

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

DEPARTAMENTO DE ELETRÔNICA DA
ESCOLA DE ENGENHARIA DA UFRJ - DEEL

ANALISADOR DE ESPECTRO POR FFT

FERNANDO DA CUNHA LUIZ

ORIENTADOR : PROF. ANTÔNIO CARLOS MOREIRÃO DE QUEIROZ

BANCA EXAMINADORA :

1. PROF. ANTÔNIO CARLOS MOREIRÃO DE QUEIROZ
2. PROF. ANTÔNIO PETRAGLIA
3. PROF. LUIZ PEREIRA CALÔBA

RIO DE JANEIRO, 12 de março de 1985.

Dedico este trabalho à meus pais, ROBERTO E LÚCIA,
que muito me ajudaram e incentivaram nos estudos.

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

-AGRADEÇO ao Prof. ANTÔNIO CARLOS MOREIRÃO DE QUEIROZ ,
pela orientação dada no projeto .

-Agradeço ao Prof. EDSON HIROKAZU WATANABE, pela libera-
ção da verba necessária ao projeto.

-Agradeço aos Técnicos do Laboratório de Eletrônica pe-
apoio dado.

-Agradeço aos Professores do Departamento de Eletrônica
pelos conhecimentos adquiridos.

RESUMO

O analisador de espectro por FFT (FAST FOURIER TRANSFORM) obtém o espectro de um sinal periódico por " SOFTWARE ", a partir de 256 pontos amostrados.

O sinal de entrada deve estar na faixa de 1V a 4V. O sinal de saída está na faixa de 1V a 4V também .

Deve ser utilizado em conjunto com um osciloscópio. A apresentação do espectro é na tela deste.

Permite também que seja utilizado como memória de osciloscópio para até 2048 pontos.

Pode ser acoplado a um " PLOTTER " X-T .

ÍNDICE

	PÁG.
CAPÍTULO 1	
INTRODUÇÃO AO ANALISADOR DE ESPECTRO	1
CAPÍTULO 2	
A FFT	
2.1 Introdução ao programa da Transformada de Fourier	3
2.2 Algoritmo de Cooley-Tukey da FFT	3
2.3 Comentários sobre representação dos números	5
2.4 Conclusões sobre o programa	6
CAPÍTULO 3	
O CONVERSOR A/D e D/A	7
3.1 Descrição do funcionamento	7
3.2 O conversor D/A	10
3.3 Escolha do comparador	14
3.4 O amostrador da entrada	15
3.5 O amostrador da saída	16
3.6 A determinação das tensões de alimentação	16
3.7 O controle de início e fim de conversão	17
3.8 Tempo de conversão	20
3.9 O oscilador de frequência variável	20
CAPÍTULO 4	
4.1 A lógica de controle do conversor	23
4.2 Os diagramas de tempo	27
CAPÍTULO 5	
O MICROCOMPUTADOR	
5.1 Escolha da CPU	35
5.2 Descrição dos sinais da CPU	35
5.3 Diagrama de tempo da CPU	38
5.4 A lógica de controle dos barramentos e das interrupções	45

	PÁG.
5.5 As memórias	46
5.6 Especificação dos tempos de acesso das memórias	46
5.6.1 Especificação do tempo de acesso da "EPROM"	48
5.6.2 Especificação do tempo de acesso da "RAM" 6116	49
5.6.3 Especificação do tempo de acesso da "RAM" 2114	50
5.7 O "CLOCK"	51
CAPÍTULO 6	
A FONTE DE ALIMENTAÇÃO	
6.1 Considerações iniciais	53
6.2 Escolha do transformador	54
CAPÍTULO 7	
CONCLUSÕES	58
APÊNDICE 1	
Demonstração da máxima amplitude possível de um harmônico de uma forma de onda.	59
APÊNDICE 2	
Referências	60
APÊNDICE 3	
Lista de material	61
APÊNDICE 4	
Listagem do programa da FFT em BASIC	65
APÊNDICE 5	
Endereços da memória	67
APÊNDICE 6	
O programa da FFT em ASSEMBLY	68
APÊNDICE 7	
Comparação dos resultados dos programas em BASIC e ASSEMBLY	106
APÊNDICE 8	
OPERAÇÃO DO ANALISADOR DE ESPECTRO	111

CAPITULO I

INTRODUÇÃO AO ANALISADOR DE ESPECTRO

O analisador é composto basicamente de 4 blocos: conversor A/D, conversor D/A, memória e microprocessador.

Na figura 1.1 temos o diagrama em blocos do analisador. O sinal de entrada é amostrado e feita a conversão analógico/digital. Ao término da conversão, o sinal de fim de conversão escreve o valor amostrado no endereço atual da memória e inicia / nova amostragem. Ao chegar novamente o pulso de inicio de conversão, o sinal de fim de conversão é retirado, avançando o contador de endereços, e iniciada nova conversão A/D.

Há uma tecla que inibe, através de um flip-flop, a escrita na memória, habilitando a leitura. Apertando esta tecla o conversor A/D para de escrever na memória e o "BUFFER 3-STATE" é desabilitado, permitindo que se observe no osciloscópio o sinal gravado.

Os pulsos de inicio de conversão devem continuar, pois é com o sinal de fim de conversão, que o endereço é mudado e o amostrador de saída é controlado.

Há uma segunda tecla que desabilita a leitura e habilita a escrita pelo conversor A/D, voltando assim a situação inicial.

Definição: 1 página = 256 BYTES.

Apertando a primeira tecla, habilitando a leitura, pode-se apertar uma terceira tecla que inicia o processamento do sinal. Após alguns segundos aparecerá na tela o espectro do sinal. O 1º ponto é o nível DC do sinal, o 2º ponto é o 1º harmônico, o 3º ponto é o 2º harmônico, e assim, por diante até o 129º ponto que é o 128º harmônico. A partir do 128º harmônico há a repetição invertida do espectro, devendo só ser considerado até o 128º harmônico.

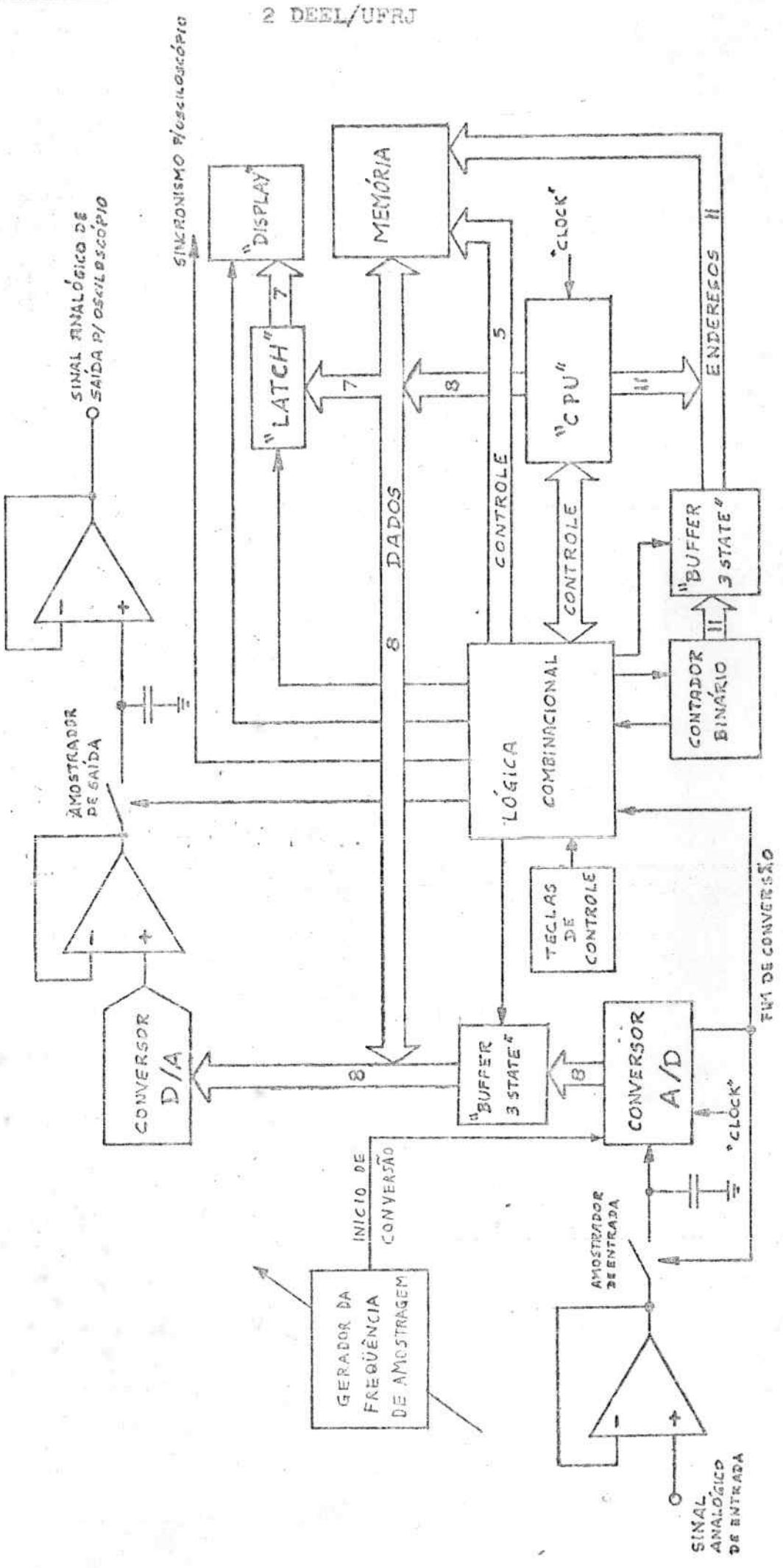


FIG.1.1 - DIAGRAMA EM BLOCOS DO ANALISADOR DE ESPECTRO POR FFT

CAPÍTULO 2

A FFT

2.1 INTRODUÇÃO AO PROGRAMA DA TRANSFORMADA DE FOURIER

Dado um vetor complexo $\alpha = (\alpha_0, \dots, \alpha_{n-1})$ este programa calcula os coeficientes da transformada direta de Fourier:

$$b_m = \frac{1}{n} \sum_k \alpha_k e^{-j2\pi \frac{m}{n} k} ; \quad 0 \leq m \leq n-1 \quad (2.1)$$

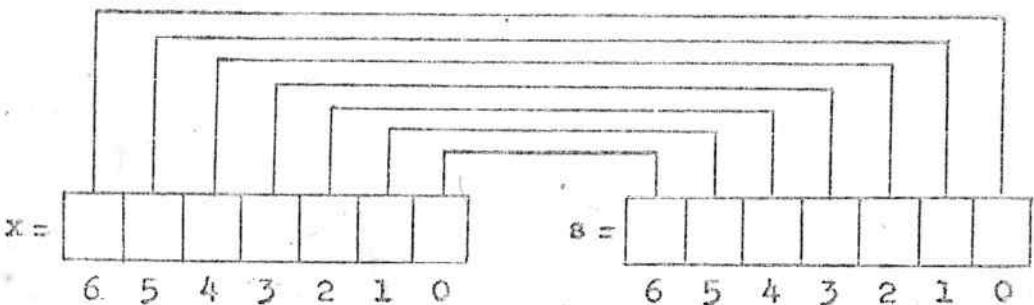
n deve ser potência inteira de 2, isto é, n pode ser: 2, 4, 8, 16, 32, ...

$$m = \log_2(n) \quad (2.2)$$

$$\alpha_k = a(k) + j b(k) \quad (2.3)$$

Define-se a função bit inverso, que é a representação de um número de m bits com os bits mais significativos à direita e os menos significativos à esquerda.

Ex: P/ $n=128 \Rightarrow m=7 \Rightarrow 7$ bits



A função bit inverso de x é $r(x)$.

$$s = R(x) \quad (2.4)$$

2.2 ALGORITMO DE COOLEY-TUKEY DA FFT (Ref. 1 e 2).

Entre com o número de pontos, N .

Entre com os pontos $A(I)$ e $B(I)$ para I de 0 a N . $A(I)$ é a parte real e $B(I)$ é a parte imaginária.

$$M = \log_2(N)$$

Definir função de BIT inverso R(K)

Para K=M até 1 passo -1 :

Para P=0 até N-1 passo 2^K :

$$X = P / 2^{K-1}$$

$$S = R(X)$$

$$X_1 = \cos(2\pi S/N)$$

$$Y_1 = \sin(2\pi S/N)$$

Para J=0 até $2^K - 1$ passo 1 :

$$X_2 = X_1 * A(J+P+2^K) + Y_1 * B(J+P+2^K)$$

$$Y_2 = -Y_1 * A(J+P+2^K) + X_1 * B(J+P+2^K)$$

$$A(J+P+2^K) = A(J+P) - X_2$$

$$B(J+P+2^K) = B(J+P) - Y_2$$

$$A(J+P) = A(J+P) + X_2$$

$$B(J+P) = B(J+P) + Y_2$$

PRÓXIMO J

PRÓXIMO P

PRÓXIMO K

Para I=1 até N-1 passo 1 :

$$S = R(I)$$

Se $S > I$, então :

$$X_3 = A(I)$$

$$Y_3 = B(I)$$

$$A(I) = A(S)$$

$$B(I) = B(S)$$

$$A(S) = X_3$$

$$B(S) = Y_3$$

PRÓXIMO I

Para I=0 até N-1 passo 1 :

$$M(I) = 2 * \text{SQR}(A(I)^2 + B(I)^2) / N$$

PROXIMO I

$$M(\phi) = M(\phi)/2$$

$$M(N/2) = N(N/2)/2$$

FIM.

2.3 COMENTÁRIOS SOBRE A REPRESENTAÇÃO DOS NÚMEROS:

O programa de FFT foi desenvolvido em "ASSEMBLY". Sendo assim há necessidade de se representar o sinal amostrado em complemento 2 para as subrotinas aritméticas. Os pontos amostrados assumem valores de ϕ a 255. Podemos assumir que o nível ϕ corresponde à 128. Subtraindo 128 de todos os pontos amostrados, então estes estarão com valores entre -128 e +127. Após este passo os pontos também estarão na representação complemento 2.

As equações que dão os coeficientes reais e imaginários da transformada de Fourier são respectivamente:

$$a_n = \frac{2}{T} \int_{t_0}^{t_0+T} f(t) \cdot \cos(nw_0 t) \cdot dt, \quad T = 2\pi/w_0 \quad (2.5)$$

$$b_n = \frac{2}{T} \int_{t_0}^{t_0+T} f(t) \cdot \sin(nw_0 t) \cdot dt, \quad T = 2\pi/w_0 \quad (2.6)$$

Pode ser demonstrado (apêndice 1) que o maior harmônico possível é:

$$2.(A-B)/\pi, \quad (2.7)$$

onde A é o nível máximo do sinal e B é o mínimo. Para o nosso caso $A=127$ e $B=-128$ e o máximo harmônico possível dado pela eq. 2.7 é :

$$\frac{127+128}{2} \cdot \frac{4}{\pi} = \frac{510}{\pi} \approx 162,338$$

Observando o algoritmo de FFT observamos que este resultado é obtido após ser dividido por $N/2$. Multiplicando por $N/2$, encontramos que o maior resultado intermediário durante o processamento para $N=256$ é :

$$128 \times 162,338 = 20.779,27$$

Este valor foi encontrado por simulação em BASIC, para encontrar o maior valor processado.

Chegamos então à conclusão de que necessitamos representar A(I) e B(I) com 16 BITS. Porém com a representação de A(I) e B(I) por números inteiros, o resultado do programa não é preciso. Incluimos então, mais 16 BITS após a vírgula, ficando a representação com 32 BITS. A representação dos senos e co-senos é satisfatória com apenas 16 BITS.

2.4 CONCLUSÕES SOBRE O PROGRAMA

Em diversas simulações com a apresentação do resultado com 8 BITS observou-se que o erro é aproximadamente de 1 BIT menos significativo para 256 pontos.

O programa desenvolvido em "ASSEMBLY" calcula o espectro / com precisão de 15 BITS, devido à rotina de raiz quadrada ser de 15 BITS. O programa em BASIC apresenta os resultados com precisão de 32 BITS em ponto flutuante. Uma comparação entre os resultados do programa em "ASSEMBLY" e do programa em BASIC mostrou que o erro relativo é menor que 0,02 em valor absoluto. Isto significa que a maior fonte de erro é por "ALIASING". Simulações em BASIC, demonstraram que o erro é inversamente proporcional ao número de pontos.

CAPITULO 3

O CONVERSOR A/D e D/A

3.1 DESCRIÇÃO DE FUNCIONAMENTO :

O diagrama completo do conversor está na figura 3.1 . O conversor A/D é do tipo de aproximações sucessivas. A cada pulso de "clock", apenas uma saída do contador em anel, 4017, fica ativa em "1", na ordem : Q9, Q8, Q1, Q2, Q3, Q4, Q5, Q6, Q7 , Q8. A passagem do estado Q9 para Q0 é através do "RESET".

Na figura 3.2 , temos o diagrama de tempo do contador.

Por uma lógica, a ser explicada na seção 3.7, após o sinal SG , inicio de conversão, são enviados 9 pulsos de clock para o contador. O primeiro estado do contador é Q9+Q0. Último estado é Q8 indicando o fim da conversão. Este estado se mantém até que seja iniciada nova conversão.

O "OU" de Q9 e Q0 é necessário devido ao atraso do "RESET" para Q0 ser da ordem de 500ns.

Q9+Q9 causa o "SET" no flip-flop de MSB(D7) e o "RESET" / nos demais. Se o resultado do comparador for de que a aproximação digital é maior do que o sinal analógico, é dado o "RESET" no flip-flop do MSB, porém a saída deste se mantém em "1" porque o "SET" ainda é ativo. No segundo pulso de clock, o contador passa do estado Q0 para Q1. Ao ser retirado o SET e o RESET do flip-flop do MSB, o RESET fica mais tempo pelo atraso de uma porta e então, o flip-flop assume o estado "0" ; Se a aproximação digital for menor que o sinal analógico, então não é dado o RESET e o flip-flop fica em "1".

O CMOS 4043B exige tipicamente que a duração do pulso de RESET seja maior que 80ns , mas somente o atraso da retirada do RESET em relação ao SET é de aproximadamente 120 ns, atraso típico de uma porta CMOS não inversora com apenas um CMOS de car-

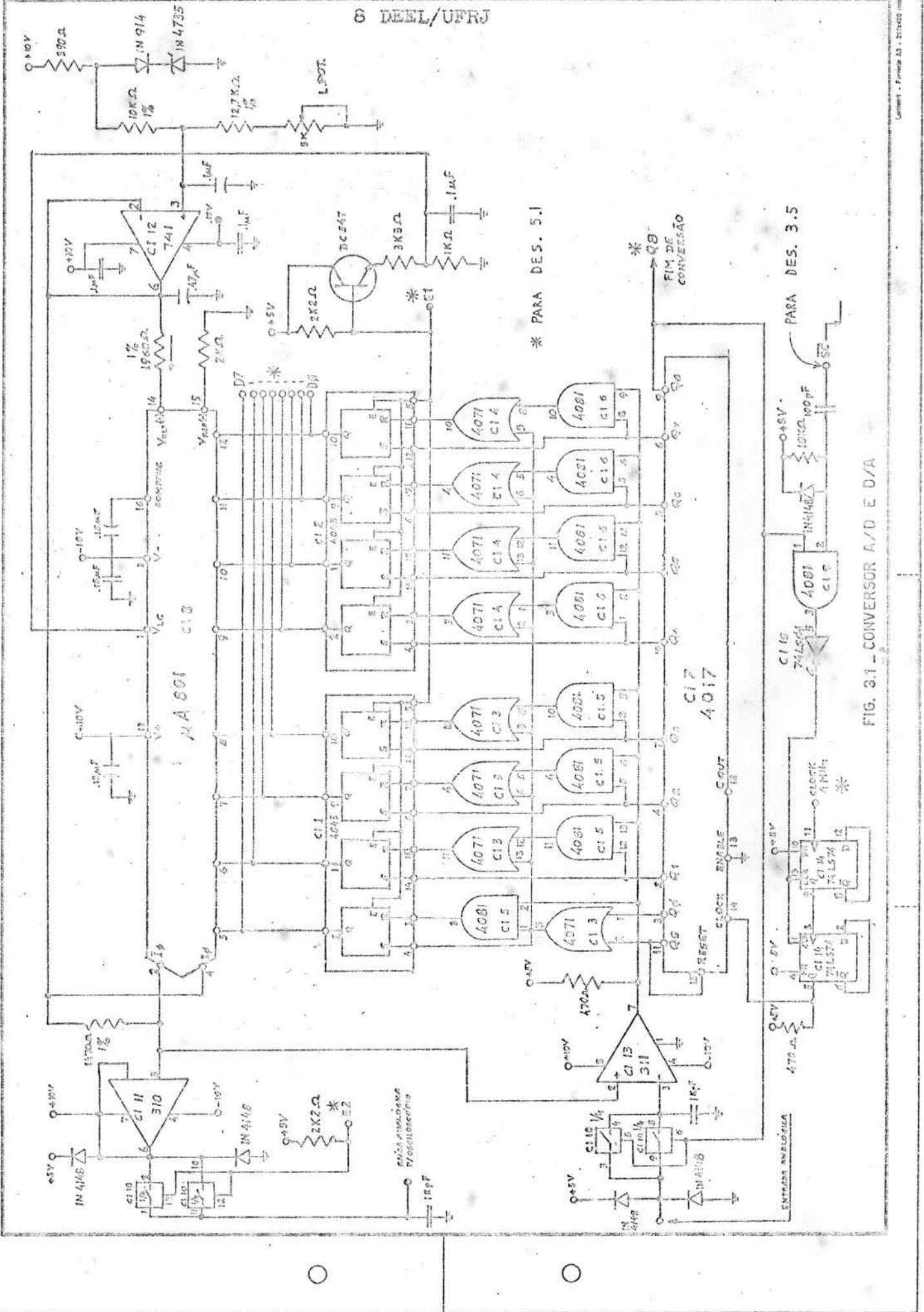


FIG. 3.1 - CONVERSOR A/D E D/A

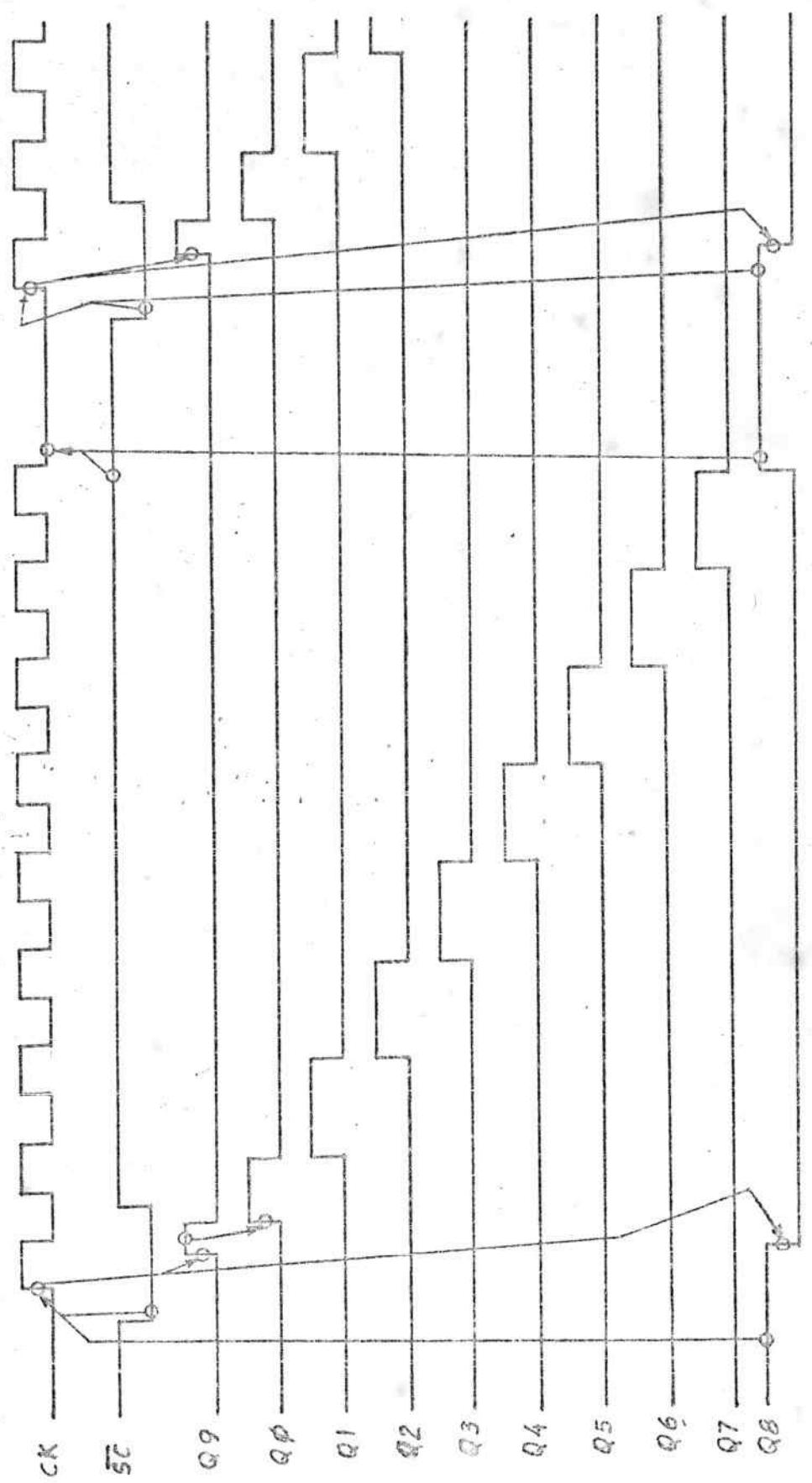


FIG. 3.2

ga (5pF).

