

1. Publicação nº <i>INPE-2476-RTR/017</i>	2. Versão	3. Data <i>Julho, 1982</i>	5. Distribuição <input type="checkbox"/> Interna <input type="checkbox"/> Externa <input checked="" type="checkbox"/> Restrita
4. Origem <i>DEE/DEL</i>	Programa <i>SISMAG/SDA</i>		
6. Palavras chaves - selecionadas pelo(s) autor(es) <i>MÁQUINA PARA PROCESSAMENTO DIGITAL</i>			
7. C.D.U.: <i>621.376.5:621.7.06</i>			
8. Título <i>ARQUITETURA DE UMA MÁQUINA PARA PROCESSAMENTO DIGITAL DE SINAIS PARA SÍNTESE E/OU ANÁLISE</i>		10. Páginas: 32	
		11. Última página: 25	
		12. Revisada por	
9. Autoria <i>Leon Lonneur</i> <i>Mario Mammoli</i> <i>Eduardo Whitaker Bergamini</i>		<i>Admir C. L. Filho</i> Admir C.L. Filho	
Assinatura responsável 		13. Autorizada por <i>Parada</i> Nelson de Jesus Parada Diretor	
14. Resumo/Notas <p><i>Este trabalho descreve a arquitetura de uma máquina capaz de realizar processamento digital de sinais para síntese e análise. A arquitetura em questão é particularmente adequada às aplicações no campo das comunicações de dados. Esta máquina é realizada com técnica de controle baseada em microprogramação ("firmware"), recurso este que permite certa flexibilidade e liberdade de aplicação. A adoção desta arquitetura aplicada à implementação dos mais importantes blocos funcionais de um modem com taxas de transmissão e recepção de 2400 e 4800 BPS, utilizando as técnicas PSK ou DPSK, é considerada neste trabalho. Finalmente apresenta-se a integração dos vários blocos funcionais, do ponto de vista do "software", de modo a constituir um dos modems do tipo citado, segundo padronização CCITT.</i></p>			
15. Observações			

ABSTRACT

This work describes a machine architecture which can execute digital signal processing for analysis and synthesis. The architecture considered is particularly adequate for applications in the field of data communications. This machine is designed with a control technique based on firmware, which permits a degree of flexibility and freedom in applications. The adoption of this architecture applied to the implementation of the most important functional blocks of a modem with transmission and reception rates of 2400 and 4800 BPS, utilizing PSK or DPSK techniques, is considered in this work. Finally, the integration of the functional blocks is shown through a flux diagram, under the software point of view, such that a modem of the mentioned type is constituted according to CCITT recommendations.

SUMÁRIO

	<u>Pág.</u>
LISTA DE FIGURAS	v
<u>CAPÍTULO 1 - INTRODUÇÃO</u>	1
<u>CAPÍTULO 2 - OS CONCEITOS BÁSICOS DE ARQUITETURA</u>	3
<u>CAPÍTULO 3 - DESCRIÇÃO DA ARQUITETURA</u>	7
<u>CAPÍTULO 4 - ESTRUTURA DA INSTRUÇÃO</u>	11
<u>CAPÍTULO 5 - VELOCIDADE DA MÁQUINA</u>	13
<u>CAPÍTULO 6 - INTEGRAÇÃO DOS BLOCOS FUNCIONAIS PARA A OBTENÇÃO DE UM MODEM DE 4800/2400 BPS</u>	19
<u>CAPÍTULO 7 - CONCLUSÃO</u>	23
REFERÊNCIAS BIBLIOGRÁFICAS	25

LISTA DE FIGURAS

	<u>Pág.</u>
3.1 - Arquitetura da máquina microprogramada	10
4.1 - Formatação da instrução	11
5.1 - Diagrama de blocos funcionais de um modem típico	14
5.2 - Diagrama de blocos de um equalizador automático	15
5.3 - Estrutura de um filtro tanque de segunda ordem	16
6.1 - Diagrama de fluxo do programa principal	20
6.2 - Diagrama de fluxo do atendimento à interrupção	21

CAPÍTULO 1

INTRODUÇÃO

Nesses últimos anos, a possibilidade de utilizar as técnicas de processamento digital de sinais no campo das comunicações de dados e sinais tem sido intensivamente investigada, com o objetivo de melhorar a confiabilidade e a relação de desempenho versus custo dos sistemas de comunicação.

A tecnologia dos semicondutores, que constitui um fator essencial na implementação física das técnicas digitais, registrou um progresso muito rápido nos últimos anos, de modo que os componentes do tipo "LSI" atuais viabilizam essas técnicas.

No entanto, a necessidade freqüente de processamento em tempo real requer não somente circuitos rápidos, mas também uma arquitetura que ofereça um alto desempenho, sem que a flexibilidade de aplicação seja prejudicada.

Esse trabalho apresenta uma solução geral, que resulta numa arquitetura baseada em componentes convencionais do tipo "LSI".

