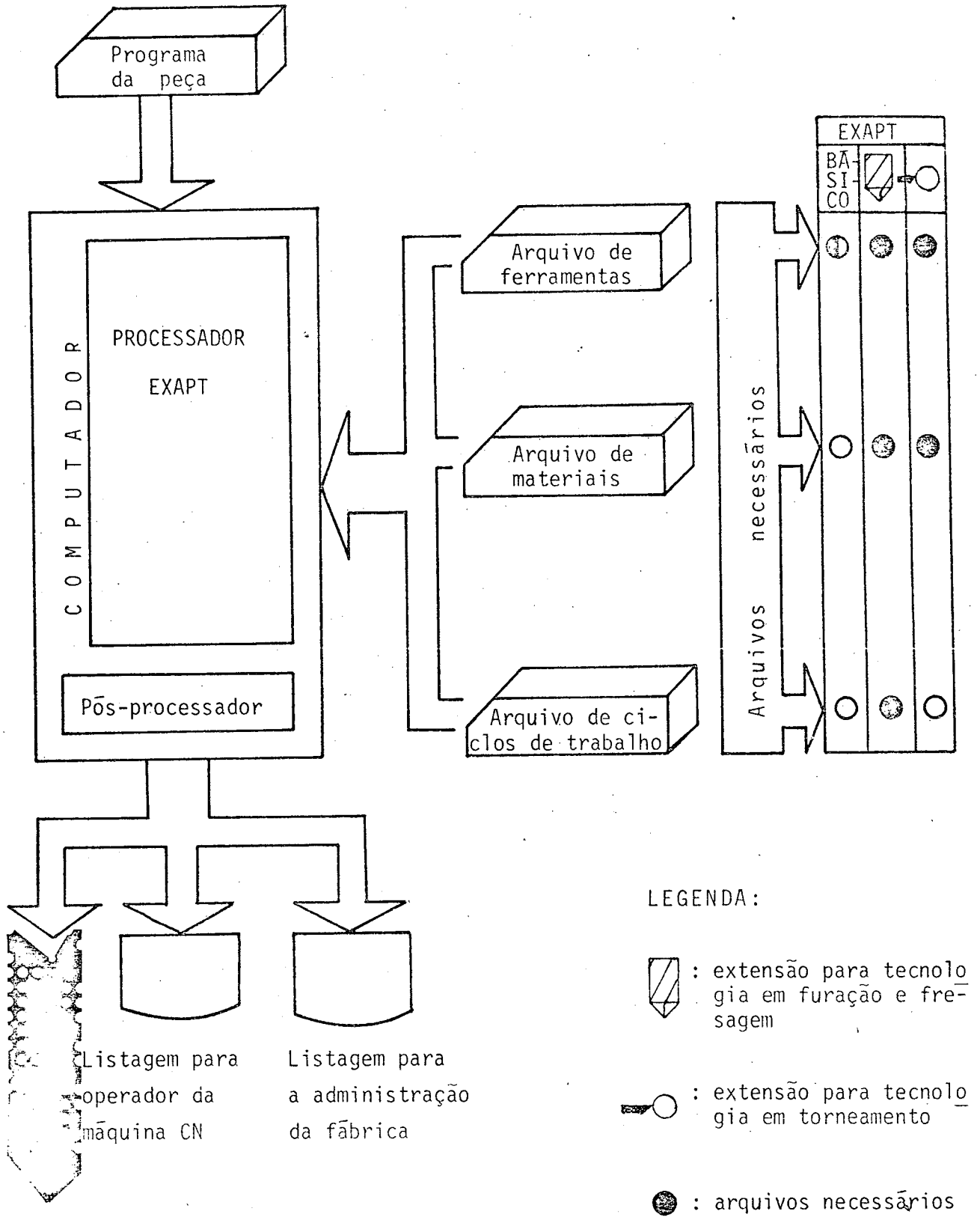


FIGURA 10 - Entradas e saídas dos processadores do sistema EXAPT de programação CN.

A formação progressiva destes arquivos é feita com módulos EXAPT auxiliares, chamados DAFES e MAPEX.

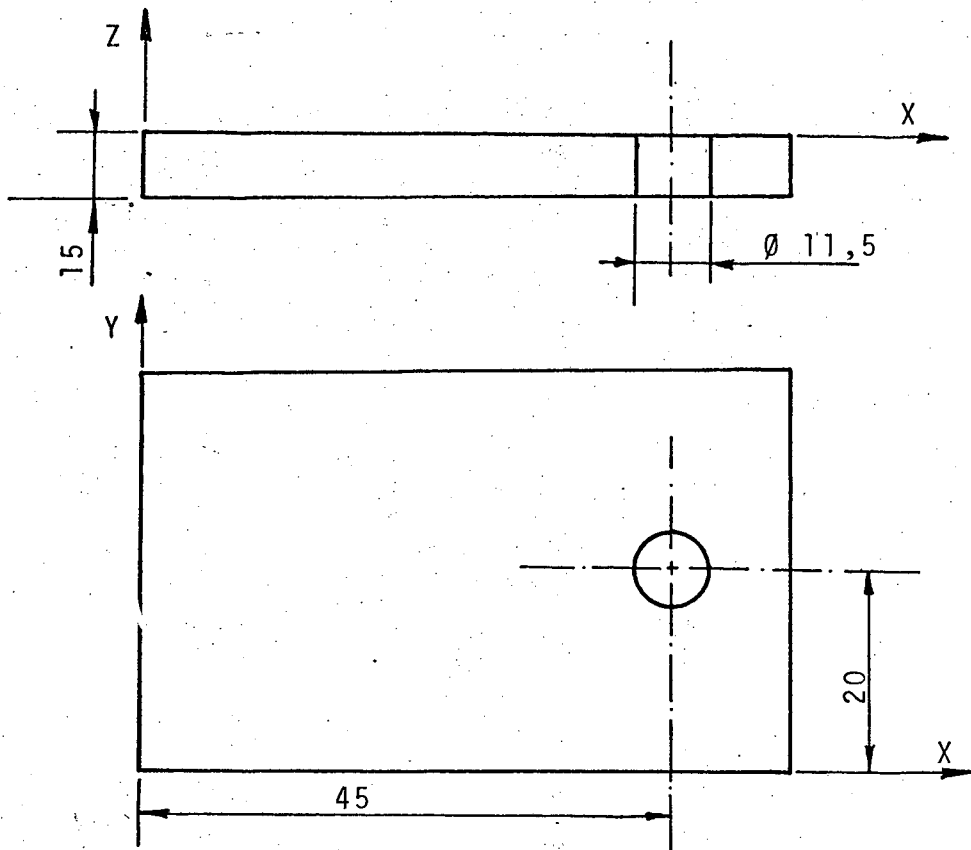
Na figura 11 supõe-se que a empresa não disponha ainda de um arquivo de suas ferramentas e de materiais usados em CN para escrever o programa da peça em EXAPT-BÁSICO.

Historicamente, devido a limitações de "hardware", existiam três conjuntos de programas independentes (processadores - ver item 2.3.2) chamados EXAPT-BÁSICO, EXAPT 1.1 e EXAPT 2, sendo os 2 últimos responsáveis pela programação com tecnologia automática de furação e fresagem ou para torneamento, respectivamente. Com a modularização do sistema, aspecto mais profundamente discutido adiante, se construiu um sistema comum, o EXAPT-BÁSICO, e, com a adição de um ou outro módulo, pode-se trabalhar com todos os termos da linguagem EXAPT, como nos antigos EXAPT 1.1 ou EXAPT 2.

Extensão do EXAPT-BÁSICO com a tecnologia automática para furação e fresagem ou torneamento

Empresas interessadas em trabalhar com tecnologia mais elaborada e automática, na área de furação e fresagem ou torneamento, utilizariam os mesmos termos de linguagem já usados no modo EXAPT-BÁSICO, aos quais acrescentaria alguns outros, mais poderosos para este tipo de trabalho, trazendo mais conforto e mais eficiência à programação e à fabricação.

Os programas mostrados nas figuras 11 e 12 ilustram esta asserção. Vê-se que se utilizam na figura 12 termos como DRILL, DIAMET, DEPTH, etc., não utilizados no programa da figura 11; entretanto os 2 programas executam exatamente o mesmo trabalho embora tenha o 2º apenas 10 linhas de comandos e o 1º, 14 linhas. Mas a redução não é, em absoluto, a característica mais importante das extensões, e sim o poder de tecnologia automática introduzido pelos novos termos.

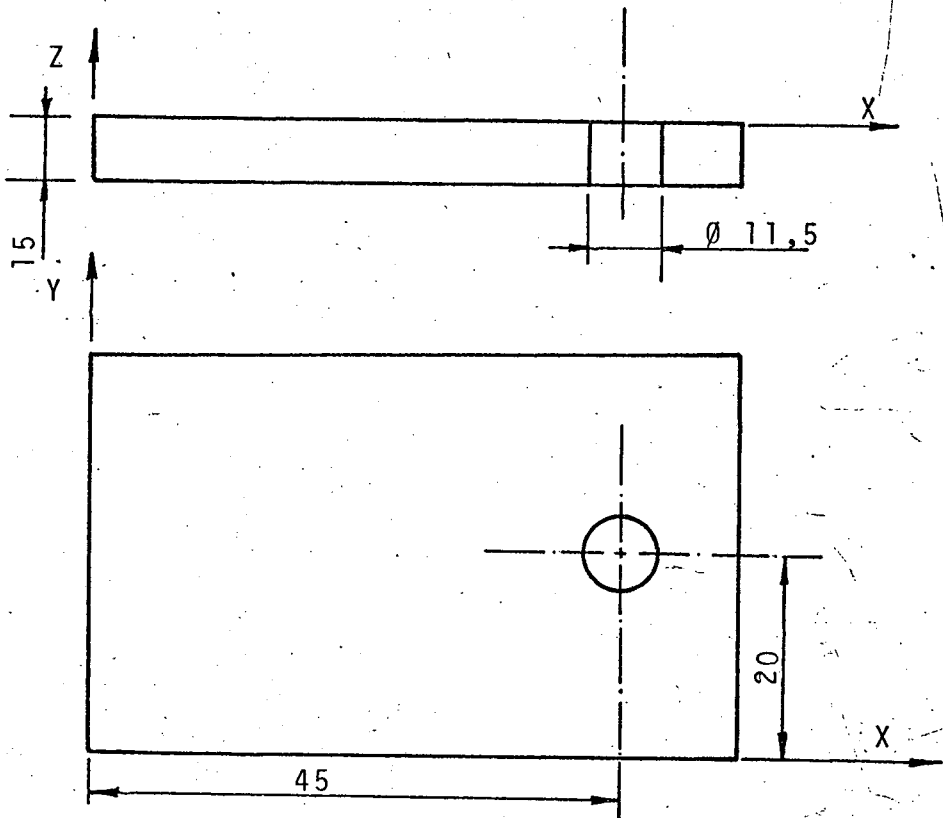


```

0001 PARTNO/ EXEMPLO 1          ##-PROGRAMACAO EM EXAPT BASICO.
0002                               ## NOME DO PROGRAMA.
0003 MACJIN/ NBH65              ##-NOME DA MAQUINA E DO
0004                               ## POS-PROCESSADOR.
0005 TRANS / 178,78,80         ##-INTER-RELACIONAMENTO ENTRE
0006                               ## O SISTEMA DE COORDENADAS
0007                               ## DA PECA COM O DA MAQUINA.
0008 P1=POINT/45,20,0          ##-DEFINICAO GEOMETRICA
0009                               ## DO PONTO P1.
0010 NEWTL/201012,1,11.5,255,0,0 ##-DEFINICAO DA BROCA
0011                               ## HELICOIDAL.
0012 TOOLNO/201012,13         ##-CHAMAR DO LUGAR 13 DO MAGA-
0013                               ## ZINE A FERRAMENTA DEFINIDA;
0014                               ## E POSICIONA-LA NA ARVORE.
0015 RAPID                       ##-AVANCO RAPIDO.
0016 GOTO/(POINT/P1, DELTA,0,0,2) ##-POSICIONAR FERRAMENTA 2MM
0017                               ## ACIMA DO PONTO P1.
0018 SRINDL/2000,CLW           ##-GIRAR ARVORE COM 2000 RPM,
0019                               ## SENTIDO HORARIO.
0020 FEDRAT/.22                 ##-AVANCO DE 0.22MM
0021                               ## POR REVOLUCAO.
0022 GODLTA/-17                 ##-BROCA DESCE 17MM, FURAN O.
0023 RAPID                       ##-AVANCO RAPIDO.
0024 GODLTA/ 17                 ##-BROCA SOBE 17MM, E SAÍ
0025                               ## DO FURO.
0026 FINI                       ##-FIM DO PROGRAMA.

```

FIGURA 11-Programa exemplo em EXAPT-BÁSICO, sem existência de arquivo de ferramentas.



```

0001 PARTNO/ EXEMPLO 2          ##-PROGRAMACAO EM EXAPT 1.1
0002                               ## NOME DO PROGRAMA.
0003 MACHIN/ NBH65              ##-NOME DA MAQUINA E DO POS-
0004                               ## PROCESSADOR.
0005 TRANS / 178,78,80          ##-INTER-RELAÇONAMENTO ENTRE
0006                               ## O SISTEMA DE COORDENADAS
0007                               ## DA PEÇA COM O DA MAQUINA.
0008 P1=POINT/45,20,0           ##-DEFINICAO GEOMETRICA
0009                               ## DO PONTO P1.
0010 PART=MATERL,6              ##-ESPECIFICACAO DO MATERIAL
0011                               ## NUMERO 6 DO ARQUIVO
0012                               ## DE MATERIAIS.
0013 FURO=DRILL/DIA 1ET, 10, DEPTH, 16 ##-DEFINICAO DE USINAGEM:
0014                               ## -FURACAO COM BROCA, SENDO,
0015                               ## - DIAMETRO: 11.5MM
0016                               ## - PROFUNDIDADE: 16MM.
0017 CLDIST/1                   ##-DISTANCIA DE SEGURANCA
0018                               ## DE 1MM, DA FERRAMENTA ATE
0019                               ## A PEÇA.
0020 WORK /FURO                 ##-CHAMADA DA OPERACAO
0021                               ## DE USINAGEM ACIMA DEFINIDA.
0022 CUT /P1                     ##-DEFINICAO DA POSICAO
0023                               ## DE USINAGEM: P1.
0024 FINI                        ##-FIM DO PROGRAMA.

```

FIGURA 12-Programa exemplo em EXAPT 1.1 usando arquivo de ferramentas e de materiais previamente construídos.

2.3.2. Processador EXAPT

Além de uma linguagem de programação, um Sistema de Programação de CN da família APT deve conter também o chamado PROCESSADOR.

O Processador EXAPT é um conjunto de programas escritos em uma linguagem de programação de alto nível (FORTRAN). Sua finalidade é:

- Aceitar como entrada o programa de uma peça escrito em linguagem de programação EXAPT.
- Fazer todos os cálculos geométricos para determinação da trajetória que a ferramenta deverá seguir para usinar a peça.
- Tomar decisões tecnológicas.
- Dar como saída um arquivo com as informações de geometria e tecnologia, que deverá ser utilizado para usinar a peça. Este arquivo é chamado de CLDATA (Cutter Location Data: dados para localização de cortes).

No processador EXAPT, o programa de cada peça passa pelos passos mostrados na figura 13, no primeiro módulo.

Inicialmente o computador transforma os comandos construídos de forma mais inteligível ao homem, em forma mais trabalhável pelo computador. A seguir são feitas as verificações quanto à correção sintática do comando (confere-se a obediência às regras da linguagem de programação). A verificação deve ser feita também quanto a lógica do comando; por exemplo, não se poderia definir um ponto como intersecção de dois círculos concêntricos.

Verificada a correção sintática e lógica do comando, este é armazenado em forma padronizada.

O processamento não segue linearmente estes passos. Na realidade, lido um cartão, seu processamento é feito sequencialmente, caráter por caráter, até ser encontrado um caráter especial, quando então são computados os caracteres lidos desde o último caráter especial, feitas as verificações sintáticas e

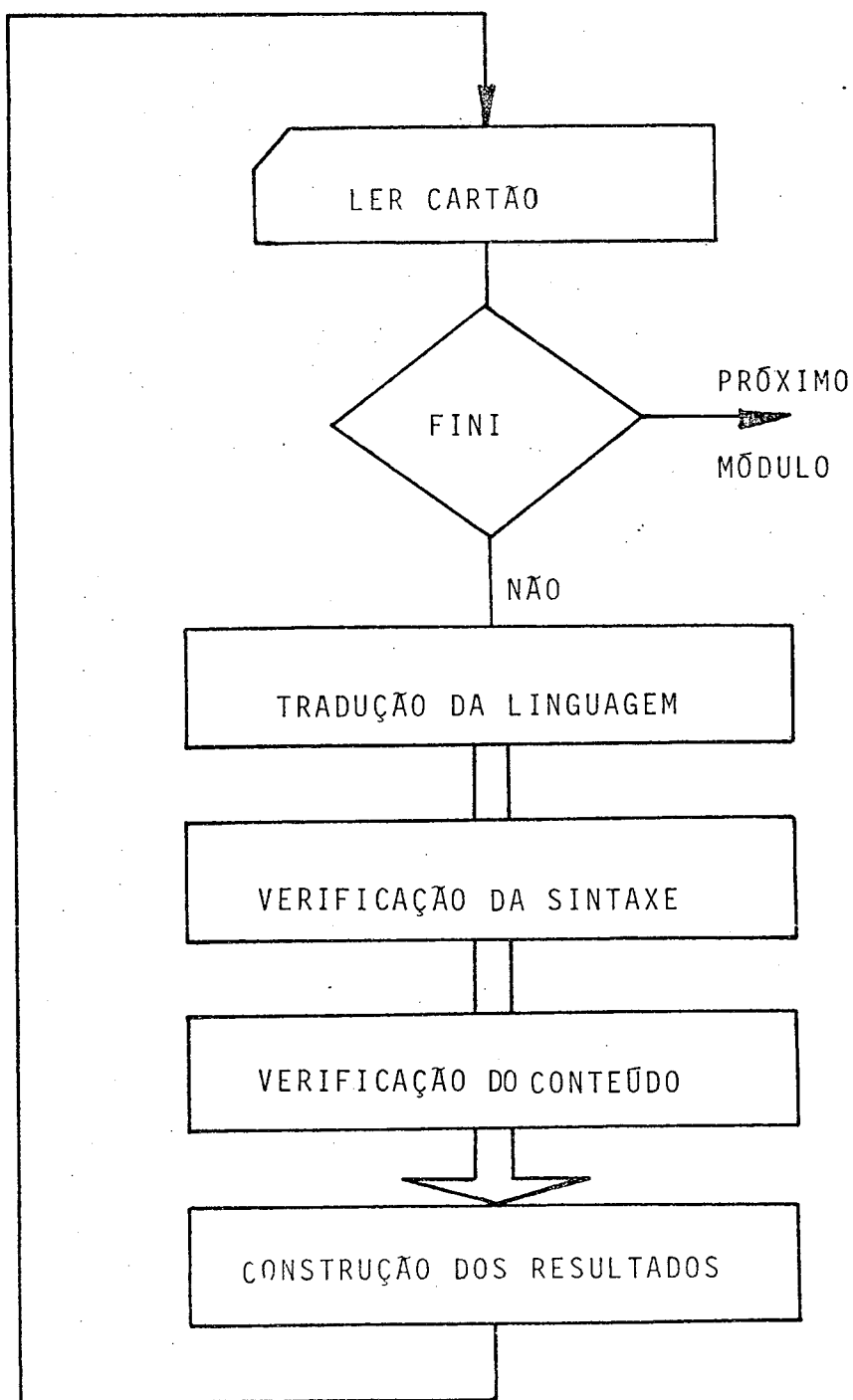


FIGURA 13 - Processador EXAPT: Estágios principais do processamento dos comandos (Módulo EINGAB).

gramaticais necessárias e o armazenamento ou emissão de mensagem de erro, conforme for o resultado da análise. No processamento, o correm as funções citadas na figura 13 de modo não necessariamente sequencial, mas global.

Nota-se que internamente, as definições geométricas de um mesmo elemento, quaisquer que sejam os modos de entrada utilizados, sempre são reduzidos às "formas canônicas", isto é, à forma padrão de trabalho. Um ponto, por exemplo, apesar de poder ser definido como intersecção de duas retas, de uma reta e um círculo, de 2 círculos, etc, sempre é armazenado internamente pelas suas coordenadas, obtidas através dos cálculos necessários feitos no módulo apropriado.

A estruturação do processador da linguagem EXAPT é modular. Cada módulo se destina à execução de uma função lógica bem caracterizada e é executada uma única vez durante o processamento de uma peça.

A figura 14 ilustra a disposição estrutural dos módulos constitutivos da linguagem EXAPT.

Os módulos, por sua vez, são constituídos de subrotinas que executam uma só tarefa (por exemplo atribuir um código numérico a cada caráter literal usado).

A tarefa 1 explica os elementos construtivos dos módulos EXAPT.

Importância da Organização Modular do Processador

Existem os módulos básicos do sistema, que em conjunto formam o EXAPT-BÁSICO, de funções já discutidas. Acrescentando-se um único módulo adicional ao EXAPT-BÁSICO, se tem o processador com tecnologia para furação e fresagem ou torneamento (TECEX 1 ou TLPATH, respectivamente. Ver figura 14).

Esta organização modular facilita enormemente os trabalhos de manutenção e desenvolvimento do sistema. Alterações de adição, subtração, modificação de seus componentes, ficam facilitadas e restritas ao módulo em questão.

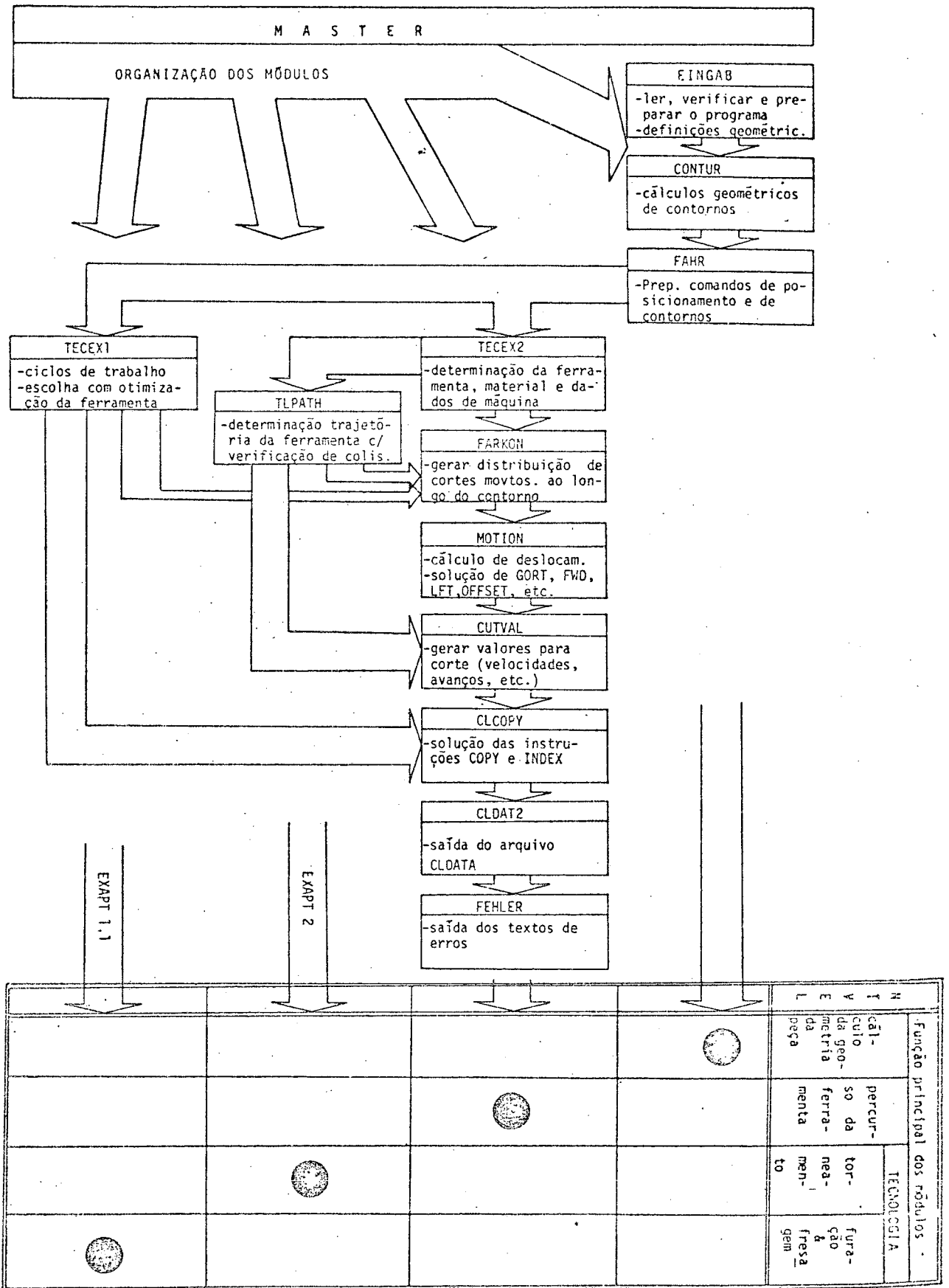


FIGURA 14 - Processador EXAPT: estrutura modular-módulos.

Fica também muito facilitado o entrosamento do sistema CN com outros sistemas computacionais como o CAD (Computer Aided Design: Projeto com auxílio de computador) e o CAM (Computer Aided Manufacturing: Fabricação com auxílio de computador).

Uma descrição sucinta da função executada por cada um dos 12 módulos do processador EXAPT é fornecida a seguir. Discute-se mais detalhadamente os módulos que estendem a capacidade de tratamento tecnológico do processador BÁSICO.

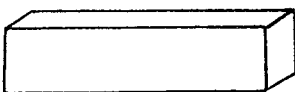
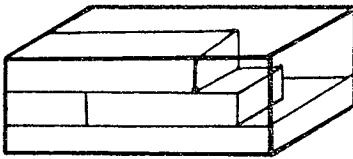
Definição	Simbologia	Descrição	Exemplo
Programa		Programa unitário ou subrotina; executa uma só tarefa definida.	Subrotina para verificar pontos de intersecção de dois elementos geométricos.
Função ou Módulo		É composto por vários programas e executa uma tarefa lógica parcial até o final.	Módulo para o cálculo de formas canônicas a partir de definições geométricas.

TABELA 1 - Elementos estruturais dos módulos EXAPT.

MÓDULO

FUNÇÃO

EINGAB

- Interpretação e controle das declarações da peça a ser executada
- Testes de sintaxe
- Cálculos aritméticos
- Execução de "Loops" e Macros
- Definições canônicas de pontos, retas, circunferências, conjuntos de pontos, matrizes

CONTUR

- Execução de contornos abertos e fechados
- Execução de cilindros tabelados

- FAHR
- Acondicionamento, isto é, relacionamento das formas canônicas de elementos geométricos para estas declarações.
 - Criação de blocos para as superfícies de delimitação ("Check-Surface") e de deslocamento ("Drive-Surface")
- TECEX 2
- Escolha do arquivo de dados de:
 - material
 - ferramenta
 - máquina
- Isto é:
- Cálculos correspondentes às declarações CLAMP e UNIT
 - Admissão de valores padrão
 - Relação entre ferramentas e suas chamadas
- FARKON
- Usinagem paralela a um contorno
 - Geração de trajetórias de corte
- MOTION
- Geração dos blocos GOTO e SUFACE a partir das superfícies de delimitação e superfície de deslocamento do módulo FAHR, para determinação da localização da ferramenta
 - Solucionar os elementos ROUND e BEVEL
 - Tratar as declarações de deslocamento
- CUTVAL
- Escolha da faixa de rotações
 - Cálculo da rotação, quando usado o comando CSRAT
 - Determinação das condições de corte
 - Dar saída aos valores de rotação e de avanço considerando as normas DIN
- CLDAT2
- Cópia dos blocos CLDATA
 - Dar saída do CLDATA
- FEHLER
- Dar saída às mensagens de erro
- TECEX 1
- Módulo usado em casos de emprego de tecnologia automática de furação e fresagem
- Funções:

- Tratamento dos ciclos de trabalho
- Definições de usinagem
- Execução dos TDATA (dados de ferramenta)
- Escolha da ferramenta com otimização

TLPTH

- Módulo usado em casos de emprego de tecnologia automática de torneamento

Função:

- Determinação das trajetórias de corte.

CLCOPY

- Tratamento das declarações INDEX e COPY.

Módulo para processamento de furação e fresagem com tecnologia automática (TECEX1)

O módulo TECEX 1 estende a linguagem do sistema EXAPT de programação, para operações de furação e de fresagem com tecnologia automática, em máquinas tipo fresas, furadeiras e centros de usinagem com até 2 1/2 eixos de controle contínuo. Com esta extensão, o processador EXAPT modular oferece as mesmas funções que o antigo EXAPT 1.1.

Os pré-requisitos para processar um programa utilizando a extensão para tecnologia em furação e fresagem, além do módulo TECEX 1 e dos pós-processadores adequados, são os arquivos tecnológicos:

- ciclos de trabalho
- materiais
- ferramentas

Neste processador estendido, furações são descritas por meio de operações de máquina e com seu intermédio pode-se determinar automaticamente (figura 9):

- seqüências de trabalho
- escolha de ferramentas
- trajetória de ferramentas
- valores de corte.

O resultado da determinação da sequência de trabalho pode ser tanto um passo de usinagem, (como na figura 12) quanto vários passos.

A determinação é baseada em regras que devem ser pré-especificadas somente uma vez no arquivo de ciclos de trabalho.

Existem sequências de trabalho tipo padrão e outras, especiais, definidas de acordo com regras específicas de cada empresa. As sequências-padrão formam uma base inicial do arquivo que pode ser estendida pelas especiais.

Programas que utilizam a extensão para tecnologia automática em furação e fresagem, contêm os chamados comandos executivos, que determinam de modo completo a sequência de usinagem da peça, incluindo a sequência de operações e a atribuição de posições. Assim, tarefas complicadas de usinagem, operações de furação com comandos de movimento e problemas de mandrilamento com barras, podem ser resolvidos em um só passo de programação.

Módulo para processamento de torneamento com tecnologia automática (TLPATH)

O módulo TLPATH estende a linguagem do sistema EXAPT de programação para operações de torneamento e furação em tornos, com uso de tecnologia automática. Com esta extensão, o processador EXAPT modular oferece as mesmas funções que o antigo EXAPT 2.

O pré-requisito para processar um programa de peça utilizando a extensão para tecnologia em torneamento, além da implementação do módulo TLPATH, e dos pós-processadores respectivos, é a existência de arquivos de ferramenta e materiais, dos quais o processador selecionará os procedimentos mais adequados.

As possibilidades da extensão para tecnologia automática em torneamento podem ser resumidos em:

1. Os dados de controle são automaticamente determinados, especialmente para sequências

de corte, trajetórias de ferramentas, velocidades de avanço e rotações.

Os seguintes aspectos são considerados:

- combinação do material da ferramenta com o da peça.
- colisão da ferramenta com a peça ou com a placa.
- acabamento superficial.
- sobremedidas de usinagem.
- dados característicos da máquina-ferramenta.

2. Os dados determináveis automaticamente, também podem ser especificados no programa da peça e então eles serão checados quanto à sua correção.

3. Os desenhos para a programação são os comuns.

4. Possibilidade de verificar os dados de controle no "plotter", onde aparece:

- a peça bruta.
- a peça acabada.
- o contorno resultante de cada operação de máquina, bem como o avanço e a velocidade rápida de aproximação.

Entradas dos Processadores EXAPT

As entradas dadas aos processadores EXAPT, para que produzam as informações que controlarão a MFCN, são constituídas de duas categorias distintas:

Entradas variáveis (programas de peças)

Entradas fixas (arquivos de dados).

Como "inputs" variáveis se denomina aqui as informações fornecidas ao sistema de forma diferente para cada nova peça fabricada, isto é, informações que são válidas para aquela determinada peça, que em suma, compõem o próprio programa da peça.

O uso de entradas fixas, que constitui um dos principais objetivos do sistema EXAPT, vem ao encontro das necessidades das empresas que têm um universo limitado de matérias primas, de ferramentas e de máquinas, que pode ser organizado em arquivos de dados utilizáveis diretamente pelo computador e chamados pelo sistema de PAC usado. Constroem-se arquivos de informações sobre as máquinas disponíveis, sobre as matérias primas utilizadas, sobre as ferramentas necessárias, sobre os dispositivos de fixação criados, e se faz algoritmos integrantes do sistema, que executem automaticamente as operações repetitivas de escolha das informações nestes arquivos, para a execução da peça programada. Em outros sistemas, com orientação geométrica, esta escolha deve ser feita a cada novo programa, pelo próprio programador, pois o sistema de PAC se limita a trabalhos matemáticos.

Dados permanentes que são fornecidos ao sistema se constituem de: (figura 15)

- a. Arquivo de dados sobre a MFCN.
- b. Arquivo de dados sobre os materiais.
- c. Arquivo de dados sobre as ferramentas.
- d. Arquivo de erros, macros e sinônimos.
- e. Arquivos de sequências de trabalho.
- a. Arquivo de Dados das Máquinas CN

Estes arquivos descrevem as variáveis relevantes das MFCN da empresa usuária do sistema.

Normalmente, as informações que integram tal arquivo, são as seguintes:

- dados para identificação e classificação da máquina
- sistema de medição (Inglês ou métrico)
- modo de variação da velocidade (contínuo)

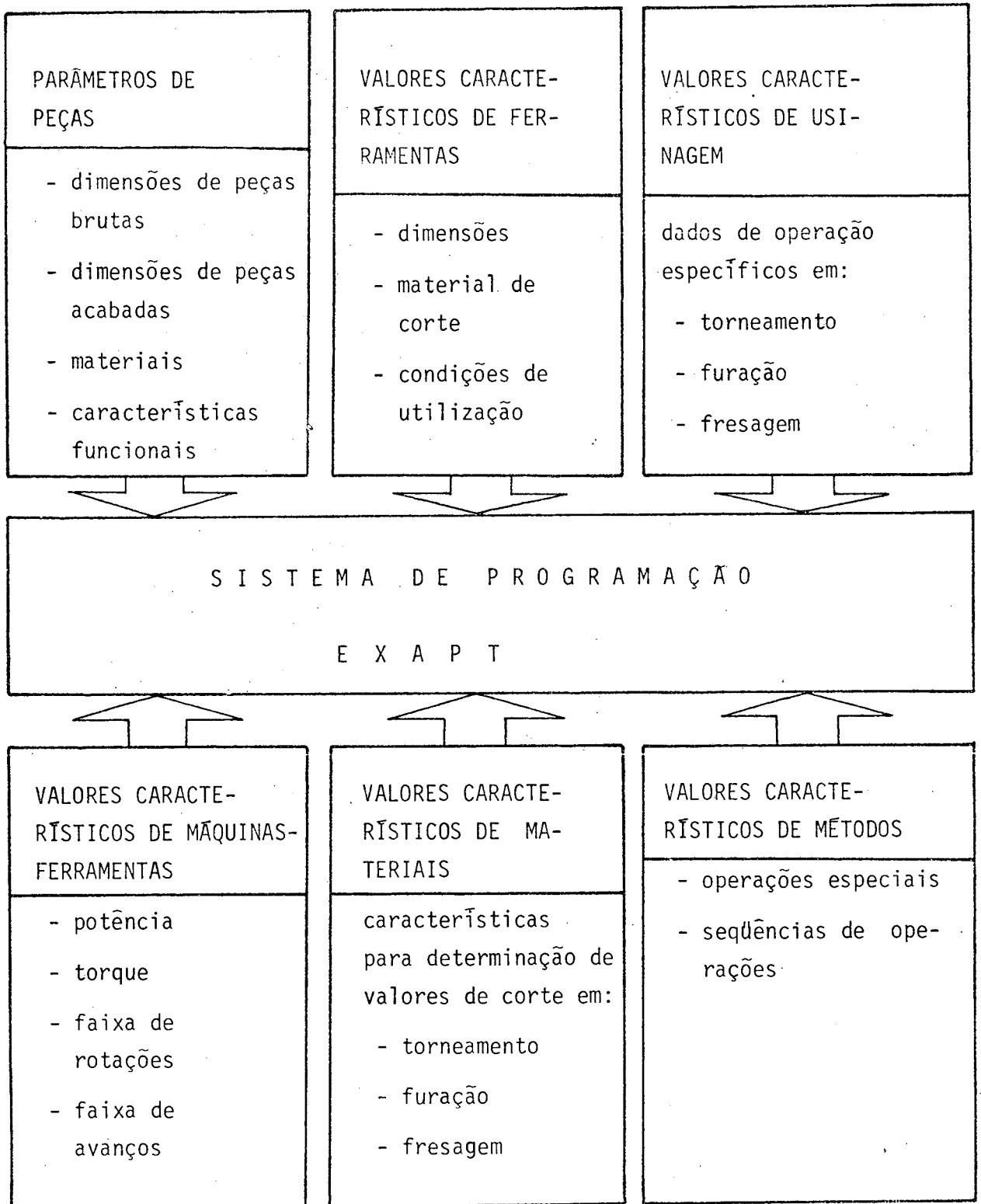


FIGURA 15 Arquivos administrados pelo sistema EXAPT de programação CN.

ou escalonado)

- velocidades disponíveis na árvore
- torques por velocidade

Objetivos básicos:

- otimização dos tempos de usinagem
- adaptação das velocidades de rotação
- informações para sistemas mais genéricos de produção como CAD/CAM (ver capítulo V).

b. Arquivo de Dados dos Materiais

Descrevem os dados pertinentes, para materiais usados pela empresa em suas máquinas.

Entre estes dados pode-se citar como exemplos:

- Índice para identificação e classificação do material
- material, tipo e geometria da ferramenta
- dados para cálculo de valores de corte
- dados para cálculo de valores de avanço.

Objetivos básicos:

- controle dos valores máximo e mínimo do avanço e profundidade de corte.
- seleção das ferramentas.

c. Arquivos de Dados de Ferramentas

Este arquivo contém as variáveis necessárias para as ferramentas usadas nas máquinas CN da empresa.

Entre estas variáveis pode-se citar como exemplos:

- Sistema de ferramentas
- Número de identificação
- Designação
- Número sistemático
- Condições de utilização (avanços, profun

didades, forças, torques, marcas de desgaste, etc.)

- Geometria de corte
- Dimensões e material da ferramenta
- Tipo de haste
- Tipo de porta-ferramenta
- Medidas de montagem

Objetivos básicos:

- definição do volume de material a ser usado na unidade de tempo
- distribuição de cortes
- determinação da trajetória da ferramenta
- checagem de colisões
- seleção das ferramentas

d. Arquivos de Macros, Erros e Sinônimos

Além de dados sobre materiais, ferramentas, máquinas, frequentemente é interessante o armazenamento de dados auxiliares para tornar mais fácil a programação. Estes dados auxiliares podem ser constituídos de MACROS, sinônimos, textos de erros, por exemplo.

No arquivo de MACROS, são armazenados os subprogramas (subrotinas) de uso geral, e ficam disponíveis para uso futuro, sem ter o trabalho de construí-los a cada vez, ou de ter que lê-los de outras fontes.