Q1 indo para "1" causa o SET no flip-flop do 2º MSB (D6), e a situação é análoga ao caso de Q0. O mesmo ocorre para os de mais flip-flops.

A conversão termina quando Q7 vai para "0" e é dado ou não o RESET no flip-flop do LSB, conforme o resultado do comparador.

Quando Q8 sobe a conversão está pronta e é iniciada nova amostragem.

A conversão leva 9 pulsos de clock, sendo que no 9º o resultado estabiliza.

A limitação do clock é dada pelos atrasos dos flip-flops, das 2 portas lógicas ligadas a cada flip-flop, do conversor D/A e do comparador. Assumindo atrasos típicos, cada porta tem atraso de 120ns, cada flip-flop tem atraso de 175ns, o conversor D/A tem atraso de 100ns e o comparador tem atraso de 200ns. Sómando, temos 715ns, o que dá uma freqüência de clock máxima de 1,4 MHz.

O conversor funciona com clock de até 1,5 MHz, porém este é o limite a partir do qual não funciona corretamente. Fixamos o clock em 1MHz, para assegurar o funcionamento correto, caso sejam trocados os componentes por alguma razão.

Os flip-flops RS contidos no CMOS 4043B, possuem saída 3-State, permitindo assim a ligação direta ao barramento de dados.

O conversor D/A é multiplexado com o registro de aproximações sucessivas e a memória.

3.2 O CONVERSOR D/A

Um dos mais rápidos conversores D/A de 8BITS encontrado no comércio, aliás o único, é o DAC 0801. É um conversor por chavea-

mento de corrente.

As principais características são :

• Resolução de 8 BITS ;

• não linearida: menor que $\pm 0,39\%$ de fim de escala;

• tempo de estabilização para $\pm 1/2$ LSB: típico de 100ns e
máximo de 150ns ;

• coeficiente de temperatura típico : de 10 ppm/ $^{\circ}\text{C}$;

• faixa de corrente de saída de 0 a 4,2 mA ;

• possui entrada de referência de nível lógico ;

• alimentação negativa de -18V a -4,5V e positiva de +4,5V a
+ 18V ;

• sensibilidade a variações da tensão de alimentação menor
que 0,01 %/ %;

• corrente típica de referência de 2mA e máxima de 4mA .

O fabricante recomenda a colocação de capacitores de 0,1 μF
próximo ao componente nos pinos 1,3,13 e 16 conforme o esquema,
para evitar ruídos que venham prejudicar o bom funcionamento do
conversor D/A.

Os únicos elementos de precisão necessários para sua utili-
zação são uma fonte de corrente de referência e um resistor de
carga.

Para a fonte de referência, foram levantadas as caracterís-
ticas de resistência dinâmica e coeficiente de temperatura de
diversos diodos Zener, devido à dificuldade de se encontrar no
comércio diodos próprios para fonte de referência, porém nenhum
satisfaz às exigências. Foram então, levantadas curvas de asso-
ciações de diodos Zener com diodos IN914. Os diodos Zener com
tensão nominal maior do que 5,5V apresentam coeficiente de tem-
peratura positiva e os diodos comuns apresentam coeficiente de
temperatura negativo.

A combinação que deu melhor resultado foi o IN4735

com

o 1N914, com acoplamento térmico. O coeficiente de temperatura é menor que $\pm 0,3 \text{ mV}/^\circ\text{C}$.

Na tabela 3.1, temos a característica da combinação dos 2 diodos.

TABELA 3.1

CORRENTE DE POLARIZAÇÃO (mA)	TENSÃO (V)	RESISTÊNCIA DINÂMICA (Ω)
5	6,730	15
7	6,755	12
9	6,776	10
11	6,794	9
13	6,811	8
15	6,825	7

Como toda a lógica foi alimentada com +5V, limitamos a tensão de referência em 4V para que as chaves analógicas CMOS funcionem adequadamente.

Devemos tirar uma fonte de tensão de referência de 4,000V a partir de 6,800V. Utilizamos o divisor resistivo associado a um trimpot multivoltas. A tensão pode ser ajustada entre 3,8V e 4,3V se a tensão na combinação dos diodos for de 6,8V. Foram utilizados resistores de 1% por causa da estabilidade.

Como BUFFER foi utilizado o C.I. LM741C. Foram colocados capacitores da entrada e na saída para eliminar ruídos da referência.

A entrada de referência, pino 14 do conversor D/A, é de corrente. A corrente típica recomendada pelo fabricante é de 2 mA. Utilizando um resistor de 1960Ω , 1%, a corrente de referência é de 2,041mA. Um resistor de aproximadamente 1960Ω deve ser ligado ao pino 15 para compensação da corrente de pola-

rização do pino 14.

A corrente máxima de saída do conversor é dada pela equação : $I_{\text{máx}} = I_{\text{ref}} \times \frac{255}{256}$ (3.1)

Para $I_{\text{ref}} = 2,041\text{mA}$, $I_{\text{máx}} = 2,033\text{mA}$.

Para que a chave CMOS funcione adequadamente, fixamos a tensão mínima em aproximadamente 1V . Com um resistor de 1470Ω , 1 %, a tensão mínima é de 1,012V dada pela equação :

$$V_{\text{min}} = V_{\text{ref}} - I_{\text{max}} \times R \quad (3.2)$$

onde $V_{\text{ref}} = 4,000$, $I_{\text{max}} = 2,033\text{mA}$ e $R = 1470\Omega$.

A variação de 1 BIT menos significativo corresponde à variação de 11,72mV dado pela eq. 3.3 , fazendo:

$$N = 1, I_{\text{ref}} = 2,041\text{mA} \quad \text{e} \quad R = 1470\Omega$$

$$\Delta V = \frac{N}{256} \times I_{\text{ref}} \times R \quad (3.3)$$

Para a mudança de nível lógico de TTL para CMOS foi incluído um transistor.

Quando E_1 é +5V devido ao " pull-up" ligado à base do transistor, o emissor apresenta 4,3V e o pino 1 do conversor apresenta 1V, elevando a tensão de mudança de nível lógico para 2,4V . Com o transistor conduzindo a corrente no emissor é de 1mA, sendo a tensão base coletor desprezível pois a corrente de base é menor que 10µA .

Quando E_1 é menor que 0,6V o transistor corta a tensão do pino 1 do conversor é 0V, devido a corrente fornecida por este ser desprezível. Neste caso a tensão de mudança de nível lógico é de 1,4V ou seja, compatível com os níveis TTL fornecidos pelas memórias .

Para a saída do conversor foi utilizada a saída $\overline{I_o}$, de modo que quando a entrada digital for nula, a corrente é máxima e a tensão é mínima (1,012V). Assim o sinal digital não sai inverso

tido na saída. A saída I_o foi ligada à fonte de referência de maneira que $I_o + I_v$ seja a corrente constante drenada da fonte de referência.

Assumindo que o valor 128 é o meio da escala a tensão de off-set é :

$$V_{os} = -I_{ref} \times \frac{128}{256} \times R + V_{ref} \quad (3.4)$$

Com $I_{ref} = 2,041 \text{ mA}$, $R = 1470 \Omega$, $V_{ref} = 4,000\text{V}$ em (3.4), então $V_{os} = 2,500\text{V}$, ou seja, na metade da tensão de alimentação da chave CMOS.

3.3 Escolha DO COMPARADOR

Desejávamos um comparador rápido, simples de se alimentar, de alta impedância de entrada para evitar o uso de um BUFFER entre o amostrador e o comparador, e que fosse encontrado no comércio com facilidade. O comparador BIPOAR LM311 satisfaz essas características, as quais são:

Máxima corrente de entrada : 250 nA

Tensão negativa de alimentação: 0 a -15V

Tensão positiva de alimentação: +5 a +15V

Faixa de tensão diferencial de entrada : $\pm 30\text{V}$

Ganho de tensão : 200 V/mV típico

Tempo de resposta: 200ns típico

Tensão de off-set da entrada : 2mV típica

Corrente mínima drenada pela saída : 50 mA

Devido a baixa tensão de off-set de entrada ser menor que $1/2$ LSB do conversor, dispensamos o ajuste de off-set. O resistor de 470Ω de pull-up da saída, é o valor apresentado nas curvas de características do fabricante. A corrente drenada pela saída fica assim limitada em aproximadamente 10mA.

3.4 O AMOSTRADOR DA ENTRADA

Assumindo que a tensão no capacitor de retenção não deve variar mais do que $1/2$ LSB durante todo o intervalo de conversão, há que se especificar um valor mínimo para este capacitor em função da corrente de polarização do comparador. O tempo é de aproximadamente 8 períodos do clock, ou seja, $8 \mu s$. $1/2$ LSB é aproximadamente 6mV de variação e a corrente máxima é de 250 nA.

$$C \geq \frac{1}{\Delta V} \int_0^t i \cdot dt \quad (3.5)$$

Com $\Delta V = 6mV$, $i = 250nA$ e $t = 8\mu s$, a eq. 3.5 nos dá que $C \geq 333pF$.

Assumindo que o período de amostragem é maior que 1μs e utilizando-se 2 chaves analógicas em paralelo podemos estimar o valor máximo do capacitor de retenção.

Cada chave CMOS apresenta uma resistência típica de 300Ω quando alimentada com +5V. Duas em paralelo apresentam resistência de 150Ω .

Com 6,7 constantes de tempo a tensão atinge 99,88 % da tensão final, ou seja, $1/3$ LSB para o pior caso.

$$\text{Logo } C \leq 6,7 \times 150 = 1,0 \text{ nF}$$

Adotamos o valor de 1nF por causa do ruído introduzido pelo sinal de controle da chave, e pelo sinal de entrada.

Foi verificado que há acoplamento "AO" da entrada para a saída com a chave aberta, causando um erro de aproximadamente 5 % no sinal amostrado, quando o sinal de entrada é uma onda quadrada de amplitude máxima. A chave CMOS utilizada não apresentou resultados satisfatórios, devendo ser substituída por um outro tipo de chave, como por exemplo JFET.

3.5 O AMOSTRADOR DA SAÍDA .

Foi utilizado um amostrador semelhante ao da entrada.

E necessário um BUFFER entre o conversor e a chave analógica, porque a saída do conversor possui uma alta impedância, 1470Ω .

O BUFFER necessita ter uma larga faixa de passagem e um alto SLEW-RATE para acompanhar a saída do conversor. Um SLEW-RATE de $3V/\mu s$ e uma faixa de passagem de $1MHz$ é satisfatório. A saída do conversor A/D varia no máximo $3V$, quando o dado vindo da memória muda bruscamente. Como o intervalo de amostragem é maior que $10 \mu s$ é razoável supor que $1\mu s$ seja o tempo de estabilização da saída. Como a frequência de amostragem é menor que 100.000 amostragens/segundo, também é razoável que se defina a faixa de passagem como muito maior que a frequência de amostragem.

O C.I. LM 310 supera muito todas as características exigidas. Suas principais características são :

Tensão de OFFSET da entrada :	$2,5mV$	típica
Corrente de polarização da entrada :	$2,0nA$	típica
Capacitância de entrada :	$1,5 pF$	típica
Canho de tensão de grandes sinais :	$0,9999V/V$	típico
Resistência de saída :	$0,75 \Omega$	típica
Deriva térmica :	$10\mu V/^\circ C$	típica
Tensão de alimentação positiva :	$+5V$	a $+18V$
Tensão de alimentação negativa :	$-5V$	a $-18V$

3.6 A DETERMINAÇÃO DAS TENSÕES DE ALIMENTAÇÃO

As tensões típicas de alimentação do conversor D/A são de $+10V$ e $-10V$. Os amplificadores operacionais, BUFFERS e COMPARADORES podem ser alimentados também com estas tensões.

A lógica foi alimentada com $+5V$ com prejuízo da velocida-

de, mas evitando-se conversões de nível lógico para interfaceamento com o microcomputador.

A estabilidade da fonte não é crítica devido a insensibilidade dos C.I.s à variação desta. Cuidados devem ser tomados quanto a corrente de polarização dos diodos de referência. Logo a fonte de +10V deve ser estável.

3.7 O CONTROLE DE INÍCIO E FIM DE CONVERSÃO

Na figura 3.3 é mostrado o diagrama de tempo relativo ao clock do conversor. O sinal de início de conversão (\overline{SC}) é gerado por um oscilador de onda quadrada externo. O clock de 1MHz é obtido por divisão do clock de 4MHz utilizado pelo microprocessador, por 2 flip-flops tipo D. Quando a conversão chega ao fim, Q8 vai para "1" e Q8 então inibe o clock de 1MHz, através / do "CLEAR" dos flip-flops. Com o clock inibido, Q8 permanece / estável. Com a descida do sinal de início de conversão (\overline{SC}) o "CLEAR" é retirado. Com o 1º pulso de "clock", Q8 desce e mantém o "CLEAR" dos flip-flops em "1". Após a descida de Q8 o sinal \overline{SC} pode voltar ao nível "1".

Foi incluído uma rede RC para garantir que o sinal \overline{SC} esteja em "1", antes do sinal de fim de conversão (Q8) ir para "1", evitando-se assim o reinício da conversão indevidamente. Na figura 3.4 temos o diagrama de tempo que relaciona o sinal / de fim de conversão (Q8) com o sinal de início de conversão / (\overline{SC}) antes e depois do capacitor. O capacitor deve manter o nível "0" na entrada do "E" por mais de 870ns para que o tempo de Q8 ir para o "0", O nível "1" deve ser atingido antes / que Q8 vá para "1", isto é, antes de 8μs. O tempo em que Q8 fica em "0" é sempre de 8μs, porque Q8 desce com o 1º pulso de clock e sobe com o 9º pulso de clock.

O "E" utilizado é CMOS, cuja mudança de nível lógico 6"

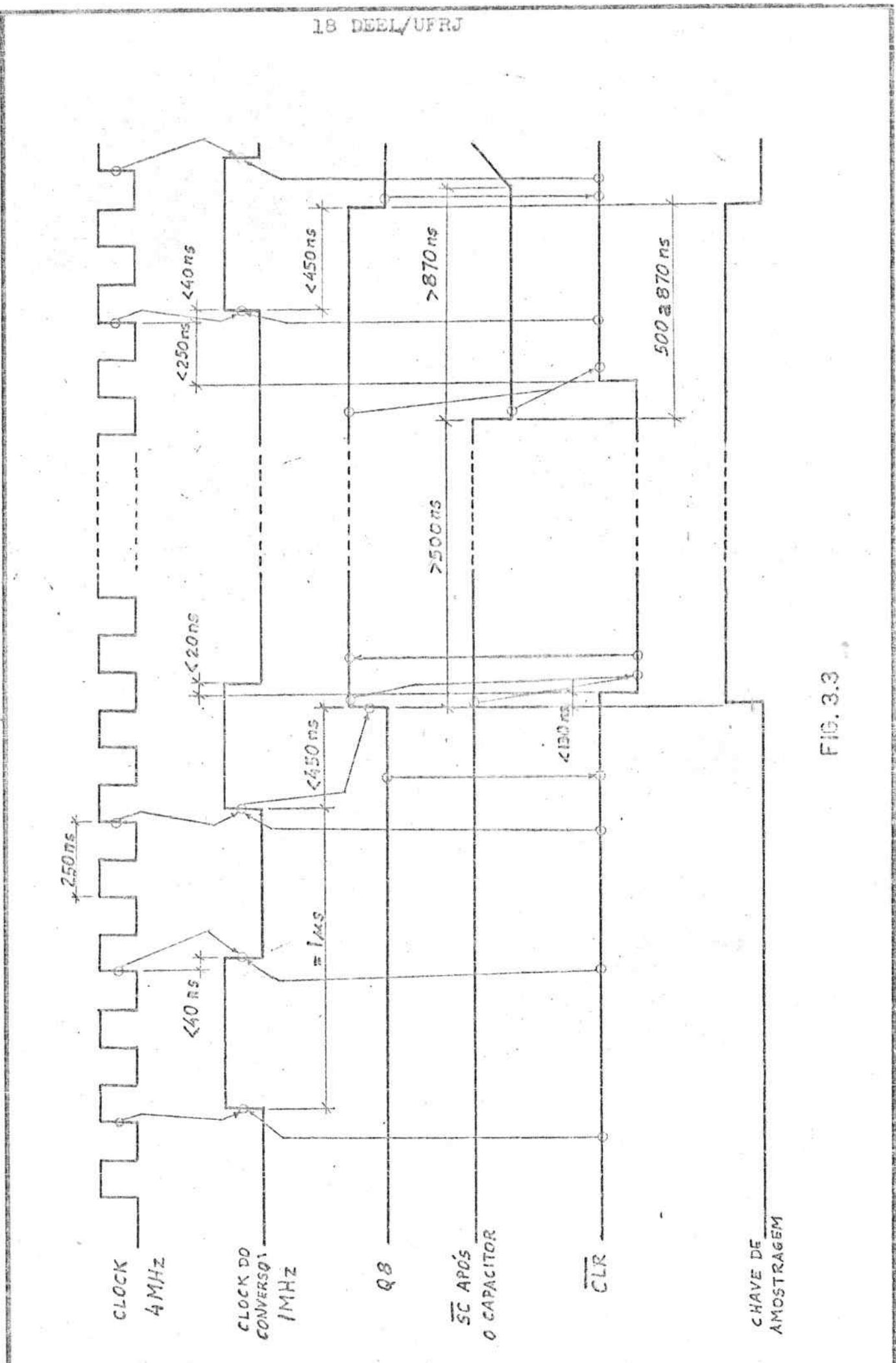


FIG. 3.3

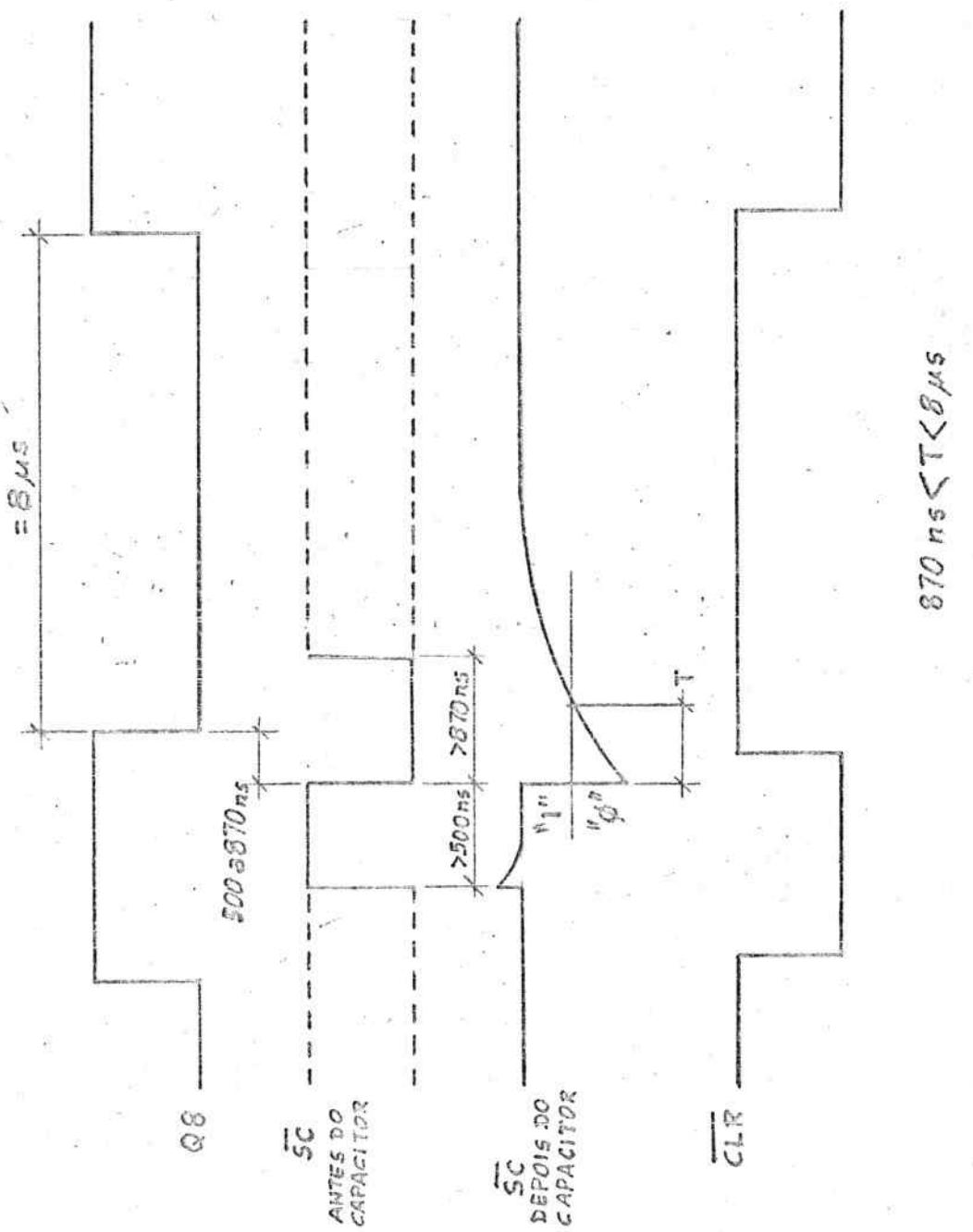


FIG. 3.4

por volta de 2,5V. O tempo necessário para a tensão no capacitor atingir 50 % do valor final (5V) é de 0,7 constante de tempo .