CAPÍTULO 2

OS CONCEITOS BÁSICOS DE ARQUITETURA

O espectro de aplicação de processamento digital de sinais (PDS) é muito extenso e diversificado. A execução de FFT (transformada rápida de Fourier), equalização automática e filtragem digital são exemplos típicos de aplicação. É possível projetar uma arquitetura de máquina voltada para cada aplicação, de modo que a máxima eficiência possa ser alcançada. Essa filosofia, no entanto, além de requerer um esforço considerável para uma aplicação, restringe a flexibilidade e a liberdade de uso da máquina. Além do mais é muito comum, dentro de um sistema, a necessidade de ter uma máquina única que cuida de tarefas múltiplas e diversificadas ao longo do tempo.

Na maioria das aplicações de processamento digital de sinais, sequências de operações aritméticas efetuadas numa ordem predefinida constituem a tarefa principal, enquanto operações de teste e desvio no fluxo do programa ocorrem com pequena frequência. Para se obter uma arquitetura básica, é necessário examinar as operações mais frequentes encontradas em processamento digital de sinais, ressaltando a importância da variável tempo.

Considerem-se os exemplos típicos:

1) FFT

$$\left. \begin{aligned} x_{n+1}(i) &= x_n(i) + W^q x_n(j) \\ x_{n+1}(j) &= x_n(i) - W^q x_n(j) \end{aligned} \right\} W = \exp. (-j 2\pi/N). \quad (1)$$

2) Equalizador automático

$$y_n = \sum C_i^n \cdot x_{n-i},$$

$$C_i^{n+1} = C_i^n - \alpha e_n x_n. \quad (2)$$

3) Filtro digital

$$y_n = \sum a_k x_{n-k} - \sum b_k y_{n-k}, \quad (3)$$

onde

$b_k = 0$ para filtros do tipo não recursivo;

$b_k \neq 0$ para filtros do tipo recursivo.

A operação básica que satisfaz as necessidades, no caso dos exemplos citados, pode ser escrita como segue:

$$A \times B + C \rightarrow D.$$

Modificações em torno desta operação básica permitem a realização de operações mais simples, normalmente necessárias, a saber,

- 1) Multiplicação: Faça $C = 0$, então $A \times B \rightarrow D$.
- 2) Adição: Faça $B = 1$, então $A + C \rightarrow D$.
- 3) Acúmulo: Faça $C = 0$, então $A \times B + D \rightarrow D$.

A sequência particular das diversas operações a serem executadas pode ser controlada por microprogramação ("firmware"). Uma análise dos exemplos já citados mostra também que a ocorrência de adições e subtrações simples é pequena em relação ao número de multiplicações; portanto, a arquitetura delineada a seguir tem necessidade fundamental de poder realizar, rápida e exhaustivamente, multiplicações sucessivas. Tendo em vista esses vários aspectos, um esquema paralelo de multiplicação parece ser o mais adequado.

Como já foi dito anteriormente, é importante ressaltar a necessidade de ter uma máquina única, apta a gerenciar e executar, em tempo real ou não, processos dos mais diversos tipos. Neste caso, além

da operação básica descrita, ela deve poder realizar, numa sequência ar
bitrária, outras operações ou funções tais como: transferências de da
dos internos, desvios no fluxo do programa, controle das entradas e
saídas de dados externos, e funções lógicas. Diante de tal fato, é na
tural que uma arquitetura com controle microprogramado seja uma boa op
ção para conservar a flexibilidade e a universalidade de sua aplicação.
A estrutura dos programas, assim como o circuito de relógio, são tam
bém diretamente beneficiados por uma arquitetura de máquina micropro
gramada.

CAPÍTULO 3

DESCRIÇÃO DA ARQUITETURA

Como pode ser visto na Figura 3.1, a máquina é constituída essencialmente por: um microprocessador MPROC, uma rede multiplicadora MULT, uma memória de leitura e escrita RAM, uma memória de constantes ROM, um conversor analógico/digital A/D, vários registros "buffer", um sequenciador de programa SEQ, uma memória de microcontrole MEM, um registro "pipeline" REGPL e um registro de status ST. Todas as transferências de dados dentro da máquina são executadas numa taxa de n bits/pulso de relógio, onde n é o número de bits contidos nas palavras manipuladas. Uma breve descrição desses blocos é dada a seguir:

- a) *MPROC*: o MPROC é um microprocessador convencional composto de elementos do tipo "bit-slice". Cada elemento consiste em uma memória RAM de dois portes com 16 palavras de 4 bits, numa ALU com seu circuito de deslocamento associado, e nos circuitos de decodificação e multiplexação, além de um registro acumulador. A sua palavra de microcontrole de nove bits é organizada em três grupos de três bits e seleciona a fonte dos operandos, a função da ALU e o registro de destino na ALU. Os elementos têm saídas do tipo "tristate" e fornecem saídas que indicam o status da ALU.
- b) *MULT*: a rede multiplicadora é constituída por vários elementos que permitem multiplicar números representados em complemento de dois, e produz um resultado em complemento de dois sem qualquer correção. O elemento é um multiplicador de 4 bits por 2 bits, e vários deles são interconectados para formar a rede. O multiplicador realiza a função $S = XY + K$ onde K é o operando de entrada usado para adicionar os produtos parciais gerados dentro da rede. No início da rede, as entradas K são disponíveis para adicionar uma constante à parte menos significativa do produto. Ao fazer um produto em complemento de dois, o bit

mais significativo da saída pode ser ignorado e as condições de "overflow" podem ser detectadas, comparando-se os dois últimos dígitos do produto.