Se uma empresa desejar criar palavras mais adequadas, segundo seu ponto de vista, para os vocábulos do EXAPT, estas palavras - SINÔNIMOS - podem ser guardadas em um arquivo que será acessado a cada programa e não será necessário ao programador criar estes sinônimos a cada novo programa, que os usar.

Os erros que são cometidos durante a programação de uma peça, são codificados em números que correspondem a textos explicativos. Estes textos podem ser encontrados em manuais especiais ou armazenados e administrados diretamente pelo sistema, e impressos automaticamente quando forem cometidos e detetados pelo processador EXAPT.

e. Arquivos de Sequências de Trabalho

Contêm regras básicas para determinação de sequências de trabalho padrão e/ou especiais.

O arquivo de sequências de trabalho é composto das seguintes partes:

a) Arquivo de constantes tecnológicas

Contêm as constantes que podem ser influenciadas pelo usuário para determinação automática de dados tecnológicos.

b) Arquivo de checagem

Contêm as regras para a programação de definições de usinagem, para sequências de trabalhos padrões e especiais.

c) Arquivo para a determinação de sequências de trabalho

Contêm as regras próprias para a determinação das sequências de usinagem a partir de uma definição de usinagem.

2.3.3. Os Pós-Processadores EXAPT

Os pós-processadores, dada sua importância peculiar no enfoque dado por este trabalho, serão tratados com mais detalhe no capítulo IV.

2.3.4. Programas Auxiliares

Além da Linguagem de Programação, do Processador e dos Pós-Processadores, um sistema de programação CN deve conter programas auxiliares que facilitem sua manipulação pelo usuário.

Na sequência discorre-se sobre alguns dos programas auxiliares disponíveis em EXAPT.

Programas de Simulação Gráfica

Para a usinagem segura e econômica de peças em um sistema de programação CN, é necessário que o programa esteja depurado de erros, isto é, a ferramenta não poderá colidir com a peça ou com os dispositivos de fixação e ainda a peça fabricada de verá coincidir com o projeto da mesma.

Uma possibilidade de verificar os dados de con trole é usinar uma peça teste, o que geralmente leva tempo e é an ti-econômico particularmente para peças únicas, lotes muito pequenos ou ainda de material de elevado preço.

Um outro método é a simulação gráfica da pe ça, trabalho feito com auxílio do "plotter".

O sistema EXAPT contém um módulo auxiliar chamado ZEIEX que traça:

- Os percursos da ferramenta programados ou automaticamente gerados.
- Secções transversais (bruta e acabada) de pe ças rotativas.
- Os dispositivos de fixação.
- Detalhes ampliados de regiões carregadas de linhas do desenho, para auxiliar a análise de processos de usinagem complexos.
- Em casos de peças de torneamento, as duas par tes simétricas ao eixo podem ser traçados.

O Anexo 5 ilustra um programa auxiliar cons truído para o "plotter" digital modelo 7210A (Hewlett Packard) do laboratório de CN da UFSC, além de exemplo de sua utilização.

Manipulação de Fitas Perfuradas

Paralelamente à preparação de dados de contro le para a máquina ferramenta CN, a administração e manipulação das fitas perfuradas já prontas, também consome um tempo que deve ser considerado.

Os problemas mais frequentemente encontrados são:

- Duplicação de fitas para assegurar que uma fita em bom estado esteja disponível em qualquer época.
- Listagem de fitas CN tanto em forma tabular, quanto codificada, para verificação do seu conteúdo.
- Armazenamento de fitas perfuradas em discos magnéticos.
- Modificação do conteúdo de fitas CN, por exemplo para atender a modificações nas condições de fabricação, sem ter que repetir todo o cálculo computacional na preparação da nova fita.
- Mudança de sistema de codificação (exemplo: ISO, EIA, ASCII, etc.).

Para resolver estes problemas foi desenvolvido o programa JANUS no sistema EXAPT.

A figura 16 mostra o funcionamento do sistema JANUS.

Conversão de Fitas Perfuradas

Outro problema de programação CN por computador é a preparação de fitas perfuradas para várias máquinas CN com a maioria das propriedades técnicas idênticas, mas com controles diferentes. Frequentemente fitas confeccionadas por sistema assistido por computador ou manualmente, são codificadas manualmente para atender às variações de controle. Isto significa que para cada nova fita a ser preparada para combinações quase idênticas de máquinas/controles, todas as modificações necessárias devem ser manualmente adicionadas.

Para resolver este problema o programa CONVERT foi desenvolvido pela EXAPT para trabalhos com diferentes portadores de dados (fita, disco, etc.).

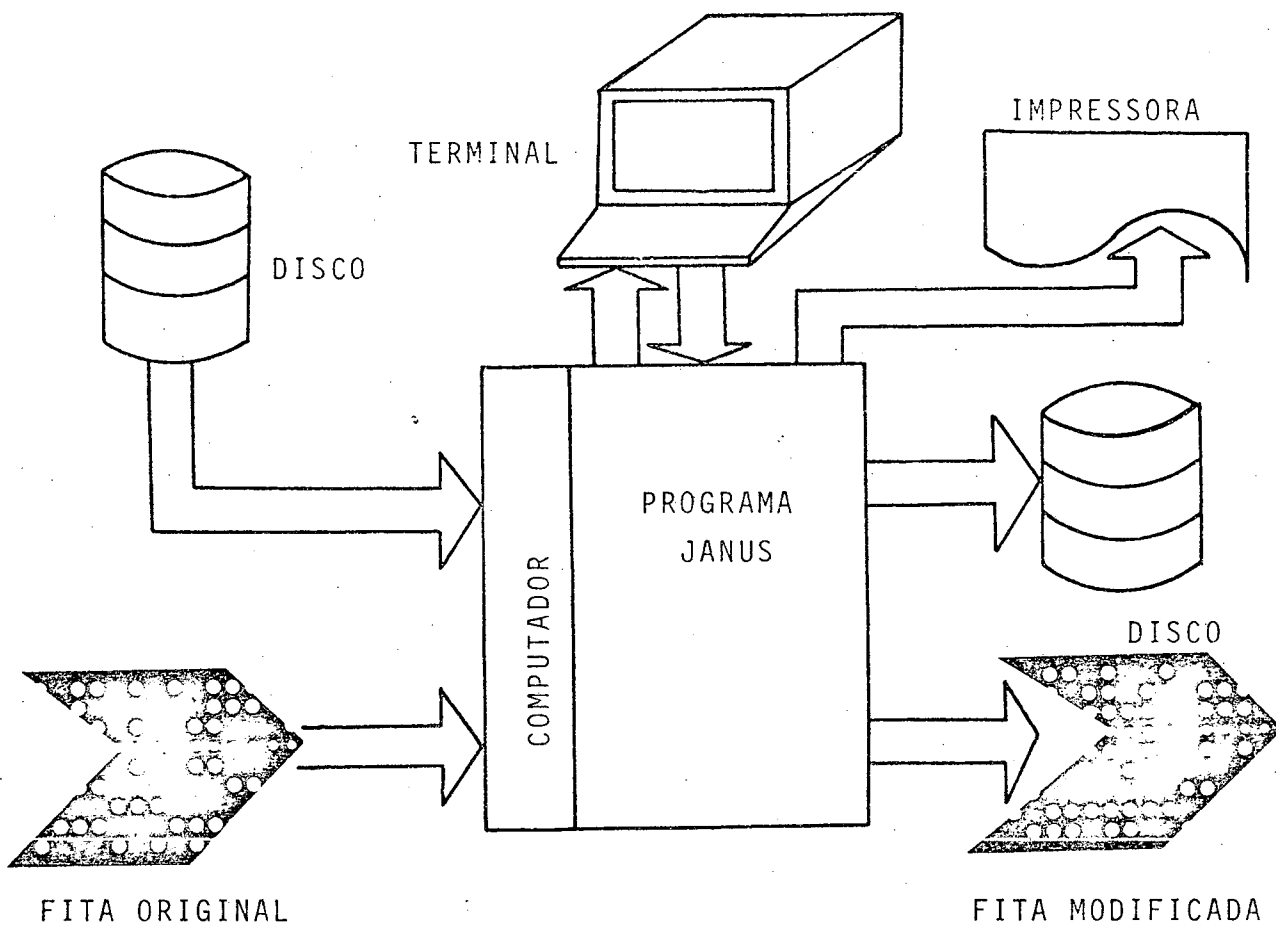


FIGURA 16 Método de trabalho do programa de manipulação de fitas e arquivos JANUS .

As características construtivas de cada configuração controle/máquina são armazenadas em forma de tabela de decisão e controle, em memórias periféricas de acesso direto. As fitas dos programas a serem convertidos são interpretadas, convertidas e transformadas em uma forma normalizada de dados, estruturados para o conteúdo dos novos programas, referentes aos novos controles.

Administração de Programas CN e de Macros

Em programação CN é comum a necessidade de armazenar programas prontos para reutilização posterior. Frequentemente estes programas devem ser readaptados para condições de trabalho diferentes como novas máquinas, novas ferramentas, etc.

Programas completos ou trechos de programas, que são utilizados amiúde, são catalogados e arquivados no sistema.

Existe um módulo EXAPT para simplificação da administração e racionalização da preparação de programas de peças, utilizando programas arquivados. Este módulo procura em cada programa novo as referências a programas reutilizados e substitui estas referências pelos próprios programas, que são copiados dos arquivos. Além disto, informações sobre materiais, ferramentas, etc., mais ou menos invariantes no tempo, são administradas no sistema EXAPT por dois módulos chamados MAPEX (Maintenance Program for EXAPT Files - Programa para manutenção de arquivos para o EXAPT) e DAFES (Datenfile Erstellungssystem: sistema de gerenciamento de arquivo de dados).

O módulo MAPEX se destina à manutenção dos arquivos do Processador EXAPT com extensão de tecnologia automática para furação e fresagem, e o módulo DAFES aos arquivos do processador EXAPT-BÁSICO e EXAPT com extensão de tecnologia automática para torneamento.

As finalidades básicas destes dois módulos são de três ordens:

- Acrescentar informações ao arquivo
- Eliminar informações do arquivo
- Modificar informações do arquivo.

Sempre que se referencie a estes módulos com os comandos próprios, primeiramente se testa a validade lógica, geral e sintática do comando e após se faz o mesmo com as informações sobre o arquivo. Este procedimento mantém alto o grau de preparação dos arquivos, assegurando um desempenho melhor do processador, que deles se alimenta.

Assim como todos os processadores EXAPT, estes módulos são escritos em FORTRAN.

A estrutura genérica dos módulos de administração de arquivos é dada na figura 17.

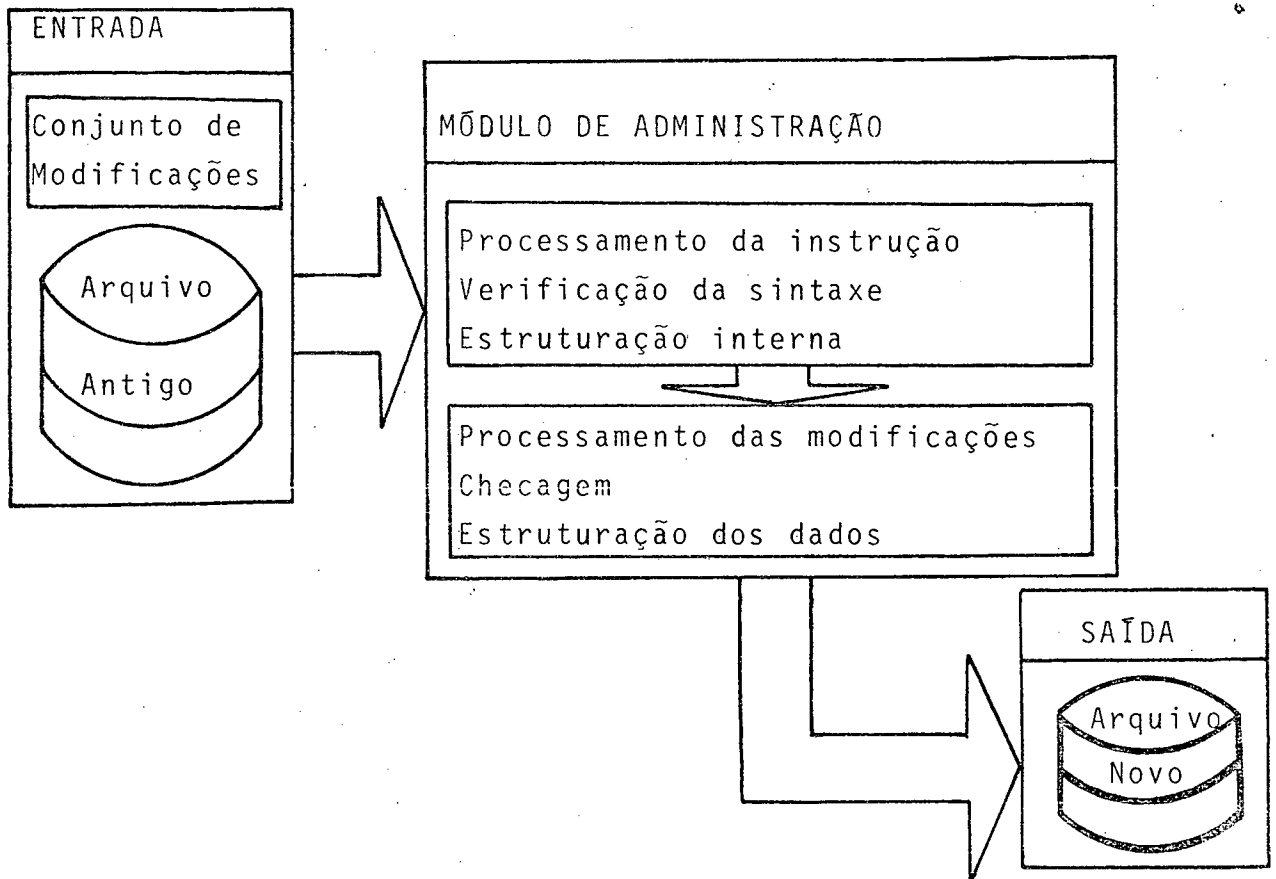


FIGURA 17 - Modo de trabalho dos módulos de gerenciamento de banco de dados MAPEX e DAFES.

No uso prático de sistemas para administração de programas CN, as seguintes vantagens podem ser conseguidas, além das vantagens adicionais de redução do esforço de administração:

- redução de informações necessárias, durante a entrada para o sistema de programação CN.
- simplificação do serviço de mudanças.
- redução de custos durante a preparação de portadores de dados (fitas, discos, etc).
devido a:
 - utilização de registros previamente preparados para peças repetidas ou similares.
 - utilização dos aspectos dos grupos de tecnologia (exemplo: procura de dados de fabricação para peças similares ou idênticas).
 - redução do tempo de processamento na preparação de partes similares ou repetidas.
 - redução da taxa de erros devido à utilização de programas já amplamente testados e otimizados.

3. A IMPLEMENTAÇÃO DO PROCESSADOR EXAPT-BÁSICO E DO SISTEMA DAFES

3.1. Introdução

Em um trabalho de implementação de um sistema de "software" num computador, existem dois aspectos básicos a enfatizar:

- o sistema a implementar
- o computador suporte

Antes do trabalho de implementação, existem dissociados um computador (com sua configuração de "hardware" e "software" definidos) e o sistema a implementar.

Feita a implementação, a instalação de processamento de dados dispõe, em seu quadro de "software", o originalmente existente e o implementado, como mostra a figura 18.

Este capítulo tem por objetivo descrever o trabalho de implementação do processador EXAPT-BÁSICO e do sistema de administração de dados para o mesmo sistema, denominado DAFES, relatar os problemas encontrados e as soluções adotadas, fornecendo adicionalmente uma seqüência de informações para operar o sistema implementado.

A implementação de um sistema de programação CN em um mini-computador envolve, entre outras coisas:

- A compatibilização dos programas que formam o seu processador com as características peculiares do compilador utilizado no mini-computador.

- A compatibilização do processador com o sistema operacional disponível no mini-computador e com as peculiaridades da CPU e das memórias auxiliares da configuração existente.

Esta compatibilização não envolve mudança na estrutura funcional ou construção dos módulos do processador - que têm geralmente mais de 10 anos de desenvolvimento contínuo, como

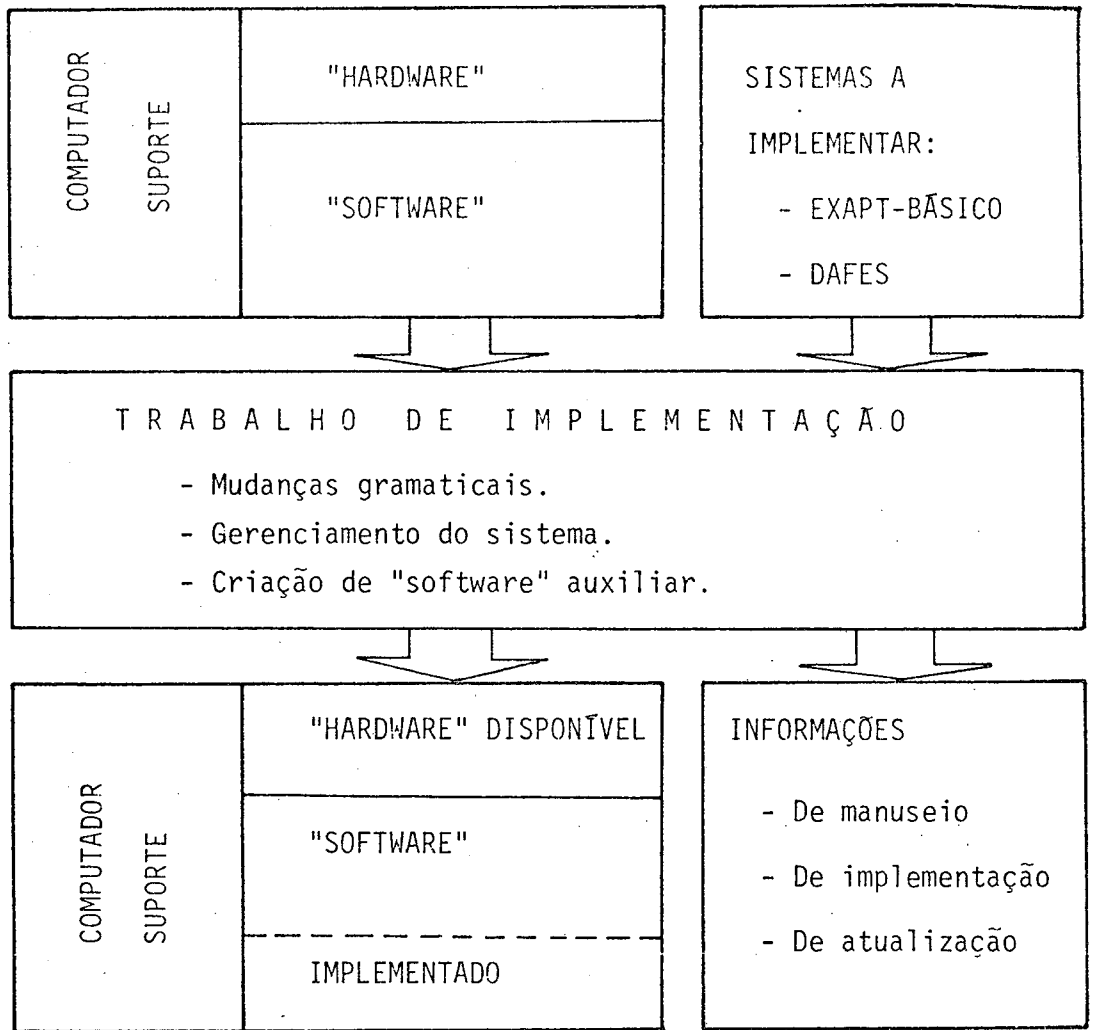


FIGURA 18 - Implementação de sistemas de "software" em computador.

o EXAPT - mas pode abranger aspectos complexos como o gerenciamento da memória através de programas em ASSEMBLER, especialmente construídos pelo implementador (ver Anexo 3).

É óbvio que as modificações exigidas são poderão ser executadas por quem tenha, ao mesmo tempo, um profundo conhecimento da construção do processador e do trabalho com o mini-computador suporte, já que a implementação depende grandemente da máquina.

3.2. O Computador Suporte

3.2.1. O "hardware"

Apesar de alguns sistemas de processamento, como o EXAPT, terem sido idealizados e construídos para serem utilizados em mini-computadores, eles são, na verdade, constituídos de programas relativamente longos.

Os módulos constitutivos do EXAPT-BÁSICO têm, por exemplo, um total que gira em torno de 40.000 comandos (cartões).

Acrescentando-se os módulos de tecnologia de torneamento e de fresagem-furação, dos sistemas de Bancos de Dados e um ou dois pós-processadores, facilmente se chega a um total de 80.000 cartões.

Os elementos componentes de uma configuração desejável para conter e operar os módulos do EXAPT-BÁSICO e do DA FES são esquematizados na figura 19, sendo:

Memória Principal: deve existir no computador uma disponibilidade mínima de 128 KBytes. Maior disponibilidade melhora o desempenho do sistema EXAPT.

Memória Auxiliar

Disco Magnético: A configuração deve dispor de uma unidade contendo disco fixo e removível.

Fita Magnética: É opcional, mas altamente desejável no sistema, quando da implementação, para facilitar o ma-

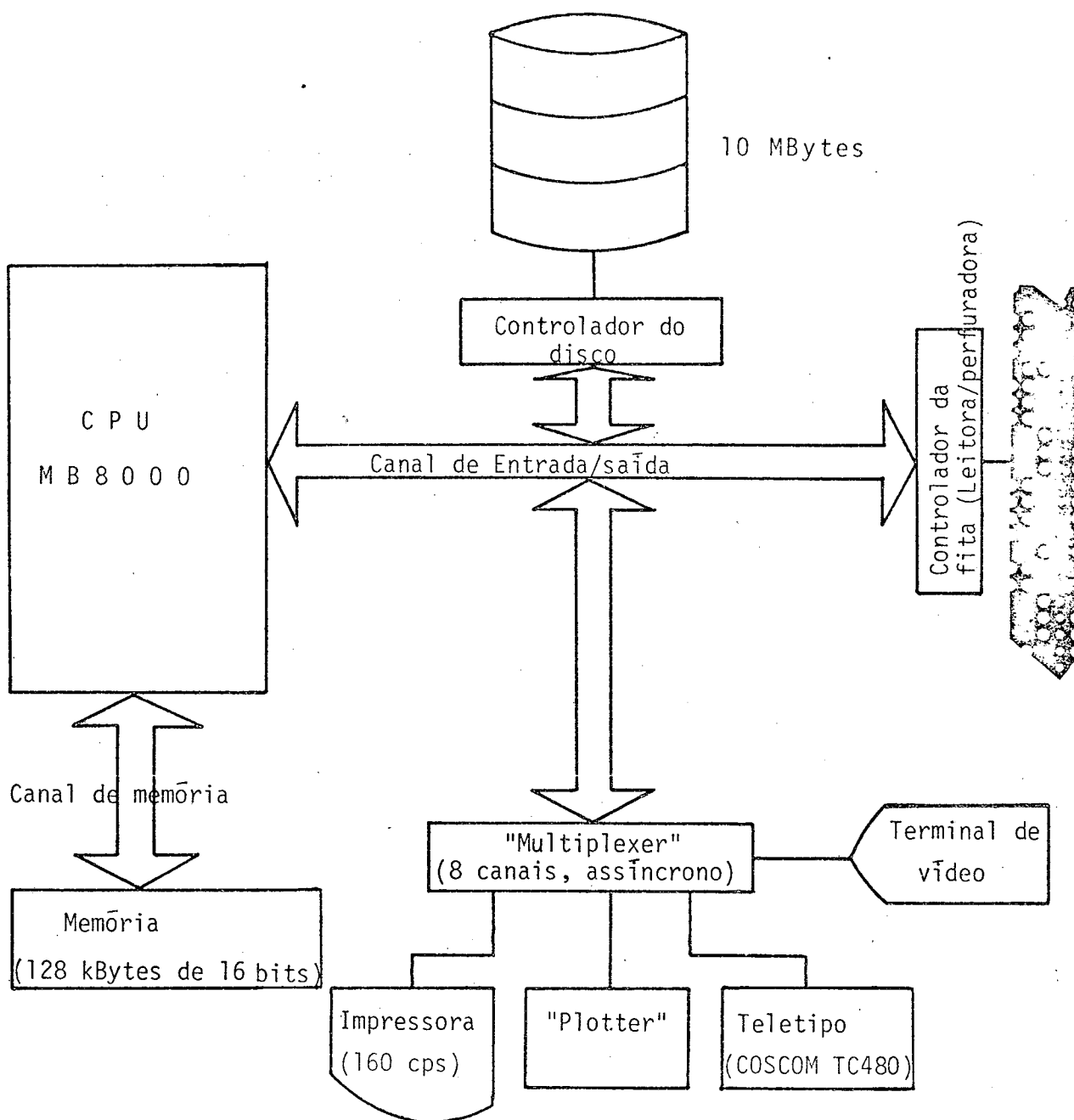


FIGURA 19 - Configuração existente na UFSC para o computador suporte.

nuseio dos arquivos. Durante a fase de produção esta unidade seria de real utilidade facilitando a intercomunicabilidade com outros centros de processamento de dados, ou de arquivo.

"Plotter": Também é opcional, sendo de grande ajuda quando da programação de peças novas. O capítulo IV trata mais em detalhe do uso do "plotter".

Leitora e Perfuradora de fita: Destinadas à produção e ao manuseio de fitas perfuradas para máquinas CN.

Impressora: Usada sobretudo na obtenção das listagens dos programas CN ou de cópias permanentes de modo geral.

Teletipo: No caso da UFSC este terminal, por estar equipado com leitora e perfuradora de fita, substitui eventualmente estes equipamentos, além de poder atuar como impressora ou no lugar da unidade de vídeo, como console da configuração, em algumas aplicações.

3.2.2. "Software"

O "software" necessário para trabalhar com o sistema EXAPT deve ser constituído de um sistema operacional que tenha suporte para processamento de programas escritos na linguagem de alto nível FORTRAN, conter biblioteca de programas utilitários e do sistema, com componentes básicos incluídos e permitir segmentação de programas ("overlay")¹.

A parte de FORTRAN deve incluir o compilador FORTRAN e a biblioteca de programas utilizados nesta linguagem de programação, tais como subrotinas para cálculos de funções trigonométricas ou exponenciais e ainda as necessárias para operações de entrada/saída, de manutenção de arquivos, de interrupção, etc.

A biblioteca de programas necessários é constituída de uma série de elementos, alguns dos quais, particularmente importantes para a implementação (ex.: "LOADER" e EDITOR).

¹Ver conceito de "overlay" pg. 73.

Um dos programas componentes da biblioteca de utilitários e do sistema é o programa "LOADER" (carregador de módulos relocáveis). Este programa carrega e liga ("link-edition") arquivos binários relocáveis produzidos pelo ASSEMBLER e outros arquivos relocáveis, selecionados da biblioteca. Deve ser ainda disponível o carregador de "overlays", recurso largamente utilizado neste tipo de implementação.

Outro programa componente da biblioteca deve ser o EDITOR, utilizado para criar e modificar textos e arquivos de modo geral.

O programa gerador de sistemas operacionais deve ser componente da biblioteca e permitir a criação de um sistema que seja ao mesmo tempo compatível com o "hardware" instalado e com os objetivos próprios da instalação.

3.3. Seqüência da Implementação

3.3.1. Organização do Sistema EXAPT no Computador

O arranjo dos diversos módulos do sistema a ser implementado é dependente do tamanho do computador disponível em termos de sua memória principal.

O processador EXAPT é propositadamente construído de módulos que, em todos os computadores, independentemente de sua organização estrutural, são executados uma única vez para cada programa de peça processado. Isto permite a adoção de um arranjo tal, que o processador seja utilizável tanto em grandes computadores como em "minis". Naqueles, todo o processador poderia estar contido na memória, durante a execução. Entretanto, em ambiente de multi-tarefas, onde mais de um usuário requer os recursos do sistema, como os de entrada/saída, "overlays", controle de CPU, etc., se o processador estiver completo na memória, o usuário do sistema EXAPT obrigará os demais a grandes esperas e o sistema não será utilizado racionalmente. Devido a este fato, mesmo em grandes computadores, o sistema fica seccionado, como esquematiza

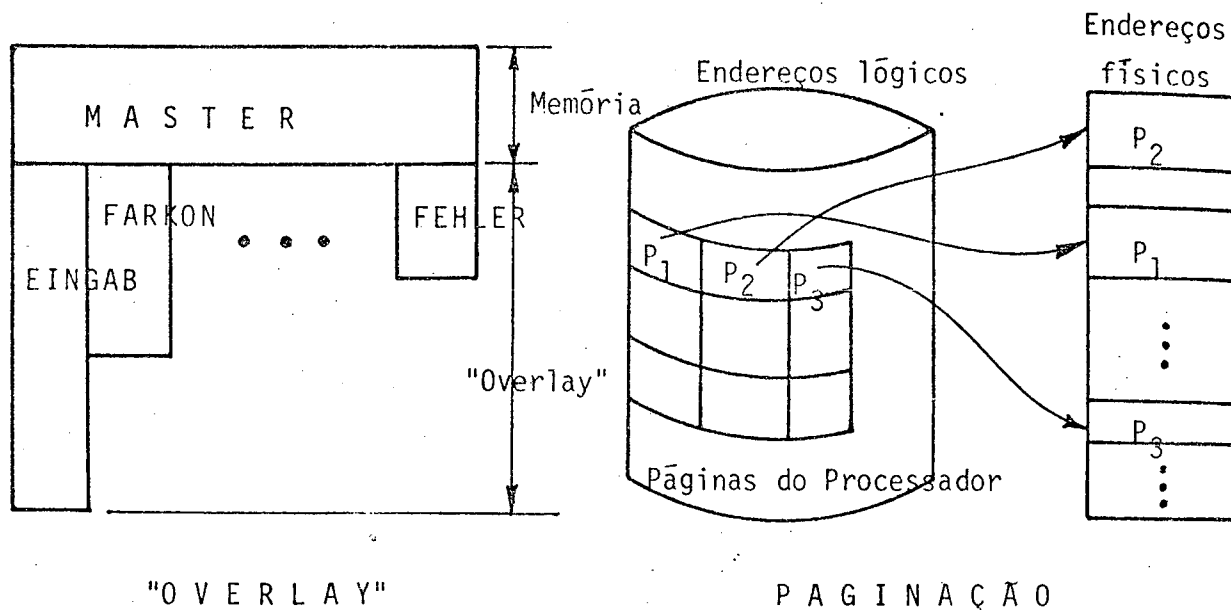


FIGURA 20-Tipos de organização do Processador EXAPT em grandes computadores.

do na figura 20. Outra alternativa é a paginação do sistema, quando se trabalha com memória virtual, e neste caso o sistema operacional carregará na memória uma determinada página - parte definida do processo - que será substituída por outra, que se encontrava em disco (existe mecanismo automático para carregamento das partes não residentes na memória).

Em mini-computadores, cada módulo origina uma tarefa. Os resultados de cada tarefa, que serão as entradas para a tarefa posterior, são gerenciados por um programa chamado MODIOS que escreve estes resultados em disco e os lê em seguida para a próxima tarefa, como se vê nas figuras 21, 22 e 23.

Na estrutura do processador EXAPT-BÁSICO, cada módulo pode ter seus arquivos temporários (figura 22), administrados por ele próprio. Entretanto arquivos inter-módulos são gerenciados pelo sistema MODIOS (figura 23).

Obedecendo às restrições ditadas pelo tamanho da memória principal disponível e pela técnica de segmentação adotada, as seguintes possibilidades de seqüenciamento do desenrolar da computação podem ser utilizadas:

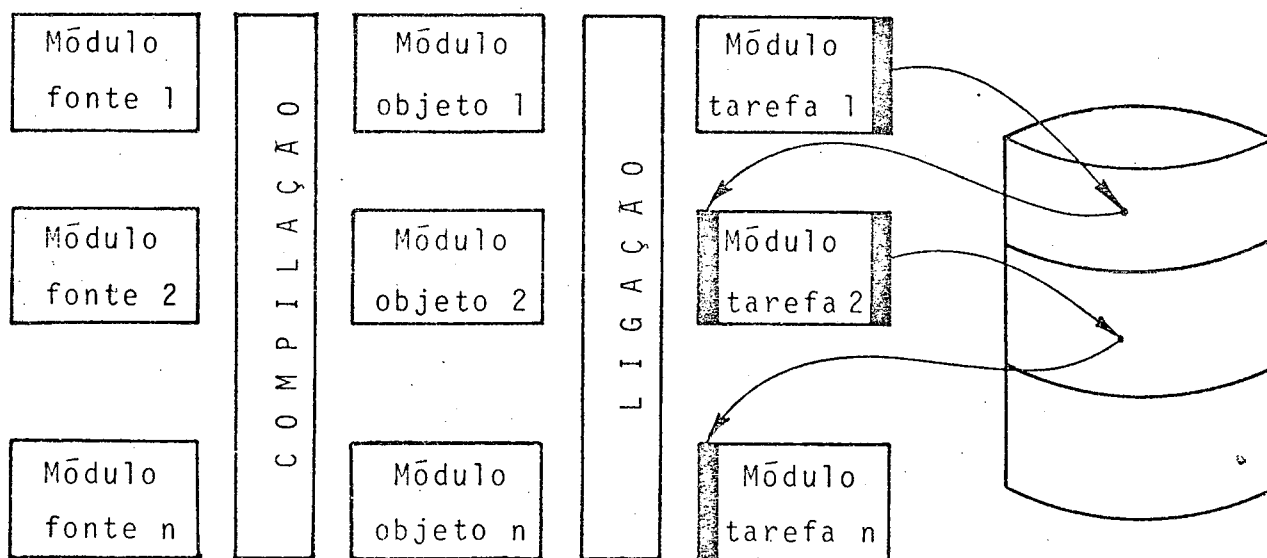


FIGURA 21 - Organização do Processador EXAPT em mini-computadores.

1. O programa "MASTER" é programado em FORTRAN. Os módulos são tratados como ramos de estruturas de "overlay" (subprogramas do MASTER).
2. O "MASTER" é em si mesmo um programa FORTRAN completo e chama os módulos, que são tratados como programas independentes. (As chamadas tarefas - "tasks").
3. O "MASTER" é escrito em uma linguagem de comando própria do computador, também chamada de procedimento ("procedure"). Neste caso os módulos também seriam chamados co tarefas, como no caso 2.

A tabela 2 fornece o processo de implementação utilizado em alguns tipos de computadores, e a técnica de implementação adotada em cada caso. A escolha apropriada do processo de implementação é dependente dos seguintes fatores:

- Capacidade da memória principal e secundária disponível.

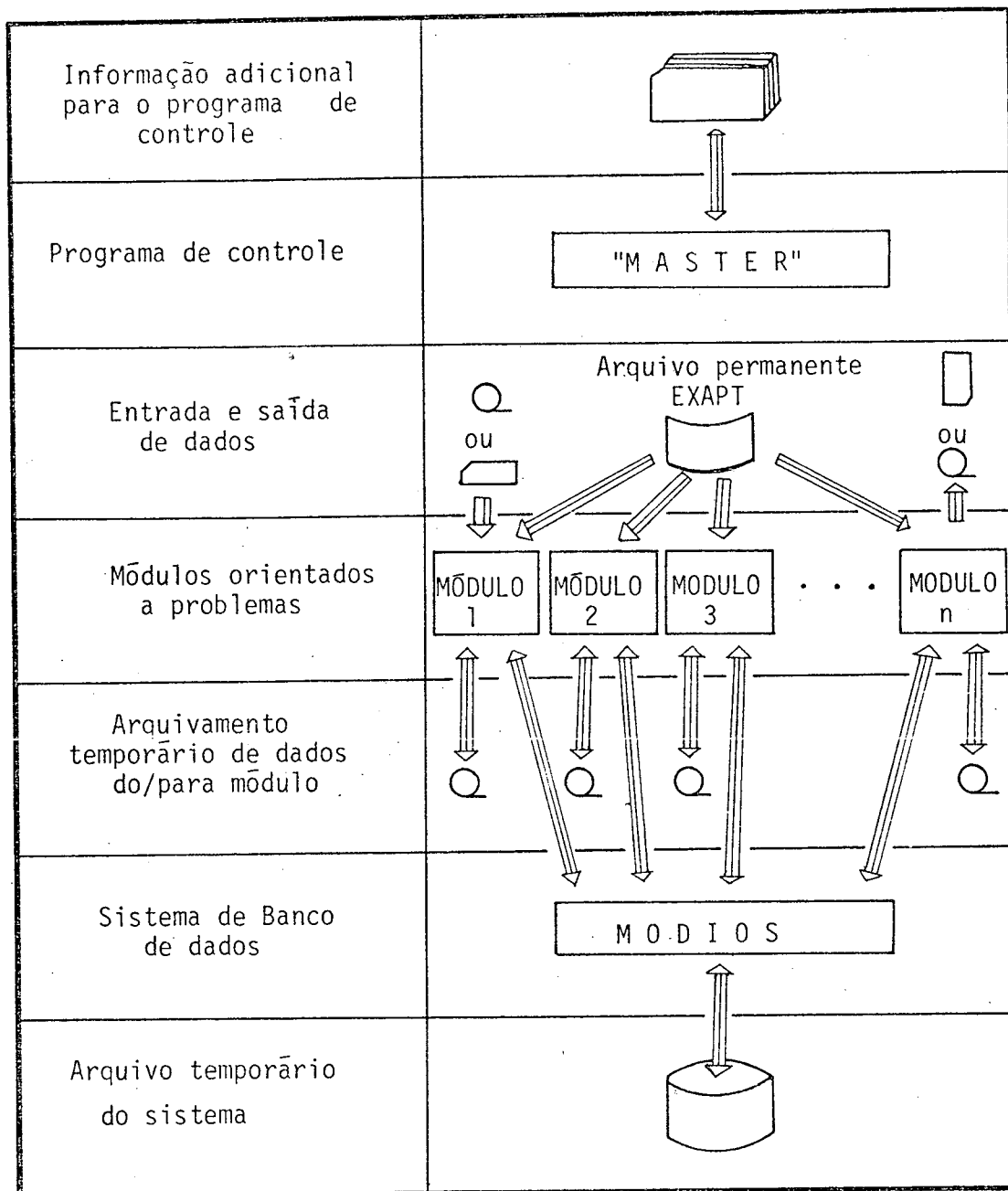


FIGURA 22 - Processador EXAPT: Estrutura do sistema modular.