Então :

$$870\text{ns} < 0,7 \cdot RC < 8\mu\text{s} \quad (3.6)$$

Da inequação 3.6 concluímos que a constante RC deve ser maior que $1,3\mu\text{s}$ e menor que $11\mu\text{s}$. Na prática uma constante de tempo de $3,3\mu\text{s}$ não funcionou com um capacitor de 100pF . A constante na prática deve estar entre $1\mu\text{s}$ e $1,5\mu\text{s}$ para $C = 100\text{pF}$. Utilizamos a constante de $1,0\mu\text{s}$ com um capacitor de 100pF e um resistor de $10\text{k}\Omega$, com resultados satisfatórios.

3.8 TEMPO DE CONVERSÃO

Pelo diagrama de tempo da fig. 3.3 , podemos notar que a conversão começa no máximo 870ns , após a descida do sinal de início de conversão .

A conversão começa no 1º pulso de clock e termina no 9º . Sendo assim , temos 8μs para a conversão . O valor final se estabiliza com a descida de Q7 , SET do flip-flop do LSB, no 9º pulso de clock com um atraso de flip-flop de 175ns .

Somando-se todos os tempos , 870ns , 8μs e 175ns , obtemos 9,045μs . Isto é o dado está estável no barramento após 9,1μs do pulso de início de conversão .

3.9 O OSCILADOR DE FREQUÊNCIA VARIÁVEL

O oscilador é utilizado para controlar a frequência de amostragem do conversor A/D.

Foi utilizado um oscilador baseado no C.I. 555. O diagrama esquemático está na figura 3.5 . A frequência do oscilador é dada por(ref. 3) :

$$f = \frac{1,443}{(R_a + 2 R_b)C} \quad (3.7)$$

R_a é a resistência entre os pinos 7 e 8 . R_b é a resistência entre os pinos 2 e 7 . C é o capacitor selecionado pela chave rotativa.

Como a frequência máxima de amostragem é de 100kHz , incluímos um trimpot para ajustar o limite da frequência em 100kHz . O potenciômetro de 100 k Ω é utilizado para o ajuste de frequência dentro da faixa selecionada pela chave . O potenciômetro de 22 k Ω em série com o resistor de 47 k Ω permite o ajuste fino da frequência do oscilador . O resistor "pull-up" no pino 3 é para garantir que a saída atinja o nível l exigido pela entrada CMOS do conversor A/D .

O circuito utilizado funciona muito bem, tanto em frequências baixas como em frequências altas. A tabela 3.2 apresenta os resultados obtidos nas faixas selecionadas.

TABELA 3.2

	Faixa	Medidas
1 Hz	a 10 Hz	0,5 Hz a 11Hz
10 Hz	a 100 Hz	5,2 Hz a 120 Hz
100 Hz	a 1kHz	52 Hz a 1,2kHz
1kHz	a 10kHz	0,67kHz a 14kHz
10kHz	a 100kHz	6,2kHz a 105kHz

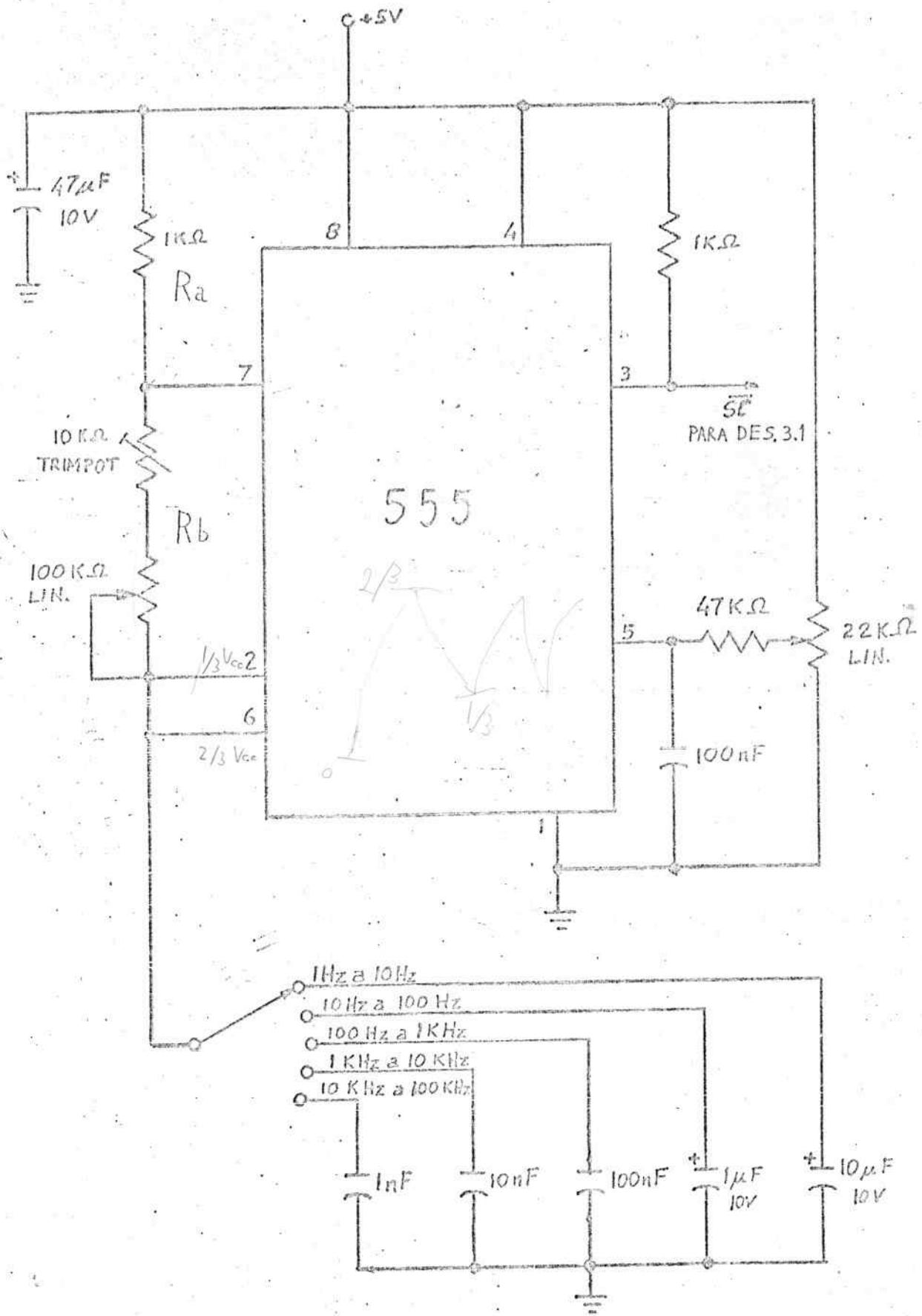


FIG. 3.5

CAPÍTULO 4

4.1 A LÓGICA DE CONTROLE DO CONVERSOR

São necessários para a lógica 3 sinais da CPU :

$\overline{\text{BUSAK}}$, $\overline{\text{RD}}$ e $\overline{\text{WR}}$.

Onde :

$\overline{\text{BUSAK}}$ - sinal de saída ativo em nível baixo que indica que a CPU está com os sinais $\overline{\text{WR}}$, $\overline{\text{RD}}$, $\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$ e com os barramentos de dados e endereços em alta impedância.

$\overline{\text{WR}}$ - sinal de saída ativo em nível baixo que indica para a memória ou dispositivo do E/S que a CPU está com os dados de saída estáveis no barramento.

$\overline{\text{RD}}$ - sinal de saída ativo em nível baixo que indica para a memória ou dispositivo do E/S que a CPU deseja ler dados.

Colocando-se resistores "PULL-UP" nas linhas $\overline{\text{RD}}$ e $\overline{\text{WR}}$, garante-se um nível lógico definido, quando a CPU estiver em alta impedância.

Introduzimos um flip-flop RS, controlado por 2 chaves de contato momentâneo, para selecionar a escrita ou leitura na memória pelo dispositivo externo. As saídas do flip-flop foram denominadas Q e \overline{Q} . QUANDO $Q = 1$ é habilitada a escrita e quando "0" é habilitada a leitura. Os sinais de leitura e escrita na memória foram denominados $\overline{\text{OE}}$ e $\overline{\text{WE}}$. O sinal vindo do conversor indicando que o dado está disponível é Q_8 .

Assim temos :

$$\overline{\text{OE}} = \overline{\text{RD}} \cdot (\overline{\text{BUSAK}} + Q) = \overline{\text{RD}} \cdot \overline{\text{BUSAK}} \cdot \overline{Q} \quad (4.1)$$

$$\overline{WE} = \overline{WR} \cdot (\overline{BUSAK} \cdot Q_7 \cdot Q_8) \quad (4.2)$$

Quando a CPU é que ocupa os barramentos ou a memória está fornecendo os dados para o conversor D/A, os flip-flops do registro de aproximações sucessivas devem estar em alta impedância, para evitar conflitos no barramento.

Assim temos :

$$E1 = BUSAK \cdot Q_7 \quad (4.3)$$

Substituindo a eq. 4.3 na Eq. 4.2, temos :

$$\overline{WE} = \overline{WR} \cdot (E1 \cdot Q_8) \quad (4.4)$$

Para que não apareça no osciloscópio o resultado de conversões D/A dos dados que estão sendo processados pela CPU, foi incluída uma lógica que inibe o amostrador de saída durante o processamento. O sinal de controle do amostrador de saída é $E2 = \overline{BUSAK} \cdot Q_8$

Dessa forma $E2 = "0"$ quando a CPU estiver ocupando o barramento de dados.

Como desejamos fazer acesso direto a memória, é necessário que a "RAM" seja selecionada quando a CPU não estiver ocupando o barramento. Sendo $\overline{Y1}$ a saída do decodificador que seleciona a "RAM", logo :

$$\overline{CS1} = \overline{Y1} \cdot \overline{BUSAK} \quad (4.6)$$

Como a escrita na memória pelo conversor D/A é quando Q_8 está em "1", a mudança para o próximo endereço pode ser com a transição de "1" para "0" do sinal Q_8 , devendo-se tomar precauções, para que o endereço não mude antes da retirada do sinal \overline{WE} . Como a lógica utilizada é TTL LS e o contador de endereços é CMOS, o atraso deste é suficiente para garantir que o sinal \overline{WE} seja desativado antes da mudança de endereço.

Utilizamos os contadores CMOS 4520. Cada C.I. possui 2 contadores síncronos de 4 BITS com entradas de clock de transi-

ção positiva e negativa. Para endereçar os 2048 bytes disponíveis para o acesso direto à memória são necessários 11 BITS.

Quando o 1º contador de 4 BITS chegar ao final da contagem e retornar a zero, deve ser dado o clock para o 2º contador. O mesmo ocorre para o 2º. Assim a saída do BIT mais significativo do 1º contador deve ser ligada ao clock de transição negativa do 2º. O mesmo deve ser feito com o 2º em relação ao 3º.

O sinal QS é clock de transição negativa do 1º contador.

Foi incluída uma chave rotativa de 2 polos e 4 posições para selecionar o número de endereços desejados: 256, 512, 1024 ou 2048. EXEMPLO para 256 pontos:

Para endereçar 256 pontos são necessários 8 BITS. O RESET nos contadores é dado quando a contagem chega à 256, ou seja, quando AB vai para "11". O RESET é dado então com A3. Como a duração do RESET é pequena, apresenta problemas de sincronismo para o osciloscópio. Assim utiliza-se a descida de A7 para sincronizar o osciloscópio.

O osciloscópio sempre inicia a varredura quando o endereço é zero.

Voltando ao FLIP-FLOP RS, é desejável que a escrita na memória seja desabilitada pela chave de fim de amostragem (FA), sómente após ter sido escrito o dado no último end. selecionado. Para desabilitar a escrita devemos dar o RESET no flip-flop. Assim : $R=FA$. Ai , onde $i = 8$ à 11 (Eq.4.7)

Para 256 pontos, $R=FA$. AB e assim por diante.

Há uma solução econômica de fazer a seleção do end. pelo contador ou pela CPU. É a utilização de resistores em série com os contadores.

Quando é a CPU que endereça, se os resistores forem de

valor razoavelmente alto, as correntes nos resistores serão baixas e o endereço enviado às memórias será fornecido pela CPU.

O fabricante garante que a corrente mínima fornecida pela CPU em nível alto (2,4V) é de 250 μ A e em nível baixo (0,4V) é -1,8mA. Podemos agora estimar um valor mínimo para os resistores.

$$R_{\min} = \frac{2,4}{250 \mu A} = 9,6 \text{ k}\Omega \quad (4.8)$$

$$R_{\min} = \frac{0,4V - V_{cc}}{-1,8mA} = \frac{-4,6V}{-1,8mA} = 2,6 \text{ k}\Omega \quad (4.9)$$

Logo :

$$R_{\min} = 10 \text{ k}\Omega \quad (4.10)$$

Da descida de Q8 de "1" para "0" até a próxima escrita (subida de Q3 de "0" para "1"), temos 8 μ s. O endereço dado pelo contador deve se estabilizar em 8 μ s no máximo. O pior caso é o de uma linha de endereço ir de "1" para "0", pois vai de 5V à apenas 0,4V.

Considerando um circuito RC passa-baixa com um degrau de 5V para 0V na entrada, a saída atingirá 0,4V após $2,5RC$, sendo C a capacidade parasita.

Cada entrada MOS tem cerca de 10pF. No caso temos uma CPU, uma EPROM e 3 RAMs como carga. Somando as capacidades obtemos um total de 50pF.

Como o tempo máximo de estabilização do endereço é de 8 μ s então a constante de tempo RC deve ser menor que 3 μ s, já que são necessárias 2,5 constantes de tempo.

Tirando o valor máximo de R:

$$R_{\max} = \frac{3\mu s}{50\text{pF}} = 60 \text{ k}\Omega \quad (4.11)$$

De (4.10) e (4.11) :

$$10 \text{ k}\Omega < R < 60 \text{ k}\Omega \quad (4.12)$$

Escolhemos $R = 22 \text{ k}\Omega$, que funcionou perfeitamente com o tempo de estabilização de $4\mu\text{s}$ na montagem em "PROTOBOARD".

Esta solução não permite no entanto que sejam ligadas entradas TTL nas linhas de endereços de A0 até A11. Foram desprezadas as correntes de fuga dos dispositivos "MOS" no cálculo de R máx.

Reescrevendo novamente as equações 4.1, 4.3, 4.4, 4.5 e 4.7 :

$$\overline{\text{OE}} = \overline{\text{RD}} \cdot (\overline{\text{BUSA}} \wedge \overline{\text{Q}}) \quad (\text{Eq. 4.1})$$

$$\overline{\text{E1}} = \text{BUSA} \wedge \text{G} \quad (\text{Eq. 4.3})$$

$$\overline{\text{WE}} = \overline{\text{WR}} \cdot (\overline{\text{E1}} \wedge \overline{\text{Q3}}) \quad (\text{Eq. 4.4})$$

$$\text{E2} = \text{BUSA} \wedge \text{Q8} \quad (\text{Eq. 4.5})$$

$$\text{R} = \text{PA} \cdot \text{A}_i \quad , \text{ onde } i = 8, 9, 10, 11 \quad (\text{Eq. 4.7})$$

O diagrama esquemático corresponde a essas equações está na figura 4.1 .

Foi aproveitado o ponto decimal do DISPLAY para indicar que o flip-flop está armado para escrita, quando o ponto está aceso . O diodo em série com o DISPLAY é para garantir que a saída Q atinja o nível "1" .

4.2 OS DIAGRAMAS DE TEMPO

Na figura 4.2 temos o diagrama de tempo para o caso de estar sendo feito acesso direto à memória pelo conversor A/D. Ao

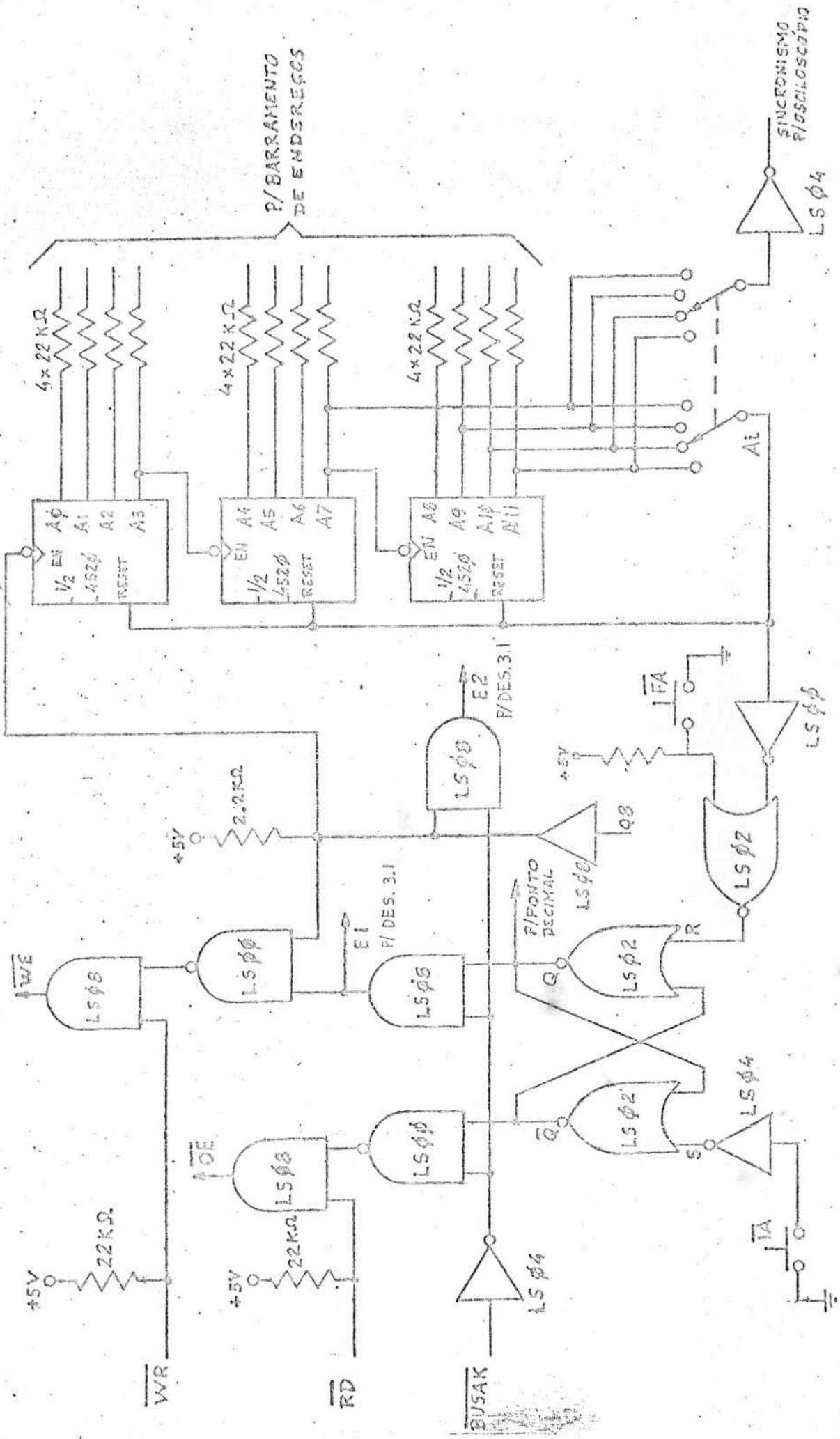


FIG. 4.1



FIG. 4.2

iniciar a conversão o sinal Q8 desce e avança o contador de endereços. Sendo o endereço anterior o último, o próximo será o endereço inicial através do RESET dado pelo BIT A_1 do endereço. Para o caso de 256 endereços, $i=8$. Sendo assim o endereço é momentaneamente 256, e volta ao endereço \emptyset . Se a chave de fim de amostragem estiver pressionada, $\overline{FA}=\emptyset$, quando A_1 ir para "1", será dado o RESET no flip-flop, desabilitando a escrita, $Q=\emptyset$. Se for pressionada a chave de início de amostragem a escrita será habilitada novamente, $Q=1$.