- c) *RAM*: é uma memória estática de escrita e leitura. O acesso é feito através de portes de entrada e saída independentes. Ela também é constituída de vários elementos com saída do tipo "tristate" e é usada para armazenar os dados variáveis (dados de entrada, coeficientes de equalizador automático, etc).
- d) *ROM*: é uma memória de leitura somente com saída do tipo "tristate", usada para armazenar dados fixos (valor inicial de coeficientes de equalizador, coeficientes de filtros não-recursivos, etc).
- e) *A/D*: o conversor A/D é do tipo de aproximação sucessiva cuja saída digital é dada na representação complemento de dois. Ele é usado na aquisição do sinal analógico, que deve ser analisado ou processado.
- f) *SEQ*: é um controlador de endereço usado para o sequenciamento das microinstruções geralmente contidas em memórias ROMs ou PROMs. Ele é constituído de um ou mais elementos com 4 bits de saída. Por exemplo, interconectando dois elementos entre si gera-se um endereço de 8 bits, o que permite acessar 256 palavras de microcontrole. O controlador permite a seleção do endereço a partir de qualquer uma das seguintes fontes:
 - 1) Entrada direta de origem externa.
 - 2) Entrada direta de origem externa temporariamente armazenada em um registro interno.
 - 3) Uma pilha de quatro palavras de profundidade.
 - 4) Um contador de programa que usualmente contém o endereço da microinstrução corrente incrementado de um.

- g) *MEM*: é uma memória de leitura que contém as microinstruções. Cada microinstrução contém os bits que controlam todos os elementos que manipulam dados dentro do sistema. O conjunto de microinstruções é denominado microprograma, o qual é elaborado de acordo com a aplicação específica.
- h) *REGPL*: normalmente o registro "pipeline" é colocado na saída da memória do microprograma, essencialmente para que a busca da nova instrução e a execução da instrução anterior sejam feitas paralelamente, o que possibilita o aumento da velocidade de processamento. O registro contém a microinstrução que está sendo executada. Enquanto os bits de controle alimentam os elementos responsáveis pela execução da instrução, uma parte da microinstrução é usada pelo sequenciador para determinar o endereço da próxima microinstrução. Este endereço é aplicado à memória de microprograma e a próxima instrução aparece na entrada do registro "pipeline".