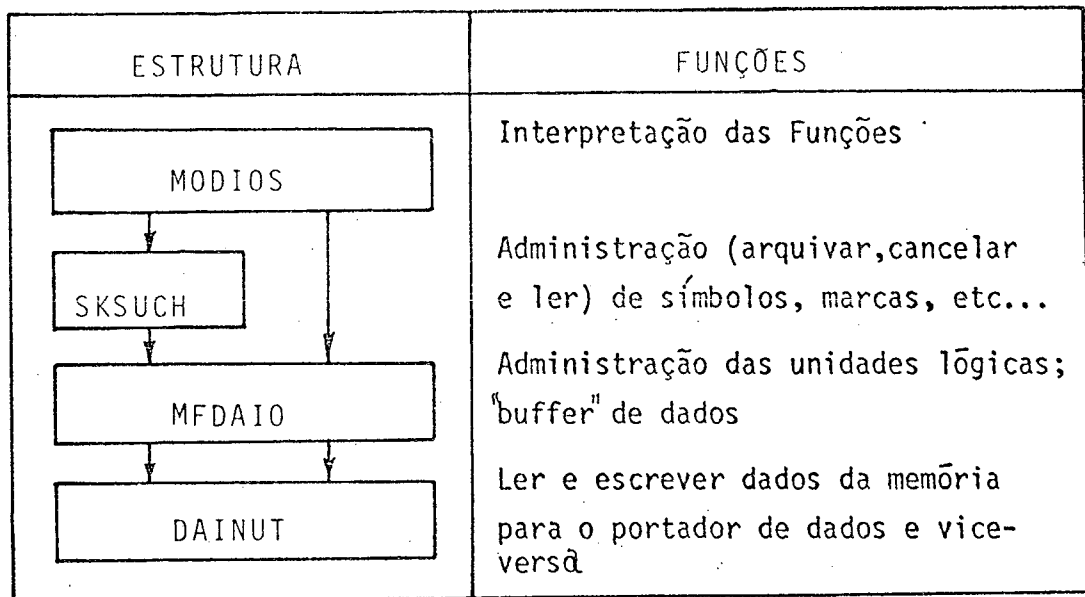


FIGURA 23 - Estrutura do módulo de manuseio de dados MODIOS (MODIOS - Modular input/output system).

- Velocidade de trabalho desejada.
- Número máximo de programas que podem ser vinculados.
- Número máximo de níveis de "overlay" aceitáveis pelo sistema.

3.3.2. Características dos Módulos a Implementar

Os módulos constituintes do processador EXAPT-BÁSICO, são formados de subrotinas que se chamam mutuamente durante o desenrolar do processamento.

A disposição destas subrotinas no computador de origem e a adotada no computador SISCO MB-8000, módulo por módulo, é fornecida no Anexo 1.

Cada módulo está, nestas tabelas, subdividido em grupos de subrotinas. Os grupos de subrotinas podem pertencer a um mesmo "overlay", formar "overlays" diferentes, ou formar a RAIZ (ROOT)² do módulo.

²Ver conceito de RAIZ pg. 73.

COMPUTADOR		TIPO DE IMPLEMENTAÇÃO
TIPO	MODELO	
	1 - IBM 370 OS 2 - IBM 370 DOS 3 - CDC 6600 4 - UNIVAC 1100 5 - SIEMENS	Estrutura de "overlays"
	6 - HP 1000 7 - DIETZ 621 8 - PRIME 9 - SISCO MB-8000 10 - NOVA (*)	"MASTER" e módulos como tarefas ("tasks")
	11 - PDP 11-34	"MASTER" como "Procedure" e módulos como tarefas ("tasks")

(*) Os computadores marcados com este sinal têm o processo de implementação em andamento.

TABELA 2 - Modos de implementação da linguagem EXAPT em diferentes computadores

O inter-relacionamento entre os diversos grupos de subrotinas pode ser melhor apreendido montando os módulos na forma da figura 24, do EINGAB, baseado na descrição de entrada dada no Anexo 1. Nesta figura, os símbolos EG-00, EG-82, etc. são os diversos nomes de "overlays" em que foi particionado o módulo EINGAB para a implementação no SISCO MB-8000.

A montagem dos diversos grupos deve obedecer a critérios de programação e às restrições de capacidade de memória do computador. Deve-se atender, de qualquer modo, à regra que se uma subrotina é chamada por outra, ambas devem estar simultaneamente disponíveis na memória principal. Então uma subrotina comum a dois ou mais "overlays" deve pertencer a todos eles ou estar na RAIZ do módulo para que esteja à mão quando chamada.

3.3.3. Passos da Implementação

Encarada sob ângulo restrito, a implementação de um sistema pode ser entendida como um processo composto da execução de cinco passos:

- 1º - Providenciar espaço para armazenar os arquivos a implementar.
- 2º - Copiar os arquivos do portador de dados (disco, fita magnética, perfurada, etc.) para o espaço reservado.
- 3º - Compilar os programas componentes do sistema.
- 4º - Ligar e carregar os diversos módulos.
- 5º - Executar os programas.

A efetivação prática destes passos subdivide-se em uma série de ações que no total formam um conjunto de complexidade razoável, dependendo sobretudo das limitações da configuração com que se trabalhe.

A figura 25 fornece uma visão global do processo, supondo que se trabalhe com um computador auxiliar.

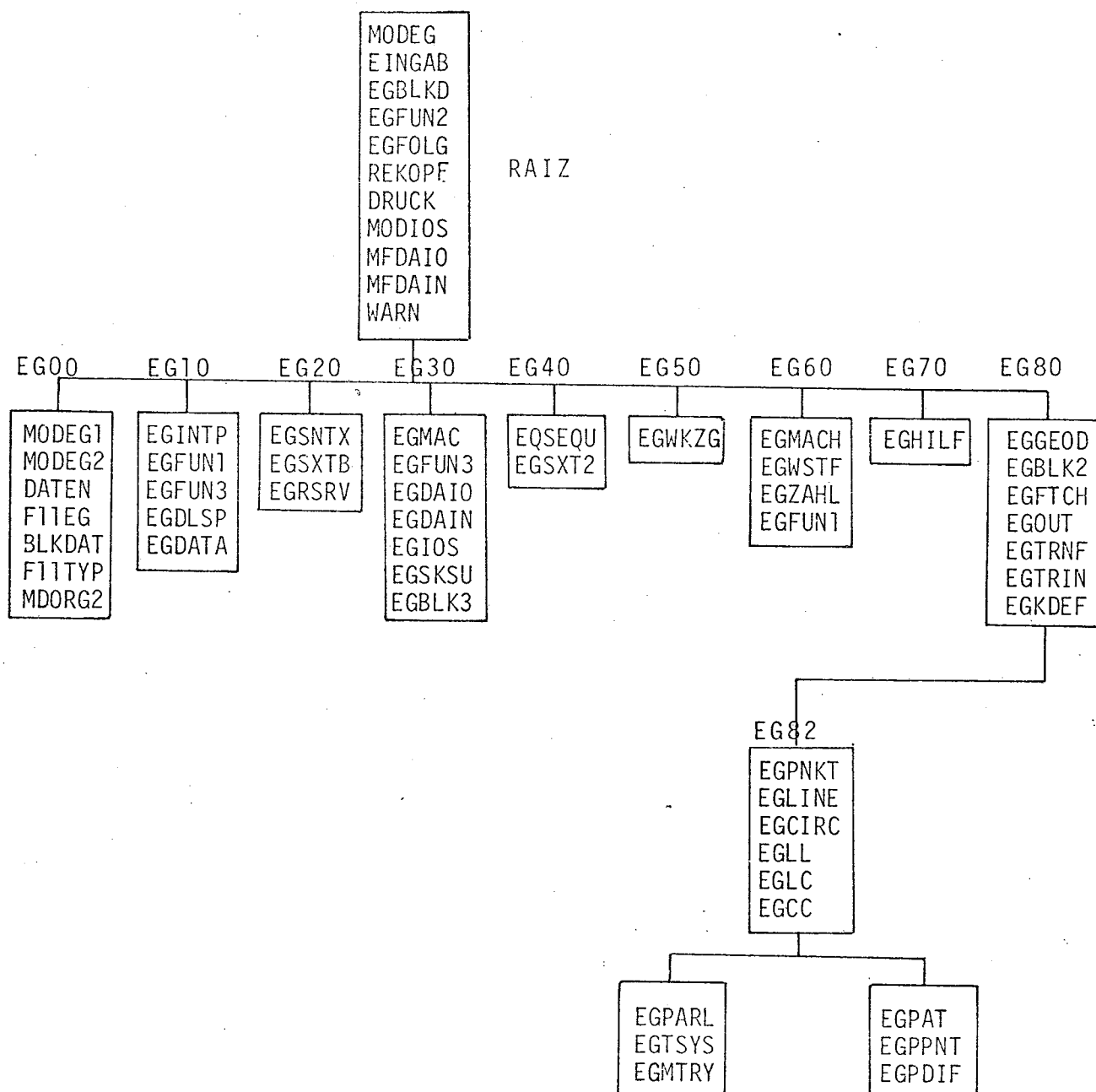


FIGURA 24 - Inter-relacionamento entre os grupos de subrotinas constituintes do módulo EINGAB.

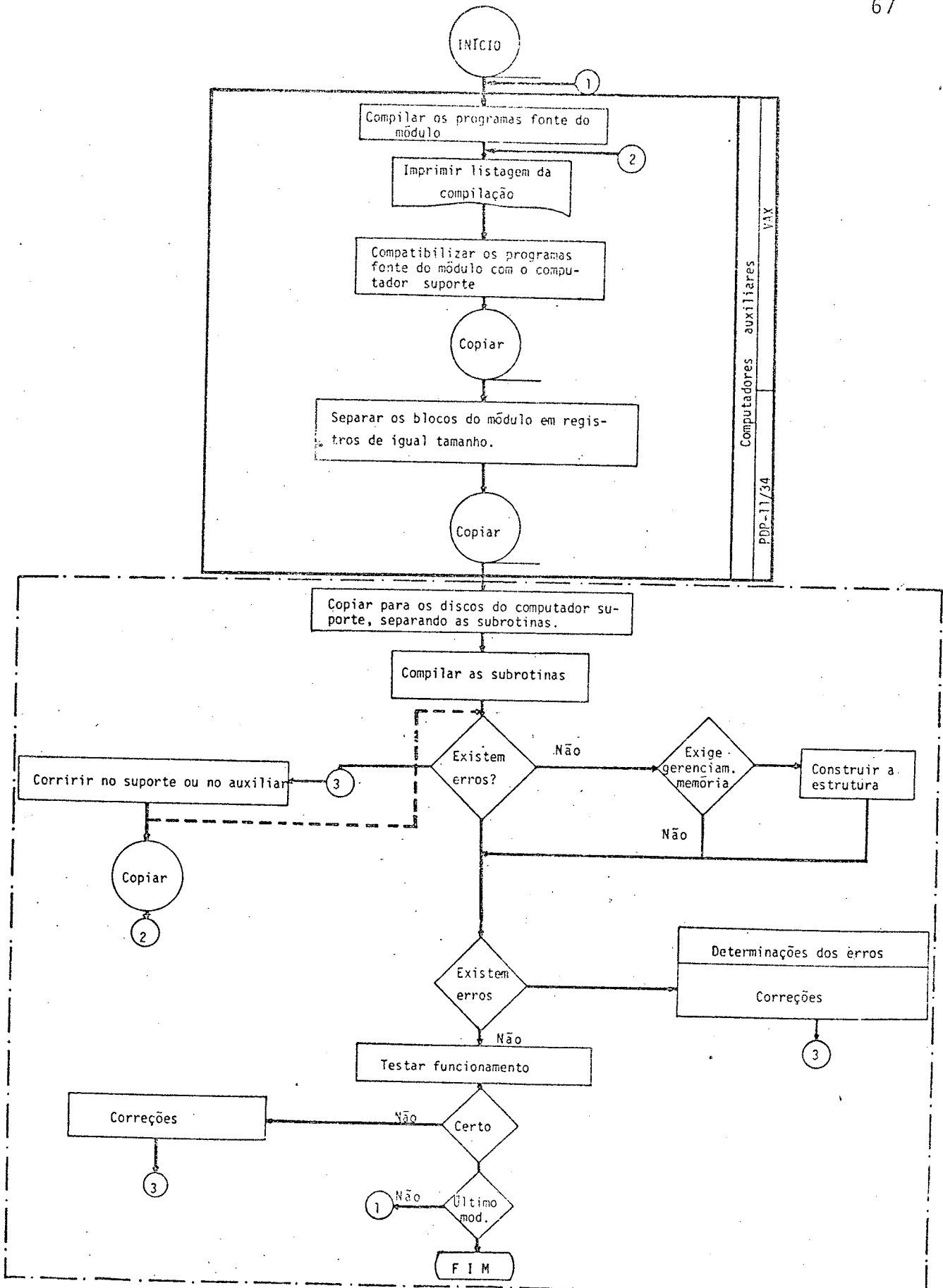


FIGURA 25 - Fluxograma adotado na implementação SISCO MB-8000.

No caso da implementação em pauta, um teste da viabilidade do projeto foi feito no computador ECLIPSE S200- idêntico ao SISCO MB-8000 - da Rheinische Westfalische Technische Hochschule, de Aachen, RFA.

O trabalho de compatibilização foi, neste teste prévio, executado no computador PDP 11/34 e no computador VAX, ambos de fabricação da Digital Equipment Corporation, de propriedade da Associação EXAPT, por questão de facilidade e de economia.

Caso não se disponha de um computador auxiliar, todo o trabalho deverá ser feito no computador suporte.

Em tomada menos panorâmica e mais detalhada, a execução destes seis passos exige as ações a seguir tratadas, embora somente as mais importantes.

a. Escolher o Módulo a Implementar

A escolha do módulo deve levar em conta a representatividade quanto ao tamanho e à constituição estrutural do mesmo.

Tomou-se inicialmente o módulo de administração de arquivos de dados DAFES, por ter tamanho significativo em relação aos demais e por exigir tratamento com "overlays".

Na implementação do primeiro módulo, se convenientemente selecionado, aparecerão todas as dificuldades a serem tratadas nos demais.

b. Tomar o Módulo Fonte da Biblioteca Original (Computador PDP 11/34, da EXAPT) e Compilá-lo no Computador Suporte

Este passo tem por finalidade a obtenção de uma listagem com as informações de tipos de variáveis utilizadas, de mapeamento, etc. para uso em futuras correções.

Os módulos escritos em linguagem de alto nível estão armazenados em um periférico qualquer de um computador auxiliar e organizados em bibliotecas.

A seguir se compila os programas do módulo escolhido e se obtêm as listagens da compilação. Trabalha-se sobre estas listagens para fazer as compatibilizações necessárias, exigidas pelo compilador do computador suporte.

c. Compatibilizar os Programas-Fonte do Módulo com o Computador Suporte

Trabalha-se no diretório do computador auxiliar, fazendo as modificações necessárias.

Com auxílio do programa editor do computador, executam-se as modificações, compatibilizando o módulo às exigências do computador suporte, como em termos de compilação, por exemplo.

É óbvio que se o computador suporte estiver disponível, todos estes ajustamentos poderão ser diretamente nele executados.

De máxima importância é manter uma cópia de todos os módulos originais em um portador de dados. Este cuidado se deve ao fato de sempre se poder contar com uma fonte segura que servirá como ponto de partida caso em algum ponto da implementação se perca algum arquivo, ou se chegue a algum ponto de estrangulamento.

Na execução das modificações deve-se elaborar uma lista de modificações a processar. Toma-se a seguir cada uma das modificações da lista e percorre-se todos os programas do módulo, procedendo a mudança, até esgotar a lista.

A lista de modificações, elaborada para a implementação do processador EXAPT no computador SISCO, é fornecida no Anexo 2.

d. Separar o Módulo em Registro-por-Registro

Os arquivos, em alguns computadores, são trabalhados por registros constantes (cartão por cartão) como é o caso do SISCO. Em outros arquivos são subdivididos em blocos de determinado número de "Bytes". Caso o módulo original venha de um computador em arquivos bloqueados, antes de submetê-lo ao computador SISCO, deve-se separá-lo em registros de um cartão cada.

Este trabalho pode ser executado no computador auxiliar. No caso do SISCO o trabalho foi feito com o auxílio de três computadores:

- VAX com blocos de 2048 "Bytes"
- PDP 11/34 com blocos de 512 "Bytes"
- SISCO com blocos de 512 "Bytes"

Quando se transfere o módulo do VAX para o PDP 11/34 a ajustagem é feita automaticamente apenas se informado ao PDP, quando da transferência, que na fita fonte os arquivos tem 2048 "Bytes".

Entretanto não se pode submeter o módulo com 512 "Bytes" do PDP diretamente ao SISCO porque este tem sua organização feita de modo diferente que no PDP, embora o tamanho dos blocos seja idêntico.

Devido a esta restrição, foi escrito o programa MTW que transforma os blocos de 512 Bytes em que se acha o módulo original, em outro constituído de registros únicos.

A figura 26 ilustra o trabalho executado no PDP 11/34 pelo programa MTW, que é fornecido no Anexo 3 deste trabalho.

e. Copiar do Módulo nos Discos do Computador Suporte

O compilador FORTRAN do computador SISCO trabalha compilando separadamente subrotina por subrotina.

O programa RTAP3 tem a função de ler a fita magnética com o módulo a implementar para os discos do SISCO, se-

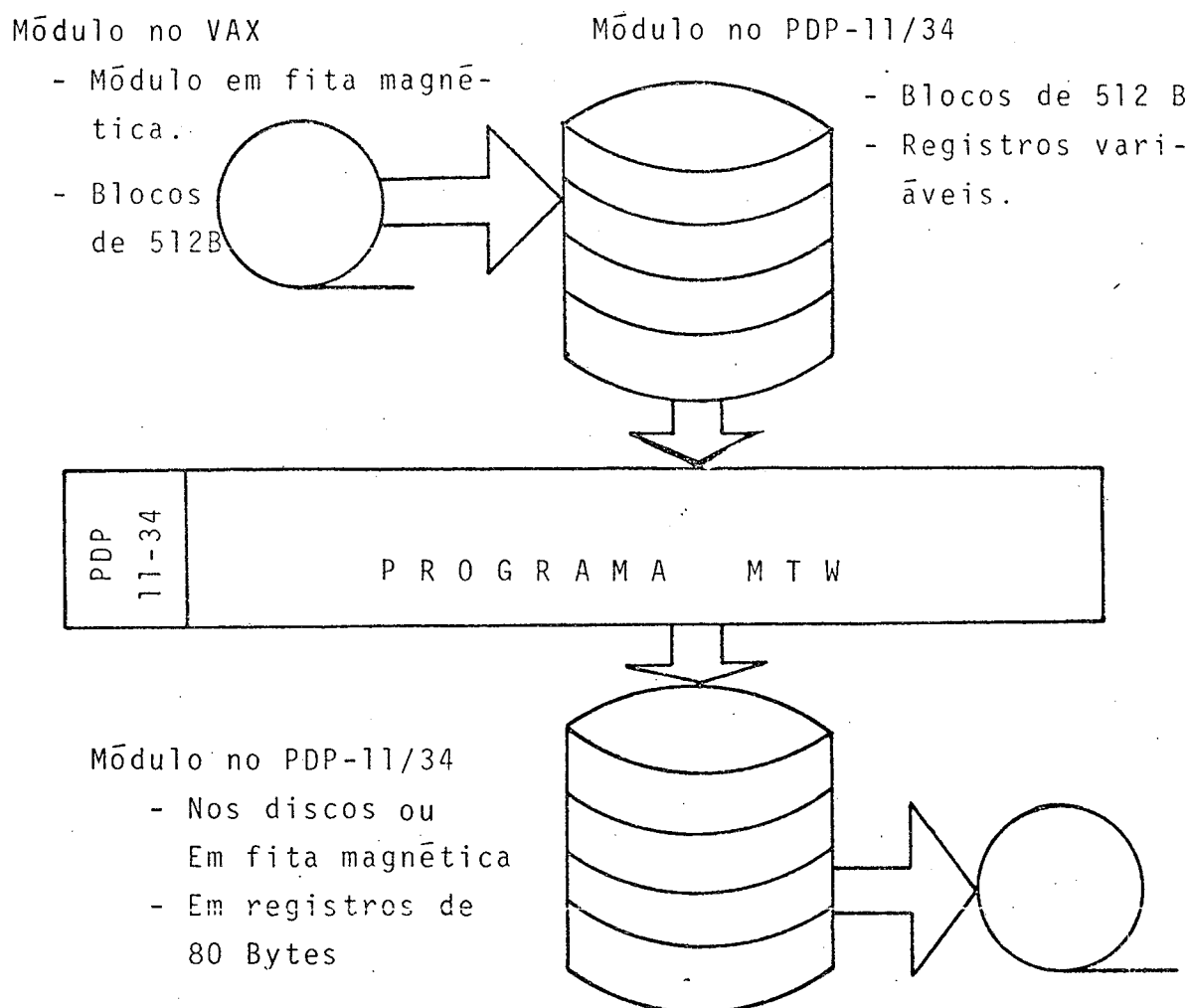


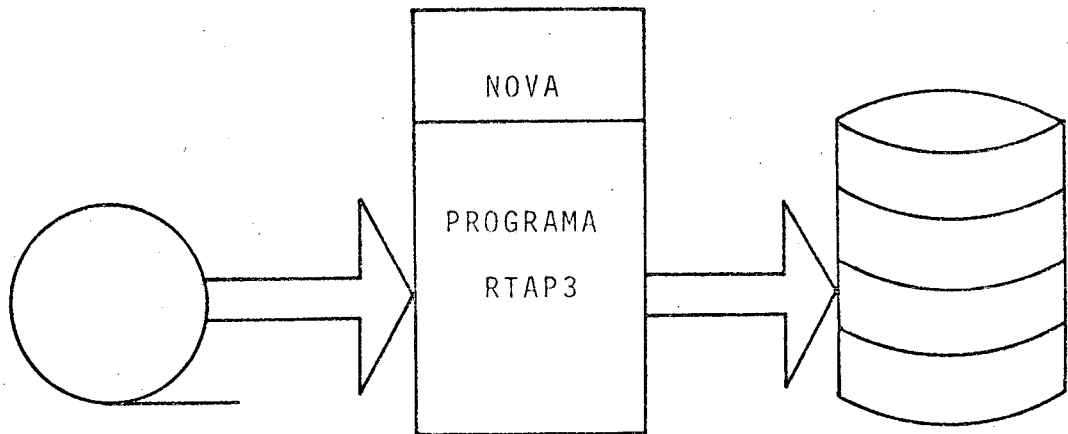
FIGURA 26 - Função do programa MTW processado no computador auxiliar PDP-11/34.

parando cada subrotina da precedente.

O programa RTAP3 identifica o início de uma subrotina quando encontra uma bandeira previamente colocada. Nesta implementação esta bandeira foi posta com o auxílio do programa editor do computador auxiliar VAX, na altura do passo descrito no item c.

A figura 27 ilustra o trabalho executado pelo programa RTAP3 no computador SISCO.

A listagem do programa RTAP3 é fornecida como parte do Anexo 3.



Módulos (originais) em
fita magnética

ORGANIZAÇÃO: [- blocos de
80 "Bytes"
- sub-rotinas
unidas em
arquivos

Módulos nos
discos do SISCO

ORGANIZAÇÃO: [- blocos de 512
Bytes
- sub-rotinas
separadas

FIGURA 27 Função do programa RTAP3 no computador de teste NOVA.

f. Compilar as Subrotinas do Módulo

Uma vez disponível o módulo a implementar nos arquivos do computador suporte, passa-se à compilação de cada uma de suas subrotinas, separadamente.

Este trabalho é executado com auxílio do Compilador FORTRAN disponível na biblioteca de subprogramas do SISCO. É interessante escolher as opções de compilação adequadas ao trabalho da implementação, como a geração de listagens e de arquivos necessários (binários relocáveis).

g. Corrigir no Computador Suporte

Caso se tenha constatado erros na compilação das subrotinas, poder-se-á adotar a política de corrigir estes erros no próprio computador suporte. Este procedimento pode ser ado

tado, quando houver muito tempo de máquina disponível. Pode-se ainda corrigir alguns no suporte e optar por corrigir no computador auxiliar os de maior exigência em termos de tempo:

h. Verificar se o Módulo Necessita Gerenciamento de Memória

Quando a memória principal de um computador não é de tamanho suficiente para acomodar, de modo completo, o programa do usuário, pode-se optar pelo método dos "overlays" para tê-lo no computador.

Neste método, fica residente na memória principal do computador suporte, apenas um trecho do programa, que se chama de RAIZ. O trecho complementar é subdividido em partes chamadas "overlays", que ficam residentes em disco. Sempre que se necessitar de um trecho de programa em área de "overlay", este é solicitado pelo programa em processamento e o sistema carrega este "overlay" na memória principal, em área para isto reservada.

Quase todos os módulos do processador, devido ao seu tamanho relativamente grande para a memória da configuração do SISCO usado, sofreram tratamento de "overlays". Neste caso já devem estar introduzidas, no passo c, as modificações próprias exigidas pelo sistema operacional para gerenciar áreas de "overlay", para o módulo.

Nos módulos EINGAB, CONTUR e CUTVAL foi adotado também o processo de troca ("swapping") entre subrotinas. Nesta técnica o programa A chama B como uma subrotina, podendo B ser do tamanho total da memória disponível. O programa A é guardado antes em disco e se copia B para a memória. O programa A retorna à memória e retoma o controle após a execução de B, ou B pode ser trocado com outro programa. Nos módulos citados acima, se usou troca para a subrotina MODIOS que trata da entrada e saída de dados nos diversos módulos (Ver Anexos 1 e 3).

i. Ligar os Programas ("LINK-EDITION")

Uma vez compiladas e depuradas as subrotinas do módulo, faz-se a ligação dos arquivos relocáveis, produzindo-se um arquivo objeto para ser executado.

A ligação deve ser executada conforme as características de cada módulo. A estrutura adotada para cada módulo, no computador suporte, está descrita no Anexo 1, e o Anexo 3 traz arquivos para auxílio desta tarefa de ligação.

j. Determinar e Corrigir Erros

A fase de ligação de programas pode acusar erros que foram cometidos em passos anteriores (de FORTRAN, de "overlays", etc.) ou na própria ligação. Novamente se deve decidir entre a correção no próprio suporte ou em outro computador.

Volta-se portanto ao passo descrito em c.

3.4. Utilização do Processador EXAPT-BÁSICO

O assunto referente a este item é tratado no Anexo 4.

3.5. Utilização do Sistema DAFES

O sistema DAFES organiza o banco de dados para os processadores EXAPT-BÁSICO e EXAPT 2. Como o processador EXAPT 2 só é operacional em computadores de grande porte, não há possibilidade de implementá-lo no SISCO MB-8000, e conseqüentemente não há interesse em implementar as rotinas do DAFES que organizam dados para este processador.

Devido a este fato, as subrotinas que administram dados de máquinas e dados de material não foram implementadas. Assim também as subrotinas para listagem dos conteúdos do banco de dados de materiais de torneamento (DFLS4), de máquinas (DFLS5) e de materiais de fresagem ou furação (DFLS7).

A omissão da implementação destas subrotinas, traz como vantagem principal a maior disponibilidade de espaço de memória principal da máquina.

O Anexo 1 mostra a estrutura do sistema DAFES como implementado no SISCO.

UTILIZAÇÃO

Assim como o processador EXAPT e o sistema operacional, o sistema DAFES está residente no disco fixo, estando o removível reservado para programas e dados do usuário.

O trabalho com o sistema DAFES é iniciado de modo dialogal.

Entrando com o comando inicial

DAFES

o sistema apresenta na tela do terminal TV-2000 a solicitação do código da unidade de entrada:

UNIDADE DE ENTRADA: nome

O operador responde a "nome" com um dos códigos:

TY se a unidade de entrada for o terminal de vídeo TV-2000 ou o terminal tipo "teletype" COSCOM.

DK se a unidade de entrada for o disco magnético.

A seguir o sistema apresenta na tela do terminal TV-2000 a solicitação do código da unidade de saída:

UNIDADE DE SAÍDA: nome

O operador responde a "nome" com um dos códigos:

TY com significado já descrito.

DK com significado já descrito.

LP se a unidade de saída for a impressora de linhas.

Quando ocorrer código DK, o sistema solicita o nome do arquivo com que vai trabalhar. Se se fornece o nome de um arquivo de saída já existente, o sistema informa este fato e pede que se entre com a letra

I

para o caso de ser ignorado, isto é, existindo tal arquivo, o comando I no terminal faz com que o arquivo existente seja apagado e gravado o novo conteúdo, com o mesmo nome.

Definidas as unidades de entrada e saída, utiliza-se o sistema DAFES como ilustrado no manual da Associação EXAPT "DAFES MANUAL-VERSION S1/2 nº 79.12.05 E". [7]

C A P Í T U L O I V

4. MANIPULAÇÃO DE UM SISTEMA CN IMPLEMENTADO EM MINI-COMPUTADOR

4.1. Introdução

Um dos parâmetros de medição da eficiência de um sistema de programação CN é o grau de utilização do computador.

A eficácia com que se trabalha em um sistema CN de processamento, pode ser modificada pela atuação tanto com ações externas ao sistema, isto é, com trabalhos que visem seu uso racional, quanto com ações internas, por exemplo, com atividades que visem aumentar suas habilitações.

Um sistema que se contente com deglutir programas escritos na linguagem de programação respectiva e expelir fitas perfuradas, sem dúvida não faz jus aos altos investimentos requeridos para implantá-lo e zomba da capacidade do computador em que foi instalado.

Embora tal nível de utilização possa ser mais alto que a programação puramente manual, deve-se aproveitar mais intensamente as potencialidades com computador.

A melhoria interna do sistema é conseguida, a nível de empresa usuária, sobretudo com a criação de pós-processadores inteligentes e de programas auxiliares úteis.

Os pós-processadores são geralmente fornecidos pelo próprio construtor da máquina-ferramenta; mas podem também ser conseguidos do fornecedor do sistema de PPC ou por escritórios de "software" especializados. De qualquer modo a responsabilidade ou o interesse máximo de seu funcionamento é do usuário final do mesmo. Ele é quem determinará a ausência ou presença de certas habilidades no programa. Por isto, do conhecimento de sua construção depende a potencialidade de modificá-lo, ou mesmo criar outro mais adequado ao seu problema.

Quanto aos programas auxiliares, que são construídos com o intuito de administrar as informações de CN já conseguidas, de transformar programas existentes em outros formatos, enfim, de auxiliar o programador em sua tarefa, promovendo um maior uso da ferramenta computacional, é tarefa de cada usuário construir ou obter os que melhor convierem ao estado particular do seu sistema.

4.2. Pós-Processadores

4.2.1. Conceituação

Pós-processador é um programa individual construído para levar em conta as peculiaridades da combinação máquina-ferramenta/unidade de controle, na qual a peça será de fato usinada.

Como o processador executa cálculos geométricos, determina o percurso da ferramenta e produz uma solução do problema de usinagem da peça, independente e geral, esta deve ser adaptada por um pós-processador, a uma combinação máquina-controle determinada para executar de fato a mesma.

A entrada do pós-processador é a própria saída do processador: o arquivo CLDATA guardado em disco ou fita magnética. A saída é a fita perfurada no código e no formato requerido pelo controle. Assim, o pós-processador determina funções G e M, por exemplo, necessárias para controlar as revoluções da árvore, avanço da ferramenta, etc., no modo exigido pela combinação máquina-controle (Figura 28).

Os sistemas de programação CN baseados na linguagem APT empregam pós-processadores independentes dos processadores, o que é vantajoso por se ter uma mesma solução genérica, o CLDATA, que pode alimentar diferentes pós-processadores obtendo-se fitas diferentes para os diferentes tipos de máquina/controler, como mostra a figura 7.

Tendo o arquivo CLDATA como entrada, os pós-processadores têm como saída:

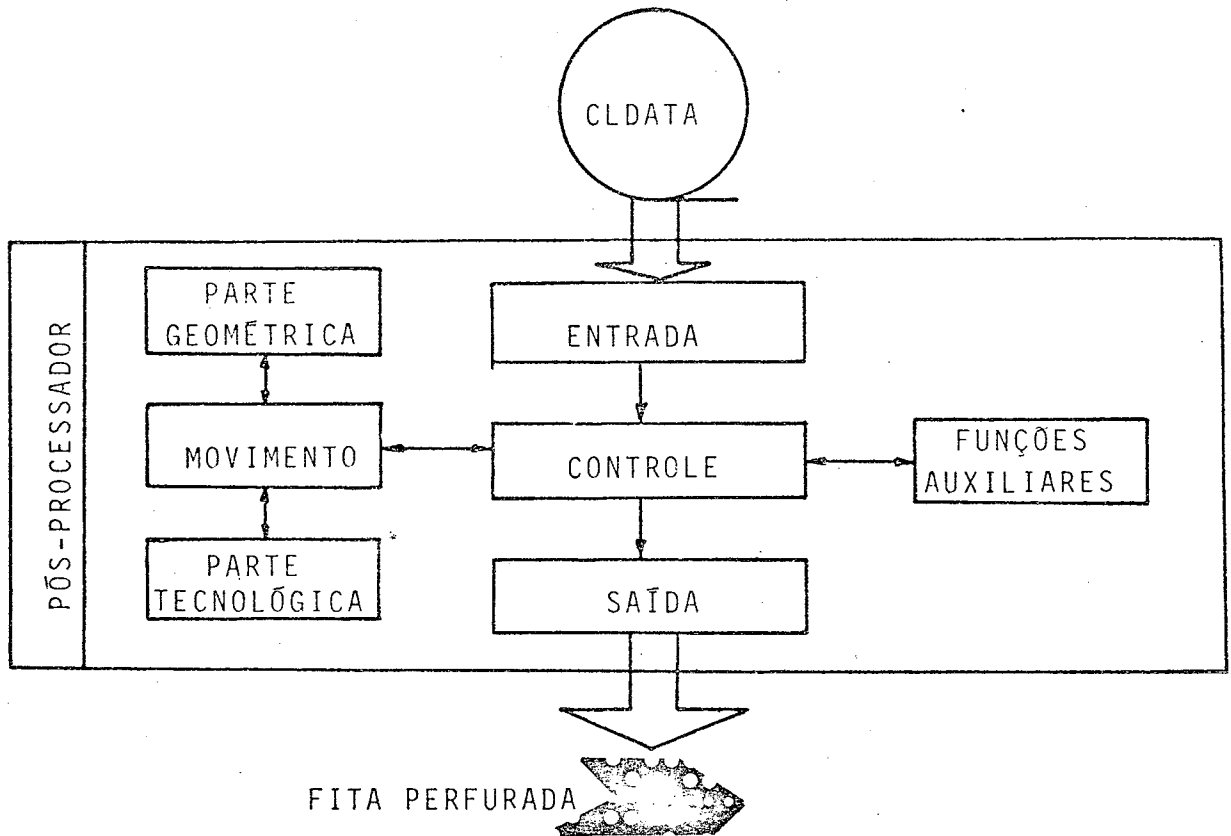


FIGURA 28 Diagrama das partes componentes dos pós-processadores.

- a fita perfurada (ou similar)
- listagem do programa em linguagem codificada.
- listagem para o operador da máquina
- listagem para a administração.

Insiste-se que embora se deva escrever um pós-processador para cada sistema máquina/controle, estes têm normalmente um décimo do comando dos processadores. Sem este recurso far-se-ia necessário um sistema completo, desde a leitura do programa em linguagem CN, cálculo da trajetória, da geometria, etc., à produção da fita perfurada, para cada novo conjunto máquina/controle.

Supondo uma peça já programada em EXAPT que venha sendo usualmente usinada no torno T-1 e que, por motivo qualquer, terá que ser usinada em outra máquina T-2. Bastaria ao usuário, com o mesmo programa EXAPT, produzir nova fita perfurada, utilizando como pós-processador o da máquina T-2.

Assim é possível às empresas padronizarem de certo modo seu maquinário CN, reduzindo seus custos. Uma empresa trabalha freqüentemente com diferentes tipos de máquinas como tor_{nos}, fresadoras, centros de usinagem e ainda com diferentes máqui_{nas} do mesmo tipo (por exemplo, diferenciados quanto à capacida_{de}, modelo, finalidade, etc.).

Mas os controles destas MFCN podem ser mais ou menos estandarizados o que facilita em muito o trabalho de p_{ós}-processamento.

A firma VWF tem, por exemplo, mais de 36 MFCN, mas trabalha apenas com 6 ou 7 p_{ós}-processadores, por ter padroni_zado seus controles em somente 2 ou 3 tipos.

Depende decisivamente da qualidade deste programa de p_{ós}-processamento se:

- O conjunto máquina/controlador é usado de modo otimizado.
- O operador e preparador da MFCN recebem todas as informações necessárias de forma clara.
- Os tempos de usinagem precisos são calculados.
- Os dados estatísticos confiáveis, como de vida da ferramenta, são obtidos.

Os p_{ós}-processadores para sistemas máquina/controlador são obtidos diretamente da Associação EXAPT, ou de firmas especializadas neste tipo de "software" ou ainda e principalmente dos próprios fabricantes da máquina. O GRUCON se encontra habilitado para gerar estes programas.

4.2.2. Variáveis Influentes nos P_{ós}-Processadores

Os p_{ós}-processadores sempre são construídos visando atender às exigências de 5 conjuntos básicos de variáveis (Figura 29).

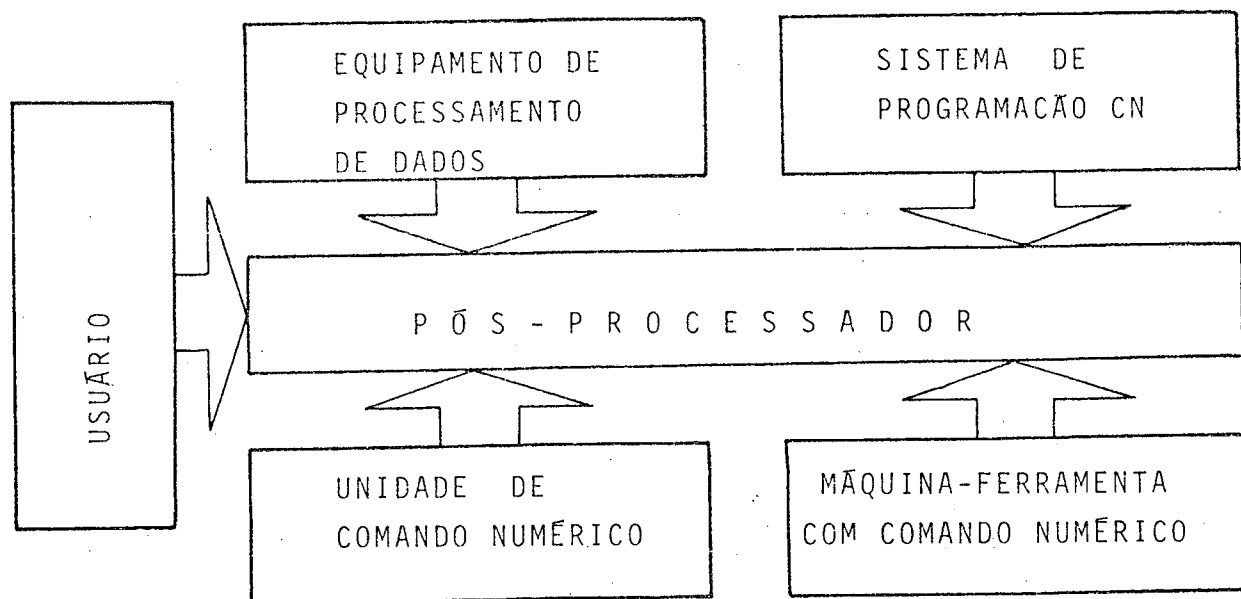


FIGURA 29 Pós-processador: variáveis de maior influência.

a. Sistema de Programação CN

O Sistema de Programação CN tem como elementos integrantes os pós-processadores e estes deverão atender às características dos demais componentes do sistema (basta lembrar que os pós-processadores recebem sua entrada do processador).