Nas figuras 4.3 e 4.4 temos os diagramas de tempo para o caso de escrita do dado na memória pelo conversor A/D e leitura do dado da memória pelo conversor D/A, respectivamente. $\overline{BUSAk} = 0$ indica que a CPU não está ocupando os barramentos. $Q=1$ permite que o dado seja escrito na memória. No caso de escrita, quando a conversão A/D termina Q8 vai para "1" e \overline{WE} vai para " \emptyset " escrevendo o dado na memória no endereço atual. Ao iniciar outra conversão, Q8 vai para " \emptyset " e \overline{WE} vai para "1" desabilitando a escrita na memória. A descida de Q8 avança o contador de endereços para o próximo endereço e a situação se repete até que seja pressionada a chave de fim de amostragem (\overline{FA}) e seja escrito o dado no último endereço selecionado conforme a figura 4.2. Após a escrita ser desativada pela chave de fim de amostragem, $Q=\emptyset$ e $\overline{OE}=0$, isto é, a leitura do dado armazenado na memória é permanente. O sinal de fim de conversão Q8 continua sendo gerado apesar de não estar sendo feita a conversão A/D pois, a saída do registro de aproximações sucessivas fica desabilitada, $E_1=0$. Q8 apenas muda o endereço neste caso. Ao mudar o endereço, a memória fornece o dado correspondente ao endereço. O dado é então, enviado ao conversor D/A. A saída do conversor possui um amostra-

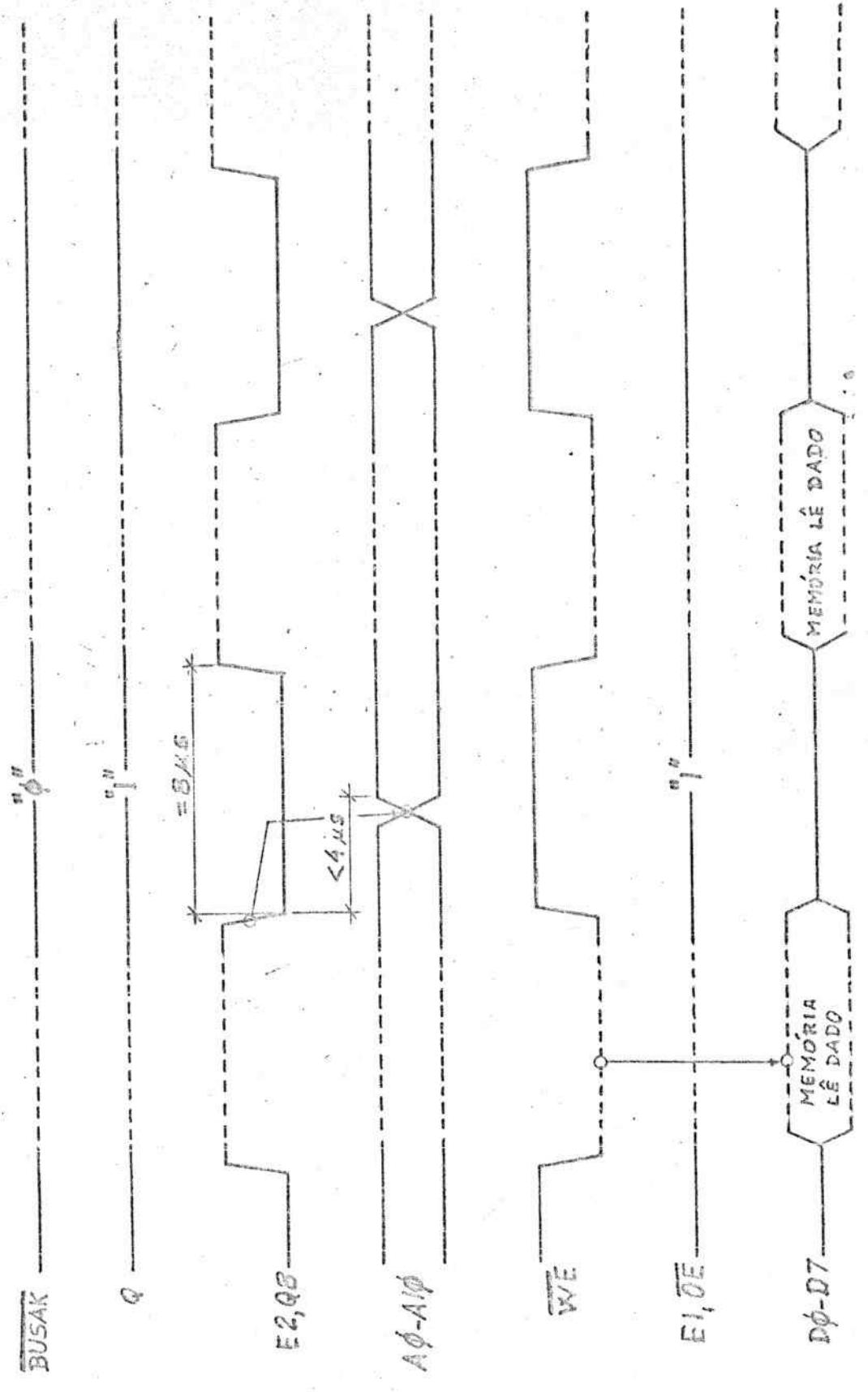


FIG. 4.3

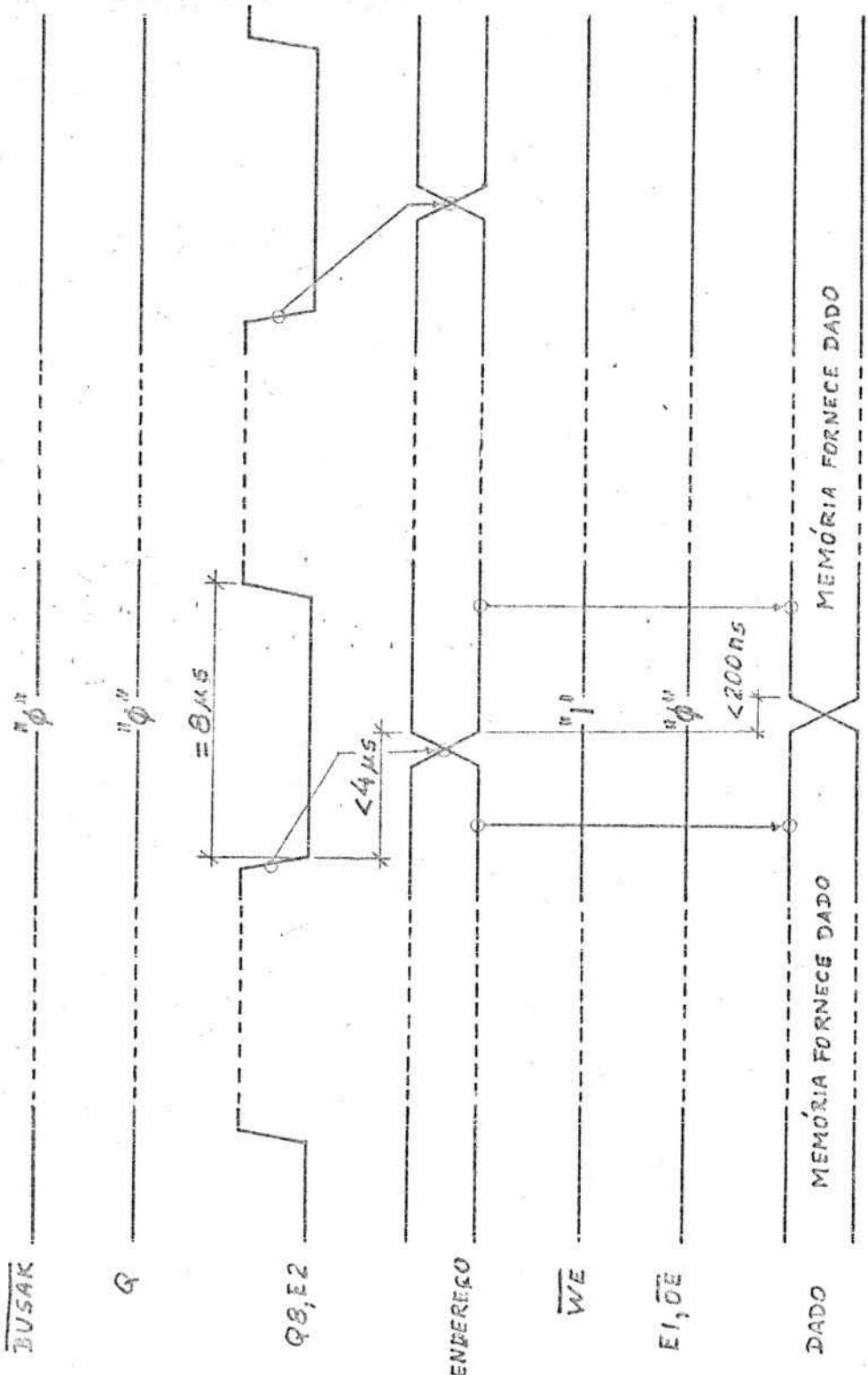


FIG. 4.4

dor que é ativado quando o dado está estável. A chave do amostrador só abre quando está sendo feita a mudança de endereços ou a conversão A/D.

Na figura 4.5 está o diagrama de tempo simplificado, quando a CPU está utilizando o barramento, $\overline{BUSA} = "1"$. Neste caso a saída do registro de aproximações sucessivas e o amostrador de saída estão desabilitados, $E1 = "0"$ e $E2 = "0"$ respectivamente. Devemos observar que neste caso os sinais \overline{WE} e \overline{OE} dependem exclusivamente da CPU. Diagramas mais detalhados para este caso encontram-se no próximo capítulo.

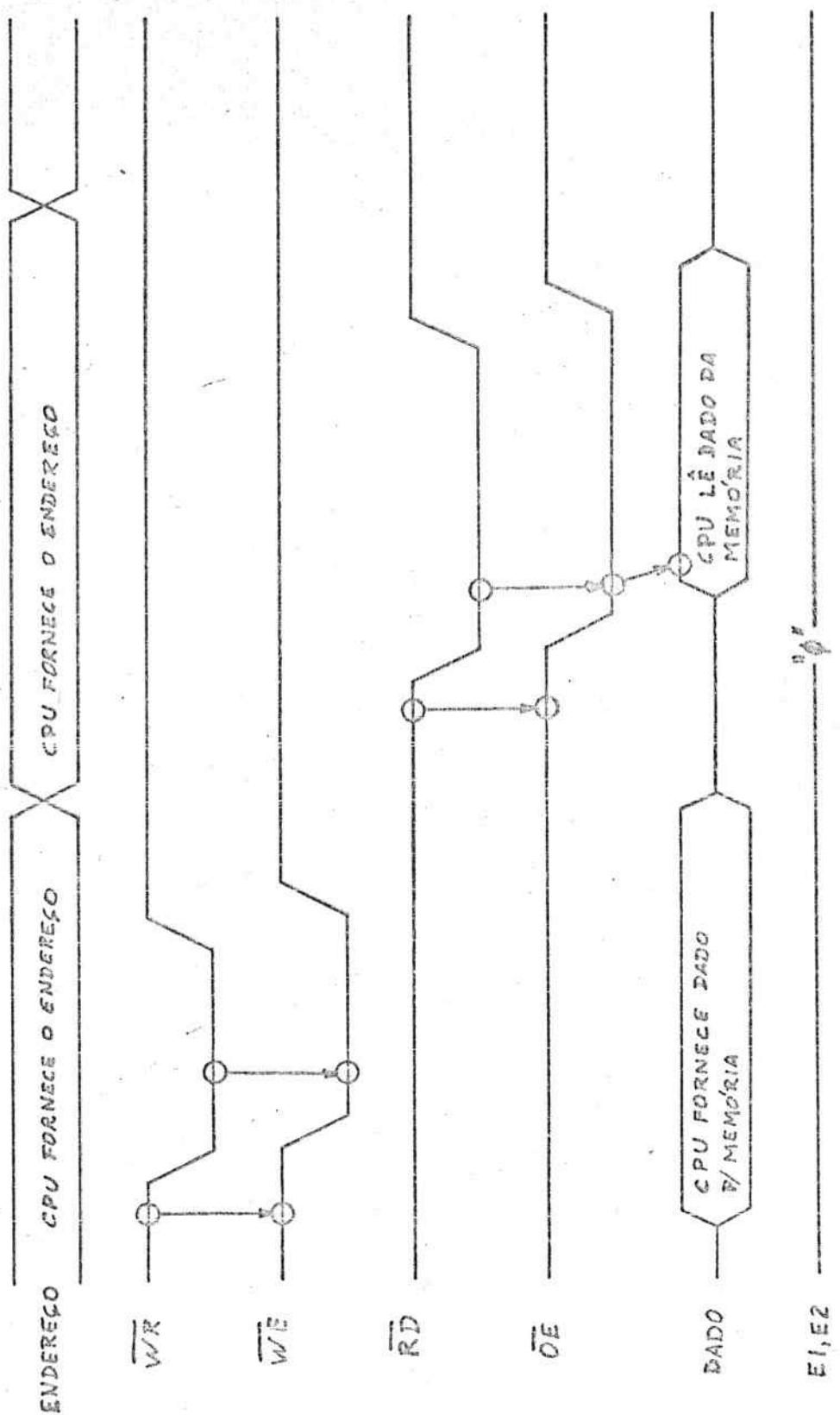


FIG. 4.5

CAPÍTULO 5

O MICROCOMPUTADOR

O diagrama do microcomputador está na figura 5.1.

5.1 ESCOLHA DA CPU

Foi escolhida para este projeto a CPU Z-80A projetada pela ZILOG INC. por diversas razões: baixo custo, facilidade de ser encontrada, necessita um "HARDWARE" externo simples, grande velocidade e grande número de instruções poderosas que facilitam o "SOFTWARE". Possui 2 conjuntos de registradores de 8Bits semelhantes ao conjunto do 8085 e mais 2 registros de 16 BITS para endereçamento indexado além do "PROGRAM COUNTER" (PC) e do "STACK POINTER" (SP). Opera com um clock de até 4MHz. Uma grande vantagem em relação a outros microprocessadores como o 6502 e 6809 é que o "SOFTWARE" pode ser desenvolvido em microcomputadores caseiros de baixo custo que também utilizam a CPU Z-80A, microcomputadores da linha "SINCLAIR".

5.2 DESCRIÇÃO DOS SINAIS DA CPU

A₀ - A₁₅ : Barramento de endereços, é "3-STATE".

D₀ - D₇ : Barramento de dados, é bi-direcional e "3-STATE".

HREQ : Sinal de saída ativo em nível baixo. Indica que o barramento de endereços, está com o endereço de memória válido.

IORQ : Sinal de saída ativo em nível baixo. Indica que o barramento de endereços está com o endereço de uma porta de entrada ou saída válido.

RD : Sinal de saída ativo em nível baixo. Indica que a CPU deseja ler um dado da memória ou de uma porta de entrada.

WR : Sinal de saída ativo em nível baixo. Indica /

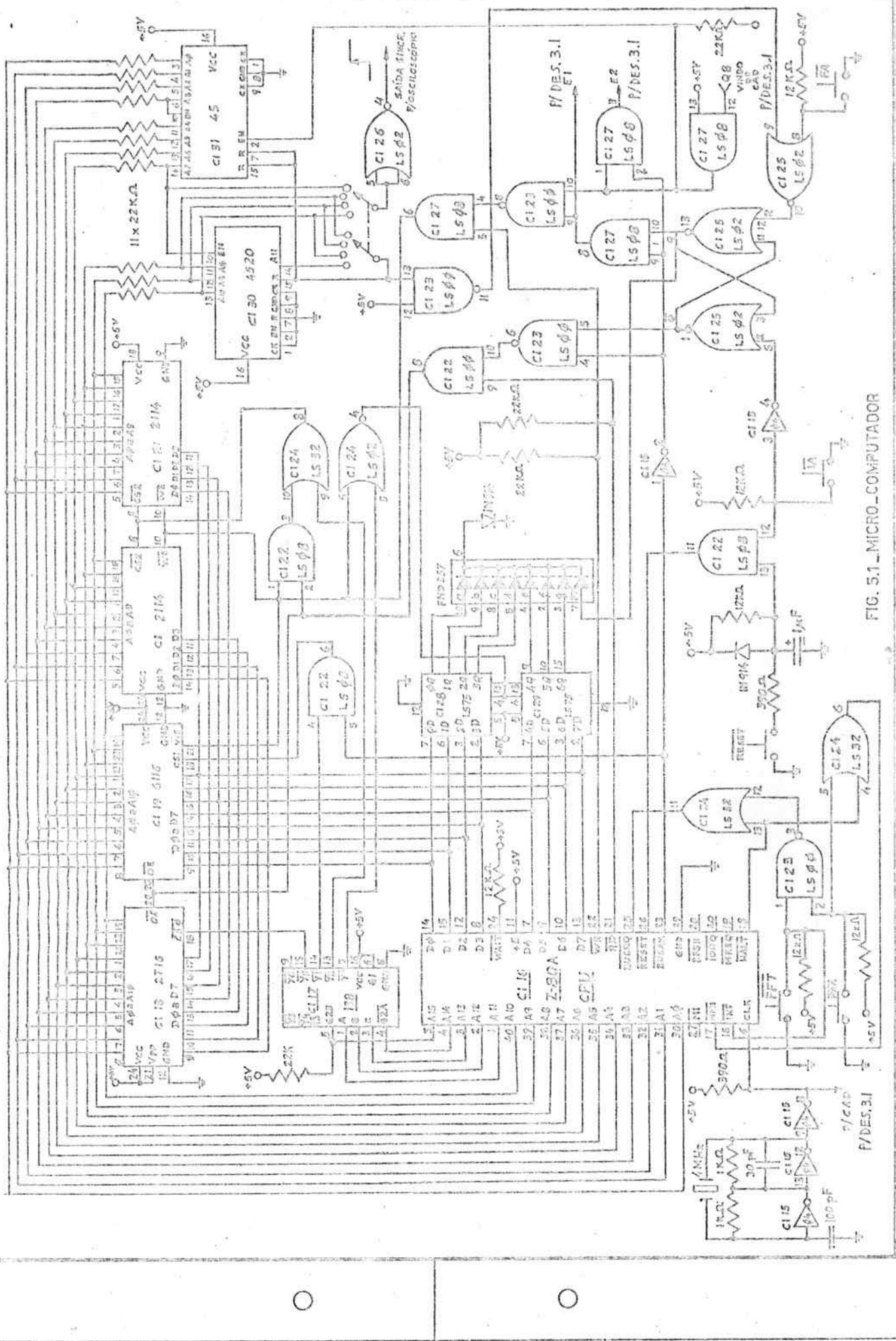


FIG. 5.1. MICRO-COMPUTADOR

- que o dado no barramento de dados está estável para ser escrito na memória ou numa porta de saída .
- MI : Sinal de saída ativo em nível baixo. Indica que a CPU está no ciclo de busca de instrução ("FETCH") .
- RFSH : Sinal de saída ativo em nível baixo . Indica que o endereço no barramento de endereços é válido para refrescar memórias dinâmicas .
- WAIT : Sinal de entrada ativo em nível baixo. É utilizado para sincronizar a CPU com memórias ou dispositivos de E/S lentos .
- BUSREQ : Sinal de entrada ativo em nível baixo. Pedido de um dispositivo externo que deseja utilizar os barramentos .
- BUSAK : Sinal de saída ativo em nível baixo. Resposta ao sinal BUSREQ indicando que os barramentos de dados e de endereços , as saídas MREQ , IORQ , RD e WR estão em alta impedância .
- HALT : Sinal de saída ativo em nível baixo. Indica que a CPU executou a instrução "HALT" e parou. A CPU só sai desse estado através de uma interrupção ou através do "RESET".
- NMI : Sinal de entrada ativo em nível baixo. É o pedido de interrupção não mascarável .
- INT : Sinal de entrada ativo em nível baixo. É o pedido de interrupção mascarável .
- RESET : Sinal de entrada ativo em nível baixo. O PC é zerado; todos os sinais de saída de controle vão para o estado inativo, nível alto. Os bar-

ramentos de dados e endereços ficam em alta impedância; o modo de interrupção zero é selecionado; a interrupção mascarável é desabilitada.

CLK : Entrada de "CLOCK".

5.3 DIAGRAMAS DE TEMPO DA CPU

Na figura 5.2 é mostrado o diagrama de tempo do ciclo de busca para a instrução "NOP". O ciclo de instrução é denominado M1 e os subsequentes M2, M3 e etc. Para o ciclo M1 são requeridos 4 períodos de clock denominados T1, T2, T3 e T4.

Na subida de T1 a CPU coloca o endereço da instrução no barramento de endereços. Na descida de T1 a CPU ativa \overline{MREQ} e \overline{RD} indicando que deseja ler dado da memória. Na descida de T2, habilita a entrada de dados. Na subida de T3 a CPU desativa \overline{MREQ} e \overline{RD} , indicando que já leu o dado. O dado deve estar disponível antes da subida de T3. Os ciclos T3 e T4 não são de nosso interesse, pois não utilizamos memórias dinâmicas.

Na fig. 5.3 é exemplificado o diagrama de tempo da execução da instrução "LD A_s (HL)", ler para o registrador A o dado da memória cujo endereço está em HL. Durante o ciclo M1 é lida e decodificada a instrução "LD A_s (HL)". Durante o ciclo de leitura da memória é colocado na subida de T1 o endereço da memória. Na descida de T1 a CPU ativa \overline{MREQ} e \overline{RD} e a memória fornece o dado. Na subida de T3 a CPU habilita a entrada do dado. Na descida de T3 a CPU desativa \overline{MREQ} e \overline{RD} , indicando para a memória que já leu o dado. O próximo ciclo é M1 novamente.

Na figura 5.4 é exemplificado o diagrama de tempo da execução da instrução "LD (HL), A", escrever na memória cujo endereço está em HL o dado contido no registrador A. Como no caso ante-