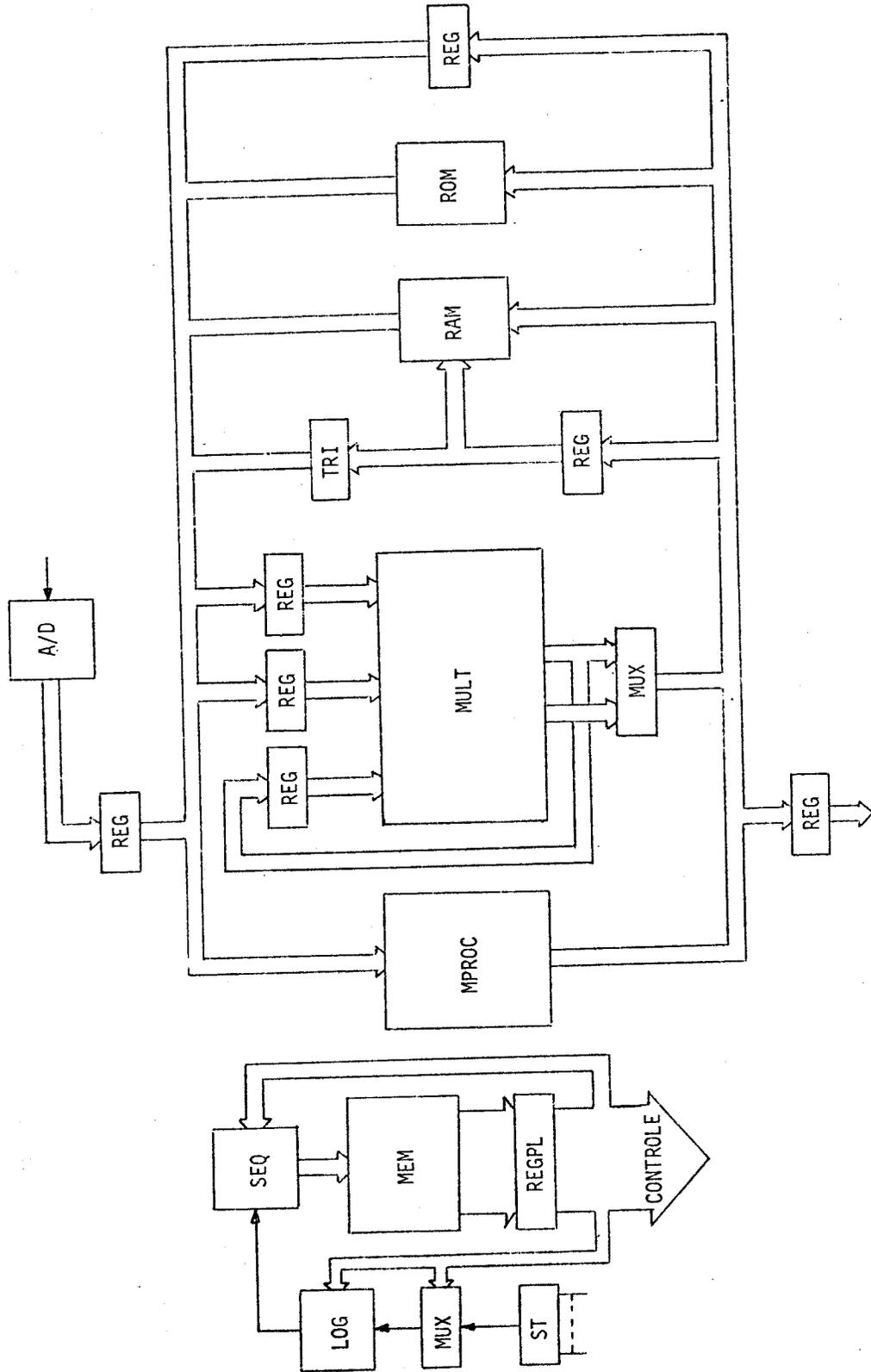


Fig. 3.1 - Arquitetura da máquina microprogramada.

CAPÍTULO 4

ESTRUTURA DA INSTRUÇÃO

A Figura 4.1 mostra a estrutura da palavra de microcontrole. Ela é constituída de vários campos, com funções bem específicas.

X	pag. RAM-ROM	end. A	end. B	função ALU	destino	fonte	end. ROM de ctes	controle portas e registros
	contr. seq.	end. desvio						

Fig. 4.1 - Formatação da instrução.

A introdução de mais de um formato na palavra de microcontrole permite reduzir o seu comprimento. Isto é possível devido à existência de operações exclusivas, próprias à arquitetura da máquina. Por exemplo, não se pode ler ou escrever na RAM enquanto se lê na ROM, nem ler na ROM enquanto se lê ou escreve na RAM. Por isso, a definição das páginas da ROM e da RAM são feitas no mesmo campo. Um outro caso típico é a impossibilidade, por imposição do projetista, de efetuar um desvio condicional ou não enquanto se utiliza um dos registros internos à ALU. Esse fato não prejudica a eficiência da máquina, mesmo por que a maioria dos testes de desvio são funções do resultado da última operação efetuada pela ALU, e não ocorrem com frequência.

CAPÍTULO 5

VELOCIDADE DA MÁQUINA

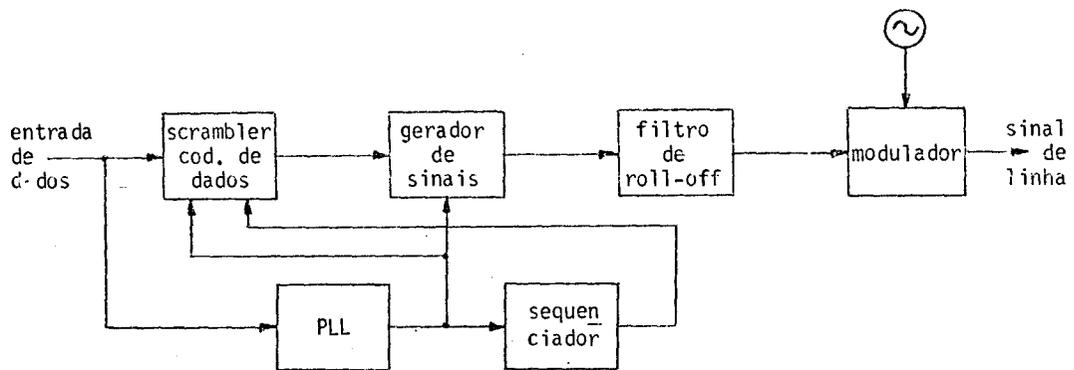
A velocidade da máquina é determinada pela arquitetura decorrente do tipo de aplicação a qual ela se destina e pela limitação natural da tecnologia dos componentes atuais do tipo LSI. Neste trabalho, discute-se a respeito da aplicação da máquina na área das transmissões de dados por canal de voz, com ênfase especial sobre um modem de 4.800 bits/seg, que utiliza, a técnica PSK ou DPSK. Um diagrama de blocos funcionais de um modem típico é mostrado na Figura 5.1. Os blocos funcionais mais apropriados para análise dos requisitos de velocidade são o equalizador, o filtro de amoldamento e o circuito de extração do sincronismo, porque eles podem ser analisados individualmente, além de consumirem a maior parte do tempo de processamento no modem em questão.