Os processadores da família APT geram CLDATA padronizado pela ISO. Entretanto CLDATA de diferentes processadores contém diferentes informações para um programa de uma mesma peça e então os pós-processadores que usam estas informações devem ser ajustados às características dos mesmos.

De outro lado, alguns processadores formam, com seus pós-processadores, blocos mais monolíticos que os da família APT e outros ainda não distinguem processadores e pós-processadores, sendo todo o conjunto escrito para cada nova combinação máquina-controle produzida.

b. Equipamento de Processamento de Dados

O equipamento de processamento de dados suporte influencia diretamente, na realidade, todo o sistema de programação, que deverá ser adaptado às suas peculiaridades.

O tamanho das memórias principal e auxiliares disponíveis, o sistema de tratamento de arquivos, a precisão dos dígitos calculados - alguns computadores (IBM por exemplo) indo a tê o 60 dígito, e outros até o 70 ou mais (DEC), - enfim todo o "software" e "hardware" a que o p^os-processador deverá atender, influenciam seu projeto.

c. Unidade de Comando Numérico

A unidade de comando numérico é quem recebe como entrada a fita perfurada que os p^os-processadores produzirão para comandar a MFCN e fabricar a peça.

Com a transformação dos controles de máquinas cujas funções são executadas por circuitos eletrônicos, em que os componentes são interligados por fios físicos, ("hardwired", para os sistemas atuais ("softwired"), onde as mesmas funções não são mais resultantes unicamente do projeto de "hardware" mais de l^ogica computacional, tornou-se possível fazer controles mais potentes, seguros e baratos.

O p^os-processador deve ser escrito para um sistema máquina-controle no qual cada elemento dispõe de habilidades ou restrições inatas, que deverão ser respeitadas e aproveitadas.

Por exemplo, certos controles não têm interpolação circular em 360° , mas por arcos de 90° . Entretanto se um p^os-processador recebe informações do CLDATA para gerar interpolação ao longo de 4 quadrantes da mesma circunferência, este pode ser construído de modo que aproveite diretamente a função interpolação de circunferência completa, se existente na unidade de controle.

③ O mesmo sucede para os ciclos de trabalho. O

pós-processador, lendo o CLDATA e detectando uma seqüência de ferramentas agindo em diferentes pontos, pode gerar, para cada ponto diretamente, o código que chama a subrotina do ciclo de trabalho respectivo, em controles com esta facilidade.

Em qualquer dos casos citados abaixo é tarefa do pós-processador gerenciar as possibilidades existentes, tais como:

- O formato padrão da fita perfurada de entrada, assim como o teste do significado das diversas letras-código, que, apesar de padronizadas por algumas normas - DIN, por exemplo - não são as mesmas de controle para controle.
- Os comandos de um bloco podem, por sua vez, ser executados da direita para a esquerda, ou vice-versa, ou de modo seletivo, como executando antes as funções auxiliares chamadas no bloco, dependendo do controle utilizado.
- As correções de ferramenta podem ser trabalhadas de diferentes modos. Alguns controles, lida a correção da ferramenta, não geram movimento da mesma e a correção só será levada a efeito quando do primeiro movimento após a correção, enquanto outros corrigem a posição imediatamente.

d. A Máquina CN

A MFCN, sendo quem, em última instância executará os comandos recebidos, influencia sobremaneira a construção do pós-processador, que deverá explorar ao máximo suas possibilidades e atender suas limitações.

Cada máquina-ferramenta tem suas próprias características. Pode ter, por exemplo, a árvore em posição vertical ou horizontal, trocador de ferramentas, mais de uma mesa, que, por

sua vez, pode ser grande ou pequena, motores de alta potência ou de potência modesta, cabeçote rotativo ou pivotante.

Hã portanto, um número alto de variáveis que deverão ser consideradas, sendo infinitamente grande, as combinações que as mesmas assumem nas diferentes máquinas.

Máquinas com elevado número de ferramentas são cada vez mais comuns. Um pós-processador eficiente, escrito para uma determinada máquina com dezenas de ferramentas fixas, talvez em um magazine ou em uma corrente, pode, tendo a coleção de ferramentas disponível no CLDATA, sugerir uma organização ótima das mesmas no magazine ou na corrente.

Na figura 30 tem-se esquematizado um magazine com cinco diferentes ferramentas de um centro de usinagem. Nota-se que, devido à grande dissimilaridade entre os comprimentos das ferramentas, se a ferramenta F1 fosse posicionada para trabalho logo após a broca helicoidal F2, ter-se-ia uma colisão daquela ferramenta com a mesa. Um pós-processador bem elaborado poderia tomar a seu cargo, quando se chama uma ferramenta curta posicionada entre duas longas adjacentes, verificar a possibilidade de colisões e emitir as mensagens pertinentes.

Numa outra situação, quando se pede ao processador a manutenção de um certo erro máximo (por exemplo, com CSRAT/99 não se deseja erro maior que 1%) na velocidade de corte, este gera uma série de informações SPINDL no CLDATA, para controlar tal velocidade no intervalo de erro desejado. Entretanto a máquina pode, na realidade, trabalhar com rotações escalonadas e o pós-processador, recebendo uma seqüência de comandos SPINDL, com parâmetros não obtíveis na máquina para que foi construído, deve reduzir toda a série para a única rotação mais próxima e mais baixa, efetivamente oferecida pelas engrenagens.

e. O Usuário do Sistema CN

O usuário, dependendo do seu problema específico, poderá solicitar do pós-processador desempenhos especiais e dispensar outros menos desejáveis.

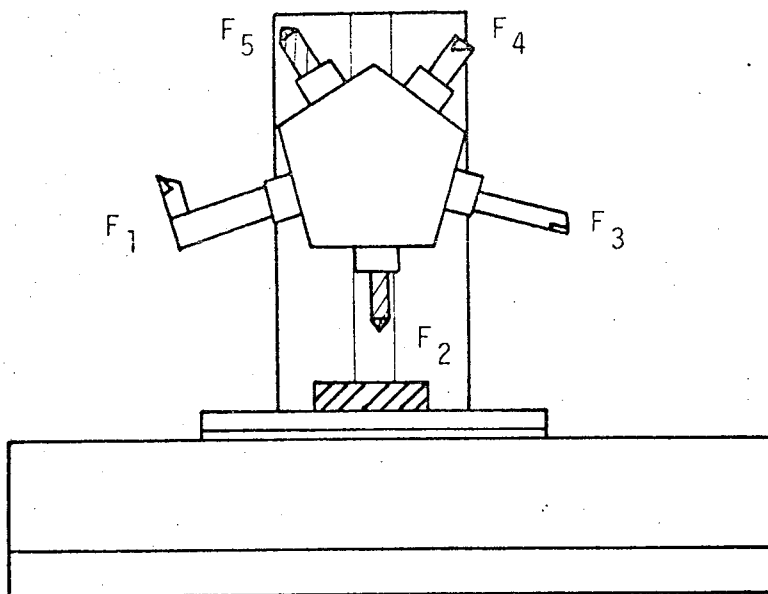


FIGURA 30 - Ferramentas de comprimentos muito diferentes ocasionam problemas que o pós-processador pode resolver.

Os pós-processadores são sempre direcionados para um conjunto máquina-controle bem determinado e real e, como tal, tem informações muito precisas sobre as funções solicitadas deste conjunto. Este programa tem a possibilidade de acumular os tempos de uso de cada ferramenta, já que ele dirige as usinagens realizadas pela mesma.

Assim, com o uso adequado destes tempos, se pode criar rotinas para:

- reafiar ferramentas após tantas horas de uso.
- trocar ferramentas após tal tempo de vida.
- etc.

Pela mesma razão, o pós-processador pode calcular os tempos principais, em que existe usinagem de fato - remoção de material - e os secundários, como: posicionamentos rápidos, trocas de ferramentas, etc.

Além da utilização convencional, estes tempos podem servir para calcular o intervalo decorrido entre duas tro-

cas de ferramentas, por exemplo. Assim, trocada a ferramenta em uma máquina, o operador, sabendo a que horas será a próxima troca, está talvez liberado para atender uma máquina vizinha.

Assim é problema dos construtores das máquinas produzir seus próprios pós-processadores de modo a torná-los atrativos aos usuários, e de modo a aproveitar ao máximo as potencialidades do sistema máquina-controle, superando suas eficiências. Tais diretivas devem ser consubstanciadas em manual de uso do pós-processador, indicando processos de otimizar a construção de programas, depurando-os de comandos executados automaticamente pelo pós-processador (por exemplo, um comando FINI pode indicar implicitamente em um determinado pós-processador, além do fim do programa, a retirada da ferramenta da árvore, fechamento do refrigerante, retirar estrado - mesa sobre a qual a peça é fixada - da área de trabalho, etc.).

4.2.3. Entrada/Saída

a. Entrada (CLDATA)

O que se visa primordialmente em um sistema CN por computador é a confecção do portador de dados, geralmente, em mais de 80% dos casos, uma fita perfurada.

Dispõe-se para isto das informações codificadas chamadas CLDATA, resultado final da cadeia de processamento que inicia com a codificação da peça em uma linguagem, passa pela fase de processamento e se transforma em informações em forma de variáveis reais, inteiras e alfanuméricas.

Estrutura do texto-CLDATA

O texto-CLDATA é composto de palavras-CLDATA e registros-CLDATA, sendo (Figura 31):

Palavra-CLDATA: composta de 2 vetores inteiros e 1 vetor real.

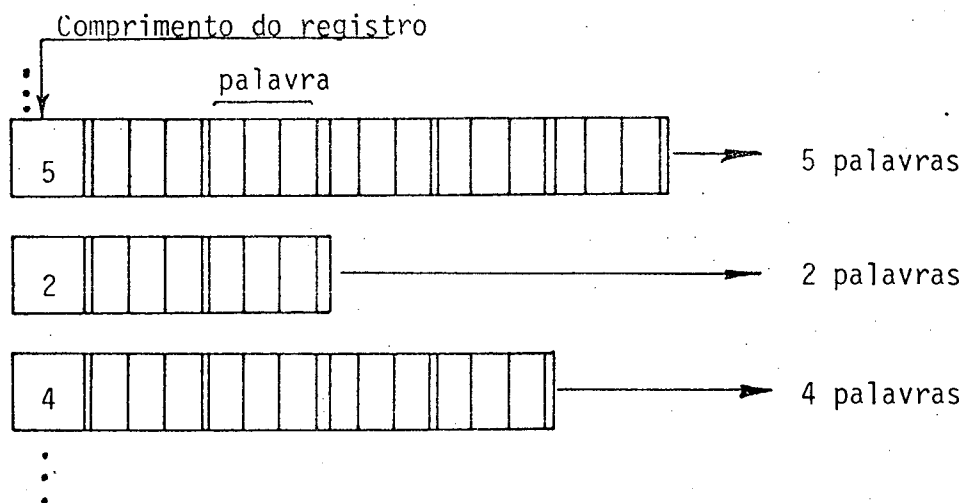


FIGURA 31 - Representação de segmento de CLDATA com 3 registros de 5, 2 e 4 palavras.

Registro-CLDATA (ou Bloco de dados-CLDATA): um conjunto de palavras-CLDATA precedidas pela especificação do número de palavras do mesmo (comprimeto do registro). Podem ser escritas no máximo 100 palavras por registro (construção atual).

Texto-CLDATA: Conjunto de registros-CLDATA.

Os textos-CLDATA são escritos pelo processador, registro por registro, em notação binária, em memórias periféricas, por exemplo discos ou fitas magnéticas. Entretanto, os textos-CLDATA podem ser também obtidos em fitas ou cartões perfurados.

Estrutura dos registros-CLDATA

A construção dos registros-CLDATA obedecem o padrão:

Número	Identif.	Inteiro	Real	Identif.	Inteiro	Real	...
1ª palavra				2ª palavra			

onde:

Número: número de palavras do registro (máximo: 100)

Identif.: identificação, caracteriza o tipo da palavra. Identif é a variável sempre inteira.

Existem as seguintes identificações e tipos correspondentes:

<u>Identificação</u>	<u>Tipo</u>
0	Sem valor escrito
1	BCD em vetor real
2	Inteiro
3	Real

Inteiro, real: são vetores de tipo FORTRAN inteiro ou real, que armazenam as informações de tipo correspondente do registro.

A primeira e a segunda palavras de um registro têm significados fixos que são:

1ª palavra (identif=2): número seqüencial do registro.

2ª palavra (identif=2): tipo de registro.

3ª a n-ésima palavra: dados e identificações que dependem do tipo de registro.

No caso de comandos do pós-processador (AUXFUN, CUTCOM, OFSTNO e PPFUN) os símbolos são armazenados como BCD, os números como reais, os valores das variáveis como reais, e o código, inteiro no caso de palavras (tipo do registro código da palavra).

Existem no CLDATA seis tipos de registros:

TIPO DE REGISTRO	OBJETIVO
1	Conter nº de seqüência da declaração no programador da peça.
2	Conter declarações de pós-processador.
3	Conter dados da superfície (circular) de deslocamento.
5	Conter dados de declarações de partida e movimento.
14	Conter registro de finalização do texto -CLDATA.
30	Conter tabelas dos contornos das peças bruta e acabada.

Um registro CLDATA tem o seguinte aspecto:

16	2	53		2	3		2	2		...	1		PTO
----	---	----	--	---	---	--	---	---	--	-----	---	--	-----

Por exemplo, o comando EXAPT

GOTO/(P3(1))=POINT/ 30,40,50

Fica em formato de registro de CLDATA (Figura 32):

	1	2	3	4	5	6	7	8								
8	2	39	2	5	2	5	1	P3	2	1	3	10.	3	40.	3	50.

onde:

Número de palavras: 8

Ordem da Palavra	Conteúdo da Palavra		Tipo de Palavra	Observação
	INTEIRA	REAL		
1	39		Inteira	Nº sequencial do registro
2	5		Inteira	Tipo de registro (ISO)
3	5		Inteira	Código(subtipo) de GOTO (ISO)
4		P3	BCD(texto)	Símbolo do ponto
5	1		Inteira	Índice do ponto
6		30	Real	Coordenada x
7		40	Real	Coordenada y
8		50	Real	Coordenada z

FIGURA 32 - Exemplo de transformação de comando EXAPT em registro CLDATA

Para cada um dos seis tipos de blocos de dados ou registros-CLDATA existem diversos subtipos (códigos): Por exemplo, para o bloco de dados tipo 5, reservado para declarações de partida e movimento, existem os subtipos 3 e 5, que definem inequivocamente os comandos FROM e GOTO, respectivamente.

A disposição e significado dos dados componentes do texto-CLDATA são padronizados pelas normas:

- ISO 3592 Numerical Control of Machines
NC Processor Output
Logical Structure (and major words)
de 15 de dezembro de 1978.
- ISO 4343 Numerical Control of Machines,
NC Processor Output
Minor Elements of 2000-type records
(Post-processor Commands)

de 01 de abril de 1978.

Assim, teoricamente, um p̄s-processador escrito para uma determinada combinaçãõ máquina-controle em um sistema tipo APT, pode ser alimentado por textos-CLDATA em qualquer dos processadores da família.

b. Saída

As saídas do p̄s-processador podem ser divididas em dois tipos:

- Saídas para controle: armazenadas em fita perfurada ou meio equivalente.
- Saídas para a administração:
 - Listagem do programa da peça, com observações para o operador.
 - Listagem das ferramentas chamadas com respectivas características, tais como: dimensões, tempos de uso, etc.
 - Dados para o Plano de Preparação, como fixações, peso, etc.
 - Informações para a administração, tais como tempos principais, secundários, etc.

A quantidade e a qualidade das informações de saída do p̄s-processador, são definidas por ocasião da sua construção. A figura 49 do Anexo 5 mostra as saídas obtidas com o p̄s-processador do torno revólver CSEPEL DECATHLON ERI 250, com controle BOSCH NC 802 da UFSC, além do programa da peça propriamente dito. A saída mais importante do p̄s-processador é, sem dúvida, a

	C L D A T A	Fita perfurada
Fonte	Processador	Pós-processador
Tipo de código	Binário	Alfa-numérico
Estrutura	Blocada seqüencialmente	Blocada seqüencialmente
Conteúdo dos blocos	Uma unidade (parte) de informação	Uma operação de máquina
Tipo de informação	Números-código padronizados (Tipo & subtipo de registro)	Letras de endereços padronizadas
Tipo de dados	Números reais Palavras chave "HOLLERITH"	Números inteiros

TABELA 3 - Paralelo entre as informações do arquivo CLDATA e as da fita perfurada.

fita perfurada que é a particularização das informações contidas no CLDATA para um determinado complexo máquina-controle.

As principais semelhanças e dissimilaridades entre a fita perfurada e o CLDATA são dados na Tabela 3.

A seqüência de informações do CLDATA especifica a seqüência dos comandos da máquina, caracterizados por números e letras-endereço que indicam os endereços do controle (as memórias G, M, T, X, Y, etc.) e quais os seus conteúdos (ver figura 33).

Um conteúdo de CLDATA resultante de vários comandos EXAPT pode ser pós-processado e condensado em uma só operação de máquina, como:

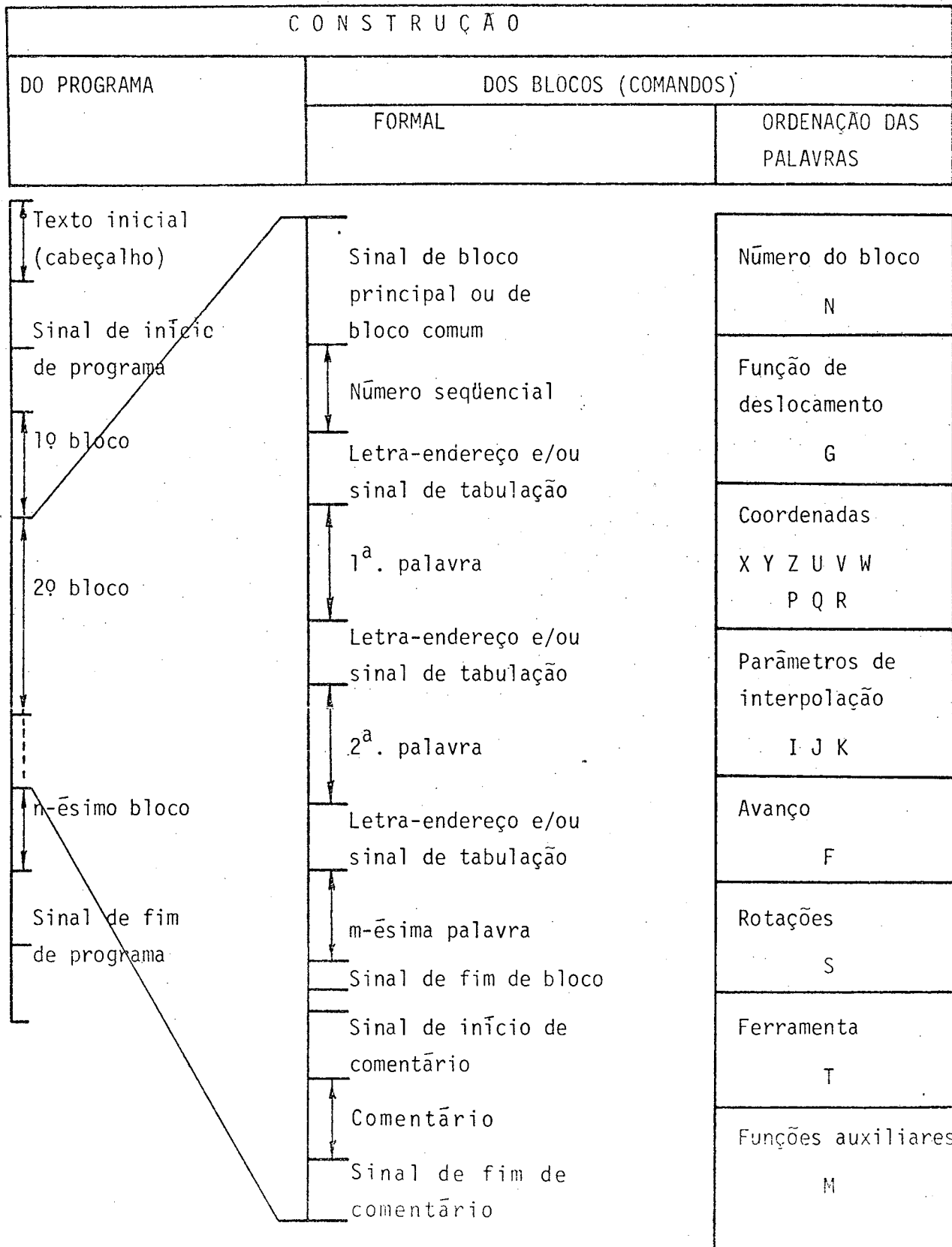


FIGURA 33 - Princípio de construção da saída de um pós-processador.

EXAPT

CLDATA

Fita Perfurada

⋮

⋮

FEDRAT/f

SPINDL/s

GOTO/g

⋮



Nn Ss Xx Yy Ff' Ss'

Onde as letras minúsculas indicam os valores das funções representadas pelas maiúsculas.

Os números do CLDATA podem ser reais ou inteiros, mas uma máquina-ferramenta, dependendo de sua precisão, trabalha com décimos de milímetro, centésimos de milímetro, etc.. Logo as informações da fita perfurada deverão ser para estas precisões transformadas. Uma informação

$$X = 150.23478 \text{ mm}$$

seria pós-processada em

$$X15023 \text{ ou } X150234$$

conforme a precisão da máquina fosse em centésimos de milímetro ou em milésimos.

A colocação das informações oriundas do pós-processador na fita perfurada, que geralmente é de 8 pistas (canais), é regida por normas como:

- ASCII RS-358
- DIN 66024
- EIA (RS 244-A)
- ISO/R840

Para elaboração dos formatos de saída das funções comandadas, são consideradas sete características a saber:

- 1) Reservada
- 2) 1: Saída sem ponto decimal
2: Saída com ponto decimal
3: Saída em formato interno

- 3) 1: Colocar sinal positivo (+) ou negativo (-) sempre
 2: Colocar s̄o sinal negativo, se for o ca
 so
 3: Saída sem sinal
- 4) 1: Preencher com zeros antes do número
 2: Não preencher com zeros antes do número
- 5) 1: Preencher com zeros ap̄os o número
 2: Não preencher com zeros ap̄os o número
- 6) Quantidade de d̄ıgıtos antes da v̄ırgula
- 7) Quantidade de d̄ıgıtos ap̄os a v̄ırgula

Exemplificando, para o p̄os-processador NBH-65 da UFSC, as funç̄oes T, F, S, X, I, D teriam os formatos da Tabela 4.

Funç̄ao	Característica						
	1	2	3	4	5	6	7
T	0	3	3	2	2	2	0
F	0	3	3	1	2	4	3
S	0	3	3	1	2	4	0
X	0	3	2	1	2	4	3
I	0	3	2	1	2	4	3
D	0	3	3	2	2	2	0

Tabela 4 - Formatos de saída de algumas funç̄oes para o p̄os-processador NBH-65 da UFSC.

As funç̄oes G e M seriam definidas em 9 e 4 formatos diferentes, conforme o mesmo número de definiç̄oes que as mesmas assumem.

4.2.4. Construção do Pós-Processador

O Arquivo de dados CLDATA preparado pelo processador é de acesso seqüencial, isto é, qualquer informação nele contida só é alcançada se antes forem lidas todas as anteriores.

Os pós-processadores atuais são construídos em duas fases distintas (Figura 34). A primeira fase tem por finalidade principal, a leitura do arquivo seqüencial original e preparação de um arquivo de acesso direto. A segunda fase, com acesso direto ao arquivo já rearrumado e sem necessitar rebobinamento da fita magnética, por exemplo, ou procura de informações adiante ou atrás da posição atual da leitura do arquivo, constrói as informações para a fita perfurada e lhes dá saída.

Em resumo, a filosofia deste tipo de construção de pós-processadores é localizar e preparar na fase 1 as informações que podem ser necessárias na fase 2, que é a fase de processamento propriamente dita e de saída.

A geração de pós-processadores pode optar por duas versões de construção:

- Na primeira versão, indicada com "(1ª)" na Figura 34, a fase de saída é chamada a cada novo registro completado, isto é, a fase 2 controla a saída através de interligação direta.
- Na segunda versão os resultados da fase 2 são colocados em um arquivo. Uma fase 3 lê este arquivo após a fase 2, registro por registro, e gera a saída final, sendo controlada pelo programa de controle e não mais diretamente, pela fase 2. Este 3º modo, por subdividir mais o pós-processador é mais indicado para mini-computadores onde haja problema de quantidade de memória.

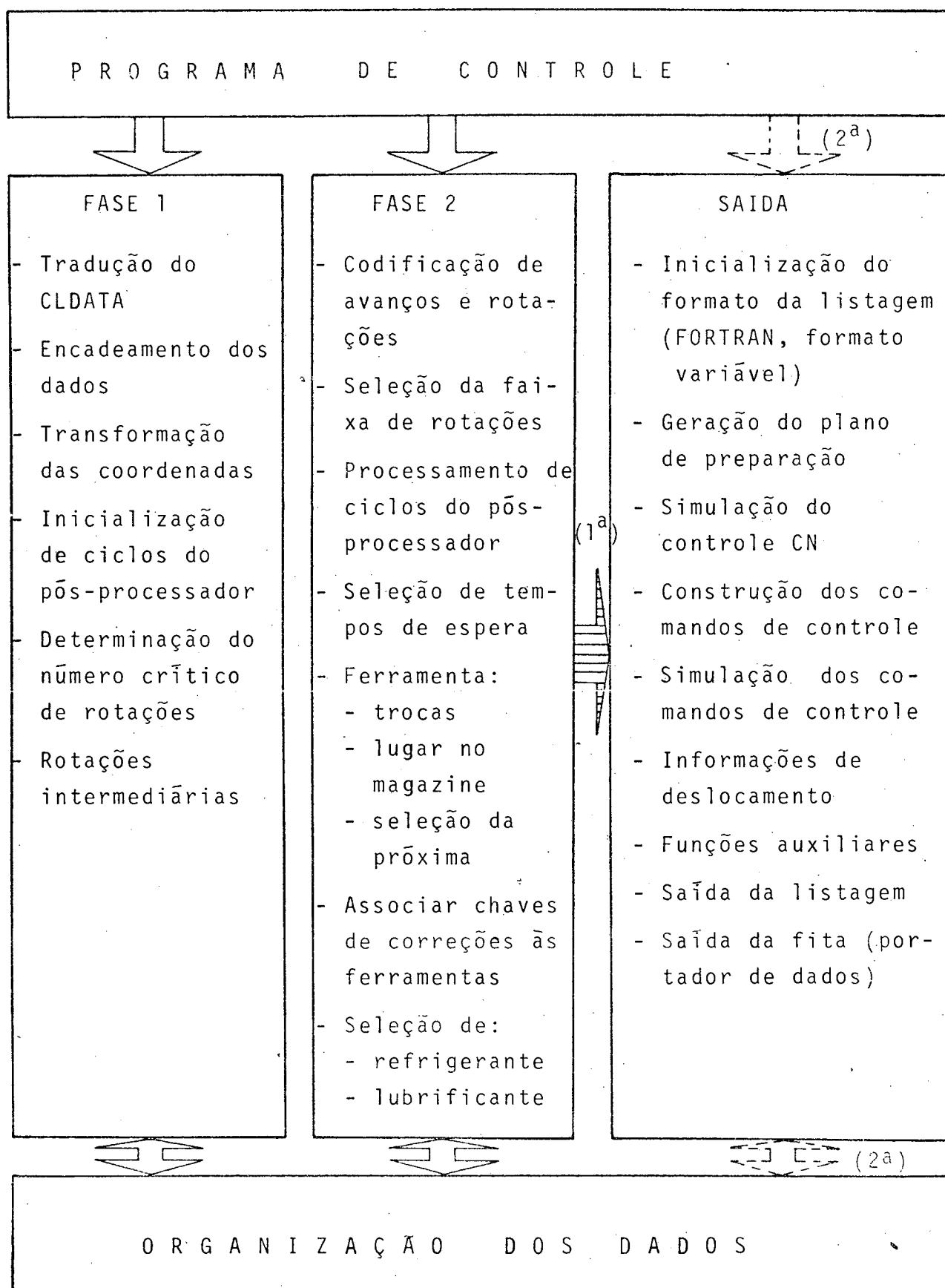


FIGURA 34. Funções desenvolvidas pelo pós-processador nas suas diversas fases de processamento.

A seguir se tem mais detalhadamente as duas fases de processamento.

Fase 1

A figura 35 esquematiza o processamento dos registros-CLDATA, que ocorre na 1ª fase, começando esta com o acesso ao arquivo CLDATA feito com a subrotina DAIO. As ações executadas são controladas por um vetor (AKTION) ou por uma chave (RKENG) conforme se trate de pós-processador de torneamento ou de furação e fresagem. Uma vez selecionada a ação codificada a executar, passa-se às transformações e interpretação necessárias para cada ação.

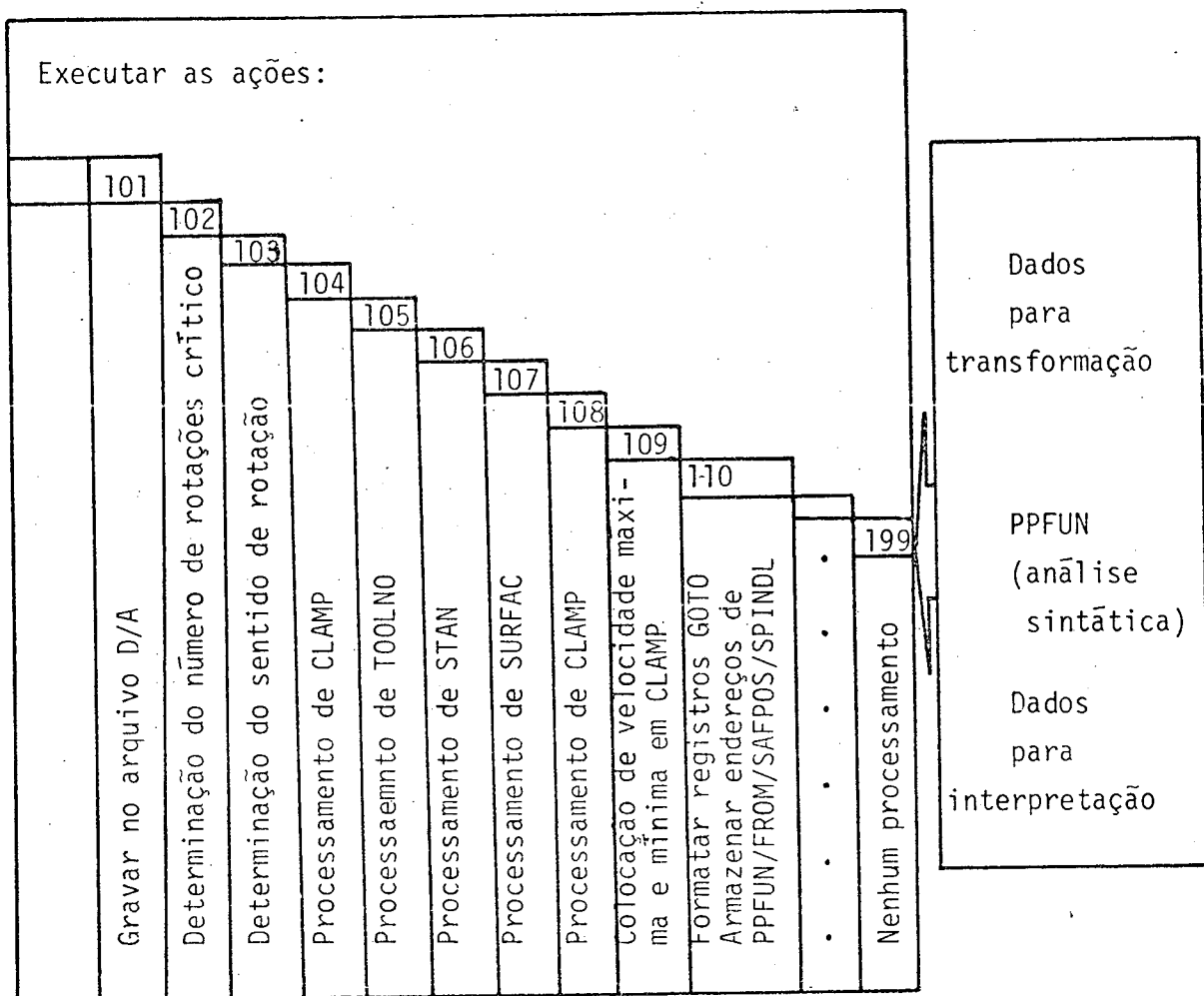
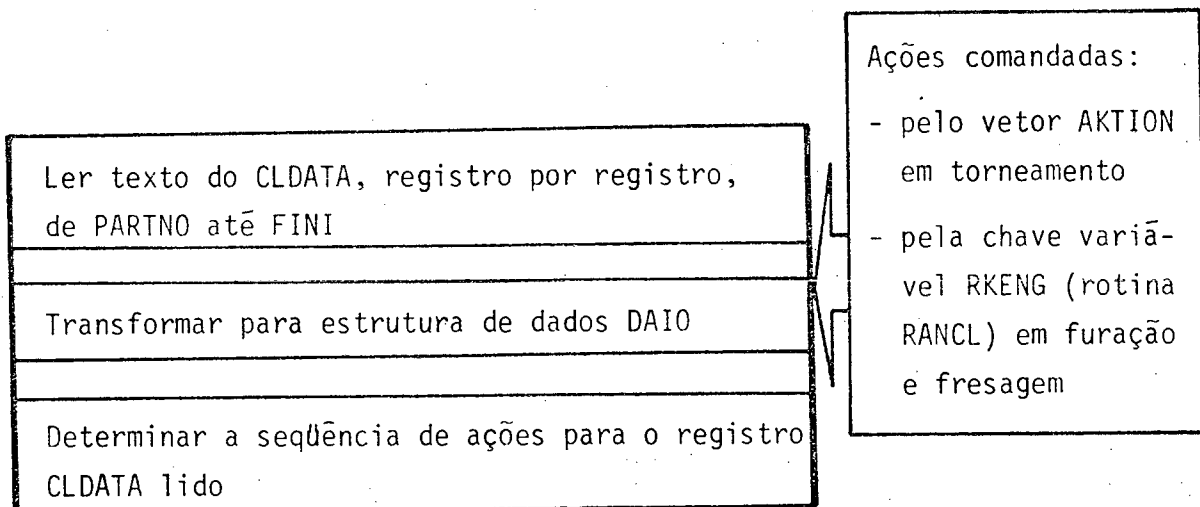
O arquivo de acesso direto produzido na fase 1, tem a parte inicial composta pelos textos de erros possíveis no pós-processador e dos seus macros (alguns pós-processadores têm rotinas ou ciclos especiais, como trocas complexas de ferramentas, em tornos especiais) e a parte de trabalho.

A parte de trabalho realiza a preparação propriamente dita do arquivo, que consiste no encadeamento da lista de registros e na execução de algumas funções f_i (Figura 36).

O encadeamento é a informação em determinados registros, dos endereços de registros anteriores ou posteriores a ele, com os quais este registro permutaria informações.

Exemplificando, com a construção de MFCN com magazines ou correntes com dezenas de ferramentas, em uma declaração (registro) de chamada de ferramenta, deve constar o endereço do registro da chamada da ferramenta seguinte. De fato, em CN existe perda de tempo se, terminado o trabalho com uma ferramenta, a máquina ficar esperando que o magazine se posicione corretamente na próxima ferramenta, para então executar a troca.

Conhecendo a próxima ferramenta, comanda-se imediatamente após o posicionamento da 1ª na árvore, o giro do magazine até aquela, de modo que esteja diretamente disponível quando de sua solicitação (em máquinas com esta possibilidade).



Gravar em arquivo de acesso direto (D/A), com DAIO, ações fora do intervalo 101 a 199.

FIGURA 35 - Trabalho de processamento na 1ª fase.

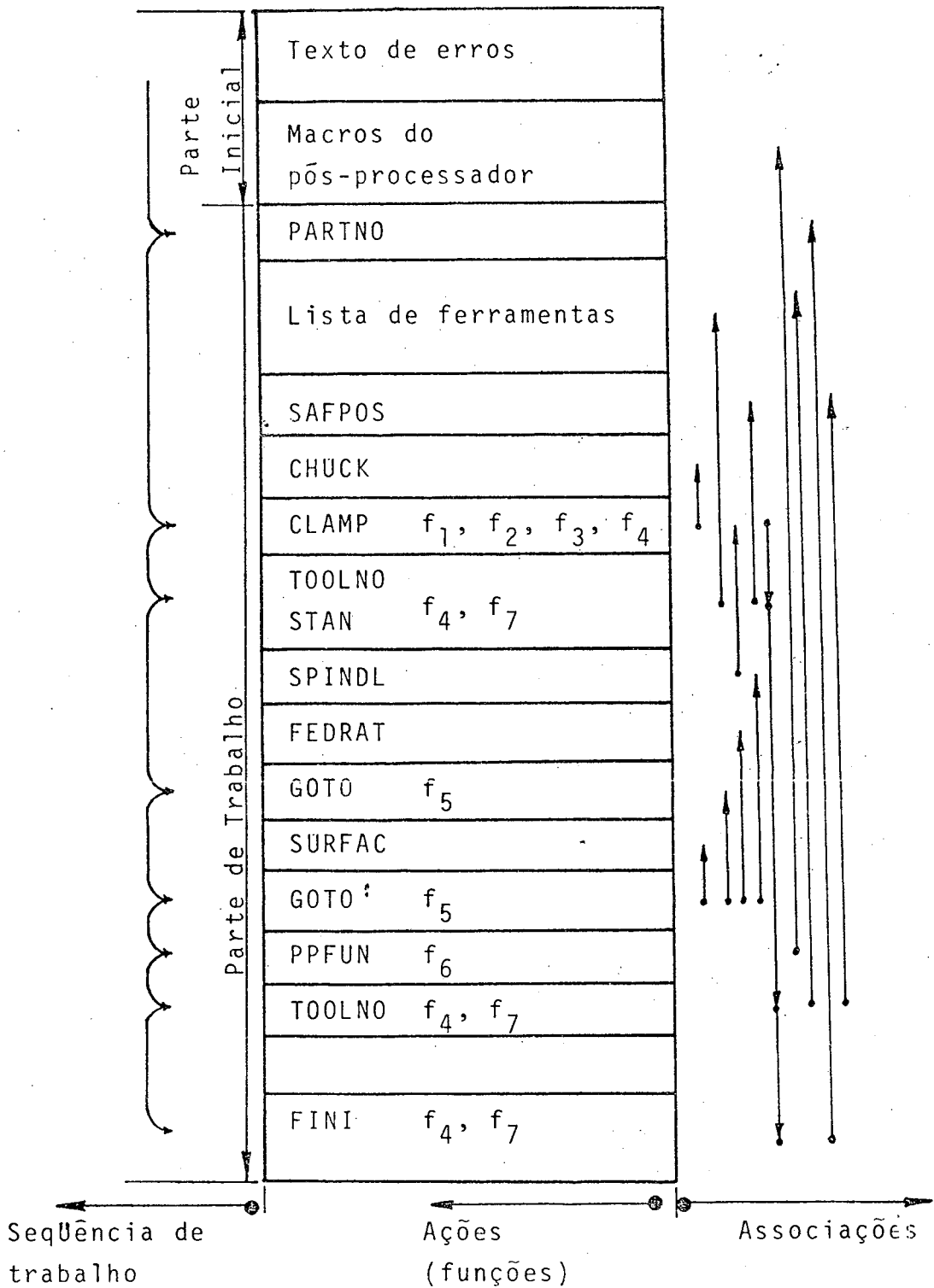


FIGURA 36 Organização em lista do armazenamento virtual intermediário (subrotina DAIO).