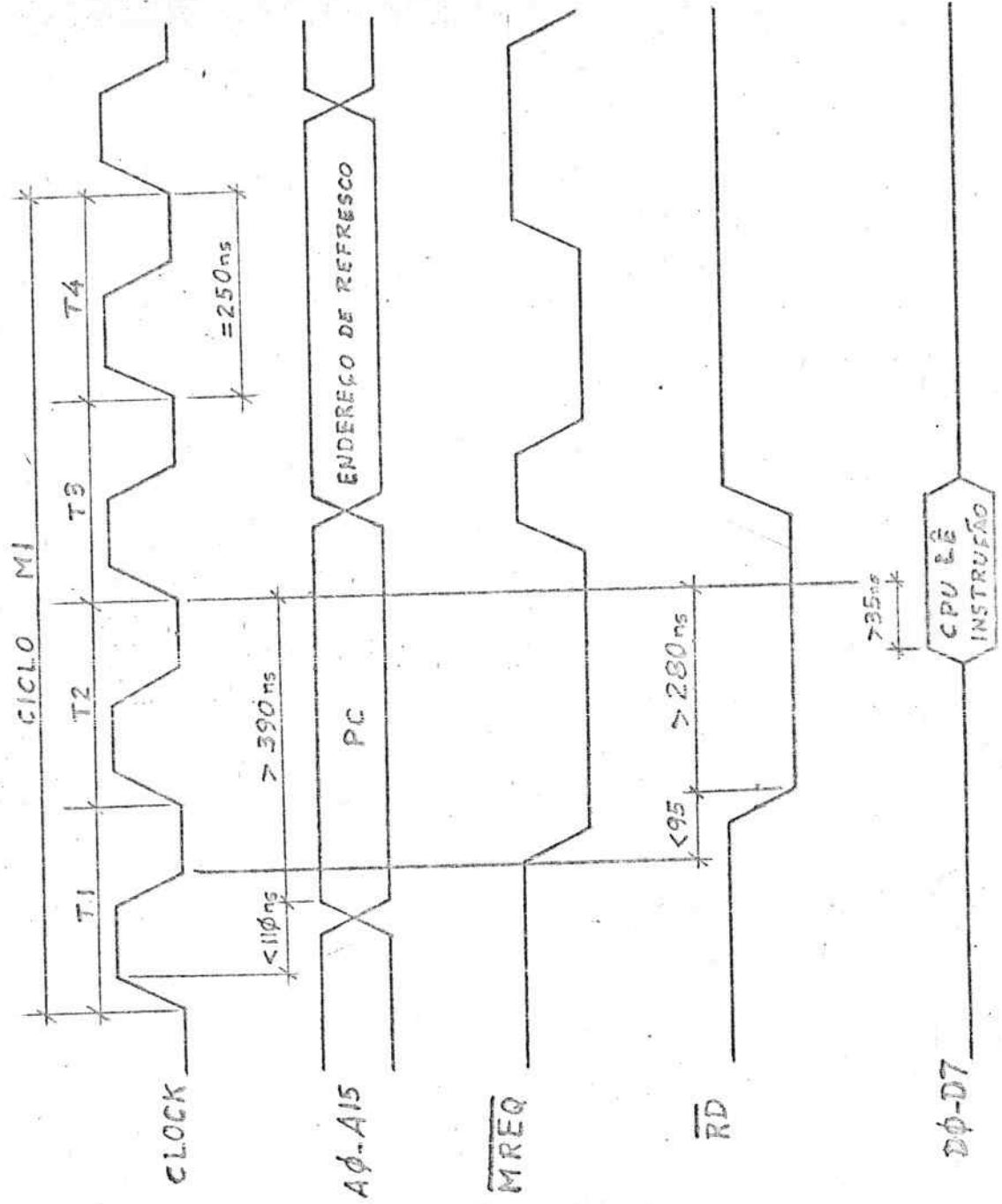


FIG. 5.2

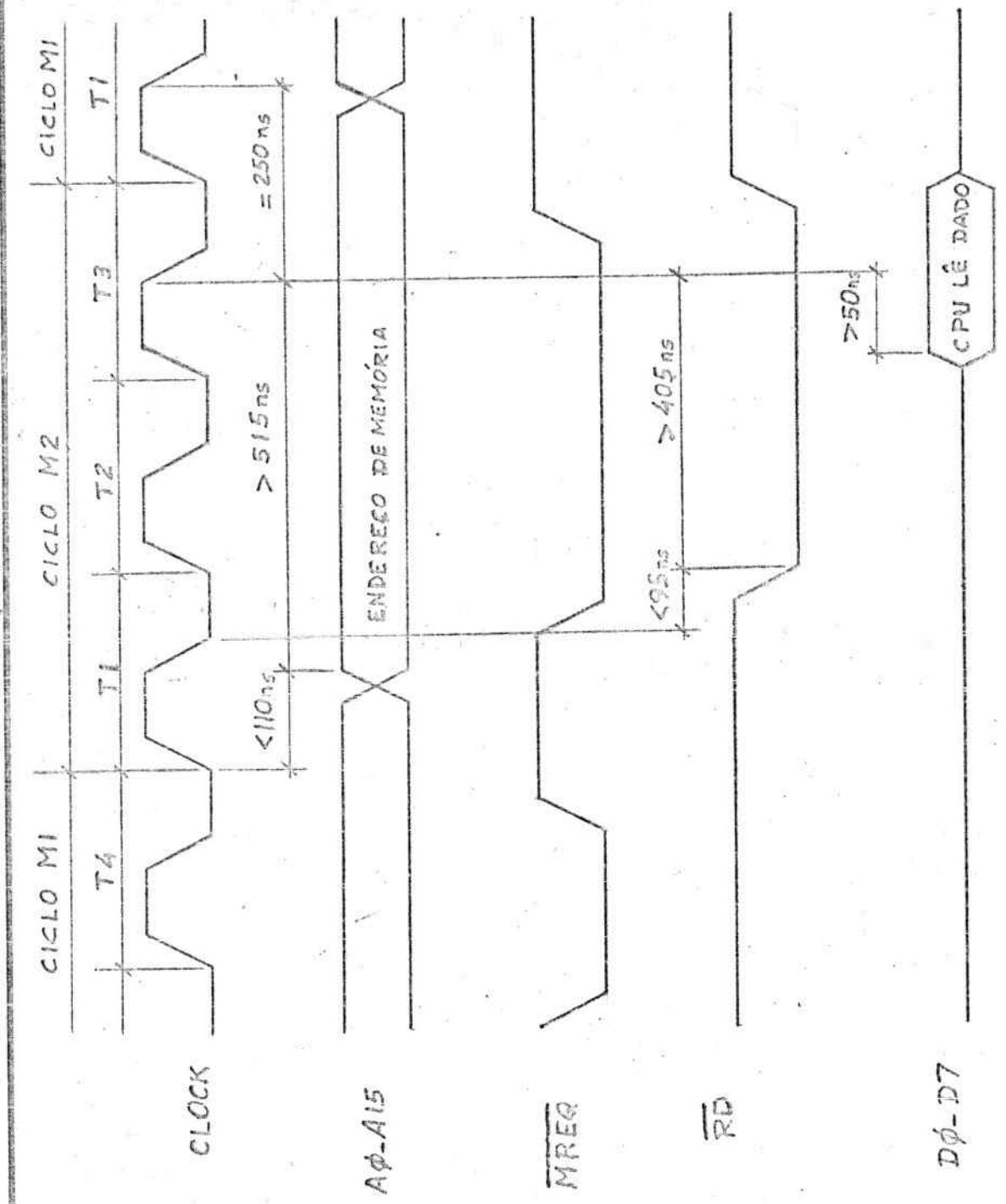


FIG. 5.3

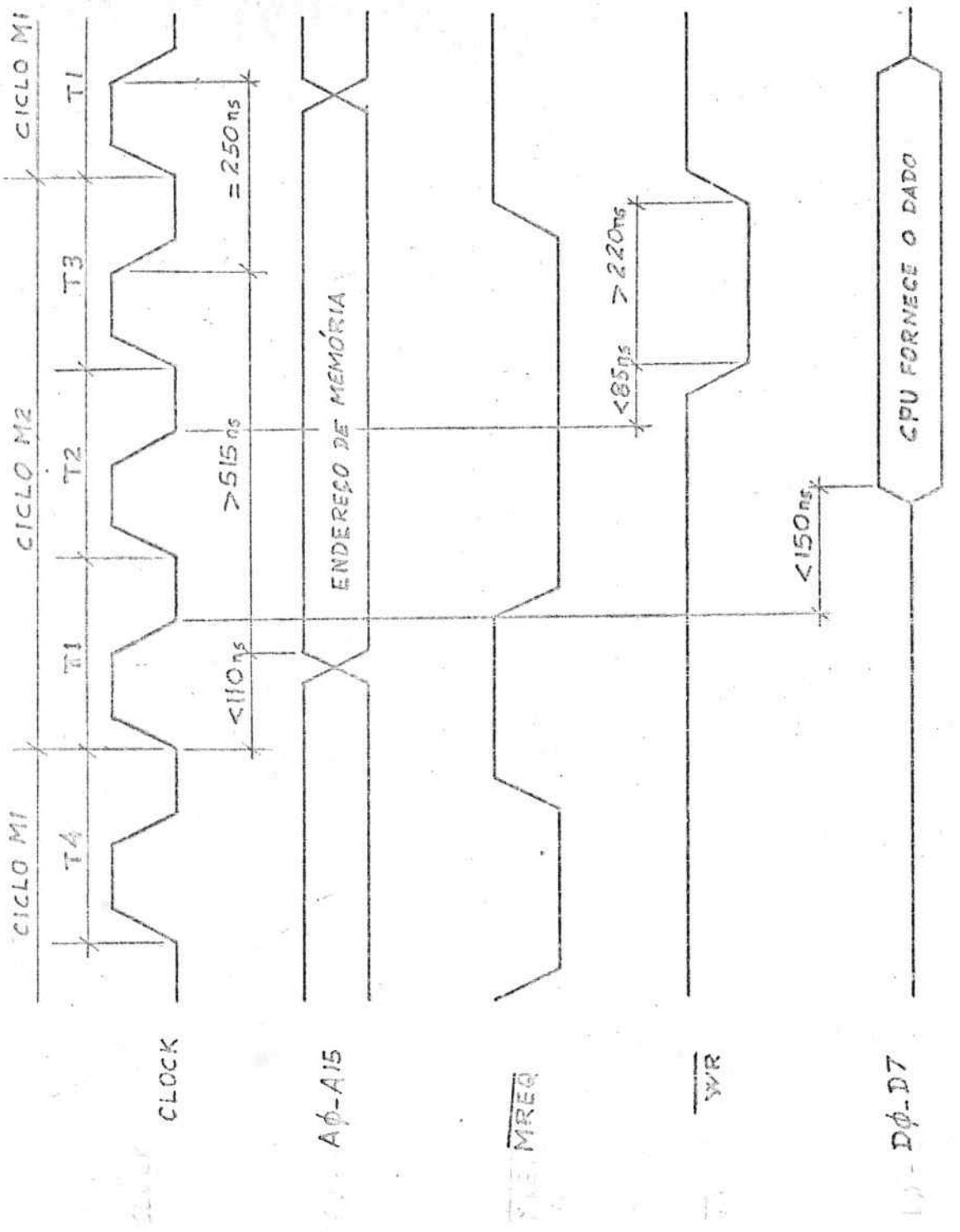


FIG. 5.4

rior é lida e decodificada a instrução no ciclo M1. Na subida de T1 a CPU coloca o endereço da memória onde será escrito o dado escrito o dado. Na descida de T1 é ativado MREQ e colocado o dado no barramento de dados. Na descida de T2 é ativado WR e a memória lê o dado. Na descida de T3 é desativado MREQ e WR e o dado está escrito na memória. Na subida de T1 do próximo ciclo M1 é retirado o dado do barramento. A CPU está sempre com DO à D7 em alta impedância, exceto no caso de escrita de dado na memória ou numa porta de saída.

Na figura 5.5 é mostrado o diagrama de tempo correspondente ao pedido dos barramentos. Na subida do clock do último período de uma instrução, a CPU verifica se BUSRL está ativo. Se BUSRL está ativo então na próxima subida do clock a CPU coloca em alta impedância os barramentos de dados e de endereços, os sinais de controle MREQ, IORD, RD, WR e ativa BUSAK indicando que os barramentos estão liberados. A CPU continua verificando na subida do clock se o pedido do barramento, BUSRL, ainda é ativo. Quando BUSRL for desativado a CPU reconhecerá na próxima subida do clock e desativará BUSAK na próxima descida do clock. Na última subida do clock a CPU iniciará um novo ciclo de instrução M1.

Na figura 5.6 é mostrado o diagrama de tempo da execução / da instrução "HALT" e a saída deste estado através de interrupção. A CPU executando a instrução "HALT", ativa na descida de T1 do próximo ciclo de máquina a saída HALT. A CPU fica executando instruções "NOP", porém o PC não é incrementado. Na subida de T4 a CPU verifica se tem algum pedido de interrupção. Caso INT ou NMI esteja ativo a CPU sai do estado de "HALT" e atende ao pedido de interrupção e após volta a continuar a executar o programa onde parou.

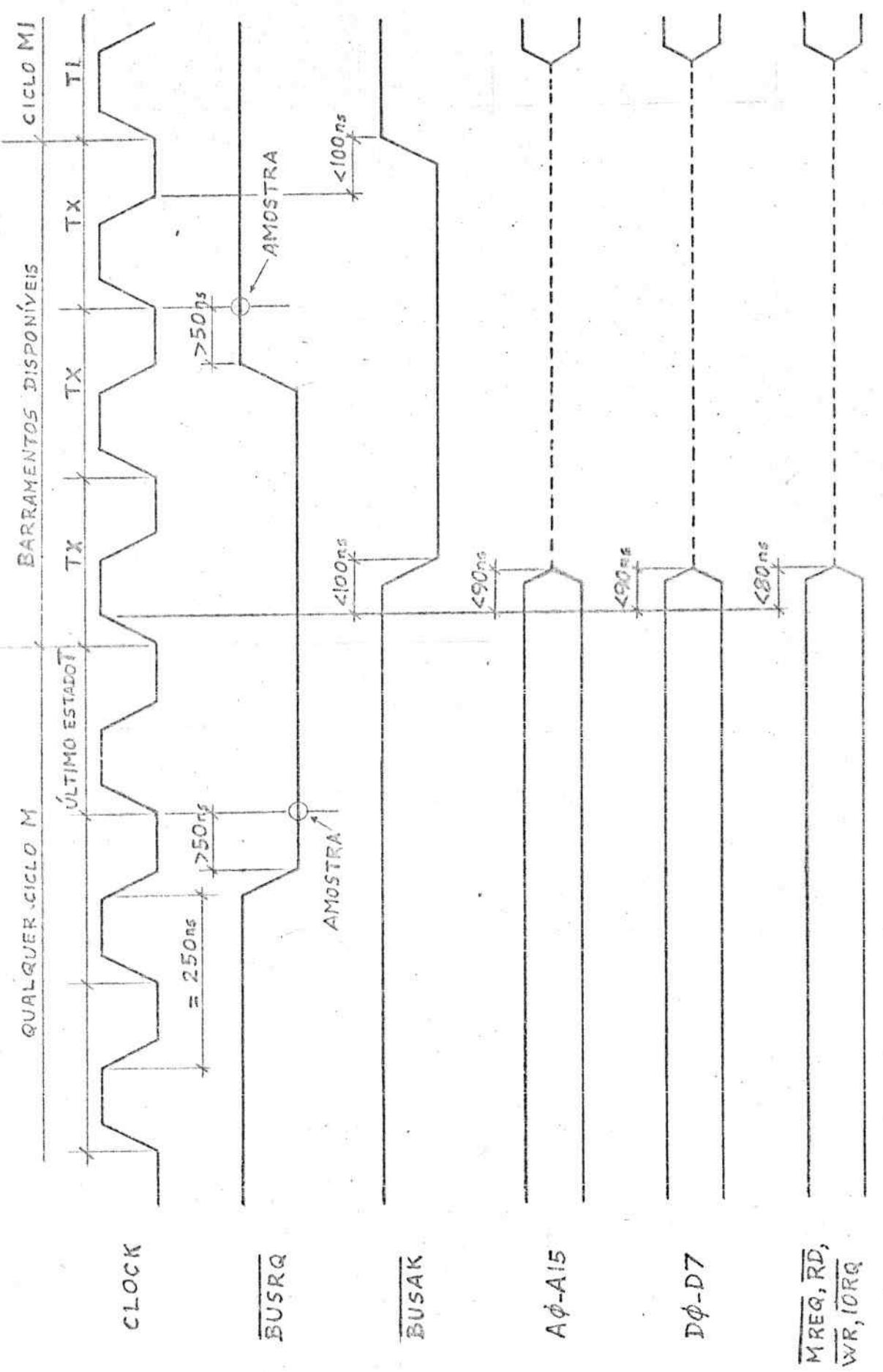


FIG. 5.5

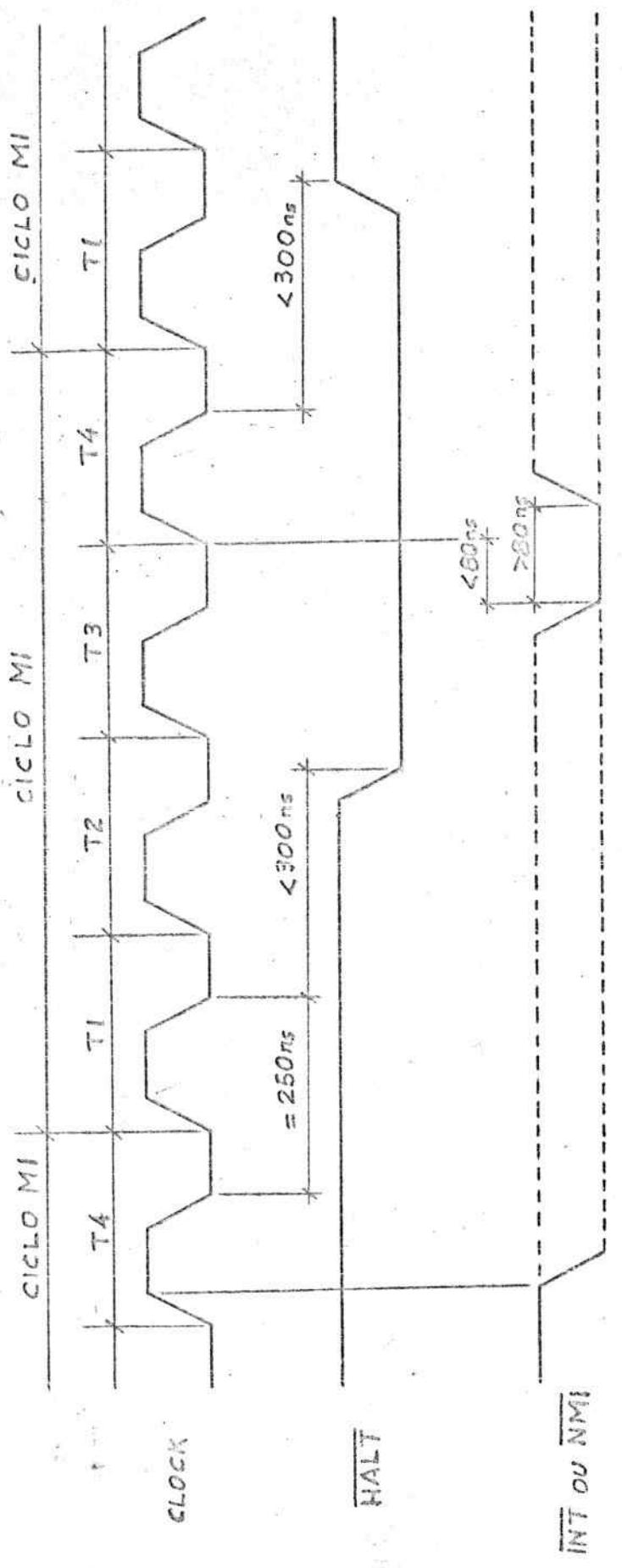


FIG. 5.6

5.4 A LÓGICA DE CONTROLE DO BARRAMENTO E DAS INTERRUPÇÕES

Desejamos comandar por "HARDWARE" a execução de 2 subrotinas, uma de troca de "página" e outra de FFT. Selecionamos a interrupção mascarável no modo 1 de interrupção do Z-80. No modo 1 a CPU executa uma subrotina que está no endereço 0038H quando INT é ativado. Quando NMI é ativado a CPU executa uma subrotina que está no endereço 0066H. A maneira encontrada foi deixar a CPU no "LOOP":

```
LOOP : HALT      ; CPU para
        JR LOOP    ; volte para LOOP .
```

Para executar uma das subrotinas, basta ativar a interrupção correspondente. A CPU executa a subrotina, volta ao "LOOP" e para novamente.

A CPU possui um flip-flop interno para a interrupção mascarável, não necessitando circuito adicional para eliminar a vibração da tecla. A interrupção não mascarável é por nível, necessitando este circuito.

Observando o diagrama de tempo da figura 5.6 notamos que quando a interrupção é reconhecida, HALT é desativado pela CPU. Assim sendo podemos utilizar este sinal para retirar a vibração da tecla. Logo:

$$\overline{\text{INT}} = \overline{\text{FFT}} \quad (5.1)$$

$$\overline{\text{NMI}} = \overline{\text{PAG}} + \overline{\text{HALT}} \quad (5.2)$$

Para que a interrupção possa ser reconhecida é necessário que a CPU saia do estado de alta impedância. Após a CPU sair de alta impedância, é necessário que ela mantenha este estado até o fim da sub-rotina de interrupção. Quando termina a sub-rotina a CPU volta para o "LOOP" de "HALT" e para. Assim sendo podemos utilizar o sinal HALT para manter a CPU no estado ativo. Logo :

$$\overline{\text{BUSRQ}} = \overline{\text{FFT}} + \overline{\text{PAG}} + \overline{\text{HALT}} = \overline{\text{FFT} \cdot \text{PAG}} + \overline{\text{HALT}} \quad (5.3)$$

O circuito é mostrado na fig. 5.7 e os diagramas de tempo, para cada tecla pressionada, são mostrados nas figuras 5.8e5.9.

O sinal BUSAK é utilizado pela lógica apresentada no capítulo 4 para o acesso direto à memória.

5.5 AS MEMÓRIAS

O "SOFTWARE" ocupa aproximadamente 1650 bytes de programa e 3 K bytes de dados. São necessários, portanto 2 K bytes de "EPROM" e K bytes do "RAM".

Para a "EPROM" foi escolhida a "2716" de 2 K bytes. Para a "RAM" foram escolhidas uma "6116" (2 K bytes) e duas "2114" (1 K nibbles), por ser a combinação mais econômica para obter 3K bytes de "RAM".

Para a decodificação de endereços foi utilizado o C.I. 74LS138, decodificador rápido de 3: 8. Na linha de endereço A15 foi colocado um resistor "PULL-UP" para garantir que o decodificador fique desabilitado durante o acesso direto à memória, para não ocorrer "conflito" no barramento de dados. Não é necessária a utilização do sinal MREQ para indicar que o endereço no barramento é de memória, porque não utilizamos os endereços de entrada e saída.

5.6 ESPECIFICAÇÃO DOS TEMPOS DE ACESSO DAS MEMÓRIAS

Para cálculo destes tempos é necessário conhecer as características da CPU para um clock de 4MHz (Ref.4).

Temos 3 casos :

1 - Leitura da instrução na "EPROM", ciclo M1 .

2 - Leitura de dado na "EPROM" ou na "RAM" , ciclo M2 em diante.

3 - Escrita de dado na "RAM" , ciclo M2 em diante.

Devemos utilizar os diagramas de tempo das figuras 5.2 ,

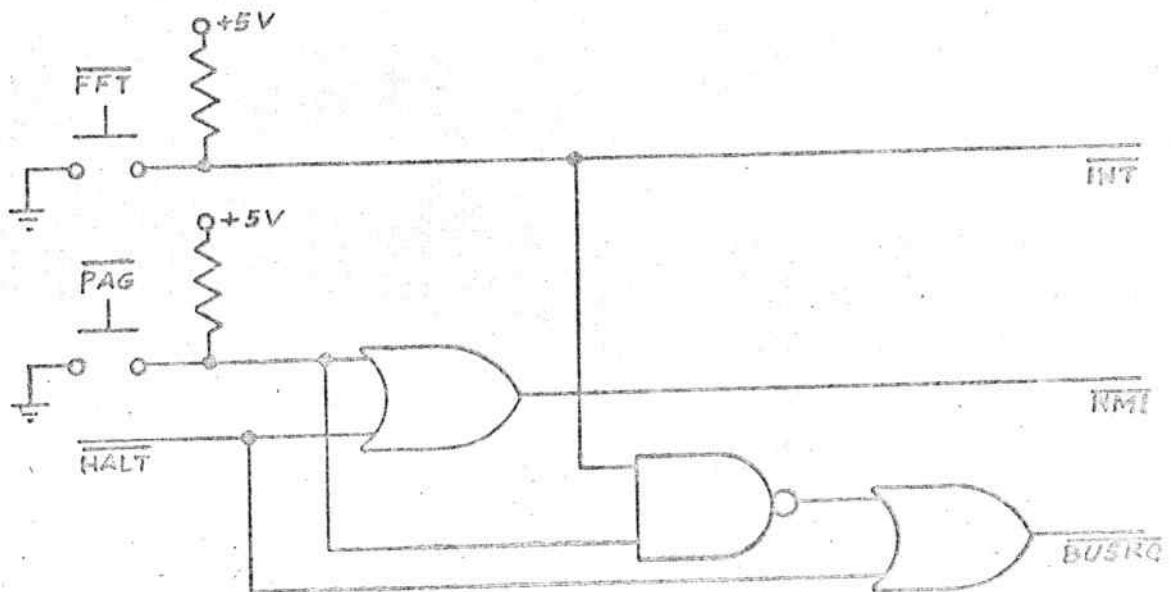
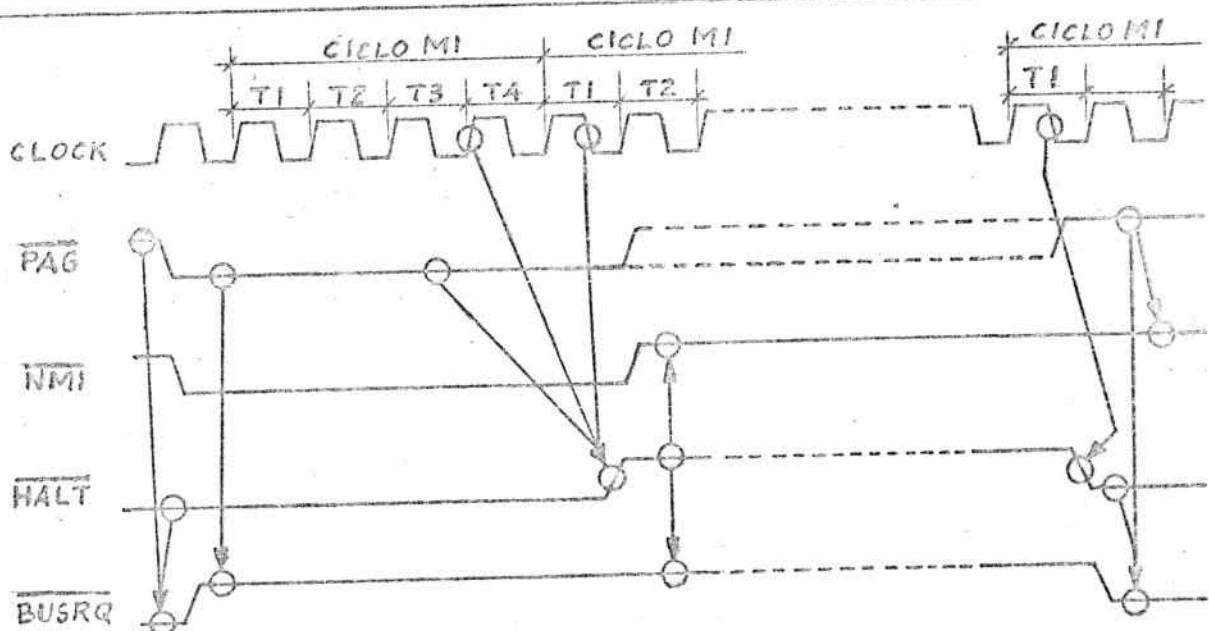
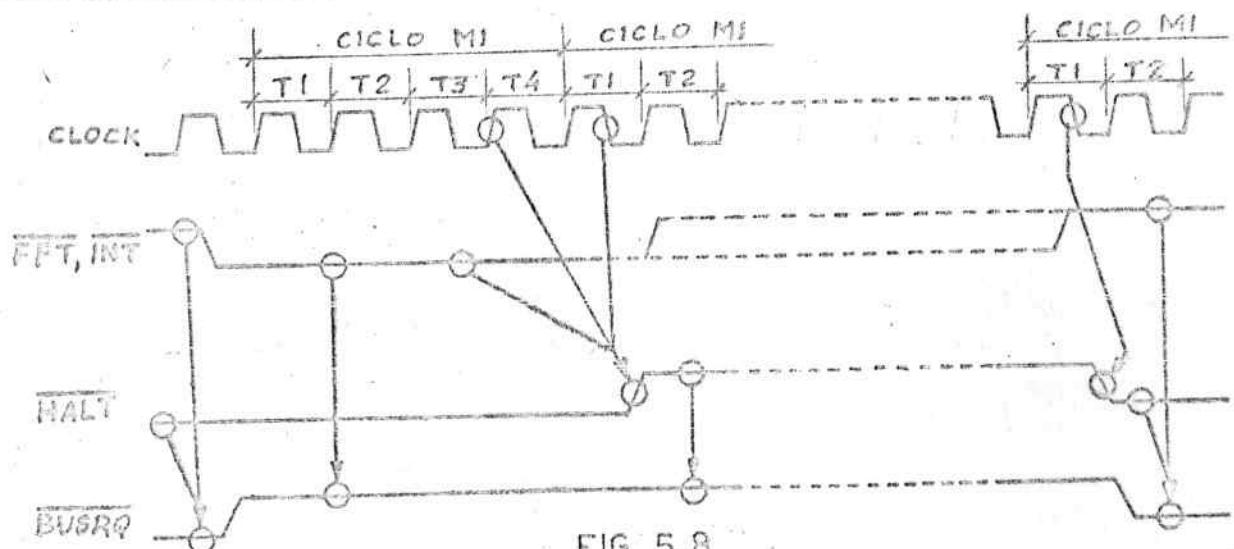


FIG. 5.7



5.3 e 5.4 para os casos 1,2 e 3 , respectivamente.