O equalizador é frequentemente implementado através da configuração mostrada na Figura 5.2, onde os valores dos coeficientes de cada "TAP" são ajustados de um modo adaptivo no caso de uma equalização automática, ou são prefixados em se tratando de uma equalização de compromisso. Para a transmissão de dados de 4.800 bits/seg, através da rede telefônica discada, a CCITT (I.T.U C.C.I.T.T, 1972) recomenda um equalizador automático. No diagrama de blocos proposto, o sinal equalizado é o sinal de banda base filtrado, o qual é obtido através da multiplicação do sinal de linha por um sinal senoidal síncrono cuja frequência é igual à frequência da portadora. Naturalmente a equalização deve ser processada em duas dimensões, já que as técnicas de modulação PSK e DPSK, assim como a QAM, são do tipo modulação em quadratura. As operações aritméticas necessárias em um processo de equalização automática que usa o algoritmo de minimização do erro quadrático médio são definidas pela Equação 2. É fácil observar a partir dessa equação que, no caso de uma equalização em duas dimensões, duas operações básicas do tipo $A \times B + C$ por "TAP" são necessários para a convolução, além de seis operações por "TAP" serem necessárias para a correção do

coeficiente de ganho relativo a cada "TAP". Portanto, para um equalizador automático com N "taps" e para uma taxa de amostragem f_a , o número total de operações básicas por segundo, J_{EA} , é estipulado por:

$$J_{EA} = 8Nf_a .$$

Para um modem de 4.800 bits/seg utilizado numa rede discada, $f_a = 1,6$ KHz e $N \approx 15$. Consequentemente, necessita-se de 192 K0/S (kiloperações por segundo) para implementar o equalizador automático.



P.L.L.: "Phase-Locked Loop"

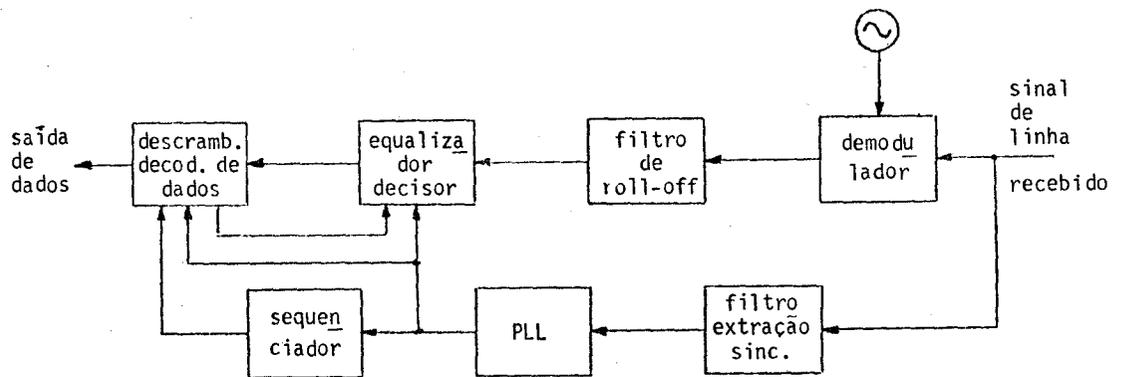


Fig. 5.1 - Diagrama de blocos funcionais de um modem típico.

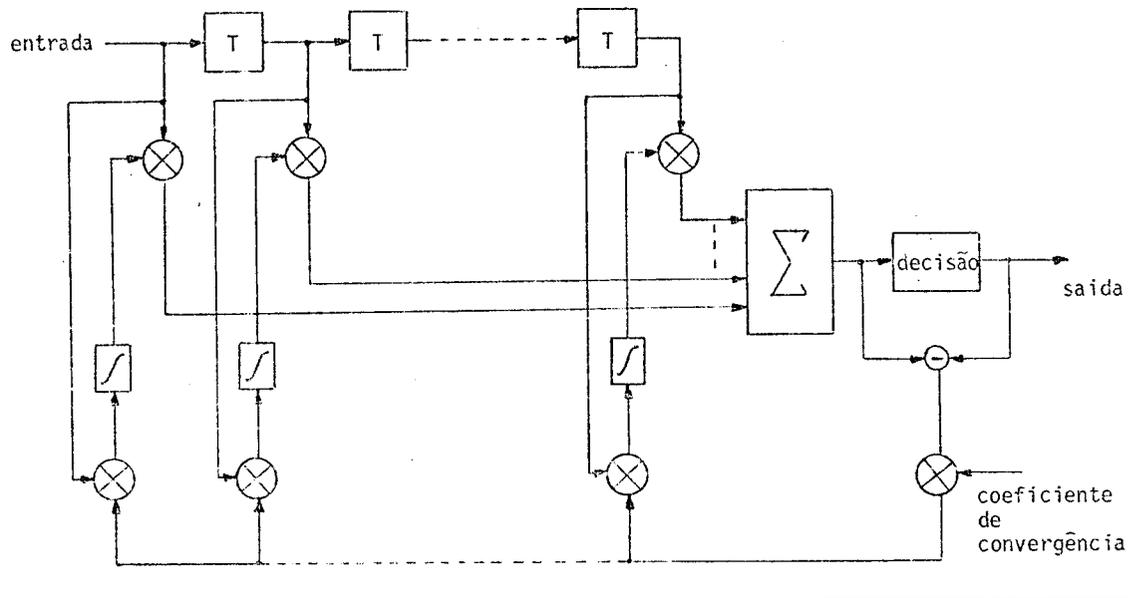


Fig. 5.2 - Diagrama de blocos de um equalizador automático.