O encadeamento na lista é feito tanto do início para o fim, quanto em sentido inverso.

As funções f_i são trabalhos como:

- f_4 : movimentar a ferramenta para posição de troca.
- f_7 : armazenar tempo de usinagem da ferramenta.

Note-se que nem todos os registros da lista geram diretamente comandos de saída (na fita), caso de SAFPOS ou SURFAC.

Fase 2

O processamento não executado na fase 1, o é na fase 2, que dispõe, para seu trabalho, de um arquivo já bem conhecido da fase inicial e com sinalizações lá definidas.

Um exemplo de processamento que ocorre na fase 2 é a seleção de domínios de rotações da árvore (Figura 37).

A fase 1 faz cálculos de logaritmação com as rotações possíveis na máquina e as solicitadas no programa, e a partir daí a fase 2 pode selecionar rotações ótimas.

Em algumas máquinas não se pode trocar bruscamente de rotações ou de ferramentas, mas exigem um tempo de reajustamento (DELAY) que é fornecido na fase 2.

4.2.5. Pós-Processador Geral (GNPP)

A construção de pós-processadores não necessita ser totalmente refeita a cada novo conjunto máquina-controle. Na realidade existe um Pós-Processador Geral que tem todas as rotinas necessárias e a partir deste sistema generalizado se monta o programa particular desejado, trabalho que comumente não muda mais que 30% do sistema geral.

A figura 38 mostra as subrotinas componentes de um sistema geral (GENPP - General Post-processor) utilizado pela Associação EXAPT. Esta Associação está estudando também um sis

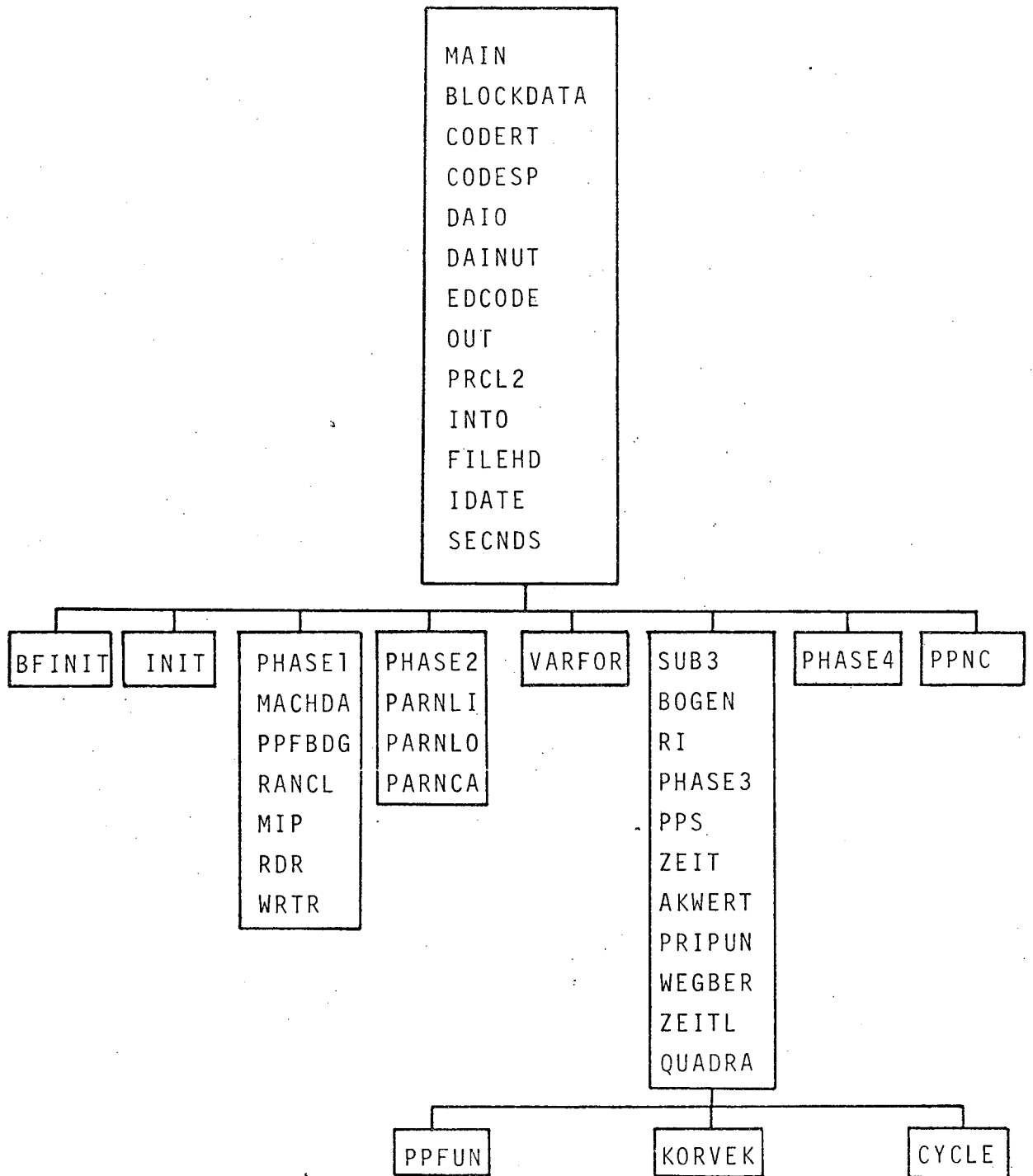


FIGURA 38 - Estrutura do programa GENPP para furação e fresagem.

tema automático de geração de pós-processadores.

A seguir descreve-se, de modo resumido, o trabalho das subrotinas componentes do sistema GNPP. Como existe um GENPP para torneamento e outro para furação e fresagem, são as rotinas comuns a ambos são citadas.

- BLOCKDATA - é um subprograma onde são inicializadas as variáveis em COMMON, os vetores que conterão os valores característicos de uma máquina e outros utilizados pelo programa. Contém também tabelas que controlarão o fluxo do programa.
- AKWERT - é uma função que calcula o valor atual dos parâmetros (X, Y, Z, F, M, etc.) do vetor que está sendo montado na linguagem da máquina.
- BFINIT - inicializa nos COMMON's os dados comuns entre pós-processadores.
- BOGEN - calcula o comprimento de um arco de círculo para o cálculo do tempo de usinagem.
- CODESP - determina a rotação da árvore e a codificação da mesma.
- CYCLE - emite dados necessários para elaborar e dar como saída controles específicos para a máquina.
- DAINUT - acessa um arquivo direto, lendo ou escrevendo algo no mesmo.
- DAIO - é utilizada para trabalhar com o arquivo de acesso direto.
- EDCODE - adapta a estrutura da palavra definida neste pós-processador para outra, caso seja utilizado outro computador com a estrutura da palavra diferente da utilizada neste.
- FILEHD - manipulação seqüencial dos arquivos contidos no disco.
- IDATE - emite a data atual do pós-processamento através de uma rotina específica.

- OUT - separa uma variável (valor) de um vetor, colocando em uma palavra.
- PARNCA - colocará do comando PARTNO o texto no vetor, na porção de saída, em linguagem de máquina no formato de cartão.
- PARNLI - executa a saída do texto do comando PARTNO.
- PARNLO - executa a saída do texto do comando PARTNO na fita perfurada num formato legível (pictórico), ou um formato padrão (ISO, EIA, etc.), de tal forma que seja desprezado pela máquina quando processado, e quando lista do fique num formato legível.
- PHASE1 - lê os registros do CLDATA (através da subrotina RANCL) colocando num arquivo de acesso direto (trabalho). Os registros do CLDATA que necessitarem de ações do pós-processador, serão nesta subrotina tratados.
- PHASE2 - subrotina de controle das subrotinas PARNCA, PARNLI, PARNLO.
- PHASE3 - esta subrotina busca parte de saída, em linguagem de máquina, colocando num vetor, controlando convenientemente sua saída.
- PHASE4 - imprime o "plano de ferramenta" (ferramentas utilizadas, suas posições no magazine, etc.).
- PPFUN - esta rotina executa e coloca no vetor de saída, na linguagem da máquina, o que foi definido pelo comando PPFUN.
- PPFBDG - verifica a correta sintaxe do comando PPFUN e prepara o programa para a próxima fase a ser executada.
- PPNC - monta a saída do comando em linguagem de máquina, no código ISO ou EIA e imprime na fita de papel perfurada (podendo ser também em qualquer outra unidade de saída).
- PRCL2 - imprime um registro do arquivo de acesso direto através de uma opção de listagem.

- PRIPUN - prepara o programa para a listagem em linguagem de máquina, e o vetor de saída para a perfuração da fita de papel.
- RANCL - lê os registros do CLDATA separando as variáveis reais e inteiras e colocando-as em vetores específicos. Calcula também o "LABEL" de um "GOTO" controlado através do tipo do registro.
- SECNDS - emite a hora do processador central através de uma rotina específica.
- SUB3 - esta rotina controla o programa gerenciado a chamada da PHASE2.
- VARFOR - monta os formatos das variáveis para a saída na listagem.
- WEGBER - calcula o caminho percorrido entre um ponto e outro.
- ZEIT - calcula o tempo principal de usinagem (corte efetivo), e o tempo secundário (tempo em que a MFCN não está cortando - movimentos de posicionamento e troca de ferramentas), deixando à disposição do programa para se impresso na listagem do pós-processador.

4.3. Programas Auxiliares

Além da Linguagem de Programação, Processador e Pós-Processadores, um sistema de programação CN deve conter programas auxiliares que facilitem ao usuário a manipulação do mesmo.

Na seqüência discorre-se sobre alguns dos programas auxiliares disponíveis em EXAPT, ou gerados no Grupo de Pesquisa e Desenvolvimento em CN da UFSC, GRUCON.

4.3.1. Programas de Simulação Gráfica

PROGRAMA PLOT - O programa Plot foi construído no GRUCON para ser processado no computador HP-1000 da UFSC, a coplado ao "Plotter" digital 7210-A, ambos de fabricação da Hewlett

-Packard.

Seu objetivo é tornar possível a obtenção do perfil da peça a ser usinada, partindo do seu programa escrito em códigos de máquina.

Pelas suas características, o sistema de fabricação por CN se envolve continuamente com programas de peças novas, que devem ser testadas antes de entrar em produção. Este teste prévio se destina à depuração, visto que, deixado para fazer na máquina o trabalho de correções implica nas seguintes desvantagens:

- Ocupa o tempo da máquina, que poderia estar produzindo.
- É inseguro, pois podem existir erros de consequências danosas no programa que é testado.
- Por exigir muita preocupação com segurança, o teste é geralmente lento.

Um programa que teste em simulador gráfico procura eliminar estas desvantagens oferecendo ao programador uma vista em planta dos trajetos percorridos pelas ferramentas.

A principal limitação de um "plotter" é que, o oferecendo apenas uma visão bidimensional do problema, simplifica-o consideravelmente e perde informações importantes.

Em peças torneadas, entretanto, onde sô o perfil usinado é relevante, o "plotter" pode descrever mais precisamente os movimentos da ferramenta.

A construção e o funcionamento deste programa são vistos no Anexo 5, que mostra também um exemplo de programação para centro de usinagem e outro para torno, ilustrando o funcionamento da implementação.

O sistema EXAPT contém um módulo auxiliar chamado ZEIEX que, partindo dos resultados do processador - CLDATA - traça:

- Os percursos da ferramenta, programados ou automaticamente gerados.

- Secções transversais de peças rotativas brutas e acabadas.
- Os dispositivos de fixação.
- Detalhes ampliados de regiões carregadas de linhas do desenho, para auxiliar a análise de processos de usinagem complexos.
- Em casos de peças de torneamento, podem ser traçadas as duas partes simétricas ao eixo.

4.3.2. Manipulação de Fitas Perfuradas

Paralelamente à preparação de dados de controle para a MFCN, a administração e manipulação das fitas perfuradas já prontas é importante fator de consumo de tempo na fabricação.

Os problemas mais freqüentemente encontrados são:

- duplicação de fitas para assegurar que uma fita em bom estado esteja disponível em qual época.
- Listagem de fitas CN tanto em forma tabular quanto codificada, para verificação do seu conteúdo.
- Armazenamento de fitas perfuradas em discos magnéticos.
- Modificação do conteúdo de fitas CN, por exemplo para atender a modificações nas condições de fabricação, sem ter que repetir todo o cálculo computacional na preparação da nova fita.
- Mudança de Sistema de codificação (exemplo: ISO, EIA, ASCII, etc.).

Para resolver estes problemas foi desenvolvido o programa JANUS no sistema EXAPT.

A figura 16 mostra o funcionamento do sistema JANUS.

PROGRAMA MERGE (do GRUCON) - Objetivo: Incluir em um programa, em linguagem EXAPT, um macro ou qualquer arquivo já existente, contendo informações necessárias ao programa.

A inclusão destes arquivos é possível através de sua solicitação pelo comando READ, inserido nos pontos do programa em que se deseja intercalar arquivos previamente criados.

Terminado o programa EXAPT executa-se o programa MERGE que procurará comandos READ e os substituirá pelos arquivos neles chamados, montando o arquivo completo resultante.

Vantagens do uso do programa MERGE:

- Comodidade na programação de peças, já que não se necessita recopiar arquivos já existentes.
- Economia de espaço em disco. Em mini-computadores este fato é muito importante e, o programa MERGE, possibilitando ao usuário a guarda de uma única cópia de cada arquivo, mesmo dos de uso geral, faz real economia.
- Quando se inclui o comando READ, se tem certeza de se estar usando a versão mais atualizada do macro em questão.

A figura 39 mostra o arquivo "ALFABETA" criado com os arquivos "ALFA" e "BETA", já existentes.

Arquivo ALFA

```

0001 O ARQUIVO 'ALFA' CONTEM O ALFABETO:
0002 *****
0003 A B C D E F G H I J K L M N O P Q R S T U V X Y W Z
0004 *****
0005
0006 FIM DO ARQUIVO 'ALFA'.

```

Arquivo BETA

```

0001 O ARQUIVO 'BETA' CONTEM OS DIGITOS 0-9:
0002 *****
0003 0 1 2 3 4 5 6 7 8 9
0004 *****
0005
0006 FIM DO ARQUIVO 'BETA'.

```

Arquivo ALFABETA

```

0001 O ARQUIVO 'ALFABETA' EH FORMADO COM OS ARQUIVOS
0002 'ALFA' E 'BETA', ATRAVES DO USO DOS
0003 COMANDOS "READ/", PROCESSADOS PELO PROGRAMA
0004 MERGE.
0005
0006 'ALFA':
0007 READ / ALFA
0008
0009 'BETA':
0010 READ / BETA
0011
0012 FIM DO ARQUIVO 'ALFABETA'

```

```

$INPUT T=00004 IS ON CR00010 USING 00005 BLKS R=0000

```

```

0001 O ARQUIVO 'ALFABETA' EH FORMADO COM OS ARQUIVOS
0002 'ALFA' E 'BETA', ATRAVES DO USO DOS
0003 COMANDOS "READ/", PROCESSADOS PELO PROGRAMA
0004 MERGE.
0005
0006 'ALFA':
0007 O ARQUIVO 'ALFA' CONTEM O ALFABETO:
0008 *****
0009 A B C D E F G H I J K L M N O P Q R S T U V X Y W Z
0010 *****
0011
0012 FIM DO ARQUIVO 'ALFA'.
0013
0014 'BETA':
0015 O ARQUIVO 'BETA' CONTEM OS DIGITOS 0-9:
0016 *****
0017 0 1 2 3 4 5 6 7 8 9
0018 *****
0019
0020 FIM DO ARQUIVO 'BETA'.
0021
0022 FIM DO ARQUIVO 'ALFABETA'

```

FIGURA 39 Construção de um arquivo, usando dois outros já existentes, através do programa auxiliar MERGE.

C A P Í T U L O V

5. RUMOS DO CN DA DIREÇÃO DO CAD/CAM

Considerando os sistemas atuais de produção de peças únicas ou de séries pequenas, nota-se que as "máquinas CN são muitas vezes uma ilha de automação isolada no campo da produção"¹.

Um grande problema a ser resolvido com auxílio do computador é o estabelecimento de uma conexão direta entre o local de produção e os de planejamento e controle.

De fato, o maior uso do computador tem sido para solução de problemas comerciais e administrativos ou para cálculos técnicos.

Entretanto, todo o processo produtivo - do projeto à peça acabada - pode ser visto como uma série de passos que visam manipular e transformar a informação necessária.

O computador é um ferramenta muito útil para este tipo de trabalho e com ele se pode simplificar e acelerar em muito a produção convencional.

A figura 40 mostra os vários passos nos quais a informação é processada. A cada passo corresponde um certo grau de automação que se tem conseguido no tratamento da informação, e este grau cresce à medida que se avança nos passos do processo, desde o projeto, colimando na fabricação. Neste último passo se alcança o maior grau de automação através do emprego de máquinas CN.

Entretanto, o nível mais alto alcançável seria a integração total dos diversos estágios individuais por que passa o fluxo da informação.

¹A Survey of DNC in Europe - Prof. Dr. Ing. G. Spur - Technical University, Berlin, RFA.



FIGURA 40 - Avanço da automatização do tratamento da informação na produção de peças usinadas.

A tecnologia está cada vez mais avançando na direção de integração destes estágios, além de sua automatização. Costuma-se classificar os sistemas que tratam desta integração e automatização, relacionando-os ao nível de abrangência e atuação na pirâmide passos da figura 40.

Existem dois sistemas de tratamento de informação chamados Projeto com Auxílio de Computador (CAD: Computer Aided Design) e Fabricação com Auxílio de Computador, (CAM: Computer Aided Manufacturing) que têm por objetivo maior esta integração.

Deve-se entender como CAD/CAM todo o sistema automatizado, usado para gerar, coletar, armazenar, manipular ou simplesmente usar informações, orientado para a produção.

O uso de mini-computadores, acoplados a sistemas que utilizam vídeo-gráficos de computador, mudou radicalmente o método com que os engenheiros analisam e projetam um componente, bem como as técnicas que podem ser usadas para transformar um desenho neste componente.

Estima-se que por volta de 1985 a indústria americana esteja gastando US\$1,5 bilhões/ano² em "hardware" para CAD/CAM, mas estes custos serão pequenos se comparados com os custos de pessoal envolvido no desenvolvimento, manutenção e aplicação de sistemas CAD/CAM.

O General Accounting Office³ editou em junho de 1976 a reportagem "Tecnologia de Fabricação - Um desafio em mudança no aumento da produtividade", onde cita os fatos seguintes:

- A taxa de aumento de produtividade nos USA é a mais baixa do mundo industrializado desde a II Guerra.

²A University-Based CAD/CAM Center - Dr. Roger S. Pressman Proceedings 17th Numerical Control Society Annual Meeting & Technical Conference, 1980, pg. 364.

³Transferring Numerical Control Technology to the Small Shop. The problems and a Solution. George P. Putnam - III Research Institute 16th Numerical Control Society Annual Meeting & Technical Conference, 1979, pg. 159.

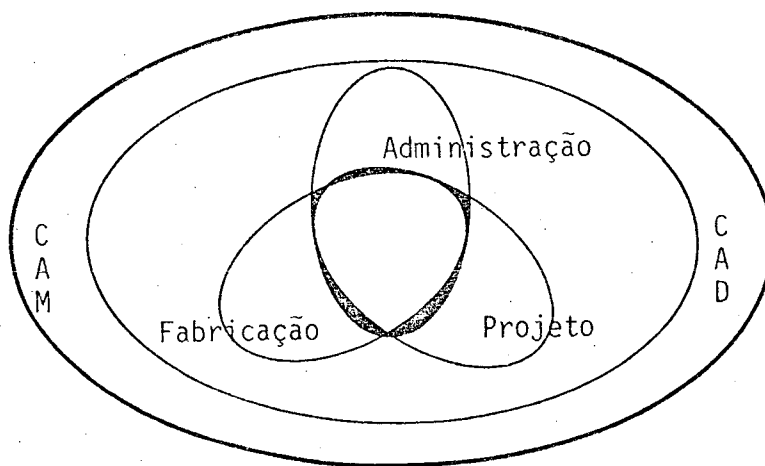
- CAM, especialmente usinagem com máquinas controladas por computador, é uma técnica avançada que poderia ser capaz de ter impacto significativo na produtividade da fabricação.
- A aplicação desta tecnologia não está progredindo de modo a sustentar o "american way of life".
- Sem esforço maior, esta tecnologia não parece ter condições de se difundir satisfatoriamente para pequenas e médias empresas.

O sistema CAD/CAM compreende largo espectro de técnicas que permitem ao engenheiro:

1. Usar o computador para projetar e analisar seus trabalhos de modo iterativo.
2. Servir de interface com a fabricação.
3. Comunicar-se com o computador em modo dinâmico, modificando a base de dados de projeto ou processo.

Do projeto à fita para máquina CN, usando um sistema CAD/CAM, o efeito mais significativo do uso dos computadores na fabricação, é o enfraquecimento dos limites tradicionais e bastante estanques entre os vários departamentos de uma empresa: há uma crescente comunicação entre os diversos departamentos porque eles utilizam fonte comum de informações. A razão é que os computadores fornecem os meios para gerar, guardar e acessar informações essenciais para a manufatura. Mais ainda, o banco de dados é acessível em qualquer lugar, via terminal, e se este participa do sistema CAD/CAM, é possível transformar iterativamente estes dados para convertê-los a uma forma adequada a determinado uso específico.

A figura 41 ilustra a participação dos diversos departamentos de uma indústria, em um banco de dados comum.



Informações comuns
 Informações compartilhadas
 Informações específicas

FIGURA 41 - Participação dos diversos departamentos de uma empresa de um banco de dados central.

5.1. Gráficos por Computador

Os desenhos convencionais de papel, representam informações sobre a geometria, material, tipo de usinagem, acabamento, etc., que em uma fabricação por CN serão transformadas em códigos de um programa.

Estes mesmos desenhos podem ser produzidos eletronicamente em vídeos de terminais de computador, armazenados, alterados, reproduzidos com maior flexibilidade.

Sistemas iterativos de gráficos por computador, quando usados para o projeto de peças, desempenham a função CAD em um sistema CAD/CAM.

Quando a usinagem CN é o segmento CAM do sistema CAD/CAM, o desenvolvimento de CLDATA via gráficos por computador, usa o CAD e o CAM.

Convencionalmente a geometria da peça é repre

sentada por um desenho que é redefinido com comandos de uma linguagem codificada, como o EXAPT. Informações de usinagem são adicionadas, e este programa fonte é processado para obtenção do CLDATA, e então pós-processado para atender as exigências de uma determinada máquina.

Mas porque redefinir a geometria da peça, se o CAD já a definiu previamente e a armazenou no banco de dados do sistema CAD/CAM?

Sistemas CAD que têm um módulo de usinagem em seu "software" (como o sistema CN EXAPT) e interligam ambos, têm benefício adicional, no sentido de que as peças podem ser diretamente programadas do terminal de computador, eliminando-se as descrições de geometria do programa CN, de vez que esta já é conhecida do sistema.

Assim podem-se criar "software" que forneçam, no vídeo do terminal, não somente o programa codificado da peça mas a própria trajetória programada da mesma, que deverá, em princípio, gerar o mesmo desenho projetado e armazenado no banco de dados.

A tecnologia CAD/CAM está em franco desenvolvimento, daí muitas de suas estruturas serem tão novas que requerem modificações para serem úteis às tarefas de CN.

Assim para implementar um sistema global de tratamento de informações para produção, diversos sistemas devem ser compatibilizados, se já não o são, através de mais pacotes de "software", a serem criados pelas empresas usuárias.

Estes pacotes de "software" que atuam como interfaces entre os sistemas, podem atender às mais diversas áreas, como:

- Compatibilização de pós-processadores.
- Compatibilização de dados tecnológicos e sua manipulação pelo sistema global.
- Organização de arquivos de dados de ferramentas, de máquinas, de materiais, etc.

- Organização de dados de projetos já prontos (aquisição ou fabricação de peças, quantidades, locais, tempos, etc.)

As áreas que mais têm aplicado o sistema CAD/CAM, têm sido:

- Projetos, análises mecânicas e eletrônicas
- Geração de instruções para sistemas CN
- "Lay-out" de fábricas
- Banco de dados para projetos e fabricação.

C A P Í T U L O V I

6. CONCLUSÕES, CONTRIBUIÇÕES E SUGESTÕES PARA FUTUROS TRABALHOS

6.1. Conclusões

Com a execução deste trabalho podem-se tirar as seguintes conclusões:

- 1^a - É possível dispor em mini-computador nacional de um sistema de PAC (Programação com auxílio de computador)
- 2^a - A indústria nacional de computadores não oferece ainda ao mercado um espectro de modelos com os quais se possa fazer uma escolha mais livre de "hardware" para configuração voltada para a área do CN.
- 3^a - A implementação do "software" de um sistema CN mais ou menos complexo, em uma configuração, só é possível com a assistência direta de pessoas que tenham conhecimento da construção do sistema. Esta necessidade advém do fato de ser imprescindível a manipulação dos componentes do mesmo sistema, para torná-lo compatível com os recursos de "software" e "hardware" existentes.
- 4^a - A implementação de um sistema CN de programação por computador só é efetiva, ou é tanto mais efetiva, quanto maior for o uso que se fizer do mesmo computador para testar, armazenar, enfim manipular o complexo de informações que são necessárias ao processo de fabricação.

5^a - A transferência de tecnologia na área de PAC, embora apresente dificuldades, é possível e bastante conveniente, desde que se tenha a possibilidade de adaptá-la às condições brasileiras e se possa contribuir com o desenvolvimento a nível internacional.

Para isso é importante que se forme um grupo de trabalho capaz de absorver toda a nova tecnologia e difundi-la.

6^a - A implantação em um computador nacional provou ser de grande importância para o desenvolvimento do CN no Brasil. Maior desenvolvimento na área de PAC poderá ser alcançado quando o sistema for implantado em outros computadores nacionais, dando ao usuário maior gama de opções.

6.2. Contribuições

As contribuições principais deste trabalho para o campo de programação de MFCN, são:

1. Constitui fonte de consulta bibliográfica em Português, praticamente inexistente.
2. Dispõe-se, como resultado prático, de um mini-computador nacional com "software" de PAC, internacionalmente reconhecido, resolvendo-se assim o problema das empresas dispostas a utilizar esta nova tecnologia, mas até então impedidas pela proibição da importação de "minis".
3. Dispõe-se de um pós-processador para traçagem de trajetória da ferramenta, antes do programa ser colocado na máquina operatriz, servindo desta forma para testar previamente a fita. Tal pós-processador serviu ain-

da de base para se estudar e desenvolver a técnica de elaboração de pós-processadores, tarefa indispensável para o uso do sistema.

4. Tem-se diretrizes para implementações de sistemas de PAC em outros mini-computadores nacionais, o que resultará numa ampliação das alternativas atualmente disponíveis, sendo portanto, extremamente salutar ao desenvolvimento e vindo ao encontro das necessidades do usuário.
5. Abrem-se novos caminhos e apontam-se outras direções para futuros trabalhos na área, sendo um destes a elaboração de pós-processadores para máquinas de fabricação nacional (já iniciado).
6. Baseado neste trabalho constituiu-se um grupo de pesquisa e treinamento na área de PAC, que absorve esta nova tecnologia, adaptando-se às condições existentes no país. Tal grupo está inclusive capacitado a continuar desenvolvendo esta área e a treinar pessoal interessado, além de dar assistência na manutenção e aperfeiçoamento do sistema implementado.

6.3. Sugestões para Trabalhos Futuros

Este trabalho partiu do pressuposto que alguém já tivesse decidido por um sistema de programação CN por computador, e, ainda mais, pelo sistema EXAPT de programação.

Na realidade a decisão pelo método de fabricação CN envolve diversas variáveis como custos, treinamento, de pessoal, reorganização estrutural de alguns setores, nova filosofia de fabricação e planejamento, etc. não tratados de nenhuma maneira nestas páginas.

Uma empresa que tenha máquinas CN como opção, pode estar em diferentes estágios tais como:

- substituição de máquinas convencionais por máquinas CN
- aquisição de máquinas CN para preencher funções em empresa existente
- aquisição de máquinas CN, para empresas novas.

A decisão do uso destas máquinas é função de fatores diversos para diferentes indústrias, para países diversos, enfim, para condições diferentes.

Um trabalho que enfocasse os diferentes pontos a serem tratados por um empresário com tais problemas, seria de real utilidade para um país como o Brasil, onde inúmeras empresas estão agora em circunstâncias justamente de engajar-se ou não em CN. Isto é, um trabalho que esclarecesse sob o ponto de vista do empresário, o que é importante considerar em sua análise e o que não o é.

Para alguém que já estivesse usando ou decidido a usar CN, um estudo que facilitasse a escolha de um ou outro método de programação, desta ou daquela linguagem, seria contribuição de valor. Enfocar-se-ia por exemplo o problema de adotar uma linguagem para cada processo, para cada máquina, ou uma genérica.

Além do mais, é fato incontestável que a cibernética está cada vez mais se tornando instrumento capaz de facilitar os métodos convencionais de produção ou mesmo revolucioná-lo completamente. Um trabalho que esclarecesse qual a amplitude de um sistema CN a implementar ou a expandir, é idéia frutífera. Isto é, considerações sobre um sistema que englobasse toda a problemática da produção, desde a idéia à peça acabada, inspecionada e estocada, ou outro que se restringisse somente a uma fração do processo e talvez sendo modularizada para futuros acoplamentos opcionais de módulos, levando à complementariedade progressiva. Parece, como mostra a figura 42 que o computador está se apoderando,

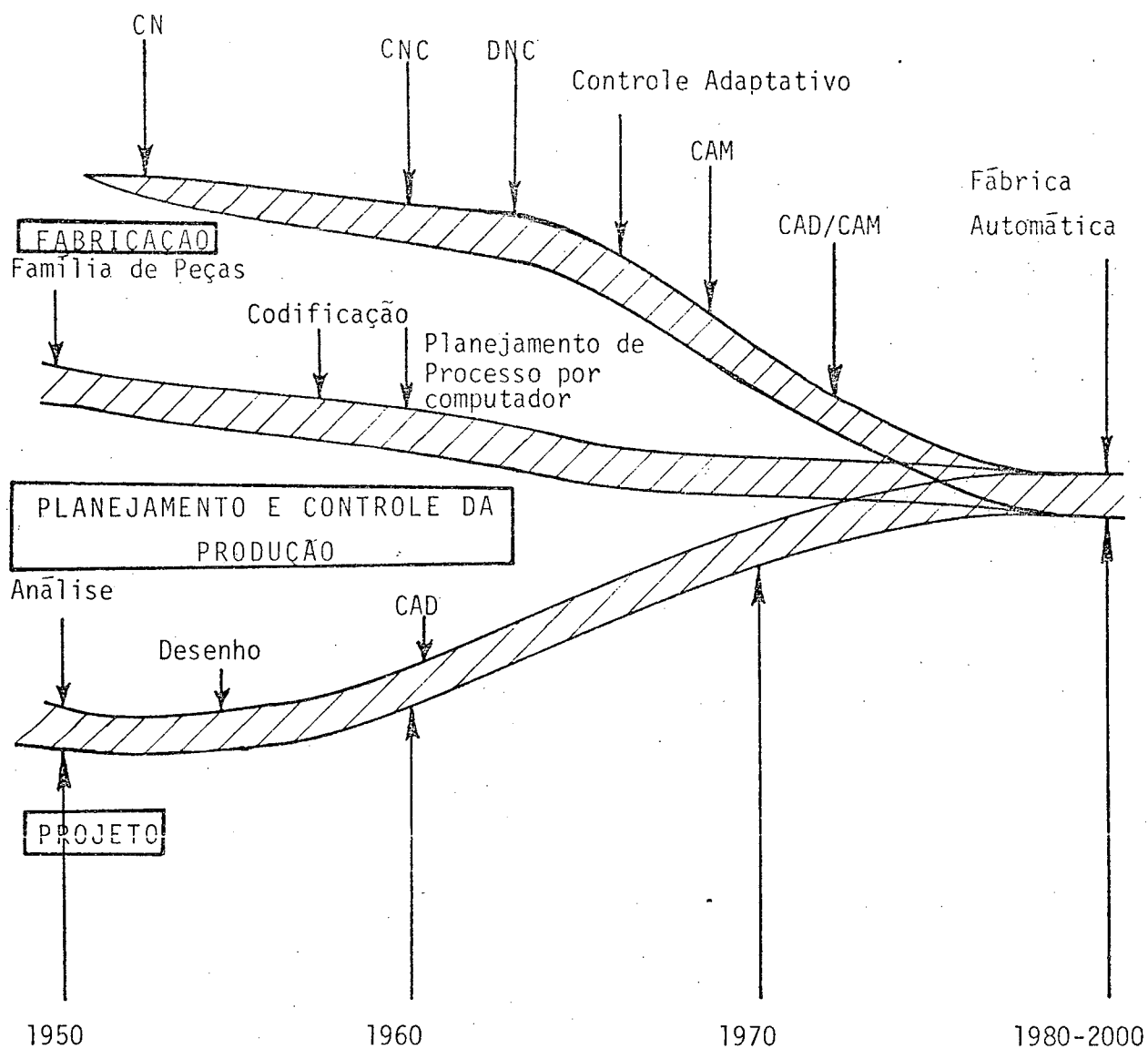


FIGURA 42 - As fases de projeto (o que é produzido), de planejamento (como e onde é produzido) e de controle (quando é produzido) da produção, se encaminham aceleradamente para a automatização.

invadindo rapidamente os domínios da fábrica.

Concretamente poderia se sugerir:

1. Implementação do sistema EXAPT em outros computadores de fabricação nacional, assim que isto for possível.
2. Desenvolvimento de um sistema menor e menos poderoso do que o EXAPT, mas compatível com este, que pudesse ser implementado em computadores menores do que o utilizado neste trabalho, para atender a um grande número de usuários que ainda não se decidiram por uma linguagem mais completa, mas já pretendem usar programação com auxílio de computador.
3. Desenvolvimento de um sistema de comando direto da MFCN com um computador nacional de pequeno porte ("mini") compatível com o sistema EXAPT implantado.
4. Implementação, em computador nacional, de um sistema CAD compatível com o EXAPT.

R E F E R E N C I A S

1. BERAN, Donald C., "Current status of NC Programming Languages in the USA", Society of Manufacturing Engineers, Dearborn, USA, 1972.
2. BUDDE, Wolfgang, "Including machining technology and NC Languages" Society of Manufacturing Engineers, Dearborn, USA, 1972.
3. DATA GENERAL CORPORATION, "Fortran IV User's Manual, Massachusetts, USA, 1977.
4. DATA GENERAL CORPORATION, "Eclipse-line Real time operating system - Reference Manual", Illinois, USA, 1975.
5. DATA GENERAL CORPORATION, "Eclipse-line Fortran IV runtime library - User's Manual", Massachusetts, USA, 1977.
6. EXAPT-VEREIN, "Aufbau eines modularen Prozessors zur Programmierung numerisch gesteuerter Arbeitsmaschinen", Aachen, RFA 1979.
7. EXAPT-VEREIN, "DAFES Manual version s1/2", Aachen, 1979.
8. EXAPT-VEREIN, "Datei-handhabung in Modulprozessor", Aachen, 1980.
9. EXAPT-VEREIN, "EXAPT-NC programming system", Aachen, RFA, 1973.
10. EXAPT-VEREIN, "EXAPT-Part programming reference manual (Working document)", Aachen, RFA, 1979.
11. EXAPT-VEREIN, "EXAPT processor CLDATA-description", Aachen, RFA, 1979.

12. EXAPT-VEREIN, "EXAPT 1.1 Part programming reference manual", Aachen, RFA, 1974.
13. EXAPT-VEREIN, "EXAPT 2 Manual", Aachen, RFA, 1974.
14. EXAPT-VEREIN, "Implementation Guide for PDP 11-34, Operating System PDP 11-34" Aachen, RFA, 1979.
15. EXAPT-VEREIN, "Modular-Processor New developments in the area of turning technology", Aachen, RFA, 1980.
16. ERNST, G., "Erarbeitung von Richtlinien und Unterlagen Zur Erstellung von Postprozessoren für APT - ähnliche Programmiersprachen, EXAPT-Verein, Aachen, RFA, 1977.
17. FERREIRA, Aureo C.; HIRSCH, Bernd E., "Programação de Máquinas CN com auxílio de Computador", Máquinas e Ferramentas, nº 22, São Paulo, 1981, pp.104-114.
18. GETTELMAN, Ken M.; ALBERT, Mark D., "Fundamentals of NC/CAM NC/CAD Guide book", Gardner. Publications, Inc., Cincinnati, USA, 1981, pp.281-446.
19. GRUCON, "Manual do Usuário de Controle Numérico de Máquinas Ferramentas", UFSC, Florianópolis, 1980.
20. GUIMARÃES, Celio C., "Princípios de Sistemas Operacionais", Campus: FENAME: CAPRE, Rio de Janeiro, 1980.
21. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO), "Numerical Control of Machines - NC processor output - Logical structure (and major words)", Switzerland, 1978.
22. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO), "Numerical Control of Machines - NC processor output - Minor elements of 2000-type records (post-processor commands)", Switzerland, 1978.

23. KIEF, Hans B., "NC Handbuch", NC-Handbuch-Verlag, Stokhein, RFA, 1979.
24. KUKLIK, H., "Erstellung eines Ablaufplanes zur Einführung und industriellen Nutzung der rechnergestützten Programmierung unter besonderer Berücksichtigung des Zugriffs zu Datenverarbeitungsanlagen in Unternehmen kleiner bis mittlerer Grösse", EXAPT-Verein, Aachen, RFA, 1977.
25. OLESTEN, Nils O., "Numerical Control", J. Wiley & Sons, Inc., New York, USA, 1970.
26. OPITZ, H.; HIRSCH, Bernd E. "Programmation automatique de machines-outil a commande numerique", CIRP Conférence Européenne de Commande Numérique des Machines-Outils, Paris, April 1968, pp. 61-70.
27. SHAH, Raymond, "Numerical Control Handbook", NCA Verlag, Zürich, Surtzerland, 1979.
28. STEMMER, Caspar E., "Estágio Atual do Comando Numérico no Brasil", Máquinas e Ferramentas, nº 22, São Paulo, 1981, pp. 18-22.
29. STEMMER, Caspar E., "Novos rumos e perspectivas na automação de indústria mecânica", UFSC, Florianópolis, 1980, pp.85-92.