Os dados do fabricante da CPU para um clock com período T de 250ns são :

- atraso do end. em relação à subida do clock em T1, menor que 110ns ;
- atraso da descida do sinal \overline{RD} em relação à descida do clock em T1 ; menor que 95ns ;
- atraso da descida do sinal \overline{WR} em relação à descida do clock em T2 ; menor que 80ns ;
- largura do sinal \overline{WR} em nível "0" ; maior que 220ns ;
- tempo em que o dado deve estar estável para leitura antes da subida do clock em T3(ciclo M1) ; maior que 35ns ;
- tempo em que o dado deve estar estável antes da descida do clock em T3(ciclo M2 em diante) ; maior que 50ns .

5.6.1 ESPECIFICAÇÃO DO TEMPO DE ACESSO DA "EPROM"

O fabricante da "EPROM" especifica o tempo de acesso em relação ao endereço, o tempo de acesso em relação a seleção do "CHIP" e o tempo de acesso em relação ao sinal \overline{OE} . As duas primeiras especificações são iguais. Logo basta calcular o tempo de acesso em relação à seleção do "CHIP" , já que o endereço é estável quando o decodificador seleciona o CHIP .

Observando a figura 5.2, concluímos que o máximo tempo de acesso em relação a seleção do "CHIP" é dado pelos seguintes : tempo mínimo em que o end. é estável antes da subida do clock em T3(390ns), tempo de atraso máximo do decodificador(20ns) , tempo mínimo em que o dado deve estar estável antes da subida do clock em T3 (35ns). Assim o máximo tempo de acesso em relação ao endereço é : $390 + 20 + 35 = 445\text{ns}$.

O tempo máximo de acesso em relação ao sinal de leitura (\overline{RD}) é dado pelos seguintes:tempo máximo de atraso de \overline{OE} em

relação à \overline{RD} (10ns), tempo mínimo em que \overline{RD} é ativo antes da subida do clock em T3 (280ns), tempo mínimo em que o dado deve estar estável antes da subida do clock em T3 (35ns). O máximo tempo de acesso em relação ao sinal \overline{OE} é :

$$280 - 10 - 35 = 235\text{ns} .$$

Pelo manual do fabricante da " EPROM " , a que satisfaz as especificações é a com máximo tempo de acesso em relação a seleção do " CHIP " de 250ns e máximo tempo de acesso de relação à \overline{OE} de 120ns . Isto é, máximo tempo de acesso de 250ns .

Para o caso da figura 5.3 as especificações seriam mais otimistas já que temos meio período de clock a mais para o acesso.

A " EPROM " deve ter um tempo máximo de acesso de 250ns.

5.6.2 ESPECIFICAÇÃO DO TEMPO DE ACESSO DA " RAM " 6116

A metodologia é a mesma que a utilizada no item 5.6.1 .

Para o tempo de acesso para leitura utilizamos a figura 5.3 pois não temos instruções na " RAM " . Para este caso o atraso do decodificador deve ser somado o atraso de uma porta TTL (veja Eq. 4.6).

O fabricante da " 6116 " especifica : o máximo tempo de acesso em relação ao endereço igual ao máximo tempo de acesso em relação à seleção do " CHIP " . Como a seleção do " CHIP " é após a estabilização do endereço, não é necessário o cálculo do tempo de acesso em relação ao endereço.

O máximo tempo de acesso em relação à seleção do " CHIP " é dado pelos seguintes : tempo mínimo em que o endereço é estável antes da descida do clock T3 (515ns), atraso máximo do decodificador (30ns), tempo mínimo em que o dado deve estar estável antes da descida do clock em T3 (50ns). O máximo tempo de acesso em relação à seleção do " CHIP " é : $515 - 30 - 50 = 435\text{ ns}$.

O máximo tempo de acesso em relação ao sinal \overline{OE} é dado pelos seguintes: atraso máximo de \overline{OE} em relação à \overline{RD} (10ns), tempo mínimo em que \overline{RD} é ativo antes da descida do clock em T3 / (405ns), tempo mínimo em que o dado deve estar estável antes da descida do clock em T3 (50ns). Logo: $405 - 10 - 50 = 345\text{ns}$ é o máximo tempo de acesso em relação ao sinal \overline{OE} .

Para a escrita, o fabricante da "6116" especifica a mínima largura do pulso de escrita \overline{WE} . O fabricante da CPU especifica este tempo como sendo maior que 220ns.

A memória "6116" mais lenta satisfaz todas as especificações acima. As especificações desta são: máximo tempo de acesso em relação ao endereço de 200ns, máximo tempo de acesso em relação a seleção do "CHIP" de 200ns, máximo tempo de acesso em relação à \overline{OE} de 120ns e mínima largura do pulso \overline{WE} de 120ns.

A 6116 deve ter um tempo mínimo de acesso de 200ns.

5.6.3 ESPECIFICAÇÃO DO TEMPO DE ACESSO DA "RAM" 2114.

A "RAM" 2114 tem o inconveniente de não possuir entrada para habilitação de sua saída de dados. Quando a CPU deseja escrever, ela coloca os dados no barramento. Se a "RAM" 2114 fosse selecionada apenas pelo decodificador 74LS138, ela também colocaria os dados no barramento até que fosse ativado o sinal de escrita (\overline{WE}). Haveria "conflito" transitório no barramento de dados. A solução para este problema é ativá-la somente quando \overline{OE} ou \overline{WE} for ativo. Assim:

$$\overline{CS2} = \overline{Y2} + \overline{WE} \cdot \overline{OE} \quad (5.4)$$

Esta solução porém, reduz o máximo tempo de acesso já que o sinal \overline{WR} é ativo por apenas 220ns.

Pelas figuras 5.3 e 5.4 podemos observar as diferenças de duração dos sinais \overline{WR} e \overline{RD} .

O fabricante da 2114 especifica: o máximo tempo de acesso

em relação ao endereço , o máximo tempo de acesso em relação a seleção do "CHIP" , a mínima largura do pulso de escrita.

O máximo tempo de acesso em relação ao endereço é dado pelos seguintes: tempo mínimo em que o endereço é estável antes da descida do clock em T3 (515ns) , tempo mínimo em que o dado deve estar estável antes da descida do clock em T3(50ns). Então o tempo máximo de acesso em relação ao endereço é :

$$515 - 50 = 465\text{ns} .$$

O tempo máximo de acesso em relação à seleção do CHIP para leitura é dado pelos seguintes: tempo mínimo em que \overline{RD} é ativo antes da descida do clock em T3 (405ns) , atraso máximo de $\overline{CS2}$ em relação à \overline{RD} (30ns) , tempo mínimo em que o dado / deve estar estável, antes da descida do clock em T3(50ns). Isto é : $405 - 30 - 50 = 325$ ns para leitura .

O fabricante da 2114 também especifica que largura mínima de $\overline{CS2}$ para escrita é igual a largura mínima do sinal \overline{WE} . A CPU fornece \overline{WE} com largura mínima de 220ns .

A 2114 mais lenta satisfaz todas as especificações acima. As especificações desta são : máximo tempo de acesso em relação ao endereço de 450ns , máximo tempo de acesso em relação à seleção do CHIP para leitura de 120ns , mínima largura dos pulsos $\overline{CS2}$ e \overline{WE} de 200ns .

A 2114 escolhida deve ter um tempo máximo de acesso de 450ns.

5.7 O "CLOCK"

A CPU exige um clock com tempos de subida e descida menor que 30ns. Foi utilizado um oscilador a cristal e inversores TTL LS . Os resistores de 1kΩ são valores típicos. O capacitor de 100pF garante que o oscilador não oscile num sub-harmônico ou harmônico de freqüência padrão do cristal. O capacitor de 30pF

foi colocado para equalizar os semi-periodos. O resistor de 390Ω é para satisfazer as exigências de C.A e A.C. do clock da CPU.

CAPITULO 6

A FONTE DE ALIMENTAÇÃO

6.1 CONSIDERAÇÕES INICIAIS

Para polarizar o "DIODO" de referência do conversor D/A é necessária uma fonte com tensão regulada, para manter a corrente de polarização aproximadamente constante. Sendo V a tensão da fonte, I a corrente de polarização, V_z a tensão do diodo de referência e R o resistor de polarização do diodo, então a sensibilidade de I em relação a V é :

$$\frac{I}{V} = \frac{\partial I}{\partial V} \cdot \frac{V}{I} = -\frac{\partial ((V-V_z)/R)}{\partial V} \cdot \frac{V}{(V-V_z)/R}$$

$$\frac{I}{V} = \frac{V}{V - V_z} \quad (6.1)$$

Da eq. 6.1 concluímos que a sensibilidade diminui com o aumento de V , sendo V maior que V_z . Sendo $V_z = 6,8V$ a sensibilidade para $V=10V$ é :

$$\frac{V}{V - V_z} = \frac{10}{10 - 6,8} \simeq 3,3$$

Admitindo uma variação de 2% para V , a variação da corrente de polarização para $I=8mA$ será :

$$\Delta I = 0,02 \times 3,3 \times 0,008 \simeq 0,53mA$$

A resistência dinâmica do "DIODO" de referência é de aproximadamente 11Ω para corrente de $8mA$. A variação da tensão será, então :

$$\Delta V = 0,53mA \times 11\Omega = 5,8mV$$

A tensão na saída do amplificador operacional da fonte de referência é de $4,0V$ para tensão $V_z = 6,8V$. Assim a variação da tensão na saída do amplificador operacional será :

$$\Delta V = 5,8 \text{mV} \times \frac{4,0}{6,8} = 3,4 \text{mV}$$

Este valor é aproximadamente 1/3 da variação causada pelo LSB do conversor D/A.

Os C.I.s utilizados podem ser alimentados com tensão positiva de 5V a 18V.

Para alimentar toda a lógica é necessária uma fonte de 5V com capacidade de 500mA, já que o consumo máximo é de 400mA.

A fonte de tensão negativa não é crítica já que os C.I.s utilizados podem ser alimentados com tensão negativa de -5V a -18V.

Necessitamos então de uma fonte com 3 tensões:

+10V a +18V, 45V e -5V a -18V.

6.2 ESCOLHA DO TRANSFORMADOR

No comércio são encontrados transformadores com secundário de 15V, 18V e 24V, todos com derivação central. Como necessitamos de fonte com tensões negativa e positiva, devemos aterrizar a derivação central. Dessa forma a tensão positiva obtida após a retificação é o valor nominal (RMS) multiplicado por $\sqrt{2}/2$.

Para obter uma tensão regulada de 10V, admitindo que uma queda de 20% na tensão do transformador, uma tensão de 11V no diodo Zener, e uma queda de 0,7V nos diodos retificados, temos:

$$11V + 0,7V = 0,8 \times \frac{\sqrt{2}}{2} \times V_{\min}$$

$\therefore V_{\min} = 20,7V$, sendo V_{\min} a tensão nominal mínima

do transformador.

O transformador escolhido foi o de 24V com derivação central.

Optamos por uma fonte com as tensões :

+10V, -10V e +5V

As correntes requeridas das fontes de +10V, -10V e +5V são respectivamente : 25mA, 20mA e 400mA.

Optamos então por um transformador com as características:

Tensão do primário : 110V e 220V

Tensão do secundário : 24V com derivação central

Corrente máxima no secundário : 750mA

O diagrama esquemático da fonte está na figura 6.1.

Os transistores escolhidos, para as fontes de +10V e -10V, suportam correntes de até 500mA e dissipam até 0,8W à temperatura ambiente de 25°C. Limitando-se a corrente em 100mA o limite da tensão entre o coletor e emissor fica em 8V.

Sem carga na fonte a tensão retificada (medida) é de 16V. O diodo Zener é de 11V. Limitando a corrente no diodo Zener em 10mA, então o resistor de polarização é de 470Ω. A corrente de base do transistor é menor que 1mA. Assim a tensão mínima no coletor deve ser de $11V + 0,47V \approx 11,5V$.

Testamos a fonte com as correntes máximas estabelecidas e observamos que os capacitores de 470μF podem ser reduzidos até 200μF, sem que haja "RIPPLE" na saída. Os capacitores de 1μF e 10μF foram colocados para eliminar ruídos da fonte.

Na fonte de 5V foi utilizado o C.I. 7805, regulador de 5V e 1A, que exige tensão de entrada de 7,3V a 35V. Como a tensão retificada é de 16V, colocamos um resistor em série para reduzir a dissipação no C.I. e aumentar a durabilidade deste. A dissipação no C.I. fica então menor que 2W e não necessita dissipador, mas colocamos um para garantir a durabilidade.

O capacitor de 2200μF também foi escolhido experimental -

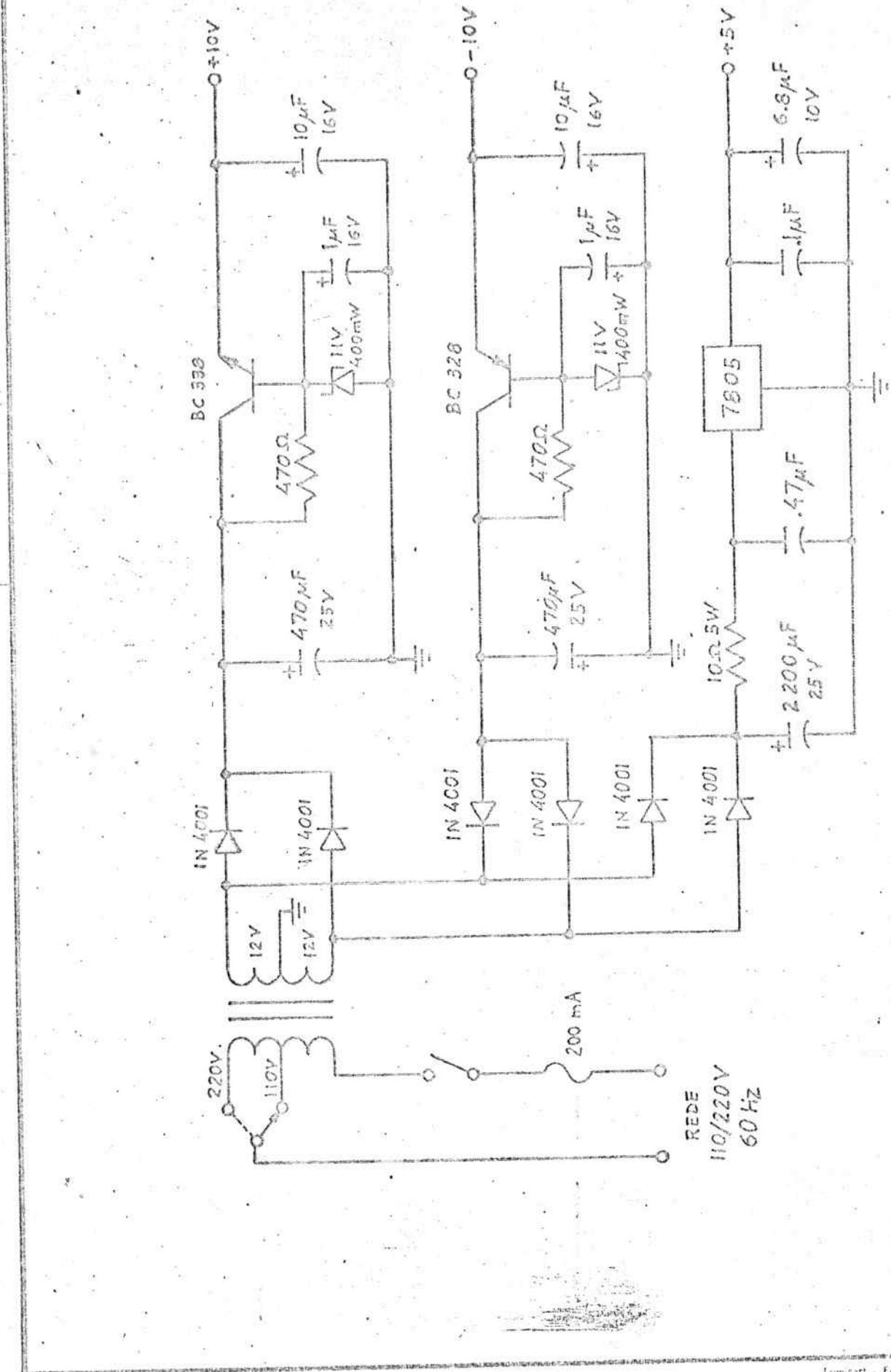


FIG. 6.1 - FONTE DE ALIMENTAÇÃO

mente. Os capacitores de 470nF e 100nF são recomendados pelo fabricante para garantir a estabilidade do C.I. e o capacitor de 6,8μF (tântalo) é normalmente usado em circuitos digitais para manter baixa a impedância da fonte em altas freqüências, pois o C.I. 7805 não tem velocidade suficiente para responder a variações de solicitação de corrente.

Os diodos retificadores são de 1A e devem ter mínima tensão reversa de $24V \times \sqrt{2} = 34V$. Foram colocados mais 2 diodos para a fonte de +5V, eliminando-se assim o "RIPPLE" causado por esta na fonte de +10V.

CAPÍTULO 7

CONCLUSÕES

O analisador de espectro funciona bem, porém necessita de algumas alterações. Algumas sugestões são dadas abaixo.

As chaves analógicas CMOS, utilizadas nos amostradores, devem ser substituídas por outro tipo de chave, como por exemplo JFET.

A utilização de um "BUFFER" na saída analógica, para evitar a carga no capacitor do amostrador/segurador, é indispensável.

A utilização de um "BUFFER" na entrada analógica, para permitir que o capacitor do amostrador/segurador se carregue, também é indispensável.

É aconselhável a inclusão de um "RESET" para os contadores de endereço, devido à dificuldade de sincronização com "PLOTTER" X-T.

Há de se incluir um circuito para somar um nível D.C. na entrada analógica e subtraí-lo na saída.

O "SOFTWARE" pode ser mudado de forma a não apresentar a repetição invertida do espectro, na segunda metade, ou para apresentação do espectro em dB .Sub-rotina TRANSF ,end. 04D4H .

Poderá ser incluído um oscilador de ajuste automático, para sincronização da amostragem com o sinal analógico de entrada , utilizando PLL (phase-locked loop).

Sugerimos que a lógica do registro de aproximações sucessivas seja feita com TTL série LS, já que o atraso da lógica é maior que o do conversor D/A e do comparador.O contador em anel CMOS poderá ser utilizado, pois funciona com um clock de até / 5MHz. Com estas modificações poderão ser atingidas facilmente 200.000 conversões A/D por segundo, com um clock de 2MHz .

APÊNDICE II.

DEMONSTRAÇÃO DA MÁXIMA AMPLITUDE POSSÍVEL DE UM HARMÔNICO DE UMA FORMA DE ONDA.

No nosso caso a amplitude do sinal é limitada entre 127 e 128. Para a integral :

$$a_n = \frac{2}{T} \cdot \int_{t_0}^{t_0+T} f(t) \cdot \cos(n \cdot w_0 \cdot t) \cdot dt, \quad T \geq \frac{2}{w_0} \quad (2.5)$$

Temos que o maior valor da integral pode ser encontrado, fazendo $f(t) = +127$ nos pontos onde $\cos(nwt)$ é positivo e $f(t) = -128$ nos pontos onde $\cos(nwt)$ é negativo. Ou seja :

$$a_{n_{\max}} = \frac{2}{T} \cdot \int_{t_0}^{t_0+T} 127,5 \cdot \cos(n \cdot w_0 \cdot t) \cdot dt = \frac{4 \cdot 127,5}{\pi} = 162,34$$

Como o seno é desfasado de $\pi/2$ em relação ao co-seno a integral da eq. 2.6 tem resultado nulo para a $f(t)$ assumida. Logo o maior harmônico tem a amplitude de 162,34.

APÊNDICE 2

REFERÉNCIAS :

- 1 - BERGLAND, G. D. - A guided tour of the fast Fourier transform , IEEE SPECTRUM , July 1969, 41-52p.
- 2 - COOLEY, J.W. , and TUKEY, J.W. - An algorithm for the machine calculation of complex Fourier series , MATHEMATICS OF COMPUTATION , Vol. 69, April 1965, 297-301p.
- 3 - BERLIN , Howard M. - Aplicações para o 555, com experiências , Editele, 1976 , 31-32p .
- 4 - ZILOG , Inc. - Z80 CPU Technical manual, Zilog Inc.
- 5 - TAUB, Herbet, and SCHILLING , Donald - Eletrônica Digital , Mc Graw - Hill , 1982 .
- 6 - BARDEN , Willian Jr. - The Z-80 Microcomputer Handbook , Howard W. Sams & Co. , Inc. , 1980
- 7 - LIMA, Delio Santos - Código de Máquina para TK e CP 200 , Micron Eletrônica Com. Ind. Ltda, 1983 .
- 8 - SANTOS , Vitoriano Ruas de Barros - Curso de Cálculo Numérico , LTC , 1977 , 78- 80 p .