FONTE: Hirsch and Wolf (1970), p. 5-11.

Um cálculo semelhante pode ser feito para o filtro de amoldamento. Normalmente, para se ter uma transmissão com uma relação S/N otimizada (Bennett and Davey, 1965), a característica total do filtro deve ser dividida igualmente entre os filtros do transmissor e do receptor. Este filtro é normalmente realizado ao utilizar-se de uma configuração do tipo transversal. Ele é portanto um filtro não recursivo, caracterizado pela Equação 3. O número de operações básicas, J_{FA} , pode ser expresso, nesse caso, por

$$J_{FA} = MJa,$$

onde M é o número de "taps" do filtro transversal e fa é a taxa de amostragem na saída do filtro. Para um modem de 4.800 bits/seg, um filtro transversal com cerca de 32 "taps" parece suficiente do ponto de vista prático (Mammoli, 1980). Considerando que dois filtros são necessários para filtrar as componentes em fase e em quadratura, respectivamente, obtêm-se 102,4 K0/S.

Finalmente, examina-se o circuito de extração do sincronismo, a qual é realizada através de uma filtragem seletiva do sinal de linha retificado. O filtro usado para esse fim possui uma faixa de passagem muito estreita, centrada na frequência de transmissão, e é denominado filtro tanque. De acordo com a frequência de transmissão e com as características dos filtros de amoldamento, mais de um filtro tanque pode ser necessário. O filtro tanque pode ser realizado por um filtro digital de segunda ordem, com a seguinte função de transferência:

$$H(z) = \frac{1 - a_2}{2} \cdot \frac{1 - z^{-2}}{1 - a_1 z^{-1} + a_2 z^{-2}}$$

A realização física deste filtro é feita através da estrutura recursiva mostrada na Figura 5.3. Fundamentalmente, este filtro requer 4 operações básicas e 2 passos suplementares para o deslocamento dos dados, o que implica 6 operações por amostra. Caso haja necessidade de dois filtros tanque em cascata, obtém-se 115,2 K0/S a uma taxa de amostragem de 9,6 kHz.

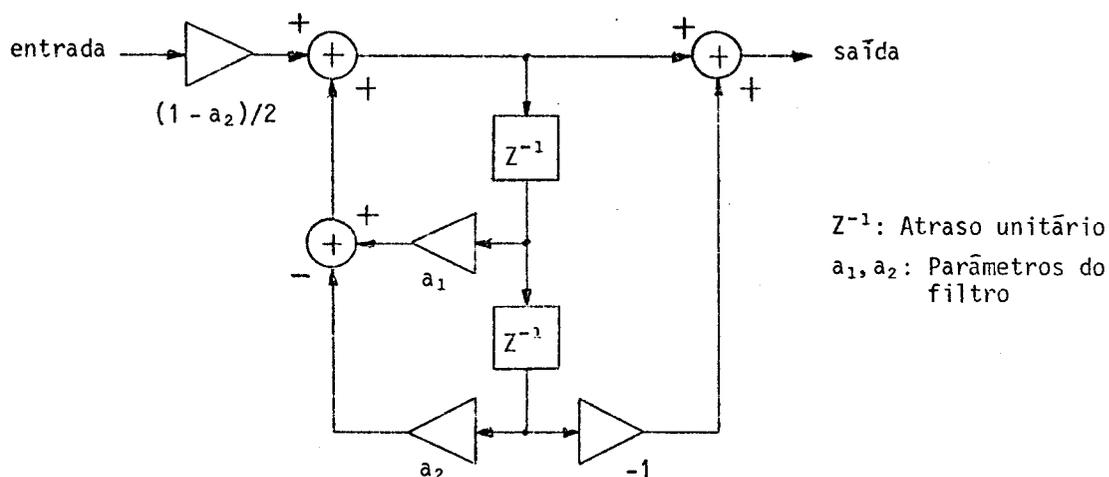


Fig. 5.3 - Estrutura de um filtro tanque de segunda ordem.

Um outro aspecto da arquitetura deve ser analisado para concluir essa discussão, de modo a traduzir o número de operações em termos realizáveis com a velocidade efetiva da máquina. O exame da arquitetura revela a necessidade de 3 a 6 pulsos de relógio de máquina para levar a cabo uma multiplicação e para armazenar o seu resultado. O pior caso operacional se dá quando os dois operandos se encontram dentro da ALU, e quando há necessidade de armazenar tanto a parte mais significativa quanto a parte menos significativa do resultado dentro da memória RAM; esse caso ocorre esporadicamente. O melhor caso se dá quando os operandos estão disponíveis diretamente no barramento de entrada e quando somente a parte mais significativa do resultado é armazenada na memória RAM. Quando se realiza uma convolução ou um processamento similar, são necessários 4 pulsos de relógio. Desenvolvendo o "firmware" de forma criteriosa, esse número de pulsos de relógio pode ser usado como base para o cálculo da velocidade da máquina. A convolução exemplifica este conceito de forma simples, através dos seguintes passos:

- 1) Busca do primeiro operando (este se encontra na memória RAM ou na memória ROM).
- 2) Busca do segundo operando (este se encontra na memória RAM ou na memória ROM).
- 3) O registro "buffer" situado entre os dois barramentos recebe a parte mais significativa do resultado, enquanto a parte menos significativa é guardada em um dos registros de entrada da rede multiplicadora, para ser somada ao produto subsequente.
- 4) As partes mais significativas dos respectivos produtos são acumuladas dentro da unidade lógica e aritmética. Durante as várias fases do processo, os ponteiros que determinam os endereços dos operandos devem ser manipulados adequadamente.