ANEXO 1

ORGANIZAÇÃO ESTRUTURAL DOS MÓDULOS DO
PROCESSADOR EXAPT E DO SISTEMA DAFES.

As estruturas da ligação ("link-edition") no computador PDP 11-34 (Digital Equipment Corporation) de origem, dos diversos módulos do processador EXAPT são fornecidos a seguir.

Para cada módulo se fornece, na primeira linha, simbolicamente, o grupo de subrotinas que compõem a RAIZ e, entre parênteses, os grupos em "overlay".

Listam-se após as subrotinas componentes de cada grupo, separadas por dois pontos.

Dentro de cada grupo, ou de grupo para grupo, podem haver ligações em série ou em paralelo, representadas por traços e vírgulas, respectivamente.

Outros nomes, que não os de subrotinas, se referem ao sistema operacional do PDP 11/34.

Basicamente, no computador SISCO, se conservou a mesma estrutura, com excessão dos módulos DAFES, EINGAB, CUTVAL e CONTUR que tem os mesmos "overlays" do PDP 11/34 mas que adicionalmente foram tratados com a técnica de troca de programas entre disco e memória ("swapping"), embora somente com a subrotina de manipulação de entrada e saída de dados (MODIOS).

Os arquivos auxiliares (Anexo 3) construídos para a ligação do módulo EINGAB ilustram a estrutura adotada no SISCO para estes módulos que foram modificados.

Na figura 43 aparecem esquematicamente a diversas fases da estrutura de "overlays" para o módulo EINGAB.

Existem duas áreas de "overlay", no arquivos EINGAB.OL e EGHELP.OL, com tabelas de carregamento nos arquivos EINGAB.SV e EGHELP.SV, respectivamente. Um quinto arquivo auxiliar chamado EGOVLY é criado pelo programa WRTOLDIR que nele escreve a tabela de "overlay" EGHELP.SV, de modo a se preservarem os endereços para carregamento.

No primeiro carregamento, os "overlays" do segundo somente são carregados para resolução de seus endereços (isto é, são mantidos "dummy") e o inverso ocorre no segundo carregamento.

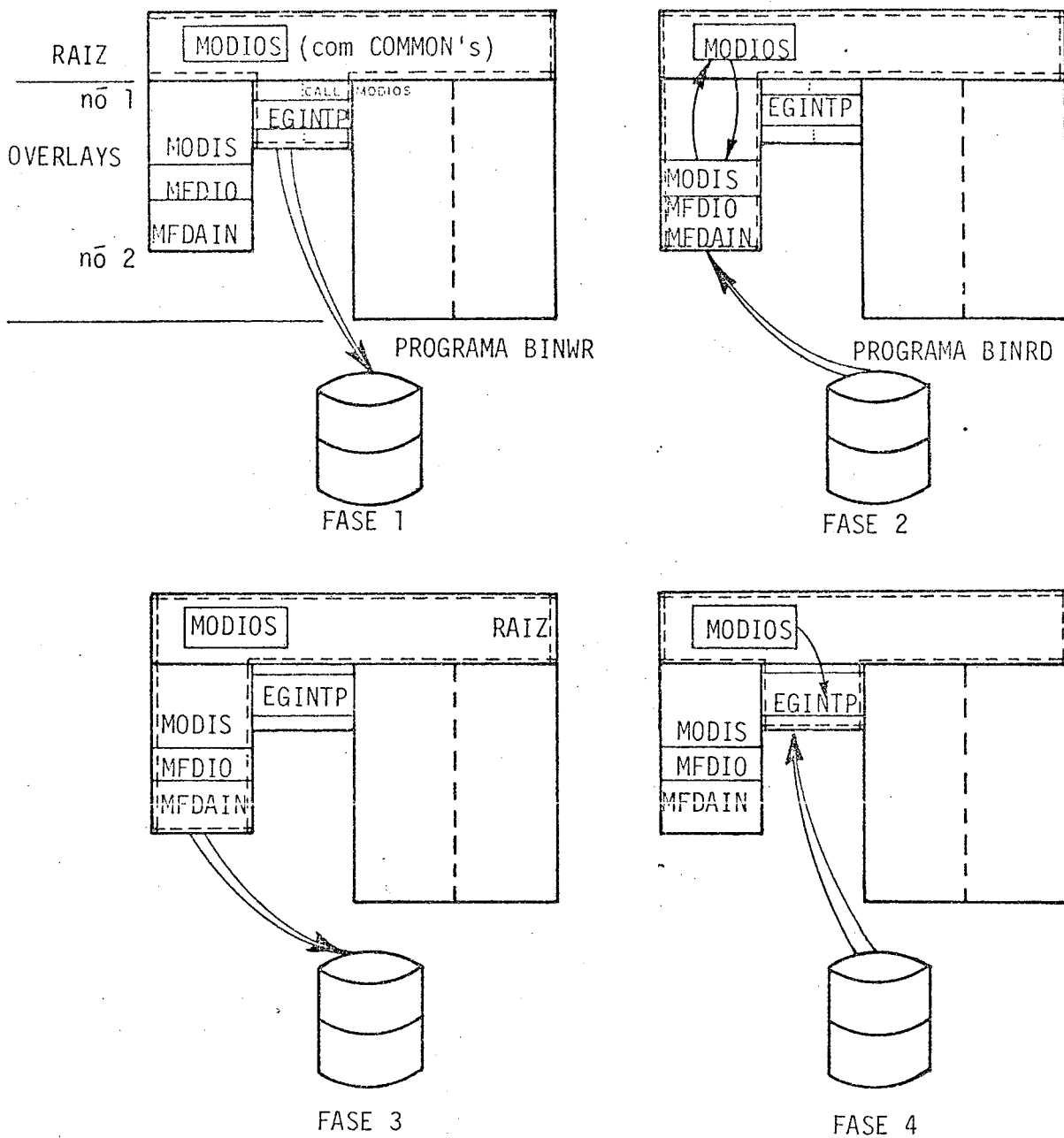


FIGURA 43 Fases do processamento do módulo EINGAB.

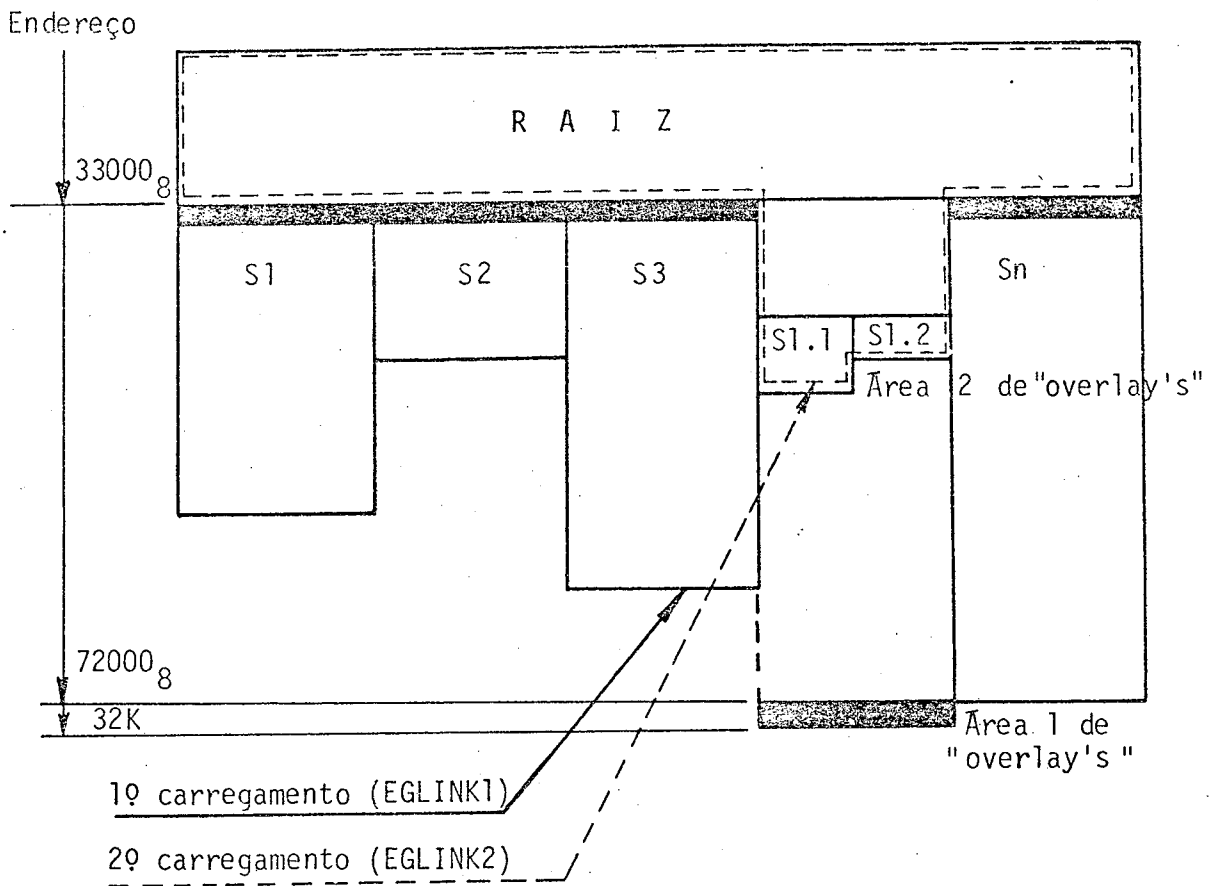


FIGURA 44- Carregamentos do módulo EINGAB.

A figura 44, onde os S_i são os diferentes "overlays", ilustra o que foi dito.

LISTEN

ROOT M2

M20: FCTR SY:LISTEN.OLB/LB:MODLI

M21: FCTR SY:LISTEN.OLB/LB:LISTEN:LIINDX

M22: FCTR SY:EXMS.OLB/LB:DRUCK:GTPSO:RETOUT:PEKOPF

M23: FCTR SY:EXMS.OLB/LB:MEDAIN:MEDAI0:MDORG1:MDORG2

M24: FCTR SY:EXMS.OLB/LB:F11MOD:F11TYP:WAR6

M25: FCTR SY:EXMS2.OLB/LB:MODIOS

M2: FCTR M20-M21-M22-M23-M24-M25

END

MOTION

ROOT MT60*(MT61,MT62,MT63)

MT60: FCTR MT600-MT601-MT602-MT603-MT604-MT605-MT606

MT600: FCTR SY:MOTION.OLB/LB:MODMT

MT601: FCTR SY:MOTION.OLB/LB:MODT:MTBKLD:MTROVE:MTAVEK:MTNKS

MT602: FCTR SY:MOTION.OLB/LB:MTSMT:MTLL:MTLC:MTCC:MTCSIM:MTAUSW

MT603: FCTR SY:EXMS.OLB/LB:PEKOPF:GTPSO:RETOUT:DRUCK

MT604: FCTR SY:EXMS.OLB/LB:MEDAIN:MEDAI0:MDORG1:MDORG2

MT605: FCTR SY:EXMS.OLB/LB:F11MOD:F11TYP:WAR6

MT606: FCTR SY:EXMS2.OLB/LB:MODIOS

MT61: FCTR SY:MOTION.OLB/LB:MTUS:MTCS

MT62: FCTR SY:MOTION.OLB/LB:MTBVR0

MT63: FCTR SY:MOTION.OLB/LB:MTGOTO:MTMON0

END

TECEX2

ROOT E2B*(E2A20,E2A3,E2A9,E2A10,E2A11,E2A40,E2A5,E2A8,E2A7,E2B4)

E2B0: FCTR SY:[210,210]TECEX2.OLB/LB:KOPF?:E2BKLD:E2PP

E2B1: FCTR SY:[210,210]EXMS.OLB/LB:MEDAIN:MEDAI0:WAR6

E2B11: FCTR SY:[210,210]EXMS2.OLB/LB:MODIOS

E2B2: FCTR SY:[210,210]EXMS.OLB/LB:PEKOPF:RETOUT:GTPSO

E2B3: FCTR SY:[210,210]TECEX2.OLB/LB:TECEX2

E2B4: FCTR SY:[210,210]EXMS.OLB/LB:F11MOD:F11TYP:MDORG1:MDORG2

E2B5: FCTR LB:[1,1]SYSIB/LB:SSHORT

E2A2: FCTR SY:[210,210]TECEX2.OLB/LB:E2FELD:E2XFU

E2A20: FCTR E2A2-E2A12

E2A3: FCTR SY:[210,210]TECEX2.OLB/LB:E2DAI0:E2DAIN:E2SKSU:E2IOS:E2WZ

E2A9: FCTR SY:[210,210]TECEX2.OLB/LB:E2ZPF

E2A10: FCTR SY:[210,210]TECEX2.OLB/LB:E2WSPF

E2A11: FCTR SY:[210,210]TECEX2.OLB/LB:E2ZCFP

E2A12: FCTR SY:[210,210]EXMS.OLB/LB:DRUCK

E2A4: FCTR SY:[210,210]TECEX2.OLB/LB:E2AUS:E2CHTP

E2A40: FCTR E2A4-E2A12

E2A5: FCTR SY:[210,210]TECEX2.OLB/LB:E2KAD

E2A8: FCTR SY:[210,210]TECEX2.OLB/LB:E2DAI0:E2DAIN:E2SKSU:E2WZ:E2IOS

E2A7: FCTR SY:[210,210]TECEX2.OLB/LB:E2ZCH:E2IOS:E2DAI0:E2DAIN:E2SKSU

E2B: FCTR E2B0-E2B1-E2B11-E2B2-E2B3-E2B5

END

:CLCOPY

.ROOT CY10

CY10:.FCTR CY01-CY02-CY03-CY04-CY05-CY06

CY01:.FCTR SY:CLCOPY.OLB/LB:MODCY

CY02:.FCTR SY:EXMS.OLB/LB:MFDAIN:MFDAIO:MDORG1:MDORG2

CY03:.FCTR SY:EXMS.OLB/LB:REKOPF:RETOUR:DRUCK:GTPSO

CY04:.FCTR SY:CLCOPY.OLB/LB:CLCOPY:CYBLKD:CYMOND

CY05:.FCTR SY:EXMS.OLB/LB:F11MOD:F11TYP:WARN

CY06:.FCTR SY:EXMS2.OLB/LB:MODIOS

:

:CLDAT2

.ROOT CD10

CD10:.FCTR CD01-CD02-CD021-CD03-CD04-CD05

CD01:.FCTR SY:CLDAT2.OLB/LB:MODCD

CD02:.FCTR SY:EXMS.OLB/LB:MFDAIN:MFDAIO:MDORG1:MDORG2

CD021:.FCTR SY:EXMS2.OLB/LB:MODIOS

CD03:.FCTR SY:EXMS.OLB/LB:REKOPF:RETOUR:DRUCK:GTPSO

CD04:.FCTR SY:CLDAT2.OLB/LB:CLDAT2:CDBLKD:CDCARD:CDOUT:CDSYMB:CDTEXT

CD05:.FCTR SY:EXMS.OLB/LB:F11MOD:F11TYP:WARN

:

:FARKON

.ROOT FK20=(F11,F12,F15,F13,F18,F19,F07,F08,F09,F25)

F21:.FCTR SY:FARKON.OLB/LB:MODFK

F22:.FCTR SY:EXMS.OLB/LB:MFDAIN:MFDAIO:WARN

F222:.FCTR SY:EXMS2.OLB/LB:MODIOS

F23:.FCTR SY:EXMS.OLB/LB:RETOUR:DRUCK:GTPSO:REKOPF

F25:.FCTR SY:EXMS.OLB/LB:MDORG1:MDORG2:F11MOD:F11TYP

F01:.FCTR SY:FARKON.OLB/LB:FARKON:FKCIRO:FKDAIO:FKCELEM:FKBLKD

F02:.FCTR SY:FARKON.OLB/LB:FKGEPD:FKGG:FKGK:FKKK:FKOBLD:FKRAF0

F03:.FCTR SY:FARKON.OLB/LB:FKKOMB:FKPRKO:FKTRIA:FKVEKK:FKERSZ

F04:.FCTR SY:FARKON.OLB/LB:FKVKP:FKTRAP:FKNAC:FKVERT

F05:.FCTR SY:FARKON.OLB/LB:FKEPSI:FKSMIT:FKUMKE:FKAUSP:FKSEQU

F07:.FCTR SY:FARKON.OLB/LB:FKARK

F08:.FCTR SY:FARKON.OLB/LB:FKGOCN

F09:.FCTR SY:FARKON.OLB/LB:FKPTCH

F06:.FCTR SY:FARKON.OLB/LB:FKCJRC:FKNORV:FKINTE:FKGS:FKKS

F10:.FCTR SY:FARKON.OLB/LB:FKAUF3:FKPAT:FKGOST:FKPARL:FKSCYC

F15:.FCTR SY:FARKON.OLB/LB:FKPRPO:FKPRPZ

F11:.FCTR SY:FARKON.OLB/LB:FKAUF1:FKFCKE:FKKVIC:FKJVPK

F12:.FCTR SY:FARKON.OLB/LB:FKKONT:FKITSE

F13:.FCTR SY:FARKON.OLB/LB:FKADSG:FKSPLE

F14:.FCTR SY:FARKON.OLB/LB:FKSNEK:FKERST:FKSECT

F16:.FCTR SY:FARKON.OLB/LB:FKJIN

F19:.FCTR SY:FARKON.OLB/LB:FKKIOV

F17:.FCTR F01-F02-F03-F04-F05-F06-F10

F18:.FCTR F14-F16

FK20:.FCTR F21-F17-F22-F23-F222

.END

```

:CONTUR
:
:ROOT CT200-*(CT210,CT220,CT230,CT240,CT250)
:
CT00:.FCTR CONTUR.OLB/LB:MODCT
CT01:.FCTR EXMS.OLB/LB:MDAID:MDAID:WARN
CT011:.FCTR EXMS2.OLB/LB:MODIOS
CT02:.FCTR EXMS.OLB/LB:GTPSO
CT03:.FCTR CONTUR.OLB/LB:CONTUR:CTBLKD
:
CT210:.FCTR EXMS.OLB/LB:MDORG1:MDORG2:F11MOD:F11TYP
:
CT10:.FCTR EXMS.OLB/LB:DRUCK
CT11:.FCTR EXMS.OLB/LB:REKOPT
CT12:.FCTR CONTUR.OLB/LB:CTCONF
CT13:.FCTR CONTUR.OLB/LB:CTPET
CT14:.FCTR CONTUR.OLB/LB:CTOUTE
CT15:.FCTR CONTUR.OLB/LB:CTPRAN
:
CT20:.FCTR CONTUR.OLB/LB:CTCONF
CT21:.FCTR CONTUR.OLB/LB:CTANFP:CTBEVL:CTCC:CTCIRC
CT22:.FCTR CONTUR.OLB/LB:CTDETC:CTELM1:CTGSM:CTKURZ
CT23:.FCTR CONTUR.OLB/LB:CTLG:CTLL:CTPNTG
CT24:.FCTR CONTUR.OLB/LB:CTTECH:CTTERM:CTTRAL
:
CT30:.FCTR CONTUR.OLB/LB:CTDOV:CTBLK3:CTDOFF:CTOVCH
CT31:.FCTR CONTUR.OLB/LB:CTDAIO:CTINTE
CT32:.FCTR CONTUR.OLB/LB:CTTSE:CTPRKH:CTTRIA
CT33:.FCTR CONTUR.OLB/LB:CTCIRO:CTELMO:CTEPSI
CT34:.FCTR CONTUR.OLB/LB:CTGRD:CTGG:CTCK:CTK
CT35:.FCTR CONTUR.OLB/LB:CTKOMB:CTSNIT:CTVEKK:CTVKP
CT36:.FCTR CONTUR.OLB/LB:CTAUFM:CTKONT:CTPREV:CTPREZ:CTVERT
CT37:.FCTR CONTUR.OLB/LB:CTAUFM:CTCIR:CTECKE:CTNEWC
CT38:.FCTR CONTUR.OLB/LB:CTKIC:CTDRV:CTVPR
:
CT40:.FCTR CONTUR.OLB/LB:CTTSTR:CTBLK2
CT41:.FCTR CONTUR.OLB/LB:CTTCL
CT42:.FCTR CONTUR.OLB/LB:CTSPLN:CTWRTX:CTWRTY
CT43:.FCTR CONTUR.OLB/LB:CTTOUT
CT44:.FCTR CONTUR.OLB/LB:CTSTRK
:
CT50:.FCTR CONTUR.OLB/LB:CTCNWC:CTCNA
:
CT200:.FCTR CT00-CT01-CT02-CT03-CT011
:
CT100:.FCTR CT20-CT21-CT22-CT23-CT24
CT101:.FCTR CT10-CT11-CT12-CT13-CT14-CT15
CT220:.FCTR CT100-CT101
:
CT110:.FCTR CT31-CT32-CT33-CT34-CT35
CT111:.FCTR CT36-CT37-CT38
CT230:.FCTR CT30-CT110-CT111-CT11-CT12
:
CT120:.FCTR CT41-CT10-CT11-CT13-CT15
CT121:.FCTR CT43-CT11-CT12-CT14
CT122:.FCTR CT42-(CT121,CT44)
CT240:.FCTR CT10 (CT120,CT122)
:
CT250:.FCTR CT50-CT11-CT12
:
:END

```

```

;CUTVAL
;
;ROOT CV20-*(CV03, CV04, CV05, CV06, CV07, CV08, CV21)
CV10: .FCTR SY:CUTVAL.OLB/LB:MODCV
CV11: .FCTR SY:EXMS.OLB/LB:MDAIN:MDAIO:VARR
CV111: .FCTR SY:EXMS2.OLB/LB:MODIOS
CV12: .FCTR SY:EXMS.OLB/LB:DRUCK:GTPSO
CV13: .FCTR SY:CUTVAL.OLB/LB:CUTVAL:CVBLKD:CVDAIO:CVVARL:CVSPCS:CVDREZ
CV14: .FCTR SY:EXMS.OLB/LB:MDORG1:MDORG2:F11:MDR:F11TYP
CV15: .FCTR SY:EXMS.OLB/LB:RETOUR
CV03: .FCTR SY:CUTVAL.OLB/LB:CVSCH:DTAN:DASIN
CV04: .FCTR SY:CUTVAL.OLB/LB:CVMCH1
CV05: .FCTR SY:CUTVAL.OLB/LB:CVCHTR:CVCHT2:CVCHTB:CVCHT3
CV06: .FCTR SY:CUTVAL.OLB/LB:CVSCHA
CV07: .FCTR SY:CUTVAL.OLB/LB:CVPASA
CV08: .FCTR SY:CUTVAL.OLB/LB:CVMCHN
;
CV20: .FCTR CV10-CV11-CV12-CV13-CV111
CV21: .FCTR CV14-CV15
;

```

```

;END
;EINGAB
;
;ROOT M1-*(L0, L1, L2, L4, L5, L6, L7, L8, P1)
M10: .FCTR SY:EINGAB.OLB/LB:MODEG
M00: .FCTR LB:(1, 1)SYSLR/LB:SSHORT
M11: .FCTR SY:EINGAB.OLB/LB:EINGAB:EGBLKD
M12: .FCTR SY:EINGAB.OLB/LB:EGFUN2:EGFOLG
M13: .FCTR SY:EXMS.OLB/LB:PEKOPT:DRUCK:GTPSO
M14: .FCTR SY:EXMS.OLB/LB:MODIOS:MEDAIO:MEDAIN
M18: .FCTR SY:EXMS.OLB/LB:VARR
M15: .FCTR SY:EINGAB.OLB/LB:MODEG1:MODEG2
M16: .FCTR SY:EINGAB.OLB/LB:DATA
M17: .FCTR SY:EINGAB.OLB/LB:F11EG
M19: .FCTR SY:EXMS.OLB/LB:BLKDAT:F11TYP:MDORG2
M1: .FCTR M10-M11-M12-M13-M14-M18-M00
L0: .FCTR M15-M16-M17-M19
L1: .FCTR SY:EINGAB.OLB/LB:EGINTP:EGFUN1:EGFUN3:EGDLSP:EGDATA
L2: .FCTR SY:EINGAB.OLB/LB:EGSNTX:EGSXTB:EGRSRV
L4: .FCTR SY:EINGAB.OLB/LB:EGMAC:EGFUN3:EGDAIO:EGDAIN:EGIOS:EGSKSU:EGBL
L5: .FCTR SY:EINGAB.OLB/LB:EGSEQ0:EGSXT2
L6: .FCTR SY:EINGAB.OLB/LB:EGWKZG
L7: .FCTR SY:EINGAB.OLB/LB:EGMACH:EGWSTF:EGZAHL:EGFUN1
L8: .FCTR SY:EINGAB.OLB/LB:EGHILE
P1 : .FCTR P10-(P11-P12, P21)
P10: .FCTR P100-P101
P100: .FCTR SY:EINGAB.OLB/LB:EGGEOD:EGBLK2:EGFTCH:EGOUT:EGTRNF:EGTRLN
P101: .FCTR SY:EINGAB.OLB/LB:EGKDEF
P11: .FCTR SY:EINGAB.OLB/LB:EGPNKT:EGLINE:EGCIRC:EGLI:EGLC:EGCC
P12: .FCTR SY:EINGAB.OLB/LB:EGPARL:EGSYS:EGTRX
P21: .FCTR SY:EINGAB.OLB/LB:EGPAT:EGPPNT:EGPDF
;
;END

```


ANEXO 2

"MUDANÇAS NO PROCESSADOR EXAPT E NO SISTEMA
DAFES, DEVIDAS AO COMPILADOR E SISTEMA OPERACIONAL DO COMPUTADOR
SISCO MB-8000"

fica: DATA INT1/100*0/ e
DATA RE1/40*0.DO/

2.4 - Mudar formatos A para S em comandos DATA

Ex.: DATA X(4)/...,A6,...,A6/ fica: DATA X(4) /...,S6,
...,S6/

2.5 - Num mesmo comando DATA não se pode atribuir valores a um vetor e a uma variável simples.

3 - Comando FORMAT

3.1 - Comando usando FORMAT nH não pode ter continuação em no vo cartão (usando coluna 6).

3.2 - Não podem existir dois delimitadores justapostos no comando FORMAT.

Ex.: FORMAT (.../,...), FORMAT (.../),
FORMAT (...//,...)

ficam: FORMAT (.../...), FORMAT (.../), FORMAT (...//...)

3.3 - Usar formatos A somente para leitura. Nos demais A_i de vem passar para S_i , $H_i > 3$ (ver item 2.4).

Ex.: FORMAT (...A6,...) fica: FORMAT (...S6,...)

4 - Comando WRITE

4.1 - Não é permitido expressão aritmética em índice de va-riáveis em WRITE.

Ex.: WRITE ... (OUT (N-1,I), I=1,IWZ)

fica: NMEN1=N-1

WRITE ... (OUT (NMEN1,I), I=1,IWZ)

4.2 - Mudar os comandos WRITE (4'IBL)KPX e READ(4'IBL)KPX

Para: CALL WRITR (4,IBL,KPX,1,IERR0) e

CALL READR (4,IBL,KPX,1,IERR0) respectivamente.

5 - Comando REAL

5.1 - Os comandos REAL*4 A(200) e REAL*4 B,C

ficam: DIMENSION A(200) e o 2º é deletado.

Em suma, não existe o comando REAL*4 neste compilador.

5.2 - O comando REAL*8 (A-L. 0-Z) não é permitido. As variáveis de precisão dupla devem ser declaradas explicitamente, como DOUBLE PRECISION.

5.3a- As constantes de precisão dupla devem necessariamente, ter um expoente precedido pela letra 'D'.

Ex.: Para atribuir o real 3 à variável de precisão dupla A, ter-se-ia: A= 3.D0

6 - Comando INTEGER

6.1 - Mudar INTEGER*2 A, B, ... para INTEGER A, B, ...

7 - Comando LOGICAL

7.1 - Mudar LOGICAL*1 A, B, ... para LOGICAL A, B, ...

8 - Comando STOP

8.1 - Mudar STOPn para STOP n

9 - Comando GOTO

9.1 - Mudar GOØTO n para GOTOØn

10 - Comando DEFINE FILE

10.1 - Mudar por ex.: DEFINE FILE 4(700,448,U,IX1)
Para: CALL OPEN (4,"EXDATA",0, IERRO,496)

11 - Comando PROGRAM

11.1 - Deve ser deletado todo o comando PROGRAM nome.

12 - DEMAIS MUDANÇAS

12.1 - Verificar as palavras reservadas.

Ex.: INT e MAXI foram mudadas para INTXXX e MAXII.

12.2 - Duas variáveis em EQUIVALENCE não devem ser parâmetros de subprogramas. (Ver variável AUS em EGFN3 de EINGAB, mudada para AAUS).

12.3 - Variáveis em EQUIVALENCE devem ter suas declarações de tipo antes do comando EQUIVALENCE.

12.4 - Não é permitido uso de nomes de subprogramas com nomes iguais em suas 5 primeiras letras. É permitido com nomes de arquivos.

Ex.: MFDAIO, MFDAIN ou MDORG1, MDORG2 foram mudados para: MFDIO, MFDAIN e MDOR1, MDOR2.

12.5 - Rótulos (labels) não usados geram WARNIG's

12.6 - Cada subrotina deve ser separadamente compilada e deve, pois, compor um arquivo.

Mudou-se as bandeiras: C:ØØØ.MACRO nome e C:ØØØ.ENDM que auxiliavam o "librarian" do PDP 11/34 para a bandeira: - nome

no início de cada subrotina, detectada pelo programa RTAP3 (Anexo 3), e divididos os "jobs" em diferentes arquivos pelo dito programa.

12.7 - Inserir COMPILER NOSTACK como primeiro comando de cada subrotina, para permitir inicializações com DATA de variáveis não em COMMON.

12.8 - Verificar funções pré-definidas ou não na biblioteca do SISCO.

Ex.: DFLOAT(I).

12.9 - Mudar a variável IANZ de 13 para 9 bytes com a finalidade de obter maior disponibilidade de espaço na memória. Mudar também as que dependem de IANZ, conforme o manual "DATEIHANDHABUNG IN MODULPROCESSOR". [8]

12.10 - Mudanças para subrotinas em "overlay".

a - Por CALL OVOPN (2, "nome. OL", Ø, IERRO) antes do 1º CALL do programa principal do módulo.

b - Inserir após cada COMPILER NOSTACK da 1ª subrotina de arquivos em "overlay":

OVERLAY nome do overlay

c - Antes de cada CALL arquivo em "overlay" inserir:

CALL OVLOD (2, nome do "overlay", -1, IERRO)

sendo: -1: se carregamento condicional.

0: se carregamento incondicional.

Deve-se analisar cada subprograma da ROOT e se nele aparece chamadas de subrotinas em "overlay", antes da chamada deverá existir o comando CALL OVLOD; o que não significa que antes de cada CALL de subrotina em "Overlay" deva ser precedida da CALL OVLOD (podem existir subrotinas do mesmo overlay sendo chamadas).

d - Inserir

CALL CLOSE (2, IERRO)

antes do comando RETURN do programa principal de cada módulo (fechar canal usado para carregamento de "overlay").

e - Os nomes dos "OVERLAY's" devem ser declarados em comandos EXTERNAL

12.11 - Ver demais mudanças na tabela de diferenças do FORTRAN ANSI para o do SISCO (ou ECLIPSE), no Manual "Fortran IV User's Guide", [3].

13 - Representação interna de caracteres (DAFES)

A tabela dos códigos internos atuais usados no processador, e os ASCII correspondentes, é fornecida a seguir.

Caracteres	Código EXAPT	Código ASC II	Caracteres	Código interno EXAPT	Código ASC II
A	1	65	0	27	48
B	2	66	1	28	49
C	3	67	2	29	50
D	4	68	3	30	51
E	5	69	4	31	52
F	6	70	5	32	53
G	7	71	6	33	54
H	8	72	7	34	55
I	9	73	8	35	56
J	10	74	9	36	57
K	11	75	,	37	44
L	12	76	.	38	46
M	13	77	\$	39	36
N	14	78	-	40	45
O	15	79	*	41	42
P	16	80	/	42	47
Q	17	81	+ &	43	38 43
R	18	82	= ≠	44	35 61
S	19	83	(%	45	37 40
T	20	84) <	46	41 40
U	21	85			
V	22	86			
W	23	87			
X	24	88			
Y	25	89			
Z	26	90			

ANEXO 3

"PROGRAMAS E ARQUIVOS AUXILIARES DA IMPLIEN-
TAÇÃO"

Tem-se abaixo alguns programas auxiliares desenvolvidos sobretudo para gerenciamento de memória. Os de título iniciado com as letras 'EG' se referem ao módulo EINGAB e são similares aos desenvolvidos para os módulos CUTVAL, CONTUR e para o sistema DAFES, não descritos aqui.

PROGRAMA BINWR (endini, endfin, canal)

Escreve no disco um arquivo binário da memória, compreendido entre 'endini' e 'endfin', aberto com 'canal'.

PROGRAMA BINRD (endini, endfin, canal):

Carrega um arquivo binário do disco para a memória, entre os endereços 'endini' e 'endfin', por 'canal'.

PROGRAMAS EGDMyxx e EXxx: (xx: dois algarismos).

Programas que fornecem os endereços para carregamento dos overlay neles citados junto a '.ENT' e '.ENTO'.

PROGRAMA LOVL e SOVL

O segundo armazena um vetor de 9 componentes na tabela de "overlay" e o primeiro carrega este mesmo vetor, da tabela. Estes componentes contêm os parâmetros do diretório dos "overlays", tais como número de áreas usadas (2 para o EINGAB), e principalmente, o endereço de memória onde inicia a área.

Arquivos EGLINK1 e EGLINK2:

Carregam a primeira e segunda área de "overlay" de EINGAB.

Arquivos EGOVLY1 e EGOVLY2:

Definem os "overlays" das áreas 1 e 2, respectivamente.

PROGRAMAS MTW e RTAP3:

Estes programas foram abordados no texto.

PROGRAMA WRTOLDIR:

Carrega em EGOVLY a tabela do "overlay"
EGHELP.SV.

	TITL	BINRD	
	.ENT	BINRD	
	.NREL		
V6.:	.BLK	1	; IANZ
	.EXTU		
	.CSIZ	0	
	FS.		
BINRD:			
	JSR	@.FARL	
	LDA	0,@T.+1,3	; IBIS
	LDA	1,@T.+0,3	; IVON
	SUB	1,0	
	STA	0,TS.+1,3	
	JSR	@.LDO	
	.C1		
	LDA	1,TS.+1,3	
	JSR	@.SDVD	
	JSR	@.ST1	
	V6.		; IANZ
	IANZ=IANZ*256+ICHAN		
	JSR	@.LDO	
	.C1		
	JSR	@.LD1	
	V6.		; IANZ
	JSR	@.SMPY	
	LDA	2,@T.+2,3	; ICHAN
	ADD	1,2	
	LDA	0,@T.+0,3	
	SUB	1,1	
	.SYSTEM		
	.RDB	77	
	JMP	.+1	
	JSR	@.FRET	
.C1:	000400		
	FS.=5		
	SFS.=0		
	T.--167		
	V.=200+T.		
	TS.=T.+2		
	FTS.=T.+0		
	V6.=V.+2		
	FVS.=V.+0		
	.END		

Programa: BINRD.


```

.TITL      BINWR
.ENT       BINWR
.TXTM      1
.NREL      1
V6.:      .BLK      1      ; IANZ
.EXTU      0
.CSIZ      0
FS.
BINWR:
JSR        @.FARL
LDA        0,@T.+1,3      ; IBIS
LDA        1,@T.+0,3      ; IVON
SUB        1,0
STA        0,TS.+1,3
JSR        @.LDO
.C1
LDA        1,TS.+1,3
JSR        @.SDVD
JSR        @.ST1
V6.:      ; IANZ
; IANZ=IANZ*256+ICHAN
JSR        @.LDO
.C1
JSR        @.LD1
V6.:      ; IANZ
JSR        @.SMPY
LDA        2,@T.+2,3      ; ICHAN
ADD        1,2
LDA        0,@T.+0,3
SUB        1,1
.SYSTM
.WRB       77
JMP        .+1
JSR        @.FRET
.C1:      000400
FS.=5
SFS.=0
T.=-167
V.=200+T.
TS.=T.+2
FTS.=T.+0
VS.=V.+2
FVS.=V.+0
.END

```

Programa: BINWR.

```

      .TITL      EGDMY00
      -----
      .ENT      MDEG1,MDEG2,DATEN,DRUCK,BLKDAT
      .EXTD     .CPYL,.FRET
      .NREL
      .CSIZ     0
      5
MDEG1:
MDEG2:
DATEN:
DRUCK:
BLKDAT: JSR      @.CPYL
        JMP      @.FRET
        .END

```

```

      .TITL      EGDMY10
      -----
      .ENTO     EG10
      .ENT      EGINT,EGINP,EGDLSF,EGDTA
      .EXTD     .CPYL,.FRET
      .NREL
      .CSIZ     0
      5
EGINT:
EGINP:
EGFN1:
EGFN3:
EGDLSF:
EGDTA: JSR      @.CPYL
        JMP      @.FRET
        .END

```

```

      .TITL      EGDMY20
      -----
      .ENTO     EG20
      .ENT      EGSNTX,EGRSRV
      .EXTD     .CPYL,.FRET
      .NREL
      .CSIZ     0
      5
EGSNTX:
EGRSRV: JSR      @.CPYL
        JMP      @.FRET
        .END

```

```

      .TITL      EGDMY21
      -----
      .ENT      EGNEXT
      .EXTD     .CPYL,.FRET
      .NREL
      .CSIZ     0
      5
EGNEXT: JSR      @.CPYL
        JMP      @.FRET
        .END

```

Programas: EGDMY00, EGDMY10, EGDMY20 e
EGDMY21.

```

      .TITL  EGDMY22
      .ENT   EGSYNT
      .EXTD  .CPYL, .FRET
      .NREL
      .CSIZ  0
      5
EGSYNT: JSR   @.CPYL
        JMP   @.FRET
        .END

```

```

      .TITL  EGDMY30
      .ENTO  EG30
      .ENT   EGMAC,EGDIO,EGDAIN,EGIOS,EGSKSU
      .EXTD  .CPYL, .FRET
      .NREL
      .CSIZ  0
      5
EGMAC:
EGFN3:
EGDIO:
EGDAIN:
EGIOS:
EGSKSU: JSR   @.CPYL
        JMP   @.FRET
        .END

```

```

      .TITL  EGDMY40
      .ENTO  EG40
      .ENT   EGSEQU,EGSXT2
      .EXTD  .CPYL, .FRET
      .NREL
      .CSIZ  0
      5
EGSEQU:
EGSXT2: JSR   @.CPYL
        JMP   @.FRET
        .END

```

```

      .TITL  EGDMY50
      .ENTO  EG50
      .ENT   EGWKZG
      .EXTD  .CPYL, .FRET
      .NREL
      .CSIZ  0
      5
EGWKZG: JSR   @.CPYL
        JMP   @.FRET
        .END

```

Programas: EGDMY22, EGDMY30, EGDMY40 e
EGDMY50.

```

      .TITL   EGDMY70
      .ENTO   E670
      .ENT    EGHILF
      .EXTD   .CPYL, .FRET
      .NREL
      .CSIZ   0
      5
EGHILF: JSR    @.CPYL
        JMP    @.FRET
        .END

      .TITL   EGDMY60
      .ENTO   E660
      .ENT    EGMAH,EGWSTF,EGZ AHL
      .EXTD   .CPYL, .FRET
      .NREL
      .CSIZ   0
      5
EGMAH:
EGWSTF:
EGZ AHL:
EGFN1: JSR    @.CPYL
        JMP    @.FRET
        .END

      .TITL   EGDMY80
      .ENT    EGGEOD,EGFTCH,EGOUT,EGTRNF,EGTRLN,EGKDEF
      .EXTD   .CPYL, .FRET
      .NREL
      .CSIZ   0
      5
EGGEOD:
EGFTCH:
EGOUT:
EGTRNF:
EGTRLN:
EGKDEF: JSR    @.CPYL
        JMP    @.FRET
        .END

      .TITL   EGDMY81
      .ENTO   E681
      .ENT    EGPAT,EGPPNT,EGPDIF
      .EXTD   .CPYL, .FRET
      .NREL
      .CSIZ   0
      5
EGPAT:
EGPPNT:
EGPDIF: JSR    @.CPYL
        JMP    @.FRET
        .END

```

Programas: EGDMY70, EGDMY60, EGDMY80 e
EGDMY81.

```

      .TITL      EX00
      .ENTO      EXMOD
      .ENT        MODIS
      .EXTD      .FRET, .CPYL
      .NREL
      .CSIZ      0
      5
MODIS: JSR      @.CPYL
      JMP      @.FRET
      .END

```

```

COMPILER NOSTACK
SUBROUTINE INIMD
COMMON / OVTEL / IARR1 (9), IARR2 (9)
EXTERNAL EXMOD
ISIZE=IARR1(9)-IARR1(5)
ISIZE=ISIZE/256
CALL DFILW ("SWAPO0",IER)
CALL DFILW ("SWAPO1",IER)
CALL CFILW ("SWAPO0",3,ISIZE,IER)
CALL CFILW ("SWAPO1",3,ISIZE,IER)
CALL OPN
CALL OVLOD (2,EXMOD,-1,IER)
CALL BINWR (IARR1(5),IARR1(9),21)
RETURN
END

```

Programa WRTOLDIR

```

INTEGER IARR (1000)
CALL OPEN (1,"EGHELP.SV",0,IERRO)
READ BINARY (1) IARR
IADR=IARR(420K-16K+1)-13
IANZ=IARR(IADR)
LANG=IANZ*4+IADR
CALL OPEN (2,"EGOVLY",0,IERRO)
TYPE "OPENING ERROR OF EGHELP.SV",IERRO
CALL OPEN (3,"$TTO",0,IERRO)
WRITE(3,9901)(I,IARR(I),I=IADR,LANG)
9901  FORMAT(" ADRESSE ",0I6," INHALT ",0I6)
WRITE BINARY (2)(IARR(I),I=IADR,LANG)
CALL CLOSE(1)
CALL CLOSE (2)
END

```

Programas: EX00, INIMD e WRTOLDIR.

```

.TITL   EGDMY82
.ENTO   E682
.ENT    EGPNKT, EGLINE, EGCIRC, EGLL, EGCC, EGLC
.ENT    EGPARL, EGTSYS, EGMTRX
.EXTD   .CPYL, .FRET
.NREL
.CSIZ   0
5
EGPNKT:
EGLINE:
EGCIRC:
EGLL:
EGCC:
EGLC:
EGPARL:
EGTSYS:
EGMTRX: JSR    @.CPYL
        JMP    @.FRET
        .END

```

```

.TITL   OPN
.ENT    OPN
.NREL
.CSIZ   0
.EXTU
.TXTM   1
SWAPO:  .+1*2
        .TXT    *SWAPO0*
SWAP1:  .+1*2
        .TXT    *SWAPO1*
        1
OPN:    JSR    @.FARL
        LDA    0, SWAPO
        SUB    1, 1
        .SYSTEM
        .OPEN   24
        JMP    .+1
        LDZ    0, SWAP1
        .SYSTEM
        .OPEN   25
        JMP    .+1
        JMP    @.FRET
        .END

```

Programas: EGDMY82 e OPN.