APÊNDICE 3

LISTA DE MATERIAL

RESISTORES:

Quantidade	Resistência(Ω)	Potência(w)	Tolerância(%)
1	1470	1/8	1
1	1960	1/8	1
1	10k	1/8	1
1	12,7k	1/8	1
1	10	5	10
2	390	1/8	10
4	470	1/8	10
5	1k	1/8	10
1	2k	1/8	5
3	2,2k	1/8	10
1	3,3k	1/8	10
1	10k	1/8	10
6	12k	1/8	10
14	22k	1/8	10
1	47k	1/8	10

POTENCIÔMETROS:

Quantidade	Resistência(Ω)	Tipo
1	5k	trimpot multivolta
1	10k	trimpot comum
1	22k	potenciômetro linear
1	100k	potenciômetro linear

CÁPACITORES :

Quantidade	Capacitância (F)	Tensão mínima(V)	Tipo
1	30p	10	cerâmico
2	100p	10	cerâmico
3	1n	10	poliéster
1	10n	10	cerâmico
12	100n	15	cerâmico
3	150n	15	cerâmico
2	470n	25	cerâmico
3	1μ	15	eletrolítico
1	6,8μ	10	tântalo
3	10μ	15	eletrolítico
1	47μ	10	eletrolítico
2	470μ	25	eletrolítico
1	2200μ	25	eletrolítico

DIODOS :

Quantidade	Tipo
2	Zener de 11V X 400mW
1	1N4735
3	1N914
4	1N4148
6	1N4001

TRANSISTORES :

Quantidade	Tipo
1	BC 547
1	BC 338
1	BC 328

CIRCUITOS INTEGRADOS :

Quantidade	Tipo
1	CD4017B
2	CD4043B
1	CD4066B
2	CD4071B
3	CD4081B
2	CD4520B
1	LM310
1	LM311
1	LM741
1	μ 40801
1	Z-80A CPU
1	HM6116P-4 , tempo de acesso \leq 200ns
2	TMS 2114-45 , tempo de acesso \leq 450ns
1	EPROM 2716 , tempo de acesso \leq 250ns
1	74LS00
2	74LS02
1	74LS04
2	74LS08
1	74LS32
1	74LS74
2	74LS75
1	74LS138
1	7805

DIVERSOS :

- 1 transformador , primário 110/220V , secundário 24V com derivação central , 750mA
- 1 cristal de 4MHz
- 1 display FND 357
- 5 chaves de 1 polo, contato momentâneo
- 1 chave de 250V , 1A
- 1 chave rotativa de 1 polo , 5 posições
- 1 chave rotativa de 2 polos, 4 posições

APÊNDICE 4

LISTAGEM DO PROGRAMA DA FFT EM BASIC :

```

10 REM FFT-FAST FOURIER TRANSFO
   RM
20 DIM A(255): DIM B(255): DIM M
   (255)
30 PI = 3.141592654
40 GOSUB 500
50 LET M = LOG (N) / LOG (2)
60 FOR K = M TO 1 STEP - 1
70 LET K1 = 2 ^ (K - 1) 128 64 32 16 8 4 2 1
80 LET K2 = 2 * K1      256 128 64 32 16 8 4 2
90 LET K3 = K1 - 1      129 63 31 15 7 3 1 0
100 FOR P = 0 TO N - 1 STEP K2 0; 0-128; 0-64-128-192; -- ; 0-2-4-
110 LET X = P / K1
120 GOSUB 390
130 LET X1 = COS (2 * PI * S /
   N)
140 LET Y1 = SIN (2 * PI * S /
   N)
150 FOR J = 0 TO K3
160 LET K4 = J + P
170 LET K5 = K4 + K1
180 LET X2 = X1 * A(K5) + Y1 * B
   (K5)
190 LET Y2 = - Y1 * A(K5) + X1 *
   B(K5)
200 LET A(K5) = A(K4) - X2
210 LET B(K5) = B(K4) - Y2
220 LET A(K4) = A(K4) + X2
230 LET B(K4) = B(K4) + Y2
240 NEXT J
250 NEXT P
260 NEXT K
270 FOR I = 1 TO N - 1
280 LET X = I
290 GOSUB 390
300 IF S < = I THEN GOTO 370
310 LET X3 = A(I)
320 LET Y3 = B(I)
330 LET A(I) = A(S)
340 LET B(I) = B(S)
350 LET A(S) = X3
360 LET B(S) = Y3
370 NEXT I
380 GOTO 550
390 LET U = 0
400 FOR L = 1 TO M
410 LET K5 = INT (X / 2)
420 LET A = X - 2 * K5
430 LET K6 = 2 ^ (M - L)
440 LET U = U + A * K6
450 LET X = K5
460 IF X < = 1 THEN GOTO 480
470 NEXT L

```

```
480 LET S = U + X * K6 / 2
490 RETURN
500 INPUT "# DE PONTOS = ";N
510 FOR I = 0 TO N / 2 - 1
520 A(I) = 0:A(I + N / 2) = 255
530 NEXT
540 RETURN
550 FOR I = 0 TO N - 1
560 LET M(I) = 2 * SQR (A(I) *
      A(I) + B(I) * B(I)) / N
570 NEXT
575 T = N / 2
580 M(0) = M(0) / 2:M(T) = M(T) /
      2
590 STOP
600 FOR I = 0 TO N - 1
610 PRINT I; TAB(10);M(I); TAB(
      30); INT (M(I) + .5)
620 PRINT
630 NEXT
640 END
```

APÊNDICE 5

ENDERECOS DA MEMÓRIA (HEXADECIMAL)

0000 a 00FF - Rotina de inicialização e interrupção .

0100 a 067F - Programa da FFT

0680 a 07FF - Tabela de seno, fim da EPROM

0800 a OBFF - A(I) armazenado na representação com 4 BYTES
em complemento a 2 .

0C00 a OFFF - B(I) armazenado na representação com 4 Bytes
em complemento a 2

1000 a 107F - Área de variáveis

1080 a 117F - Área onde é guardado o sinal amostrado do
qual foi calculada a FFT.

1180 a 127F - Área onde é guardado o espectro na represen -
tação de módulo com 8 Bits

0800 a 08FF - Área para onde é transferido o conteúdo dos
endereços de 1080 a 117F ou 1180 a 117F ,
após o cálculo da FFT

0E00 a OFFF - Área onde é guardado o espectro na representa
ção de módulo com 16 Bits

1180 a 137F - Área que é utilizada temporariamente para
guardar o módulo com 16 Bits

1380 a 137F - Área destinada ao "STACK FONITER" .

APÊNDICE 6

O PROGRAMA DA FFT EM ASSEMBLY

O programa inclui uma sub-rotina de troca de "página". Cada página são 256 Bytes. Como temos 8 páginas (2048 Bytes) disponíveis de RAM para acesso direto à memória, é interessante que se possa mostrar cada página separadamente, para maior resolução, principalmente quando estiver sendo usado como memória de osciloscópio. A sub-rotina faz a rotação de páginas. É executada através de interrupção, não mascarável, da CPU.

A rotina de cálculo da FFT inclui uma sub-rotina que salva o sinal amostrado que está nos primeiros 256 endereços da RAM. A rotina da FFT é executada através de uma interrupção, mascarável, da CPU.

Após ser executada a rotina da FFT, a sub-rotina de rotação de página não é mais executada. É executada uma sub-rotina que coloca nos primeiros 256 Bytes da RAM, ora o resultado da FFT, ora o sinal amostrado, para cada interrupção, não mascarável, da CPU.

A rotina da FFT pode ser modificada para o cálculo com menor número de pontos, simplesmente mudando a instrução que está no endereço 01COH (LD A,128).

Ex.: Para o cálculo da FFT com 128 pontos a instrução LD A,128 deverá ser substituída por LD A,64.

A instrução poderá ainda ser substituída por uma instrução de entrada. Ex.: IN A,(porta)

Nas páginas a seguir está a listagem comentada do programa em ASSEMBLY.

ENDERECOS DAS VARIÁVEIS :

Na 1^a coluna está o endereço em hexadecimal ; na 2^a coluna, o nome da variável ; na 3^a coluna , o número de BYTES da variável.

Para maior compreensão muitas variáveis tem o mesmo nome das do programa em BASIC, sendo referida ao BASIC , quando for o caso .

ENDERECO	VARIÁVEL	BYTES	COMENTARIO
1000	GN	2	N , BASIC
1002	GM	2	M , BASIC
1006	GK	1	K , BASIC
1007	GK1	1	K1 , BASIC
1008	GK2	2	K2 , BASIC
100A	GK3	1	K3 , BASIC
100B	GP	1	P , BASIC
100C	GS	1	S , BASIC
100D	Y1	2	Y1 , BASIC
100F	X1	2	X1 , BASIC
1011	GJ	1	J , BASIC
1012	AK4	2	Endereço de A(K4),BASIC
1014	AK5	2	Endereço de A(K5),BASIC
1016	BK4	2	Endereço de B(K4),BASIC
1018	BK5	2	Endereço de B(K5),BASIC
101A	MULT	4	Multiplicando
101E	C1	4	X1 . A(K5)
1022	C2	4	Y1 . B(K5)
1026	C3	4	Y1 . A(K5)
102A	C4	4	X1 . B(K5)
102E	RES	4	Resultado da multiplicação

ENDEREÇO	VARIÁVEL	BYTES	COMENTÁRIO
1032	X2	4	X2 , BASIC
1036	Y2	4	Y2 , BASIC
103A	GI	1	I , BASIC
103B	GAS	2	Endereço de A(S) , BASIC
103D	GBS	2	Endereço de B(S) , BASIC
103F	GAI	2	Endereço de A(I) , BASIC
1041	GBI	2	Endereço de B(I) , BASIC
1043	X3	4	X3 , BASIC
1047	Y3	4	Y3 , BASIC
104C	GC	1	Contador da sub-rotina de raiz quadrada
104D	GHI	2	Endereço do módulo
1060	FLAG	1	Indica se foi calculada a FFT
1061	PAG	1	Endereço relativo da tabe la do DISPLAY .

0000 31 FF 13 LD SP,13FFH ;Ultimo end. da RAM p/SP.
 a/ RAM começa no endereço 0800H .
 0003 ED 56 IM 1 ;Seleciono modo 1 de interrupção.
 0005 FB EI ;Habilito interrupção .
 0006 3E 3F LD A,3FH ;Código p/ colocar zero
 no display .
 0008 32 00 38 LD (DISP),A ;Acendo DISPLAY com zero
 000B AF XOR A ;Zero acumulador .
 000C 32 60 10 LD (FLAG),A ;Zero Flag que indica se
 a FFT já foi calculada .
 ; A seguir está a rotina de geração
 do sinal de ajuste:
 000F 01 7F 00 LD BC,127 ;Num. de BYTES a serem
 transferidos .
 0012 11 01 03 LD DE,0801H ;End. inicial destino .
 0015 21 00 03 LD HL,0800H ;End. inicial fonte .
 0018 77 LD (HL),A ;Zero p/ conteúdo do
 end. inicial fonte .
 0019 ED B0 LDIR ;Coloco zero nos primeiros 128 BYTES da RAM .
 001B 2F CPL ; 255 p/ o acumulador .
 001C 01 7F 01 LD BC,383 ;Num. de BYTES a serem
 transferidos .
 001F 11 81 08 LD DE,0881H ; End. inicial destino .
 0022 21 80 08 LD HL,0880H ; End. inicial fonte .
 0025 77 LD (HL),A ; 255 p/ conteúdo do
 end. inicial fonte .
 0026 ED B0 LDIR ; Coloco 255 nos BYTES
 de 129 até 511 da RAM .

0028 3E 80 LD A,128 ;128 p/ o acumulador .
 002A 01 7F LD BC,127 ;Num. de BYTES a serem
transferidos.
 002D 11 01 09 LD DE,0901H ;End. inicial do destino
 0030 21 00 09 LD HL,0900H ;End. inicial fonte .
 0033 77 LD (HL),A ;128 p/ primeiro end.
da fonte.
 0034 ED B0 LDIR ;Coloco 128 nos BYTES
256 a 383 da RAM .
 0036 18 CD JR PULAL ;Continua em PULAL .
 0038 3E 09 INT: LD A,9 ;End. relativo da tabela
do DISPLAY mais 1, letra A .
 003A 32 61 10 LD (PAG),A ;Guardo end. relativo .
 003D 32 60 10 LD (FLAG),A ;Ativa FLAG .
 0040 CD 00 01 CALL FFT ;Executa a FFT e second
DISPLAY com F no final .
 0043 ED 7D FININT: RETI ;Retorno da interrupção.
 0045 01 7E 00 PULAL: LD BC,126 ;Num. de BYTES a serem
transferidos .
 0048 11 82 09 LD DE,0982H ;End. inicial do destino
 004B 21 80 09 LD HL,0980H ;End. inicial fonte .
 004E AF XOR A ;Zero acumulador .
 004F 77 LD (HL),A ;0980H contém zero e
0981H contém 255 .
 0050 ED B0 LDIR ;Coloco zero nos BYTES
pares e 255 nos BYTES ímpares.
a partir do endereço 384 até 511 da
RAM .
 0052 00 NOP

0053 00	NOP	;
0054 01 00 06	LD BC,1536	;Num. de BYTES a transferir.
0057 11 00 0A	LD DE,0AOOH	;End. inicial destino .
005A 21 00 08	LD HL,0800H	;End. inicial fonte .
005D ED B0	LDIR	;Copia o conteúdo dos primeiros 512 BYTES da RAM p/ os próximos 1536 BYTES da RAM .
005F AF	XOR A	;Zero acumulador .
0060 32 61 10	LD (PAG),A	;Zero end. relativo da tabela do DISPLAY .
0063 76	PARE:	HALT ;Para a CPU e libera o barramento por HARDWARE .
0064 18 FD	JR PARE	;Após executada uma interrupção volta ao estado anterior .
0066 3A 60 10	NMI :	LD A,(FLAG) ;Se FLAG diferente de zero a FFT já foi calculada .
0069 E7	OR A	;Verifica se já foi calculada a FFT .
006A 20 05	JR NZ,PUL42	;Se já pulou. Uma página = 256 BYTES
006C CD 70 00	CALL TROCAS	;Faz a rotação das primeiras 8 páginas da RAM .
006F 18 03	JR PUL43	
0071 CD AB 00	PUL42 :	CALL TROC42 ;Coloca no end. inicial da RAM, o sinal amostrado ou o espectro .
0074 CD CD 00	PUL43 :	CALL DISPLAY ;Indica qual a página da RAM, está na página ZERO, ou se é o

espectro ou o sinal amostrado de qual foi calculada a FFT .

0077 CD DA 00	CALL ESPERA ;Chama rotina p/DEBOUNCE	
	Espera 426 milisegundos .	
007A ED 45	FIMNMI:	RETN ;Fim da NMI .
007C 01 00 01	TROCAS:	LD BC,256
007F 11 80 10		LD DE,1080H ;Inicio destino temporário .
0082 21 00 08		LD HL,0800H ;Inicio fonte .
0085 ED B0	LDIR	;Salva os 256 BYTES da página zero da RAM .
0087 C1 00 07		LD BC,1792
008A 11 00 03		LD DE,0800H ;Inicio destino .
008D 21 00 09		LD HL,0900H ;Inicio fonte .
0090 ED B0	LDIR	;Desloca 1792 BYTES das páginas 1 até 7, para as páginas zero até 6 da RAM .
0092 C1 00 01		LD BC,256
0095 11 00 0E		LD DE,0F00H ;Inicio destino final .
0098 21 80 10		LD HL,1080H ;Inicio destino temporário .
009B ED B0	LDIR	;Coloca o antigo conteúdo da pag. zero na pag. 7 da RAM .
009D 3A C1 10	LD A,(PAG)	;End. relativo da tabela do DISPLAY p/ acumulador .
00A0 30	INC A	;próximo caracter do DISPLAY .
00A1 FE 08	CP 08H	;Verifica se o anterior era o end. relativo do caracter 7 .

00A3 20 02 JR NZ,PUL4 ;Se não pule.
 00A5 3E 00 LD A, \varnothing ;Se sim então end. relativo do caracter zero p/ acumulador .
 00A7 32 61 10 PUL4 : LD (PAG),A ;Guarda end. relativo do caracter atual .
 00AA C9 RET ;Fim da troca .
 00AB 01 00 01 TROCA2 : LD BC, 256 ;Num. de BYTES a serem transferidos .
 00AE 11 00 08 LD DE,0800H ;End. inicial da página zero da RAM .
 00B1 34 61 10 LD A,(PAG) ;Aponta end. relativo do último caracter mostrado .
 00B4 3C INC A ;Próximo caracter .
 00B5 FE 0A CP 0AH
 00B7 20 0B JR NZ,Pula5 ;Se o caracter anterior era letra A, então pule .
 00B9 3E 08 LD A,8 ;End. relativo da letra A na tabela do DISPLAY .
 00BB 32 61 10 LD (PAG),A ;Guarda end. relativo do caracter a ser mostrado no DISPLAY .
 00BE 21 80 10 LD HL,1080H ;End. inicial dos pontos utilizados pela rotina de FFT.
 00C1 ED B0 LDIR ;Copia para a página zero da RAM , o sinal do qual foi calculado o espectro .
 00C3 C9 RET
 00C4 32 61 10 PUL5 : LD (PAG),A ;Guarda end. relativo do caracter F .
 00C7 21 80 11 LD HL,1180H ;End. inic.espectro calc.

00CA ED BO	LDIR	;Transfere para a página zero da RAM .
00CC C9	RET	
00CD 3A 61 10	DISPLAY: LD A,(PAG)	;End. relativo do caracte r a ser mostrado .
00D0 21 E3 00	LD HL,TDISP	;End. do inicio da tabela do DISPLAY .
00D3 85	ADD A,L	;Acumulador aponta para o end. do caracter .
00D4 6F	LD L,A	;HL aponta para o end. do caracter H=0
00D5 7E	LD A,(HL)	;Código do caracter para o acumulador .
00D6 32 00 38	LD (DISP),A	;Acende DISPLAY com novo caracter .
00D9 C9	RET	
00DA 01 FF FF	ESPERA:	LD BC,65535 ;Espera 426 milisegundos para um clock de 4MHz .
00DD 0B	LOOP1:	DEC BC
00DE 78		LD A,B
00DF B1		OR C
00E0 20 FB		JR NZ, LOOP1
00E2 C9	RET	
		;A seguir está a tabela do DISPLAY :
00E3 3F	TDISP :	DB 3FH ;Código p/ acender zero no DISPLAY .
00E4 06	DB 06H	;Idem p/ 1 .
00E5 5B	DB 5BH	;Idem p/ 2 .
00E6 4F	DB 4FH	;Idem p/ 3 .
00E7 66	DB 66H	;Idem p/ 4 .

00E8 6D	DB 6DH	;Idem p/ 5 .
00E9 7D	DB 7DH	;Idem p/ 6 .
00EA 07	DB 07H	;Idem p/ 7 .
00EB 77	DB 77H	;Idem p/ 8 .
00EC 71	DB 71H	;Idem p/ F .
00ED	DS 13H	;Área não utilizada .
0100 3E 80	FFT :	LD A,128 ;Coloco metade do num. de pontos em A .
0102 6F	LD L,A	
0103 26 00	LD H,O	;Transfiro p/ HL .
0105 29	ADD HL,HL	;Encontro o número de Pontos N .
0106 22 00 10	LD (GM),HL	;Guardo em GM . ;O trecho a seguir calcula : $N = (\log N) / \log 2 .$
0109 21 00 00	LD HL,0	;Zero HL ;N/2 está em A .
010C 24	LOOP2 : INC H	
010D 1F	RRA	;SHIFT de A p/ a direita
010E 30 FC	JR NC,LOOP2	;Volte se BIT=0
0110 22 02 10	LD (GM),HL	;Coloco H em GM+1 e zero (GM) para facilitar subrotinas de BIT inverso .
0113 01 00 01	LD BC, 256	
0116 11 80 10	LD DE,1080H	;End. destino dos pontos amostrados que estão na página zero da RAM.
0119 21 00 08	LD HL,AI	;End. inic.pág.zero RAM.

0110 ED B0	LDIR	;Transfere 256 BYTES.
011E 01 FF 07	LD BC,2047	;Desejo zerar os primeiros 2 Kbytes da RAM.
0121 11 C1 08	LD DE, A+1	
0124 21 00 08	LD HL, A1	
0127 36 00	LD (HL), 0	;Zero primeiro BYTE da RAM .
0129 ED B0	LDIR	;Zero mais 2047 BYTES.
012B 11 7F 11	LD DE,117FH	;DE aponta para ultimo BYTE salvo .
012E 21 FF 0B	LD HL, A1<1023;HL aponta para ultimo BYTE da parte real .	
0131 0E 00	LD C,0	;Inicialize contador com 256 .
0133 1A	LOOP3 :	LD A,(DE) ;Coloco em A o valor a ser representado com 4 BYTES .
0134 06 80	ADD A, 128	;Subtrai nível DC .
0136 F2 3D 01	JP P, PULAG	;Se positivo, zero BYTE mais significativo.
0139 36 FF	LD (HL),FFH	;Sendo negativo coloco FFH no BYTE mais significativo .
013B 18 02	JR, PULAT	;Pule .
013D 36 00	PULAG :	LD (HL), 0 ;Zero BYTE mais significativo .
013F 2B	PULAT :	DEC HL ;HL aponta para o 2º BYTE mais significativo .
0140 77	LD (HL),A	;Coloco o valor no 2º BYTE mais significativo
0141 2B	DEC HL	;Zero para o 3º BYTE mais significativo .

0142 2B	DEC HL	;Zero para o BYTE menos significativo .
0143 2B	DEC HL	;HL aponta para o BYTE mais significativo do próximo ponto com 4 BYTES .
0144 1B	DEC DE	;DE aponta para o próximo ponto representado por um BYTE.
0145 0D	DEC C	;Decremento contador .
0146 02 33 01	JP NZ,LOOP3	;Se não acabou , volte.
0149 3A 03 10	LD A,(GM+1)	;Coloco M em A .
014C 32 06 10	LOOP4 : LD (GX),A	;Salvo K . ;A seguir faço $K1=2^{(K-1)}$
014F 47	LD B,A	;Coloco K em B .
0150 AF	XOR A	;Zero A .
0151 37	SCF	;CY = 1
0152 17	LOOP5 : RLA	;Retenção para a esquerda através de CY .
0153 05	DEC B	;Decremento contador .
0154 02 52 01	JP NZ,LOOP5	;Se K1 ≠ $2^{(K-1)}$, volte.
0157 32 07 10	LD (GX1),A	; Salvo K1 .
015A 26 00	LD H,0	;
015C 6F	LD L,A	;K1 está em HL .
015D 29	ADD HL,HL	;Faço $K2=2*K1$.
015E 3D	DEC A	;Faço $K3=K1-1$.
015F 22 08 10	LD (GX2),HL	;Salvo K2 .
0162 32 0A 10	LD (GX3),A	;Salvo K3 .
0165 AF	XOR A	;Inicializo P
0166 32 0B 10	LOOP6 : LD (GP),A	;Salvo P . ; X = P .