O resumo dos cálculos feitos é visto na Tabela 5.1.

TABELA 5.1

TOTAL DE OPERAÇÕES POR PROCESSO

PROCESSO	FREQUÊNCIA DE AMOSTRAGEM	Nº DE OPERAÇÕES BÁSICAS POR AMOSTRA	TOTAL DE OPERAÇÕES BÁSICAS
Equalizador automático 4800 BPS	1,6 kHz	120	192 KOPS
Filtro de amoldamento 4800 BPS	1,6 kHz (9,6 kHz)	64	102,4 KOPS
Filtro de extração de sincronismo	9,6 kHz	12	115,2 KOPS

O número total de operações básicas para esses três processos é de aproximadamente 410 KOPS. Considerando que, em média, 4 pulsos de relógio de máquina são necessários para levar a cabo uma operação, a máquina deverá trabalhar com uma frequência mínima de 1,64 MHz. É possível implementar a arquitetura proposta usando componentes convencionais do tipo LSI, de forma a se obter um ciclo de máquina de duração aproximada de 250 nanossegundos. Existe portanto um tempo disponível muito grande para executar outras funções, tais como:

- gerenciar os diversos processos;
- monitorar sinais;
- tomar decisões;
- adquirir dados.

Sugere-se que, de modo geral, o "software" seja desenvolvido de forma a ter um programa principal encarregado de gerenciar todos os processos ou tarefas. Os processos seriam então executados sob forma de rotinas ou sub-rotinas referenciadas pelo programa.

CAPÍTULO 6

INTEGRAÇÃO DOS BLOCOS FUNCIONAIS PARA A OBTENÇÃO DE UM MODEM DE 4800/2400 BPS

Neste capítulo, são descritos de forma sucinta os programas que gerenciam os vários processos envolvidos na demodulação do sinal de linha. Antes de prosseguir, considerações de ordem geral fazem-se necessárias. Basicamente, o modem é um equipamento que viabiliza a troca de informação ponto-a-ponto entre dois equipamentos terminais. Quando se dá o início da transmissão por parte de um terminal, os equipamentos de comunicação, assim como o terminal receptor, devem permitir o escoamento da informação a uma velocidade única, previamente estabelecida. Neste caso, o processamento próprio à execução da comunicação deve ser feito em tempo real. Isto significa que o modulador e o demodulador devem adquirir e fornecer periodicamente dados a intervalos rigorosamente iguais, geralmente definidos por um oscilador de alta estabilidade. Por isso, o processamento propriamente dito deve se dar entre os respectivos intervalos de fornecimento de dados. Existem várias abordagens para resolver este problema, tal como definir todas as tarefas e calcular o tempo necessário para cada uma delas. Alocar tais tarefas nos intervalos prescritos é uma possibilidade, embora carente de flexibilidade. Um outro esquema aqui adotado trabalha num regime de interrupções igualmente espaçadas, de forma que o processo corrente seja interrompido para ceder a vez às rotinas de entrada e saída. As interrupções são detetadas por "hardware". Os diagramas de fluxo mostrados nas Figuras 6.1 e 6.2 caracterizam o procedimento utilizado na demodulação do sinal de linha. A mesma filosofia pode ser empregada no modulador.