FORTRAN IV VOIC-03 TUE 02-DEC-80 15:28:56 PAGE 001
 CORE 23K, UIC (250,212) , REHQ.LST=MTW.FTN

Programa MTW

```

0001      BYTE BBUF(4096),ISBP(4),EBL(2),BHILF(32)
0002      BYTE BCR(2),BRECRD(136)
0003      INTEGER IBUF(2048),IPARM(6),ISBP(2),PFLAG,TIMFLG,HILF(16)
0004      INTEGER ANSW,YES,RECLNG,IRECRD(68)

C
0005      EQUIVALENCE (IBUF(1),BBUF(1))
0006      EQUIVALENCE (IBLANK,EBL(1))
0007      EQUIVALENCE (ISBP(1),BSP(1))
0008      EQUIVALENCE (HILF(1),BHILF(1))
0009      EQUIVALENCE(IRECRD(1),BRECRD(1))

C
C *** QIO FUNCTION CODES
C *** READ LOGICAL BLOCK IO,RLB
0010      DATA IORLB /*001000/
C *** WRITE LOGICAL BLOCK IO,WLB
0011      DATA IWLB /*000400/
C *** ATTACH DEVICE IO,ATT
0012      DATA IOATT /*001400/
C *** REWIND TAPE
0013      DATA IORWD /*002400/
C *** GET STATUS
0014      DATA IOSEC /*002520/
C *** SET STATUS
0015      DATA IOSTC /*002500/
C *** WRITE END-OF-FILE MARK
0016      DATA IOEOF /*003000/
C *** EVENT FLAG NUMBER
0017      DATA PFLAG /1/
0018      DATA TIMFLG/27/
C *** MAGNETBAND EINHEIT
0019      DATA MIO /27/
C *** LAUFENDE KARTENNUMMER
0020      DATA KARTE /0/
0021      DATA IBLANK /2H /
0022      DATA YES /1HY/
0023      DATA HILF /16*0/
0024      DATA BCR/13,9/
C *** CALL ASSLUN (2,'MT',0)
0025      CALL GETADR (IPARM(1),BRECRD)
0026      CALL QIO (IOATT,MTO,PFLAG,,ISBP,IPARM,IDSP)
0027      CALL WAITER (PFLAG)
0028      IF(ISBP(1) .NE. 1) GOTO 10000
0029      WRITE (3,3000)
0030      READ (3,3001) ANSW
0031      3000 FORMAT (' REWIND [Y/N] :',*)
0032      READ (3,3001) ANSW
0033      3001 FORMAT (A1)
0034      IF(ANSW.NE.YES) GOTO 3301
0035      CALL QIO (IORWD,MTO,PFLAG,,ISBP,,IDSP)
0036      CALL WAITER (PFLAG)
0037      3301 CONTINUE
0038      10 CONTINUE
0039      10 CONTINUE
0040      READ (1,9901,END=1010)BRECRD
0041      J=81
0042      12 CONTINUE

```

Programa: MTW.

FORTRAN IV V01C-03 TUE 02-DEC-80 15:26:56 PAGE 002
 CORE=23K, UIC=[250,212] ,BEHO.LST-MTW.FIN

```

0043            J=J-1
0044            IF(J.EQ.0)GOTO 10
0046            IF(FRECKR(J).EQ.BBL(1)) GOTO 12
0048            IF(J.LE.14)J=15
0050            IF(MOD(J,2).EQ.0)J=JJ
0052            J=J+1
0053            FRECKR(J)=FCKR(J)
0054            IPARM(2)=J
0055            CALL RIO (IOWLB,MTO,PFLAG,ISBP,IPARM,IDSP)
0056            CALL WAITFR (PFLAG)
0057            IF(ISBP(1).NE.0) GOTO 11
0059            CALL MARK (TIMFLG,10,2)
0060            CALL WAITFR(TIMFLG)
0061            11 CONTINUE
0062            IF(ISBP(1).NE.1) GOTO 10000
0064            KARTE=KARTE+1
0065            GOTO 10
0066            9901 FORMAT(136A1)
0067            10000 CONTINUE
0068            IF(ISBP(1).EQ.248) GOTO 1010
C *** ABNORMALER STOP (MTW NOT READY,...)
0070            WRITE(3,9902) BSBP(2),BSBP(1),ISBP(2),IDSP
0071            RECLNG = ISBP(2)
C            WRITE (3,3007) (BBUF(JJ),JJ=1,RECLNG)
0072            DD 3310 J=15,RECLNG,12
0073            JA=J
0074            JE=JJ+5
0075            IF(JE.GT.RECLNG) JE=RECLNG
0077            WRITE (3,3008) JA,JE,(BBUF(JJ),JJ=JA,JE)
0078            KK= -1
0079            DD 3311 JJ=JA,JE
0080            KK = KK+2
0081            BFILE(KK) = BBUF(JJ)
0082            3311 CONTINUE
0083            WRITE (3,3009) (FILE(KK),KK=1,16)
0084            3310 CONTINUE
0085            3008 FORMAT (' DMP ',I4,1X,I4,' ',I5,16(2X,A1,1X))
0086            3009 FORMAT (' OCTAL            :', 16( 04 ))
C3007 FORMAT (' DMP ',136A1)
0087            9902 FORMAT(' MTW STOPPED WITH STATUS :',2I4,2X,I4,2X,I4)
0088            CALL EXIT
0089            1010 CONTINUE
C *** NORMALER STOP , EOF ENCOUNTERED
0090            CALL RIO (IOEOF,MTO,PFLAG,ISBP,IDSP)
0091            CALL WAITFR (PFLAG)
0092            IF(ISBP(1).NE.1) GOTO 10000
0094            WRITE(3,9903) KARTE
0095            9903 FORMAT(' MTW STOPPED AT E-O-F AFTER ',I5,' CARDS WRITTEN ')
0096            CALL EXIT
0097            END

```

Programa: MTW (continuação).

 Programa RTAP3

```

DIMENSION IEB(59), IAS(59), II(5000)
DIMENSION IF (100), IE(200)
DIMENSION IEE (199)
EQUIVALENCE(IE(2), IEE(1))
ACCEPT "INPUT-FILE:"
READ (11, 200) IF(1)
CALL MTOFD (1, IF, 0, IER)
ACCEPT "OUTPUT-FILE:"
READ (11, 200) IF(1)
OPEN 2, IF
200  FORMAT (520)
1   CALL MTDIO (1, 0, II, ISTAT, IP, IER)
    IF (IER, EQ. 1) GO TO 3
    TYPE IER, ISTAT
    CLOSE 1
    CLOSE 2
    STOP
    3   IX=0
    DO 2 I=1, IP
      IH=II(I)/256
      IL=II(I)-IH*256
      IX=IX+1
      IE(IX)=IH*256
      IF (IH, EQ. 13) GO TO 99
      IX=IX+1
      IE(IX)=IL*256
      IF (IL, EQ. 13) GO TO 99
2   CONTINUE
    GO TO 1
99  IX=IX-1
    IF(IE(1), NE. 1H+) GOTO 98
    CLOSE 2
    JJ=2
97  CONTINUE
    JJ=JJ+1
    IF(IE(JJ), NE. 1H ) GOTO 97
    IE(JJ)=1H.
    JJ=JJ+1
    IE(JJ)=1HF
    JJ=JJ+1
    IE(JJ)=1HR
    DO 96 I=1, 10, 2
    IE(I/2+1)=IEE(I)/256*256+IEE(I+1)/256
96  CONTINUE
    WRITE(10, 9901)(IE(I), I=1, 5)
9901 FORMAT(" FILE NAME --->", 5A2)
    OPEN 2, IE
    GOTO 1
98  CONTINUE
    WRITE (2, 100)(IE(I), I=1, IX)
    GO TO 1
100 FORMAT (100A1)
    END

```

Programa: RTAP3,

DELETE/V EGMAP1

LODR/P 77/C EGMAP1/L EINGAB/S EINGAB EGBLKD ^
 EXMSFIX:SWAPMODIOS EGFN1 EGFN2 EGFN3 ^
 EGFOLG REKOPF DRUCK WARN SOVL INIMD OPN BINRD BINWR ^
 @EGFORTLIB@ @EGOVLY1@ 73000/N FORT.LB

EGLINK1

DELETE/V EGMAP2

LODR/P 77/C EGMAP2/L EGHELP/S EINGAB EGBLKD ^
 EXMSFIX:SWAPMODIOS EGFN1 EGFN2 EGFN3 ^
 EGFOLG REKOPF DRUCK WARN SOVL INIMD OPN BINRD BINWR ^
 @EGFORTLIB@ @EGOVLY2@ 73000/N FORT.LB

EGLINK2

33000/N CMODIS MFDIO MFDAIN, ^
 MDEG1 MDEG2 MDOR2 DATEN BLKDAT, ^
 EGINT EGINP EGDLSF EGDTA, ^
 EGSNTX EGRSRV EGNEXT EGSYNT, ^
 EGMAC EGDIO EGDAIN EGIOS EGSKSU EGBLK3, ^
 EGSEQU EGSXT2, ^
 EGWKZG, ^
 EGMAH EGWSTF EGZAH, ^
 EGHILF, ^
 EGGEOD EGBLK2 EGFTCH EGOUT EGTRNF EGTRLN EGKDEFJ ^
 72000/N (EGDMYB1,EGDMYB2) ^

EGOVLY1

33000/N (EX00, ^
 MDEG1 MDEG2 MDOR2 DATEN BLKDAT, ^
 EGDY10, ^
 EGDY20, ^
 EGDY30, ^
 EGDY40, ^
 EGDY50, ^
 EGDY60, ^
 EGDY70, ^
 EGGEOD EGBLK2 EGFTCH EGOUT EGTRNF EGTRLN EGKDEFJ ^
 CEPAT EGPPNT EGPDIF, ^
 EGPNTY EGLINE EGCIRC EGCC EGLL EGLC EGPART EGTSYS EGMTRXJA

EGOVLY2

Arquivos: EGLINK1, EGLINK2, EGOVLY1 e EGOVLY2,
 para carregamento do módulo EINGAB.

0001 SOVL MACRO REV 06.00.

16:40:51 12/01/80

	TITL	SOVL
02	. ENT	SOVL, LOVL
03	. NREL	
04	. EXT0	. FARL, . FRET
05	. F1:	
06	00000'000000 . F2:	0
07	000000	. CSIZ 0
08	00001'000001	FS.
09	00002'006002\$LOVL:	JSR @. FARL
10	00003'035611	LDA 3, T, 3
11	00004'032423	LDA 2, @C420
12	00005'150015	COM# 2, 2, SNR
13	00006'002001\$	JMP @. FRET
14	00007'025000 SLOW:	LDA 1, 0, 2
15	00010'045400	STA 1, 0, 3
16	00011'151400	INC 2, 2
17	00012'175400	INC 3, 3
18	00013'104110 SOLP:	SBI 1, 1
19	00014'124015	COM# 1, 1, SNR
20	00015'002001\$	JMP @. FRET
21	00016'020412	LDA 0, C4
22	00017'040412	STA 0, CNT
23	00020'021000	LDA 0, 0, 2
24	00021'041400	STA 0, 0, 3
25	00022'151400	INC 2, 2
26	00023'175400	INC 3, 3
27	00024'014405	DSZ CNT
28	00025'000773	JMP . -5
29	00026'000765	JMP SOLP
30	000001	FS. = 1
31	177611	T. = -167
32	000011	V. = 200+T.
33	00027'000420 C420:	420
34	00030'000004 C4:	4
35	00031'000000 CNT:	0
36	00032'000001	FS.
37	00033'006002\$SOVL:	JSR @. FARL
38	00034'031611	LDA 2, T, 3
39	00035'036772	LDA 3, @C420
40	00036'174015	COM# 3, 3, SNR
41	00037'002001\$	JMP @. FRET
42	00040'000747	JMP SLOW
43	. END	SOVL

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS

6

ANEXO 4

MÉTODO DE UTILIZAÇÃO DO PROCESSADOR EXAPT

MÉTODO DE UTILIZAÇÃO DO PROCESSADOR EXAPT

Partindo dos programas fonte do sistema EXAPT, em FORTRAN, a utilização exige 3 passos:

- 1 - Compilação
- 2 - Ligação e Carregamento
- 3 - Execução.

No primeiro passo cada subrotina é compilada separadamente.

No segundo é feita a ligação e carregamento dos arquivos gerados no primeiro passo, conforme as estruturas próprias de cada módulo, tratadas no Anexo 1, com Auxílio dos programas de tratamento de endereços e de arquivos do Anexo 3.

A execução pode ser feita com chamada individual na console dos módulos necessários ou com chamamento automático.

Na execução automática o passo 3 se subdivide em:

- 3.1 - Posicionamento do usuário em seu diretório.

A entrada do usuário em diretório de trabalho se faz com:

Comando: DIR partição: diretório

onde partição é:

DPØF: se o diretório está no disco fixo.

DPØ: se o diretório está no disco removível

diretório é: nome do diretório dos programas do usuário.

- 3.2 - Ligar o arquivo de entrada do Sistema EXAPT (nome: INPUT) ao arquivo onde se encontra o programa de peça a ser processado.

Comandos: DELINK INPUT

LINK INPUT EXAPT: arquivo

onde arquivo é: nome do arquivo em que se encontra o programa de

peça. Este arquivo pode ser criado com a ajuda dos periféricos disponíveis na configuração (console, teletipo, leitora fita perfuradora, etc.)

3.3 - Executar o processador EXAPT

Comandos: 1) diretório: EXAPT

ou

2) EXAPT

onde diretório é: nome do diretório onde se encontra os programas-objeto do processador EXAPT, caso estes não estejam no próprio diretório do usuário, quando se usaria o 2º comando.

O programa principal de nome EXAPT chama cada módulo do disco para a memória, assim que se faça necessário sua execução.

Cada vez que inicia o processamento de um determinado módulo, aparece na console a mensagem:

hh mm ss MÓDULO EM PROCESSAMENTO: nome

onde hh mm ss são: horas, minutos e segundos do início da execução do módulo.

nome é: nome do módulo.

A interrupção da execução é conseguida em qualquer instante pressionando as teclas CTRL e A (simultaneamente) ou a tecla PF0.

A segunda possibilidade de execução é a individual, módulo a módulo, pois um programa de uma determinada peça não exige necessariamente passar por todos os módulos que compõem o sistema EXAPT de programação. Por exemplo um programa que não utilize a função contorno não será processado nos módulos CONTUR e FARKON.

Os módulos necessários para processar um determinado programa de peça são determinados pelo primeiro módulo do sistema: EINGAB. Este gera um arquivo chamado MSCC.CM onde escreve ordenadamente os módulos a serem chamados após EINGAB.

Executado EINGAB, lista-se o arquivo MSCC.CM e se chama os módulos contidos neste arquivo, na ordem em que foram dados (Figura 45).

Na execução automática, utiliza-se um programa principal de nome EXAPT que, para poupar trabalho de se executar individualmente cada módulo necessário, faz esta função programadamente. Este programa principal carrega em disco o módulo que terminou seu processamento e leva para a memória o próximo módulo a ser executado, utilizando a técnica de troca de programas ("swapping").

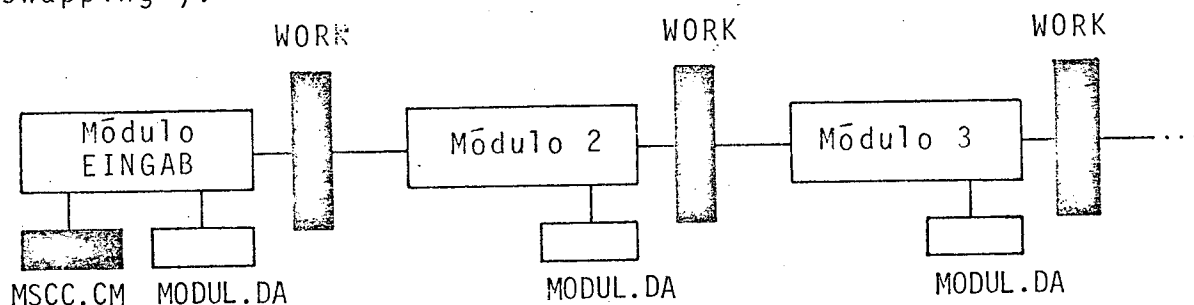


FIGURA 45 -- Arquivos produzidos pela execução dos módulos do processador EXAPT.

Resultados Intermediários dos Módulos

Arquivo MODUL DA.

Guarda conteúdo de algumas variáveis COMMON rotuladas (MODINF, ZUGRIFF, MFDAT, DATORG) de especial interesse para o sistema, passando de módulo a módulo (Figura 45).

Arquivo Work

Os resultados de cálculo de cada módulo são escritos em um arquivo de nome WORK que é lido pelo módulo seguinte. Com isto, para fins de teste, pode-se começar o processamento por um módulo intermediário, tendo o cuidado de tomar o arquivo WORK criado pelo módulo anterior a ele.

O arquivo WORK pode ser listado com auxílio do módulo LISTEN, se no programa de peça se inserir o comando: PRINT/12, ON, número do módulo o onde número do módulo é encontrado em "Record Beschreibung zum EXAPT Modul Processor".

[8], assim como a descrição da construção do arquivo WORK.

De outro lado os resultados intermediários dos diversos módulos podem ser impressos no arquivo PRINT, criado pelo processador, se se atribuir valor lógico .TRUE. As variáveis lógicas que, em cada módulo, controlam os comandos WRITE.

O arquivo PRINT pode ter saída em um dos periféricos disponíveis, como: impressora de linhas (\$LPT), vídeo da console (\$TTO), no COSCOM (QTY:Ø), etc., ligando o arquivo PRINT a um destes arquivos periféricos.

ANEXO 5

PROGRAMA PLOT E EXEMPLOS DE PEÇA FRESADA E
USINADA, PROGRAMADAS NO SISTEMA EXAPT IMPLI-
MENTADA NO SISCO MB-8000

O programa PLOT foi construído no GRUCON, com objetivo de se ter um instrumento de teste dos programas EXAPT novos, além do auto-treinamento necessário para geração de pós-processadores. Sua configuração estrutural está exposta na figura 46.

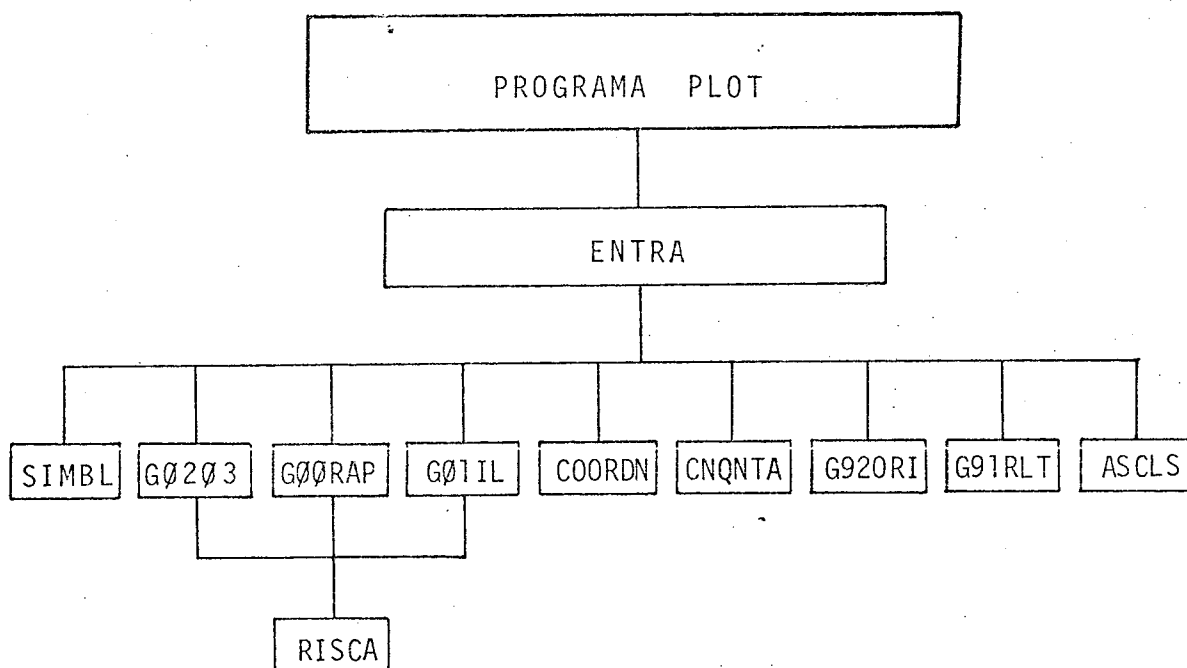


FIGURA 46 - Esquema da estrutura dos módulos componentes do Programa PLOT.

O programa auxiliar PLOT obedece a mesma filosofia de todo o sistema EXAPT: tem forma modularizada.

As funções executadas por cada módulo estão esquematizadas na tabela 5:

FUNÇÃO	MÓDULO RESPONSÁVEL
Administração do fluxo do processamento	Master & ENTRA
Interpretação - Circular horária/anti-horária - Movimento retilíneos rápidos - Interpolação linear em avanço	G0203 GOORAP G01IL
Leitura e interpretação dos dados de entrada	Master+ASCLS+ENTRA+SIMBL
Montagem dos números lidos	COORDN
Correções dos números lidos (obtenção em coord. absolutas)	G91RLT+G92ORI+CNQNTA
Controle do "plotter" (traçado)	RISCA

Tabela 5 - Funções e módulos do programa auxiliar PLOT.

O trabalho do programa PLOT é a repetição das fases descritas na tabela acima, para cada um dos registros lidos.

Partindo da peça em fita perfurada é feita sua leitura, registro por registro. Cada registro lido é compactado, eliminando caracteres inúteis (brancos) e interpretado, caráter por caráter.

A seguir faz-se a construção dos resultados, isto é, a montagem dos números e das funções, que são plotadas.

Existem subrotinas específicas para executar interpolação linear em movimento rápido (G00RAP), gerando linhas tracejadas no "plotter", em avanço (G01IL), gerando linhas contínuas e interpolação circular (G0203), uma vez que o "software" do "plotter" só executa a função de levar a pena em linha reta baixada (traçando) ou levantada (não traçando) a um ponto definido por suas coordenadas cartesianas.

Um exemplo de gráfico produzido a partir de um programa de peça é ilustrado a seguir (Figura 47).

```

( EXEMPLO 90 PARA PLOTAR )
N05 G00 G17 G40 G53 G60 G90 Y150000 Z600000 M05
N10 T1 M06
N15 X17000 Y17000 Z192000
N20 X49000 Y40000
N25 Y00000
N30 X57660 Y35000
N35 Y45000
N40 X49000 Y60000
N45 X91000 Y63000
N50 X79000 Y40000
N55 X98000 Y000
N60 Y65000
N65 X00000 Y80000
N70 X000
N75 Y20000
N80 X20000 Y000
N85 X98000
N90 Z197000
N95 X000 Z192000
N100 Z374000
N105 G00 G53 Y150000 Z600000
N110 T99 M06
N115 M30

```

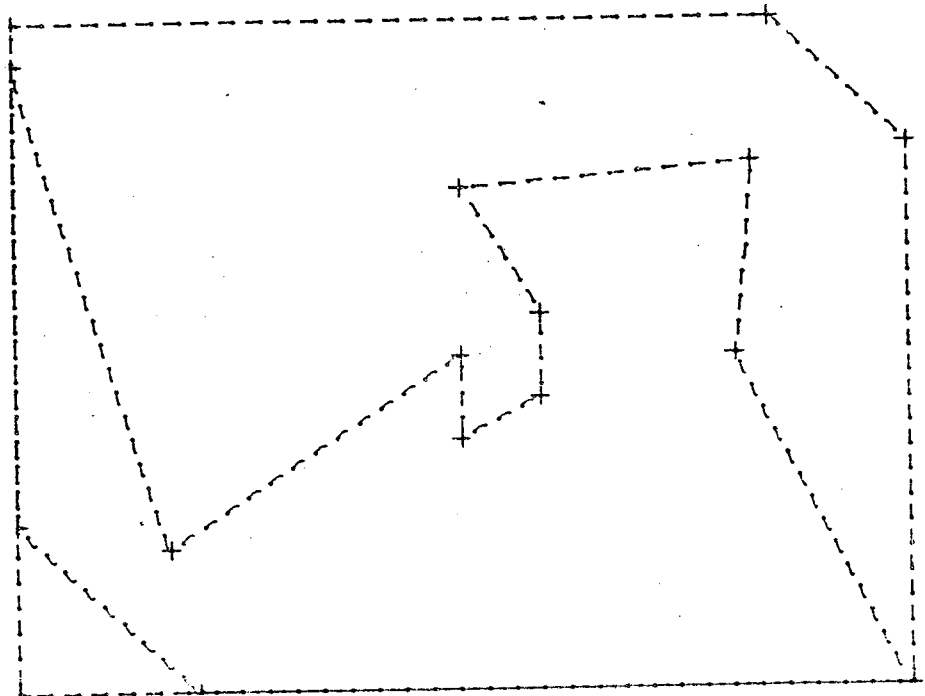


FIGURA 47 - Exemplo de programa de peça com posicionamento e seu traçado no "plotter" com o programa PLOT .

Fornece-se a seguir o programa EXAPT para torneamento da peça HASTE ISO-40, suporte de ferramentas para a árvore do centro de usinagem NBH-65, processada e pós-processada com o sistema EXAPT implementado no SISCO MB-8000. Ilustra-se também o Plano de Preparação (além de parte do programa de peça em linguagem manual), mostrando como o programa pós-processador pode ser adaptado às condições exigidas pelo usuário, máquina e controle. Apresenta-se também o programa EXAPT da peça fresada CRUZ, juntamente com a saída do pós-processador NBH-65.

Finalmente, a foto das peças usinadas CRUZ e HASTE ISO-40 são mostradas abaixo.

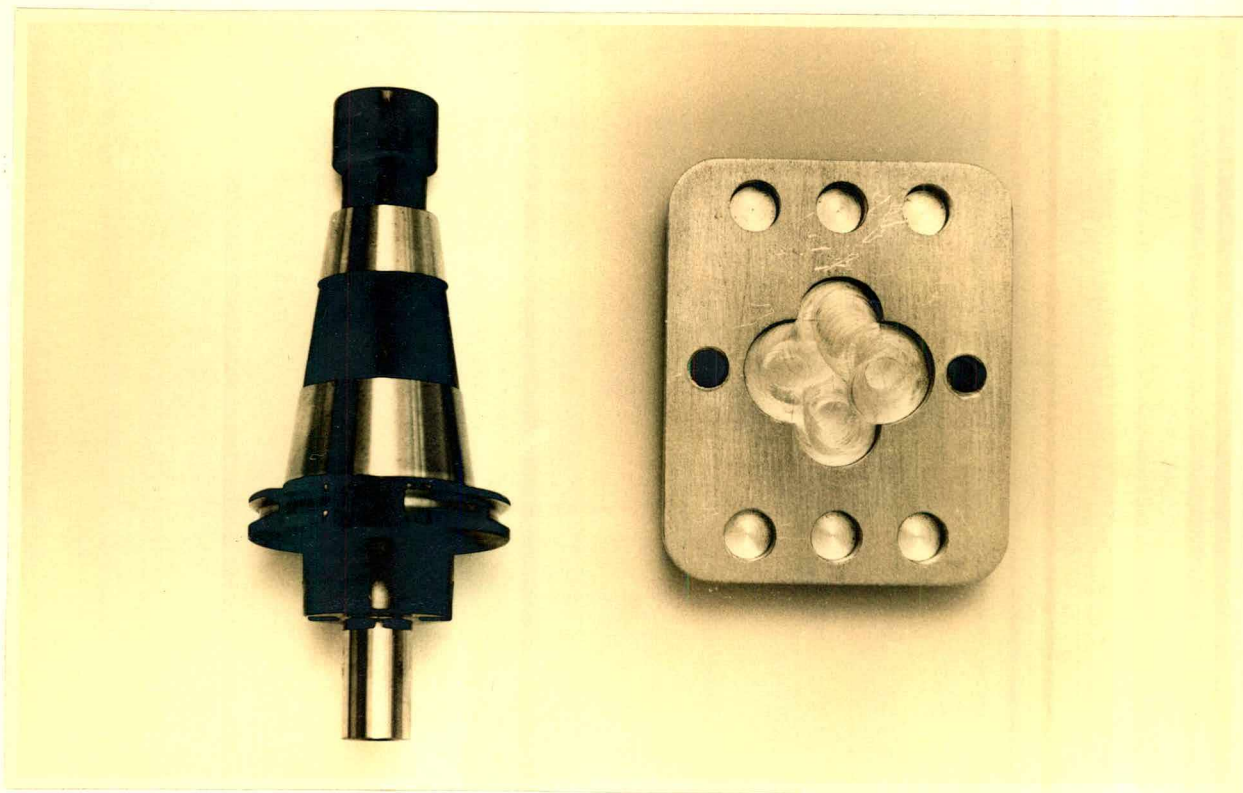


FIGURA 48 - Foto das peças HASTE ISO-40 e CRUZ, usinadas com programas obtidos da implementação no SISCO MB-8000.

```

PARTNO/F1023 HASTE ISO-40, NBH65
MACHIN/CSEPEL
CHUCK/1,48,6
CLAMP/-13
FROM/708,100
CUTLOC/MIRROR
1.1. DEFINICOES GEOMETRICAS
ANGLO=8+17780+50/3600
L1=LINE/(POINT/(94,9+9,7-28),(25/2)),ATANGL,(180-ANGLO)
L2=LINE/PARREL,L1,YLARGE,0,55
L25=LINE/PARREL,L1,YLARGE,1
L3=LINE/PARREL,L1,YSMALL,0,30
L4=LINE/(POINT/(9,7+94,9),(23/2)),ATANGL,60
P1=POINT/INTOF,L4,(LINE/0,10,10,10)
L5=LINE/P1,ATANGL,90
P2=POINT/INTOF,L5,(LINE/0,(17/2),10,(17/2))
L6=LINE/P2,ATANGL,90
PPRINT/BROCA DE CENTRO * 8 MM
NEWTL/101012,1,3,340,0,0
TOOLNO/101012,3,0SETNO,3
STAN/180
COOLNT/ON
GOTO/((100+9,7),0
SPINDL/1000,CCLW
FEDRAT/0,05
GODLTA/-14,0
RAPID
GODLTA/14,0
PPRINT/PLANAR-FRENTE E DEBASTAR
NEWTL/737003,4,0,8,(340-0,8),(-35+0,8),0
TOOLNO/737003,4,0SETNO,4
STAN/180
GOTO/((100+9,7),(70/2)
GO/T0,PLAN,(94,9+9,7)
CSPEED/100
CSRAT/70
FEDRAT/0,2
GO/ON,DIA,0
GODLTA/2,0
RAPID
GO/PAST,DIA,61
FEDRAT/0,2
GO/ON,PLAN,-1
GODLTA/1,1
RAPID
GO/PAST,PLAN,(96+9,7)
GO/T0,DIA,55
FEDRAT/0,2
GO/T0,PLAN,10
GODLTA/1,1

```

Programa EXAPT de peça HASTE ISO-40.

```

RAPID
GO/PAST,PLAN,(96+9.7)
GO/TO,DIA,50
FEDRAT/0.2
GO/TO,PLAN,10
GODLTA/1,1
RAPID
GO/PAST,PLAN,(96+9.7)
GO/TO,DIA,46
FEDRAT/0.2
GO/TO,PLAN,10
GODLTA/1,1
RAPID
GO/PAST,PLAN,(96+9.7)
GO/TO,DIA,39
FEDRAT/0.2
GORGT/DIA,39,TO,L25
GODLTA/1,1
RAPID
GO/PAST,PLAN,(96+9.7)
GO/TO,DIA,33
FEDRAT/0.2
GORGT/DIA,33,TO,L25
GODLTA/1,1
RAPID
GO/PAST,PLAN,(96+9.7)
GO/TO,DIA,26
FEDRAT/0.2
GORGT/DIA,26,TO,L25
GORGT/L25,TO,PLAN,10
GODLTA/1,1
RAPID
GO/PAST,PLAN,120
PPRINT/SANGRIA DA PONTA DA PECA 12 MM
NEWTL/000001,4,0,425,-40
TOOLNO/000001,5,0SETNO,5
STAN/180
GOTO/(9.7+94.9+5),(32/2)
CSPEED/17
GO/ON,PLAN,(9.7+94.9-16)
FEDRAT/0.1
GO/TO,DIA,20
RAPID
GO/PAST,DIA,30
GODLTA/-3.5,0
FEDRAT/0.1
GO/TO,DIA,20
RAPID
GO/PAST,DIA,30
GODLTA/-3.5,0
FEDRAT/0.1
GO/TO,DIA,20

```



```

RAPID
GO/PAST,DIA,30
GO/PAST,PLAN,120
PPRINT/FINALIZAR E SANGRAK 0.2 MM
NEWTL/737018,4,0.4,(340-0.4),(-55+0.4)
TOOLNO/737018,6,0SETNO,2
CSPEED/100
CSRAT/70
STAN/180
GOTO/(9.7+96),15
FEDRAT/0.1
GO/TO,DIA,20,TO,PLAN,(94.9+9.7)
GORG/PLAN,(94.9+9.7),ROUND,1
GOLFT/DIA,25,BEVEL,0.5
GOLFT/PLAN,(94.9+9.7-18.5)
GORG/DIA,21
GORG/PLAN,(94.9+9.7-28),ROUND,1.8
GOLFT/L2,PAST,PLAN,(9.7+94.9-45)
GOLFT/PLAN,(9.7+94.9-45)
GORG/L3
GORG/PLAN,(9.7+94.9-70),BEVEL,0.5
GOLFT/L2
GOLFT/DIA,44.45
GORG/PLAN,9.7,ROUND,1
GOLFT/DIA,60,PAST,PLAN,-1
GOLTA/1,1
RAPID
GO/PAST,PLAN,120
PPRINT/ABRIR RANHURA
NEWTL/109013,4,0,340,-30
TOOLNO/109013,7,0SETNO,7
CSPEED/21
CSRAT/70
STAN/180
GOTO/(9.7+96),(62/2)
GO/ON,PLAN,(4.7+2.11/2)
FEDRAT/0.1
GO/TO,DIA,55.5
GO/TO,PLAN,(4.7+0.9)
RAPID
GO/PAST,DIA,62
GO/PAST,PLAN,120
PPRINT/FAZER BURACO
NEWTL/000002,1,14,440
TOOLNO/000002,8,0SETNO,8
STAN/180
GOTO/107,0
SPINDL/250,CLW
FEDRAT/0.15
GO/TO,PLAN,(9.7+94.9-48.5)

```

```

RAPID
GO/PAST,PLAN,120
PPRINT/ACABAR BURACO
NEWIL/730002,4,0.8,(339.5-0.8),(16.2-0.8)
TOOLNO/730002,9,0SETNO,9
CSPEED/50
CSRAT/70
STAN/180
GOTO/(9.7+94.9+5),(24/2)
FEDRAT/0.1
INDIRV/-1,0
GO/T0,L4
GOLFT/L4
GORGTL5,PAST,L4
GODLTA/1,-1
RAPID
GO/PAST,PLAN,(9.7+94.9+5)
FINI
    
```

POS-PROCESSADOR E X A P T - GENPP PARA MAQUINAS DE TORNEAR
 F1023 HASTE ISO-40, NBH65
 CSEPEL NC BOSCH NC 802
 FIXACAO 1

11/12/81 15:39:51
 PAGINA 4

N	G	G	X	Z	I	K	F	S	T	M	M	M
116	40							10	07000	09		
117				306400								
118			70000									
119									08000	06		
120	92		100000	206400								
121	00	43		107000				10	08008	03	08	
122			0									
123								12				
124	01		0	56100			3750					
125	00			120000								
ACABAR BURACO												
126	40							10	08000	09		
127				206400								
128			100000									
129									09100	06		
130	92		114600	306900								
131	43							10	09109	03	08	
132	00			108800				15				
133			11200									
134	01		11200	105012			7100					
135			8614	103520								
136			6314	99536								
137			5314	100536			10000	16				
138	00			109600								
139	40											
140				306900				10	09100	09		
141			114600									
142												
143									03100	06		
144										05		
										30		

Parte da saída do pós-processador CSEPEL para a peça HASTE ISO-40

PLANO DE PREPARACAO

*F1023 HASTE ISO-40, NBH65
 *FIXACAO 1
 *MAQUINA CSEPEL NC PESO BRUTO
 *CONTROLE BOSCH NC 802 MATERIAL
 *MAQUINA NR. ERI250 PECA BRUTA
 PAGINA 5
 DATA
 NOME

 * Z X * DESENHO DA FIXACAO DA PECA
 *PT. DE REFERENCIA. -21.120 . 709.000 *
 *PT. DE INICIO . 100.000 . 708.000 *
 *AVANCO . 100.000% *
 *REFRIGERANTE . SIM *

 *PLACA *
 *NUMERO DA PLACA . 1.000 *
 *DIAMETRO . 0.000 *
 *COMPR. CASTANHA . 0.000 *

LISTA DAS FERRAMENTAS UTILIZADAS

FERRAMENTA	FERR. -NR.	COMANDO DE TROCA	CORR.	LUGAR NO. MAGAZINE.	PRE-AJUSTE L	PRE-AJUSTE Q
*FER. P/ FUR..	101012.	3100.	3.	3.	340.00	0.00
*FER. P/ TOR..	737003.	4000.	4.	4.	340.00	35.00
*FER. P/ TOR..	1.	5100.	5.	5.	425.00	40.00
*FER. P/ TOR..	737018.	6100.	2.	6.	340.00	55.00
*FER. P/ TOR..	109013.	7000.	7.	7.	340.00	30.00
*FER. P/ FUR..	2.	8000.	8.	8.	440.00	0.00
*FER. P/ TOR..	730002.	9100.	9.	9.	339.50	14.60

OBSERVACOES

NR. DA SENTENCA	CHAVE DE CORRECAO	X	Z	OBSERVACOES
3	3			
14	4			
60	5			
77	2			
107	7			
121	8			
131	9			

* TEMPO TOTAL = 10.45'
 * TEMPO DE CORTE = 9.51'
 * TEMPO IMPRODUTIVO = 0.94'

FIGURA 49 - Saída do pós-processador CSEPEL (peça HASTE ISO-40).

```

PARTNO/CRUZ
REMARK/*** DEFINICAO DOS PONTOS ***
ZSURF / 18
P1 = POINT/((98-76)/2),((80-40)/2)
PCENTR= POINT/(98/2),(80/2)
PSOBCR= POINT/PCENTR,DELTA,0,0,-10
ESCONL= PATERN/LINEAR,P1,ATANGL,90,INCR,2,AT,(40-((80-40)/2))
LSIMET= LINE /((98/2),0,(98/2),1
DICONL= PATERN/MIRROR,LSIMET,ESCONL
TODCON= PATERN/RANDOM,ESCONL,DICONL
REMARK/*** CHAMADA DO POS-PROCESSADOR DA MAQUINA ***
MACHIN/ NBH65
TRANS / 274,170,78
REMARK/*** DEFINICAO DAS FERRAMENTAS ***
NEWTL / 204041,5,11.7,190,0,0
NEWTL / 204043,5,16,195,0,0
$$$** DEFINICAO DA SUBROTINA -CICLO-***
CICLO = MACRO/PROFUN=13
FEDRAT/100,PERMIN
GODLTA/--PROFUN
RAPID
GODLTA/PROFUN
TERMAC
PPFUN /PALA,ON
ROTABL/180
COOLNT/ 1
TOOLNO/204041,3
SPINDL/4000
I=0
PROXIM)I=I+1
PPRINT/*** NOVO PONTO DE -TODCON- ***
GOTO /TODCON,RETAIN,I
CALL /CICLO
IF(6-I)PRONTO,PRONTO,PROXIM
PRONTO)PPRINT/*** PREP. DA ENTRADA NO REBAIXO ***
GOTO /PCENTR
CALL /CICLO,PROFUN=10

```

Programa EXAPT de peça fresada CRUZ

```

$$$** DEFINICAO DO REBAIXO ***
C1 =CIRCLE/49,50,12
C2 =CIRCLE/61,40,10
PC3 =POINT /49,30
C3 =CIRCLE/CENTER,PC3,RADIUS,12
C4 =CIRCLE/37,40,10
PPRINT/*** EXECUCAO DO REBAIXO ***
TOOLNO/204043,4
CSPEED/200,CCLW
GOTO /PCENTR
FEDRAT/100,PERMIN
GOTO /PSOBCR
GODLTA/0,10,0
GODLTA/0,-20,0
GOTO /PSOBCR
GODLTA/-12,0,0
GO /PAST,C2
GOLFT /C2,2,INTOF,C3,ROUND,0
GOLFT /C3,ROUND,0
GOLFT /C4,ROUND,0
GOLFT /C1,ON,XPAR,50
RAPID
GOTO /PCENTR
$$$** DEFINICOES DO CONTORNO EXTERIOR ***
L3 =LINE/0,0,1,0
L5 =LINE/0,0,0,1
L6 =LINE/0,80,1,80
L7 =LINE/98,0,98,1
PPRINT/*** EXECUCAO DO CONTORNO EXTERIOR ***
GOTO /-15,-15,18
GOTO /0
FEDRAT/100,PERMIN
GO /T0,L3,T0,L5
GORGT /L3,PAST,L7,ROUND,12
GOLFT /L7,ROUND,12
GOLFT /L6,ROUND,12
GOLFT /L5,ROUND,12
GOLFT /L3,ON,YPAR,13
RAPID
GODLTA/0,-5,18
COOLNT/OFF
PPFUN/PALA,OFF
$$$===CLPRNT $$$** DECLARACAO PARA IMPRIMIR A SAIDA DO PROCESSADOR.
$$$===PRINT /3,ALL
FINI

```

```

%
( CRUZ )
N05 G00 G17 G40 G50 G60 G90 Y150000 Z600000 M05
N10 G00 G50 X000
N15 M21
N20 B180000
N25 G00 G50 X450000 M00
N30 T3 M00
( NOVO PONTO DE -TODCON- )
N35 X187000 Y190000 Z286000 S4000 M03
N40 G01 Z273000 F100
N45 G00 Z286000
( NOVO PONTO DE -TODCON- )
N50 Y210000
N55 G01 Z273000
N60 G00 Z286000
( NOVO PONTO DE -TODCON- )
N65 Y230000
N70 G01 Z273000
N75 G00 Z286000
( NOVO PONTO DE -TODCON- )
N80 X263000 Y190000
N85 G01 Z273000
N90 G00 Z286000
( NOVO PONTO DE -TODCON- )
N95 Y210000
N100 G01 Z273000
N105 G00 Z286000
( NOVO PONTO DE -TODCON- )
N110 Y230000
N115 G01 Z273000
N120 G00 Z286000
( PREP DA ENTRADA NO REBAIXO )
N125 X225000 Y210000
N130 G01 Z276000
N135 G00 Z286000

```

Programa de saída do pós-processador de peça CRUZ

(EXEC DO REBAIXO)

N140 G00 G53 X450000 Y150000 Z600000 M05
 N145 T4 M06
 N150 X225000 Y210000 Z291000 G3550 M03
 N155 G01 Z201000
 N160 Y220000
 N165 Y200000
 N170 Y210000
 N175 X210000
 N180 X235000
 N185 G02 X237000 Y212000 I2000 J000
 N190 X239000 Y210000 I000 J-2000
 N195 X237000 Y208000 I-2000 J000
 N200 G03 X229000 Y200000 I000 J-8000
 N205 G02 X225000 Y196000 I-4000 J000
 N210 X221000 Y200000 I000 J4000
 N215 G03 X213000 Y208000 I-8000 J000
 N220 G02 X211000 Y210000 I000 J2000
 N225 X213000 Y212000 I2000 J000
 N230 G03 X221000 Y220000 I000 J8000
 N235 G02 X225000 Y224000 I4000 J000
 N240 X229000 Y220000 I000 J-4000
 N245 G00 X225000 Y210000 Z291000
 X EXECUCAO DO CONTORNO EXTERIOR
 N250 X161000 Y155000
 N255 Z273000
 N260 G01 X168000 Y162000
 N265 X262000
 N270 G03 X282000 Y182000 I000 J20000
 N275 G01 Y238000
 N280 G03 X262000 Y258000 I-20000 J000
 N285 G01 X188000
 N290 G03 X168000 Y238000 I000 J-20000
 N295 G01 Y182000
 N300 G03 X188000 Y162000 I20000 J000
 N305 G01 X189000
 N310 G00 Y157000 Z291000
 N315 G00 G53 Y150000 Z600000 M05 M09
 N320 G00 G53 X450000
 N325 G000
 N330 T99 M06
 N335 G00 G53 X000
 N340 M02
 N345 G00 G53 X450000
 N350 M30

ANEXO 6

A HISTÓRIA DO COMANDO NUMÉRICO E O MÉTODO DE
FABRICAÇÃO CN.

a.1. Desenvolvimento no Exterior

A história do CN é relativamente nova, se for comparada sua idade com o alto grau de desenvolvimento que logrou alcançar. Aliás tal fato tem se repetido com todas as áreas que utilizam eletrônica e cibernética como ferramentas.

Provavelmente o primeiro passo histórico na direção do CN tenha sido dado por Joseph Jacquard, que, com cartões perfurados como meio de controle, conseguiu produzir diversos padrões florais em seu tear mecânico, em 1801, França [25]. O fato de colocar as informações do controle de um processo em um meio externo à mente do controlador é muito importante, e a esta idéia se deve muito do grau de automatização que muitos processos hoje alcançaram em diversos ramos, não são os computacionais.

O passo realmente pioneiro foi sem dúvida dado na indústria aeronáutica americana.

Um empresa produtora de pás para rotores de helicópteros, nos Estados Unidos tinha o seguinte problema para cada novo tipo de pá a ser construído [25]:

- Calcular matematicamente o perfil da pá.
- Produzir mais de 50 modelos de perfis para servirem de moldes à manufatura e à inspeção.
- Operar manualmente a máquina para produzir os moldes dos perfis e as peças.
- Dar acabamento manual final à peça.

Em 1949 esta empresa fez uma proposta à Força Aérea Americana para utilização de rotinas de computador para controlar a máquina que moveria sua ferramenta em incrementos tão pequenos ao longo do percurso desejado, que geraria em sua trajetória o contorno desejado para a peça.

Foi feito um contrato para realizar os trabalhos, tendo participação bilateral empresa-governo. Mais tarde este trabalho foi encampado pelo Massachusetts Institute of Technology que em 1952 completou o protótipo de uma fresadora CN.

Em lugar de um apalpador que seguia o perfil de um modelo, colocou-se um sistema CN que informava um sistema de servomecanismos, que por sua vez comandava a máquina.

Os movimentos dos três eixos comandados eram registrados e dados como realimentação ("feed-back") ao controle CN.

O sistema gerou grande interesse, principalmente na indústria aeronáutica e em 1957 teve lugar a primeira instalação industrial.

Nos dez primeiros anos de sua história, somente a indústria aeronáutica dos Estados Unidos, grandemente encorajada por sua Força Aérea, trabalhou com CN.

Apenas por volta de 1960 o processo começou a ser empregado em uma escala mais comercial embora produzindo apenas máquinas de controle ponto a ponto.

Os anos sessenta marcaram também o início da aceleração da produção de máquinas com controle contínuo. (Ver item b.2.1.). Em 1960 apareceu nos EUA (Massachusetts Institute of Technology - MIT) a primeira versão da linguagem APT, que é o tronco de onde partem várias linguagens dela derivadas, formando a família APT (ver item 2.1).

Na Europa, o desenvolvimento no campo do CN não foi devido, em suas origens, especificamente à indústria aeronáutica, mas à indústria de um modo genérico. A direção mais perseguida em sua pesquisa foi sobretudo a da introdução de tecnologia já na altura da programação das peças, isto é, a do desenvolvimento de linguagens de programação com poder descritivo geométrico e tecnológico [26].

O número de máquinas instaladas nos EUA e na RFA ano por ano é mostrado na figura 50 [29].

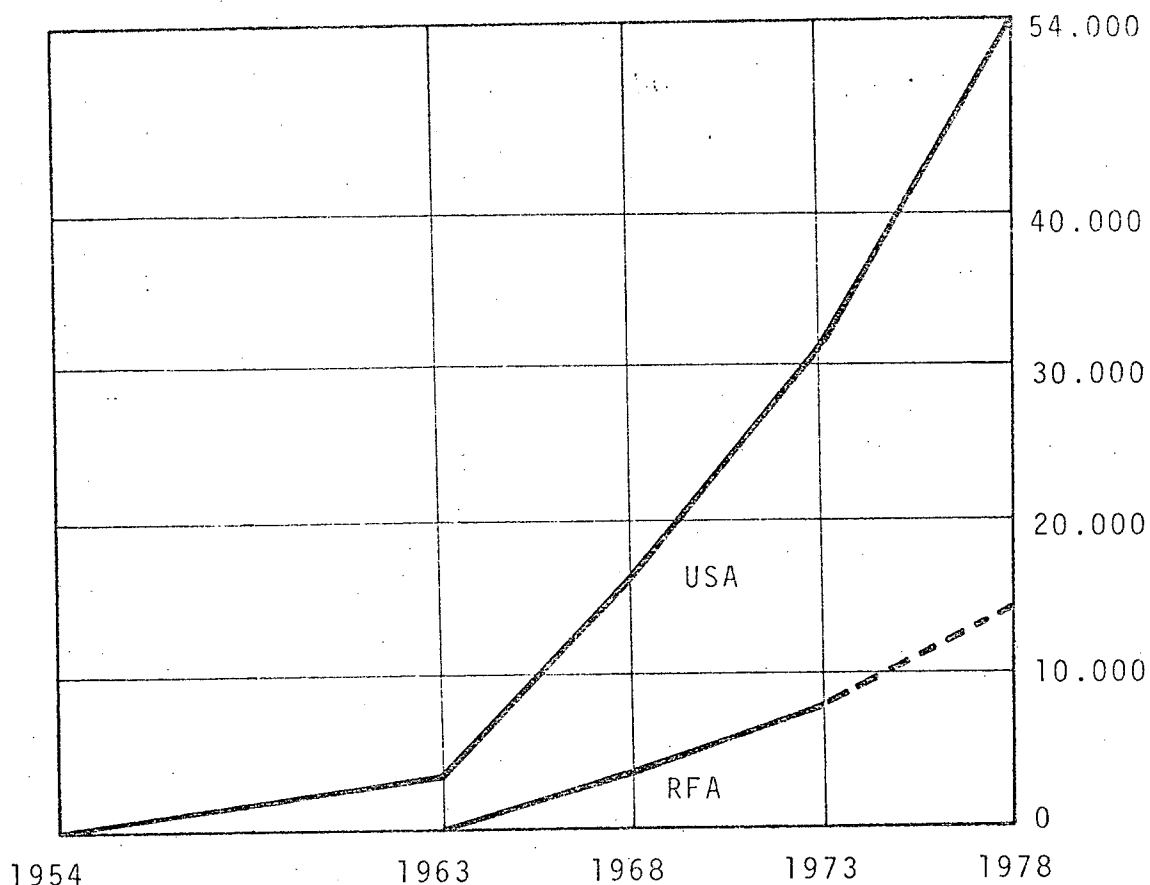


FIGURA 50 - Máquinas CN instaladas nos USA e na RFA entre os anos de 1954 e 1978.

a.2. O Desenvolvimento no Brasil

O CN começou no Brasil, tanto em nível industrial como acadêmico, no final da década de 60 e início da de 70.

As primeiras máquinas foram instaladas em indústrias filiais de empresas americanas.

Em 1969 o Curso de Pós-Graduação em Engenharia Mecânica da UFSC começou a oferecer a disciplina Comando Numérico com 45 horas-aula.

Conforme dados obtidos de pesquisa efetuada pela Sociedade Brasileira de Comando Numérico (SOBRACON) através de questionários enviados às empresas, com os quais se espera ter atingido mais de 80% de todas as indústrias possuidoras de máquina CN, e ainda de informações de outras fontes, como CACEX, pode-

-se estimar em 650 o número de máquinas CN instaladas no Brasil.

O número de instalações ano a ano, a partir de 1976 é:

1976	53	1978	85	1980	46
1977	63	1979	63		

Entre os usuários que já trabalham com programação por computador pode-se notar no Brasil uma dicotomia bem pronunciada:

- Usuários de grandes computadores, com sistemas gerais de programação
- Usuários de pequenos computadores, com sistemas de programação CN particularizados.

A tabela 6 [17] ilustra esta asserção.

Três empresas iniciaram no Brasil a produção de controles para MFCN (Diadur, Numericon/Digicon, Siemens) e atualmente nove indústrias fabricam tais máquinas (Doerries, Index, Itabras, Nardini, Pittler, Romi, Thyssen-Hueller, Traubomatic, Wotan).

A SOBACON foi criada em 24 de março de 1980 para reunir os esforços dispendidos neste campo. É constituída de pessoas físicas e jurídicas interessadas em CN bem como em projeto e fabricação com apoio de computador.

Os objetivos principais desta sociedade são:

- I - Desenvolvimento e divulgação de conhecimentos.
- II - Intercâmbio de experiências.
- III - Contribuição para a normalização nacional e internacional.
- IV - Relacionamento com outras instituições similares e órgãos do governo.

Fundada com doze sócios-empresas, oito sócios-instituições e doze sócios-individuais, a SOBACON conta hoje com: mais de cem sócios-individuais, quase duas dezenas de empre-

SISTEMA DE PROGRAMAÇÃO	USUÁRIO	COMPUTADOR (*)	OBSERVAÇÃO
ADAPT	CATERPILLAR EMBRAER	IBM 370 (G)	
APT	CSN CTA EMBRAER	IBM 370 (G)	
EASYPROG	ENGEX	PDP 8 (P)	Torneamento
ELAN	EBERLE ERICSSON	HP 9825 (P)	Eletro-erosão
EXAPT	BROWN BOVERI SIEMENS UFSC VW MERCEDES	IBM 4341 (G) R. 20 (M) IBM 4341 (G) HP 1000 (M) SISCO MB 8000 (M) IBM 370 (G) PDP 11-34 (M) IBM 3032 (G)	
H-100 H-400	INDEX MAQUINASA INDEX	PDP 11-03 (P)	Torneamento Fresagem
OLIVETTI GTL-3 GTL-T	BOSCH BRASEIXOS COBRASMA IPT JUNTAS FLEXA MOLLINS OLIVETTI POSITRON SENAI	P 652 (P) P 6060 (P) P 6060 P 6040 (P) P 6060 (P)	
SAPT	VIBASA	CANON (M)	
SRG	IPT	HP 9830 (P)	
T-APT	ERICSSON THYSSEN	HP 9825 (P) PDP 11-03 (P)	Eletro-erosão

(*) G = Grande

M = Médio

P = Pequeno

TABELA.6 - Programação por computador. Sistemas instalados no Brasil.

sas filiadas e mais de um dezena de instituições de ensino.

Espera-se que esta sociedade seja um marco im-
portante do desenvolvimento nacional na esfera do CN.

a.3. Desenvolvimento na Universidade Federal de Santa Catarina

No ano de 1977, foi implementado o primeiro processador EXAPT e pós-processador para o torno revólver CSEPEL DECATHLON ERI-250, com controle numérico BOSCH NC 802, no computador IBM 360/40 do Núcleo de Processamento de Dados da UFSC.

Este trabalho foi executado e vem sendo desenvolvido dentro do âmbito do convênio de cooperação científica que a UFSC mantém com a RWTH, de Aachen.

Alguns professores conveniados trabalharam na construção de sistemas de programação de CN por computador e se dispõem a transferir este "know-how", criando assim condições excepcionais para a UFSC absorver conhecimentos técnico-científicos do método de construção de sistemas de PAC.

Mais tarde o sistema EXAPT foi transferido para o computador Hewlett-Packard HP 1000 e foi ampliado com a implementação de pós-processador para o centro de usinagem NBH-65 da Thyssen Hueller. Hoje o sistema EXAPT está implementado no computador IBM 4341 da UFSC e, com o presente trabalho, no SISCO MB-8000.

A UFSC conta com um grupo de trabalho denominado: Grupo de Pesquisa e Treinamento em Comando Numérico (GRUCON), constituído de Professores e alunos da área de Engenharia Mecânica e Computação.

Este grupo tem dispendido grande esforço na absorção do "know-how" do pessoal proveniente do Convênio UFSC-RWTH, na elaboração de material de consulta e cursos de treinamento.

b. O Método de Fabricação CN

b.1. Introdução

No processo convencional de fabricação, de posse do projeto da peça, caracterizada pelo seu desenho cotado, material a ser usado e outras informações, o setor de produção deve preparar:

- Uma ficha de produção detalhando cada operação necessária para executar a peça.

- Requisições das ferramentas e, quando for o caso, de dispositivos e ferramentas especiais necessárias para as fixações, usinagens, etc.

- Desenho mostrando as operações de máquina necessárias, as áreas a serem usinadas, velocidades de avanço, de rotação e cortes para cada operação.

Mas, a partir da introdução do conceito de CN, entra em jogo um novo personagem: o programa, isto é uma seqüência de instruções simbólicas de controle da máquina, para execução de um determinado trabalho.

Então, em lugar de requisições, desenhos, fichas escritas, usadas no processo convencional, o encarregado da produção escreve um programa que contém estas mesmas informações em linguagem simbólica.

O Comando Numérico é um método de monitorar -acompanhar e influir - através de números, letras e outros símbolos, um certo processo. A programação é através de símbolos e o objetivo - o controle - é conseguido através de informações armazenadas em um portador de dados.

A filosofia do Controle Numérico não se restringe às máquinas-ferramentas, mas encampa o planejamento, a produção, bem como os métodos e processos.

A transferência das instruções necessárias ao gabinete de controle para governar a máquina, pode ser manual ou automaticamente (através de um portador intermediário como fita

perfurada, fita magnética, cartões perfurados, etc.) ou outros aparelhos periféricos, podendo ser mesmo o próprio computador que estaria então diretamente ligado à MFCN.

Todo o sistema de produção, desde o planejamento do produto à peça acabada, pode ser visto como uma série de passos que incluem preparação, manuseio e processamento de informações. Este fato implica que no processo convencional de produção muitos passos de trabalho repetitivos e demorados devem ser executados pelo homem. A figura 51 ilustra os vários passos que sofre a informação no processo convencional, no CN e num processo totalmente automatizado, ainda inexistente.

O desenvolvimento tende a fornecer máquinas que podem processar dados, como as máquinas CN e os computadores. Estas máquinas vão progressivamente sendo adaptadas à realização de todo o processo de produção, tendendo a torná-lo totalmente automático.

b.2. O Sistema Máquina-Ferramenta/Gabinete de Controle CN

A MFCN é um sistema formado pelos seguintes elementos:

- Gabinete de Controle
- Máquina Ferramenta
- Sistema de Medida
- Sistema de Realimentação ("Feed-back")

A figura 52 ilustra esquematicamente um sistema MFCN com seus componentes principais interligados.

b.2.1. Sistemas de Controle

A filosofia da fabricação por Comando Numérico evoluiu, dos sistemas iniciais, onde o gabinete de controle praticamente só transmitia à máquina os dados de controle pré-inseridos no portador de dados, aos sistemas modernos, onde um computador controla direta e simultaneamente diversas máquinas.

No início da história das MFCN cada controle

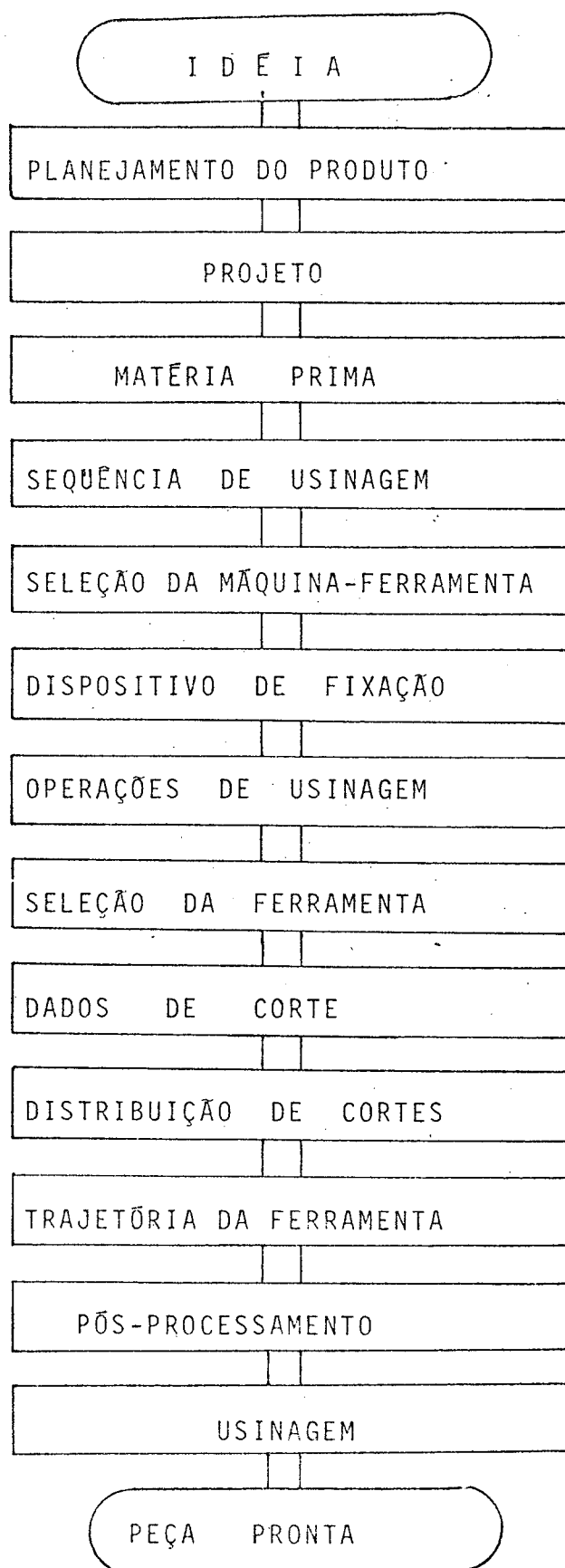


FIGURA 51 Processamento integrado de informações na produção.

era construído especificamente para determinada máquina-ferramenta. O "hardware" era planejado de modo particular. A cada peça produzida a fita perfurada, se fosse o caso, teria que retornar ao início do programa para a usinagem da peça seguinte, porque o programa não era armazenado pelo controle. Hoje a grande maioria das MFCN são dotadas do chamado CNC: Comando Numérico Computorizado. Nestes controles, o trabalho é executado por mini-computador, fato que traz benefícios tanto para o construtor da máquina, quanto para o do controle.

A modularidade é uma das principais vantagens, já que com o mesmo "hardware" básico pode-se fazer controles para diferentes máquinas (fresas, tornos, etc.) ou para um mesmo tipo de máquina com características diferentes (número de eixos controlados, dimensões, etc.). As adaptações são exclusivamente de "software", o que é mais barato, muito mais flexível e de menor risco.

A modularidade permite a expansão também das funções do sistema. Além do mais se tem uma maior taxa de utilização do equipamento, menor tempo de programação, menor índice de falhas (elimina-se leituras repetidas) e facilidade de manutenção (existem programas-diagnóstico para máquina e controle).

Os CNC estão cada vez mais baseados em microprocessadores e têm geralmente os seguintes elementos:

- Unidade central de processamento, para execução das operações aritméticas e lógicas.
- Memória para armazenamento de programa e de dados.
- Unidade de entrada e saída para interface.

O software de um CNC compreende:

1. Programa operacional.
 - Executa operações fixas e básicas.
2. Programa de interface
 - Para adaptação do CNC a uma MFCN específica.

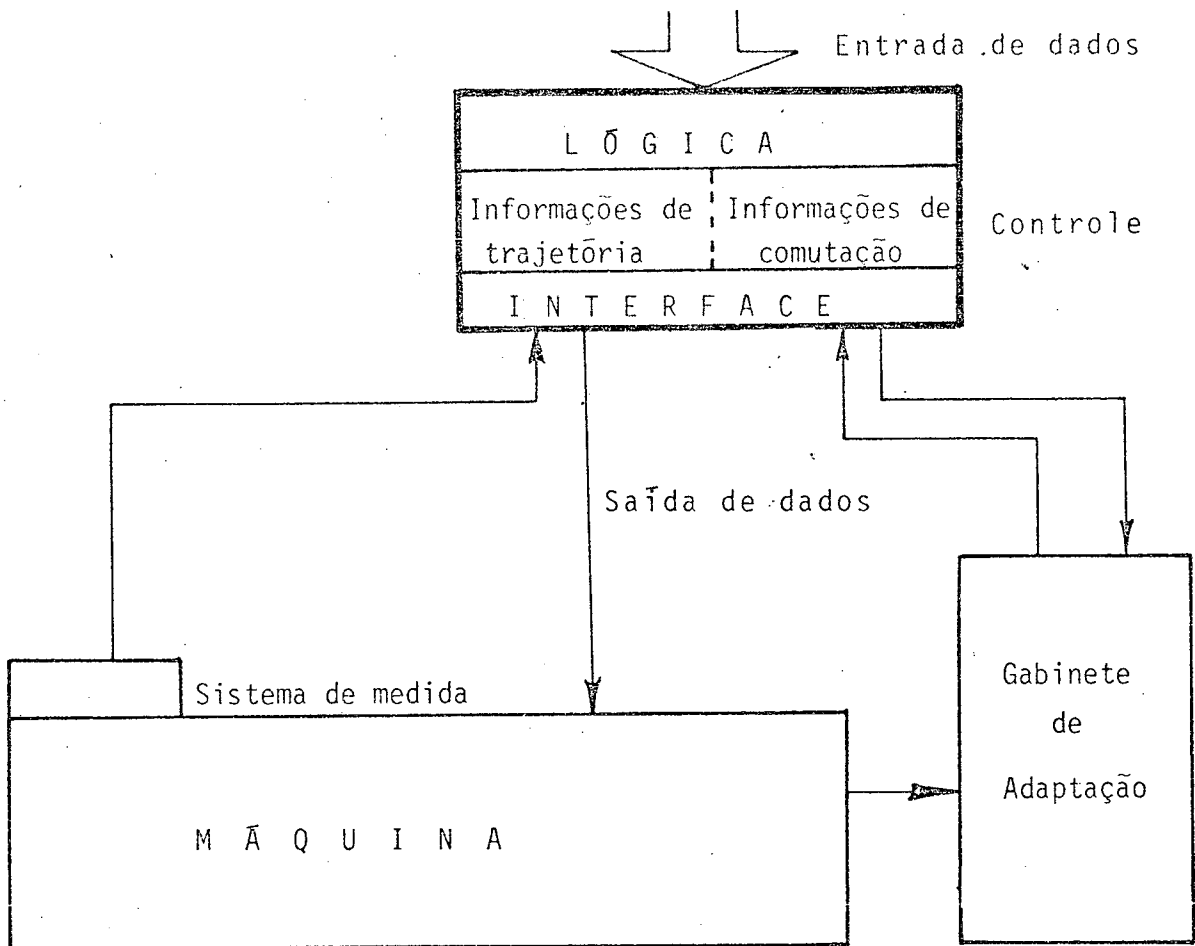


FIGURA 52 - Componentes principais de um sistema CN.

3. Programas de aplicação orientada (especiais), compostos de subrotinas e rotina de diagnóstico, podendo ser modificado para atender aplicações específicas.

Em resumo, além de ser controles industriais os CNC são comparáveis a verdadeiros mini-computadores que podem ser adaptados para diferentes tipos de MFCN, somente com trabalhos de "software", agindo sobre os programas de interface.

Por outro lado, um determinado CNC pode ser adaptado para executar operações específicas e definidas pelo usuário ou construtor da máquina, bastando para isto que se faça as transformações desejadas sobre os programas de aplicação.

Embora os controles possam servir em diferentes tipos de máquinas ferramenta, podem ser classificados em três grandes grupos básicos quanto ao tipo de controle exercido sobre a ferramenta:

- 1º - Controle de posicionamento ou controle ponto a ponto.
- 2º - Controle de corte reto.
- 3º - Controle de corte em contorno.

Nos sistemas de controle por posicionamento, a ferramenta é posicionada em um ponto, numericamente definido, e então um ciclo de operações tais como: centrar, furar, escarear e dar acabamento, é executado. As velocidades de rotação, de corte e o ciclo de operações podem ser ou não controlados. Neste método de controle não há controle ou usinagem no caminho percorrido pela ferramenta entre o ponto de partida e o de chegada. Aliás nestes controles o caminho percorrido pela ferramenta é irrelevante porque a usinagem só começa quando o ponto de chegada é atingido.

No segundo tipo de controle, de corte reto, a ferramenta pode usinar ao longo de cada eixo comandado, mas somente em linha reta, o que permite, em princípio, a usinagem de peças de faces planas ou cilindros.

Os sistemas de corte em contorno podem usinar peças com qualquer configuração, ao longo dos dois ou mais eixos comandados.

Deve-se observar que como a ferramenta se move paralelamente aos eixos. Linhas inclinadas em relação a eles, ou linhas circulares, só podem ser obtidas por interpolação linear ou circular, ou ainda parabólicos, em alguns casos. Esta interpolação consiste em subdividir o percurso retilíneo ou curvilíneo a usinar, em percursos menores, simultâneos e paralelos aos eixos. Esta interpolação pode ser programada remotamente, no portador de dados, o que daria um comprimento considerável de fita perfurada. Por isto, os controles têm "hardware" próprio, interpolador, para esta finalidade, e o portador de dados somente informa os parâmetros da linha ou do círculo.

b.2.2. Máquina-Ferramenta

Introdução

As MFCN se caracterizam principalmente por:

- Movimentos não produtivos mais rápidos que nas convencionais.
- Para que haja um aproveitamento mais eficiente do investimento inicial elevado, elas devem estar o máximo possível em produção.
- Para se ter rendimentos mais significativos, procura-se trabalhar com os motores nos limites de potência.

Estes fatores fazem com que as MFCN tenham uma concepção construtiva diversa das convencionais.

Entre as características de construção mencionam-se os seguintes aspectos relevantes:

1. Projetos robustos para suportar altas cargas, vibrações, velocidades, e para, ao mesmo tempo, manter as deformações consequentes dentro dos limites exigidos pelas peças produzidas.
2. Ter elementos de posicionamento de baixo atrito e alta precisão, mesmo com as mudanças de temperatura a que são sujeitos (os parafusos de esferas recirculares, pré-tensionados e de grande diâmetro são geralmente a solução para este problema). Também as guias devem ter atrito reduzido e ser projetadas para eliminar o fenômeno conhecido como "stick slip".
3. Sistema eficiente de remoção de cavacos, pois que estes são produzidos a altas taxas.

4. Motores de alta potência, geralmente com variação contínua da velocidade, rápidos, havendo um para cada eixo comandado.

Influência do Método CN na Ferramenta

Como na programação da MFCN o programador deve inserir parâmetros de ferramenta previamente definidos, é razoável adotar uma sistemática geral para isso. As ferramentas de uso comum têm ainda mais importância e devem ter suas definições internas ao sistema - no banco de dados pertinente - cuidadosamente feitas, para reduzir os investimentos em ferramentas e tornar mais simples e segura a programação e otimizar os parâmetros de uso.

A padronização de ferramentas permite também adaptações nos projetos de modos que, a médio prazo, se poderá ter um número limitado e mais ou menos fixo de ferramentas padronizadas e especiais.

Já existem ferramentas de construção padronizadas para MFCN e também ferramentas especiais. Assim, por exemplo, alguns construtores de tornos CN adotaram um sistema unificado de porta-ferramentas que se acha normalizado pelo VDI (VEREIN DEUTSCHER INGENIEURE) [23].

Este sistema de porta-ferramenta preenche as seguintes exigências:

- Sistema de troca rápida da ferramenta. Conseqüentemente, os porta-ferramentas não podem ser presos à máquina por parafusos, solda, etc., mas fixados por ajuste com interferência.
- Permitir preparação inicial por meios externos (ajustagem dos parâmetros iniciais).
- Permitir troca manual e/ou automática da castanha.
- Construídas com material de alta tenacidade.

- Permitir trabalho sob refrigeração intensa.

Influência do Método CN nos Motores

Devido à alta solicitação que se faz das MFCN para recuperar seu alto investimento, e das altas cargas para aumentar seu rendimento, seus elementos motores, transmissores de força, de comando, eixos, elementos de lubrificação e assim sucessivamente, tiveram que sofrer não somente um reforço mas também um redimensionamento; os parafusos de movimento, providos com esferas recirculantes, caixas de velocidade com variação contínua de velocidade (não escalonada).

As velocidades e acelerações são sempre maiores, resultando conseqüentemente em forças maiores sobre os diversos elementos que deverão ser adequadamente projetados e criteriosamente construídos para suportar sobre-cargas.

Sistema de Realimentação

O controle dos movimentos e outras ações de uma máquina-ferramenta CN pode ser feito de dois modos:

- Circuito aberto
- Circuito com realimentação ou fechado.

Circuito aberto: Neste tipo de circuito, se um dado comando é dirigido a um elemento para que este vá na posição 1 à posição 2, não existe realimentação para verificar se a posição 2 foi ou não atingida; em suma, é dado um comando mas não é feito o controle do mesmo (exemplo: motores passo a passo).

Circuito fechado: Neste tipo de circuito, existe realimentação feita pelo sistema de medição, e a diferença entre a posição real e a posição 2 é informada ao controle, que comanda novo sinal de movimento, até que o erro entre a posição real e a desejada seja inferior à tolerância admitida.

Sistema de Medida

No caso do circuito fechado, cada eixo de uma MFCN necessita de um sistema de medição que forneça sinais capazes de serem processados eletronicamente. Estes sinais são levados ao comparador, para verificação de seu valor em relação ao valor do deslocamento comandado.

Medidas de ângulos, rotações, comprimentos, etc., são feitas por sistemas de medição que podem se dividir em:

- a) Quanto ao local de medição diretos ou indiretos.
- b) Quanto ao método de detecção da medida, analógicos ou digitais.
- c) Quanto ao processo de medição, absolutos ou incrementais.
- d) Quanto ao fenômeno físico empregado para medição, indutivos ou foto-elétricos.