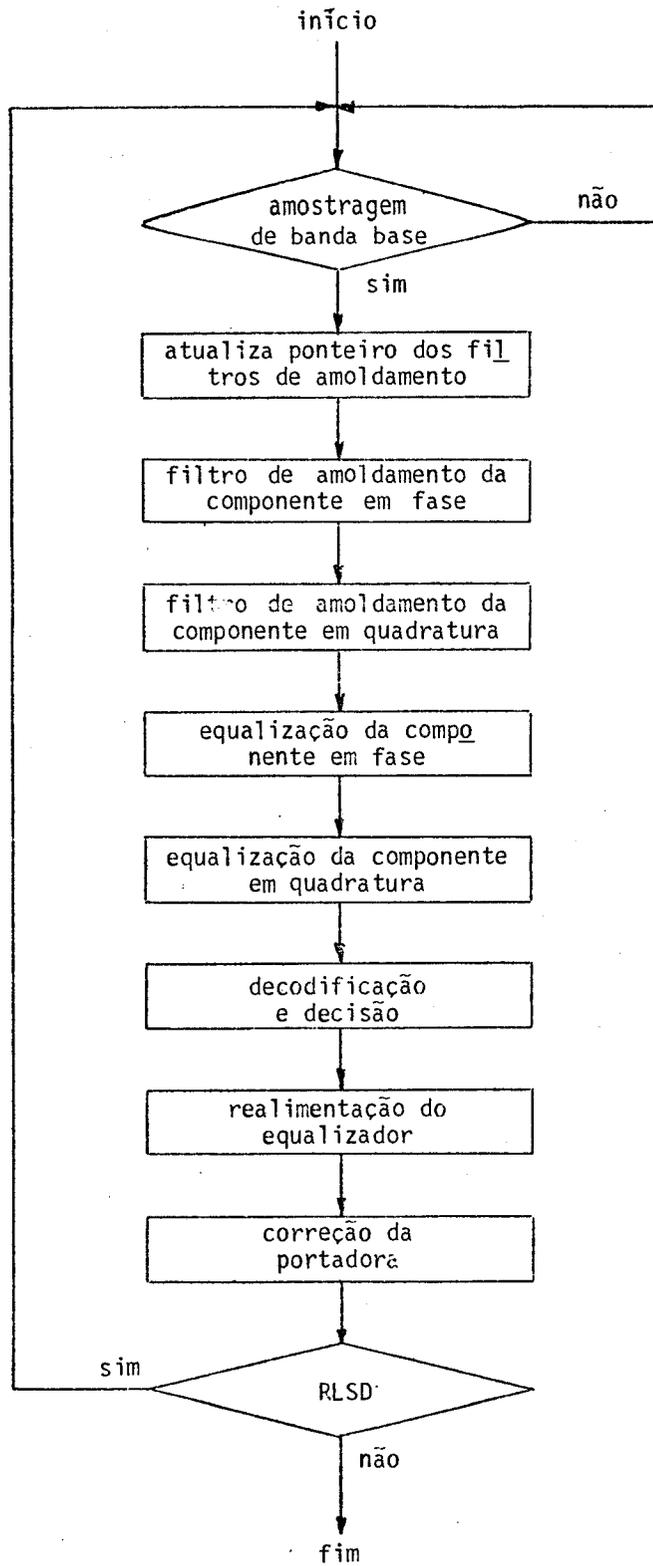


Fig. 6.1 - Diagrama de fluxo do programa principal.

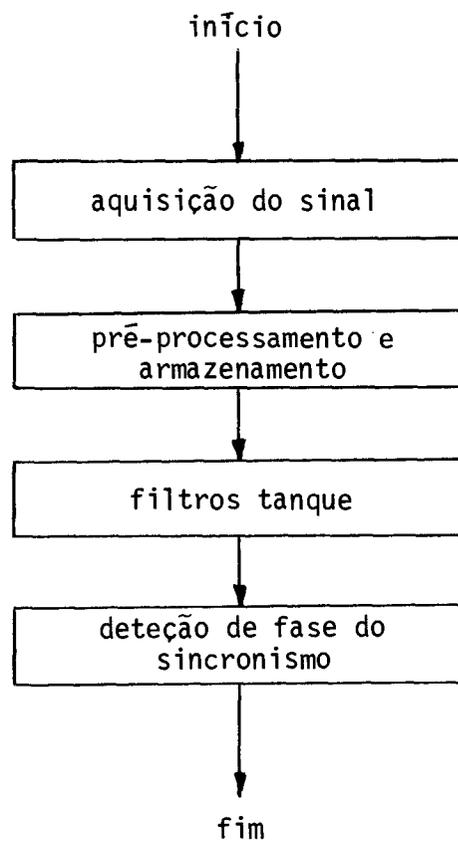


Fig. 6.2 - Diagrama de fluxo do atendimento à interrupção.

CAPÍTULO 7

CONCLUSÃO

Este trabalho descreveu a arquitetura e as funções de uma máquina baseada em um processador de propósito geral, ao qual são adicionados outros componentes do tipo LSI, o que a torna particularmente adequada ao processamento digital de sinais no campo das comunicações de sinais e dados. Devido ao seu esquema de controle por "firmware" e à sua grande capacidade de manipulação de dados, a flexibilidade e a universalidade de aplicação são as principais características desta máquina.

A sua utilização para realizar um modem de 4800 bits/segundo foi mostrada a partir das características de vários blocos funcionais; finalmente, foi visto como a máquina pode vir a constituir um modem completo.



REFERÊNCIAS BIBLIOGRÁFICAS

- I.T.U C.C.I.T.T. *4800 bits per Second Modem Standardized for Use on Leased Circuit*. Genève, 1972 (serie V.27 Recommendation).
- W.R. BENNETT, J.R. DAVEY. *Data Transmission*. Mc Graw-Hill 1965.
- MAMMOLI, M. *Simulação de um Modem Digital para Comunicação a 4800 bits/s*. Tese de Mestrado em Eletrônica e Telecomunicações/Sistemas Digitais. São José dos Campos, INPE, out., 1980 (INPE - 1928 - TDL/032)
- D. HIRSCH, W.J. WOLF. *A Simple Adaptive Equalizer for Efficient Data Transmission*. *IEEE Transactions on Communication Technology*, Vol. COM-18, nº 1, February 1970.