0169 ED 4B 06 10 LD BC,(BK) ;Recupero K .
 ;A seguir faço X = P / (2 * KL)
 016D CB 3F LOOP7 : SRL A ;X = X/2
 016F OD DEC C ;Decremento K .
 0170 C2 6D 01 JP NZ, LOOP7 ;Se X ≠ P/(2 * KL) volte.
 0173 17 RLA ;X = P / KL .
 0174 CD 32 03 CALL BINV1 ;Represento X em BIT in-
 verso e faço S = X * 256/N , para busca
 na tabela de seno . S está em HL .
 0177 7D LD A,HL
 0178 32 00 10 LD (GS), A ;Salvo S .
 017E 29 ADD HL,HL ;Multiplico S por 2 por-
 que o seno está representado com 2
 BYTES.
 017F EB EX DE,HL ;Coloco em DE .
 0180 21 80 06 LD HL,TABSEN ;HL aponta para o inicio
 da tabela .
 0180 19 ADD HL, DE ;HL aponta para:
 SEN(2*PI*S/N)/2 .
 0181 11 0D 10 LD DE, YL ;DE aponta para YL .
 0184 ED A0 LDI
 0186 ED A0 LDI ;Faço :
 (YL) = SEN(2*PI*S/N)/2
 ;DE aponta para YL .
 0188 01 7E 00 LD BC,126 ;O Cosseno está 128 BYTES
 após o Seno .
 018B 09 ADD HL,BC ;HL aponta para :
 COS(2*PI*S/N)/2 .

018C ED A0	LDI	
018E ED A0	LDI	;Faço :
	X1 = COS (2*PI*S/N)/2	
0190 3E FF	LD A,FF	;Inicializo J com zero.
0192 3C	LOOPS : INC A	;Próximo J .
0193 32 11 10	LD (GJ),A	;Salvo J .
0196 2A 0B 10	LD HL,(GP)	;Recupero P em L .
0199 85	ADD A,L	
019A 6F	LD L,A	;Faço K4 (J+P) no regis- tro L.
019B AF	XOR A	;Zero A .
019C 67	LD H,A	;Zero H. J+P < 256
019D 01 00 08	LD BC,AL	;BC aponta para primeiro BYTE do vetor A(I).
01A0 29	ADD HL,HL	
01A1 29	ADD HL,HL	;Multiplico J+P por 4 porque a representação de A(I) e B(I) é com 4 BYTES .
01A2 09	ADD HL,BC	;HL aponta para BYTE me- nos significativo de A(K4) .
01A3 22 12 10	LD (AK4),HL	;Salvo end. de A(K4) .
01A6 EB	EX DE,HL	;Coloco K4 em DE .
01A7 2A 07 10	LD HL,(GK1)	;Coloco K1 em L
01AA 67	LD H,A	;Zero H .
01AB 29	ADD HL,HL	
01AC 29	ADD HL,HL	;Multiplico K1 por 4 por causa representação de A(I) e B(I) .
01AD 19	ADD HL,DE	;Faço K5 = K4 + K1 ;HL aponta para A(K5) .

01AE 22 14 10 LD (AK5),HL ;Salvo end. de A(K5).
 01B1 01 00 04 LD BC,1024 ;O end. de B(I) está
1024 BYTES acima de A(I).
 01B4 09 ADD HL,BC ;HL aponta p/ B(K5).
 01B5 22 18 10 LD (BK5),HL ;Salvo end. de B(K5).
 01B8 EB EX DE,HL ;Coloco K4 em HL.
 01B9 09 ADD HL,BC ;HL aponta p/ B(K4).
 01BA 22 16 10 LD (BK4),HL ;Salvo end. de B(K4).
 01BD 11 1A 10 LD DE,MULT ;End. destino do multi-
plicador em módulo e sinal.
 01C0 24 14 10 LD HL,(AK5) ;HL aponte para A(K5),
em complemento a 2.
 01C3 CD B0 03 CALL COMP ;Representa A(K5) em módulo e sinal e coloca em (DE).
 01C6 ED 4B 0F 10 LD BC,(X1) ;Coloca o
COS(2*PI*S/N)/2 em BC, multiplicando.
 01CA CD 70 03 CALL ROT ;FAZ:
C1 = A(K5)*COS(2*PI*S/N) e coloca o re-
sultado em (RES), na representação do
módulo e sinal.
 01CD 11 1E 10 LD DE,C1 ;End. destino.
 01D0 21 2E 10 LD HL,RES ;End. fonte.
 01D3 CD B0 03 CALL COMP ;FAZ:
(C1) = A(K5)*COS(2*PI*S/N), na repre-
sentação complemento a 2.
 01D6 ED 4B 0D 10 LD BC,(Y1) ;SEN(2*PI*S/N)/2 p/ BC
 01DA CD 70 03 CALL ROT ;FAZ A(K5)*SEN(2*PI*S/N)
Resultado em (RES) em módulo e sinal.
 01DD 11 26 10 LD DE,C3 ;End. destino.

01E0 21 2E 10 LD HL,RES ;End. fonte .

 01E3 CD B0 03 CALL COMP ;FAZ:

 $(C3) = A(K5) * \text{SEN}(2\pi S/N)$, em complemento 2 .

 01E6 11 1A 10 LD DE,MULT ;End. destino do multiplicador em módulo e sinal .

 01E9 2A 18 10 LD HL,(BK5) ;HL aponta para B(K5) em complemento 2 .

 01EC CD B0 03 CALL COMP ;Representa B(K5) em módulo e sinal e coloca em (MULT) .

 01EF ED 4B 0D 10 LD BC,(Y1) ;SEN(2\pi S/N)/2 p/ BC .

 01F3 CD 70 03 CALL ROT ;FAZ B(K5)*SEN(2\pi S/N)

 01F6 11 22 10 LD DE,C2 ;End. destino .

 01F9 21 2E 10 LD HL,RES ;End. fonte .

 01FC CD B0 03 CALL COMP ;FAZ :

 $(C2) = B(K5) * \text{SEN}(2\pi S/N)$, em complemento 2 .

 01FF ED 4B OF 10 LD BC,(X1) ;FAZ:BC-COS(2\pi S/N)/2

 0203 CD 70 03 CALL ROT ;FAZ:B(K5) COS(2\pi S/N)

 0206 11 2A 10 LD DE,C4 ;End. destino .

 0209 21 2E 10 LD HL,RES ;End. fonte .

 020C CD B0 03 CALL COMP ;FAZ :

 $(C4) = B(K5) * \text{COS}(2\pi S/N)$, em complemento 2 .

 020F DD 21 32 10 LD IX,X2 ;End. do resultado da soma .

 0213 21 1E 10 LD HL,C1 ;End. de uma parcela .

 0216 11 22 10 LD DE,C2 ;End. de outra parcela .

 0219 CD 10 03 CALL MAIS ;FAZ: $(X2) = (C1) + (C2)$.

021C DD 21 36 10	LD IX,Y2	;End. do resultado da subtração .
0220 21 26 10	LD HL, C3	;End. do subtraendo .
0223 11 2A 10	LD DE,C4	;End. do minuendo .
0226 CD 21 03	CALL MENOS	;FAZ: $(Y2)=-(C3)+(C4)$.
0229 DD 2A 14 10	LD IX,(AK5)	;End. do resultado da subtração .
022D ED 5B 12 10	LD DE,(AK4)	;End. de A(k4),minuendo
0231 21 32 10	LD HL, X2	;End. do subtraendo .
0234 CD 21 03	CALL MENOS	;FAZ: $A(k5)=A(k4)-X2$.
0237 DD 2A 13 10	LD IX,(BK5)	;End. do resultado da subtração .
023B ED 5B 16 10	LD DE,(BK4)	;End. de B(k4),minuendo
023F 21 36 10	LD HL,Y2	;End. do subtraendo .
0242 CD 21 03	CALL MENOS	;FAZ $B(k5)=B(k4)-Y2$.
0245 DD 2A 12 10	LD IX,(AK4)	;End. da soma .
0249 ED 5B 12 10	LD DE, X2	;End. de uma parcela .
024D 21 32 10	LD HL,(AK4)	;End. de outra parcela.
0250 CD 10 03	CALL MAIS	;FAZ $A(k4)=A(k4)+X2$.
0253 DD 2A 16 10	LD IX,(BK4)	;End. da soma .
0257 ED 5B 16 10	LD DE , Y2	;End. de uma parcela .
025B 21 36 10	LD HL,(BK4)	;End. de outra parcela
025E CD 10 03	CALL MAIS	;FAZ: $B(k4)=B(k4)+Y2$.
0261 3A 11 10	LD A,(GJ)	;Recupera J .
0264 21 0A 10	LD HL,(CK3)	;End. de K3 em HL .
0267 BE	CP (HL)	;Verifica se J = K3.
0268 02 92 01	JP NZ,LOOP8	;Senão volte .
026B 3A 0B 10	LD A,(GP)	;Recupera P .
026E 16 00	LD D, Ø	

0270 5F	LD E,A	;Coloco P em DE .
0271 2A 08 10	LD HL,(GK2)	;Coloco step em HL,K2
0274 19	ADD HL,DE	;Próximo J .
0275 ED 4B 00 10	LD BC,(GN)	;N para BC .
0279 7D	LD A,L	;J em A .
027A B9	CPI C	;COMparo J com N .
027B C2 66 01	JP NZ,LOOP6	;Se J < N- 1 Volte.
027E 3A 06 10	LD A,(GK)	;Recupero K .
0281 3D	DEC A	;K= K- 1 .
0282 C2 4C 01	JP NZ,LOOP4	;Se K ≠ 0 volte .
0285 3E 01	LD A,l	;Inicializo I .
0287 32 3A 10	LOOP9 :	LD (GI),A ;Salvo I .
028A CD 37 03	CALL BINV2	;Represento em BIT inv. os M BITS menos significativos, S está em HL .
028D 3A 3A 10	LD A,(GI)	;Recupero I .
0290 DD	CP L	;Compara I com S .
0291 D2 EC 02	JP NC,PULAS	;Se I > S , pule .
0294 29	ADD HL, HL	
0295 29	ADD HL, HL	;S=4* S porque A(S) e B(S) são de 4 BYTES .
0296 01 00 03	LD BC, AI	;End. do primeiro BYTE de A(0) .
0299 11 00 04	LD DE, 1024	;B(0) está 1024 BYTES depois .
029C 09	ADD HL,BC	;HL aponta para A(S) .
029D 22 3B 10	LD (GAS),HL	;Salvo end. de A(S)
02A0 19	ADD HL, DE	;HL aponta para B(S) .
02A1 22 3D 10	LD (GBS),HL	;Salvo end. de B(S) .

02A4 26 00	LD H,0	
02A6 6F	LD L,A	;Coloco I em HL
02A7 29	ADD HL,HL	
02A8 29	ADD HL, HL	;I=I*4 porque A(I) e B(I) sao de 4 BYTES
02A9 09	ADD HL,BC	;HL aponta para A(I)
02AA 22 3F 10	LD (GAI),HL	;Salvo end. de A(I)
02AD 19	ADD HL, DE	;HL aponta para B(I)
02AE 22 41 10	LD (GBI),HL	;Salvo end. de B(I)
02B1 01 04 00	LD BC,4	;NUM. de BYTES a transferir
02B4 11 43 10	LD DE, X3	;End. destino
02B7 24 3F 10	LD HL,(GAI)	;End. fonte
02BA ED BO	LDIR	;FAZ X3 = A(I)
02BC 0E 04	LD C,4	;NUM. de BYTES a transferir
02BD 2A 43 10	LD HL,(GBI)	;End. fonte
	DE aponta para End. destino Y3	
02C1 ED BO	LDIR	;FAZ Y3 = B(I)
02C3 0E 04	LD C,4	;BYTES a transferir
02C5 ED 5B 3F 10	LD DE,(GAI)	;End. destino
02C9 2A 3B 10	LD HL,(GAS)	;End. fonte
02CC ED BO	LDIR	;FAZ A(I) = A(S)
02CE 0E 04	LD C,4	;BYTES a transferir
02D0 ED 5B 41 10	LD DE,(GBI)	;End. destino
02D4 2A 3D 10	LD HL,(GBS)	;End. fonte
02D7 ED BO	LDIR	;FAZ B(I) = B(S)
02D9 0E 04	LD C,4	;BYTES a transferir
02DB ED 5B 3B 10	LD DE,(GAS)	;End. destino

02DF 21 43 10	LD HL,(X3)	;End. fonte .
02E2 ED B0	LDIR	;FAZ A(S)= X3 .
02E4 0E 04	LD C,4	;BYTES a transferir .
02E6 ED 5B 3D 10	LD DE,(GBS)	;End. destino .
		End. fonte em HL, Y3
02EA ED B0	LDIR	;FAZ B(S)= Y3
02EC 3A 3A 10 PULAS :	LD A,(GI)	;Recupera I .
02EF 3C	INC A	;Próximo I .
02F0 21 00 10	LD HL,0N	;End. de N .
02F3 BE	CP (HL)	;Compara I com BYTES menos significativa de N .
02F4 C2 87 02	JP NZ, LOOP1 ;Se I ≠ N , volte .	
02F7 CD 30 04	CALL MODULO	;FAZ M(0) = A(0) N + M(I)=2*SQRT(A(I)^2+B(I)^2)/N para I=1 à N-1 com 16 BITS .
02FA CD D4 04	CALL TRANSF	;Representa M(I) e/ 8 BITS correspondendo e transfere para MIS e também p/ REG. zero da RAM . Aconde DISPLAY com F .
02FD C9	RET	;Fim da FFT .
02FE	DS 13H	;Área não utilizada .
	MAIS :	;FAZ (IX)=(DE)+(HL) com 4 BYTES .
0310 0E 04	LD C,4	;Nº de BYTES de cada parcela .
0312 B7	OR A	;Zero CARRY .
0313 1A LOOP10 :	LD A,(DE)	;BYTE da 1ª parcela .
0314 8E	ADC A,(HL)	;Soma e/ BYTE da segun- da parcela e com o CARRY anterior.
0315 DD 77 00	LD (IX+0),A	;Guarda soma em (IX) .

0318 13 INC DE ;Próximo BYTE da 1ª
 parcela .
 0319 23 INC HL ;Próximo BYTE da segun-
 da parcela.
 031A DD 23 INC IX ;Próximo Byte da soma .
 031C 0D DEC C ;Decrementa contador .
 031D C2 13 03 JP NZ,LOOP10 ;Se não terminou , volte
 0320 C9 RET ;Fim da soma .
 MENOS : ;FAZ (IX)=(DE)-(HL) com 4 BYTES .
 0321 0E 04 LD C, 4 ;Num. de BYTES .
 0323 E7 OR A ; Zero CARRY .
 0324 1A LOOP11 : LD A,(DE) ;Minuendo .
 0325 9E SBC A,(HL) ;Minuendo=subtraendo-
 -CARRY .
 0326 DD 77 00 LD (IX+0),A ;Diferença para (IX) .
 0329 13 INC DE ;Próximo BYTE do minuen-
 do .
 032A 23 INC HL ;Próximo BYTE do sub -
 traendo .
 032B DD 23 INC IX ;Próximo BYTE da dife -
 rença .
 032D 0D DEC C ;Decremento contador .
 032E C2 24 03 JP NZ,LOOP11 ;Se não terminou , volte.
 0331 C9 RET ;Fim .
 0332 26 08 BINV1 : LD H,8 ;Para inverter os 8 BITS,
 0334 C3 3A 03 JP,LOOP12 ;Pule .
 0337 2A 02 10 BINV2: LD HL,(GM) ;Para inverter o M BITS
 menos significativos,M está em H .
 033A 0F LOOP12 : RRCA ;BIT menos significa-
 vos p/ CY .
 ;" " " "

033B CB 15	RL L	; CY p/ BIT menos significativo .
033D 25	DEC H	; Decremento contador de BITS .
033E C2 3A 03	JP NZ,LOOP12	; Se não terminou , volte.
0341 C9	RET	; Fim, H=0 e L contém S .
	MULTIP:	; Multiplica BC por DE e coloca resultado em DEHL , não destroi BC .
0342 3E 10	LD A,16	; NUM. de BITS do multiplicando e do multiplicador .
0344 21 00 00	LD HL,0	; Limpo HL .
0347 29	LOOP13:	ADD HL,HL ; 2 vezes HL .
0348 EB	EX DE,HL	
0349 ED 6A	ADC HL,HL	
034B EB	EX DE,HL	; 2 vezes DE+CY de HL
034C D2 54 03	JP NC,Pula9	; Continua se nenhum CY
034F C9	ADD HL,BC	; Soma multiplicando .
0350 E2 54 03	JP NC,Pula9	; Se não CY , pule .
0353 13	INC DE	; DE+CY
0354 3D	PULA9:	DEC A ; Decremento contador .
0355 02 47 03	JP NZ,LOOP13	; Se não terminou , volte.
0358 C9	RET	; Fim .
	DIVIDE:	; Divide HLDE por BC, quociente em DE e resto em HL. NÃO destroi BC .
0359 3E 10	LD A,16	; Num. de Bits do divisor
035B EB	LOOP14:	EX DE,HL
035C 29	ADD HL, HL	
035D EB	EX DE,HL	; 2 vezes DE .
035E ED 6A	ADC HL,HL	; 2 vezes HL+CY de DE .

0360 B7	OR A	;Limpo CY .
0361 ED 42	SBC HL,BC	;Subtraia divisor .
0363 F2 68 03	JP P,PULAO	;Se deu positivo pulo para incrementar DE .
0366 09	ADD HL,BC	;Soma divisor .
0367 1B	DEC DE	;Decremento DE .
0368 13	PULAO: INC DE	;Incremento DE .
0369 3D	DEC A	;Decremento contador .
036A C2 5B 03	JP NZ,LOOP14	;Se não terminou, volte.
036D C9	RET	;Fim .
036E	DU 2	;Não utilizado .
	ROT :	;Multiplica BC por (MULT), 4 BYTES , dando um resultado com 6 BYTES, desprozendo os 2 menos significativos e guardando o resultado em (RES), com 4 BYTES .
0370 78	LD A,B	
0371 E5	PUSH AF	;Salva sinal do multiplicando, BIT 7 de A .
0372 CB 21	SLA C	
0374 CB 10	RL B	;Recupera módulo do Seno ou Cosseno .
0376 ED 5B 1A 10	LD DE,(MULT)	;Os 2 BYTES menos significativos do multiplicador .
037A CD 42 03	CALL MULTIP	;Multiplica BC por DE e coloca o resultado em DEHL.
037D ED 53 2E 10	LD(RES),DE	;Guarda 2 BYTES menos significativos do produto .
0381 ED 5B 1C 10	LD DE,(MULT+2)	;2 BYTES mais significativos do multiplicador .

0385 F1	POP AF	;Recupera sinal do multiplicador no BIT 7 de A.
0386 AA	XOR D	;Coloca no FLAG de sinal o sinal do produto.
0387 F5	PUSH AF	;Salva sinal do produto.
0388 CB B A	RES 7,D	;Torna multiplicador positivo.
038A CD 42 03	CALL MULTIP	;Multiplica BC por DE e coloca o produto em DEHL.
038D DD 53 30 10	LD(RES+2),DE	;Salva 2 BYTES mais significativos do produto.
0391 ED 5B 2E 10	LD DE,(RES)	;Recupera 2 BYTES da multiplicação anterior.
0395 19	ADD HL,DE	;Soma 2 BYTES das multiplicações.
0396 22 2E 10	LD(RES),HL	;Salva 2 BYTES menos significativos do produto.
0399 2A 30 10	LD HL,(RES+2)	;Recupera 2 BYTES mais significativos do produto.
039C D2 A0 03	JP NC,PULALL	;Se não houve CARRY da soma , pulo .
039F 23	INC HL	;Havendo CARRY, incrementa 2 BYTES mais significativos do produto .
03A0 F1 PULALL:	POP AF	;Recupera sinal do produto.
03A1 F2 A6 03	JP P,PULAL2	;Se positivo, Pule .
03A4 CB FC	SET 7,H	;Sendo negativo seto BIT mais significativo do BYTE mais significativo do produto .

03A6 22 30 10 PULAI2 LD(RES+2),HL ;Guardo 2 BYTES mais significativos do produto .

03A9 C9 RET ;Fim. O produto se encontra em DEHL representado em módulo e sinal .

03AA DS 6 ;Não utilizado .

COMP: ;Representa um num. de 4 BYTES inicialmente em complemento 2 em módulo e sinal ou vice-versa .

03B0 D5 PUSH DE ;Salvo end. destino .
;End. fonte está em HL .

03B1 01 04 00 LD BG,h ;Num. de BYTES ocupados pelo número .

03B4 79 LD A,G ;Salvo Num. de BYTES .

03B5 E9 B0 LDIR ;transfiro 4 BYTES de (HL) para (DE) .

03B7 AF LD C,A ;Recupero num. de BYTES .

03B8 1B DEC DE ;DE aponta para BYTE mais significativo .

03B9 1A LD A,(DE) ;Pego BYTE mais significativo .

03BA 07 RLCA ;Coloco sinal em CY e no BIT 0 de A .

03BB D2 CC 03 JP NC,PULAI3 ;Pulo, se positivo .

03BE 3F CCF ;Torno positivo .

03BF 1F RRA ;Coloco sinal positivo no num. e torno CY=1 .

03C0 12 LD (DE),A ;Salvo BYTE mais significativo .

03C1 D1	POP DE	;Recupero End. destino para apanhar BYTE menos significativo.
		O CARRY é zero .
03C2 1A	LOOP15: LD A,(DE)	;Apanho BYTE .
03C3 2F	CPL	;Complemento BYTE . B contém zero .
03C4 88	ADC A,B	;Soma BYTE com o CARRY .
03C5 12	LD(DE),A	;Salvo BYTE .
03C6 13	INC DE	;Aponço para o próximo BYTE .
03C7 0B	DEC C	;Decremento contador de BYTES .
03C8 C2 C2 05	JP NZ,LOOP15	;Se não terminou, volte.
03CB C9	RET	;Fim. Se o num. é negativo .
03CC D1	PULALP: POP DE	;Recupero posição do STACK POINTER .
03CD C9	RET	;Fim. Se o num. é positivo .
03CE	DS 2	;Não utilizado .
	RAIZ:	Faz raiz quadrada de um num. colocado em HLDE e coloca o resultado em BC .
03D0 3E 02	LD A,2	;Preciso dividir o num. por 4 para não ocorrer OVERFLOW .
03D2 CB 30	LOOP16: SRL H	;Divido H por 2 e coloco BIT 0 de H em CY .
03D4 CB 1D	RR L	;Divido L por 2 e coloco CY no BIT 7 de L .
03D6 CB 1A	RR D	;Idem para D .

03D8 CB 1B	RR E	;Idem para E.
03DA 3D	DEC A	;Decremento contador .
03DB 20 F5	JR NZ,LOOP16	;Se não terminou, volte.
03DD D5	PUSH DE	;Salvo 2 BYTES menos significativos do num .
03DE E5	PUSH HL	;Salvo 2 BYTES mais significativos do num.
03DF 01 FF FF	LD BC,FFFEH	;Primeira aproximação do resultado .
03E2 AF	XOR A	;Limpou CARRY e A .
03E3 B4	OR H	
03E4 B5	OR L	
03E5 B2	OR D	
03E6 B3	OR E	;Verifico se o num. é saída a raiz é zero .
03E7 C2 EF C3	JP NZ,PULAH	;Se não é zero, calcula a raiz .
03EA 05	INC BC	;Torno BC = zero .
03EB EB	POP HL	
03EC D1	POP DE	;Recupero posição do STACK POINTER .
03ED 09	RET	;Fim, se a raiz é zero .
03EE 00	NOP	
03EF 37	PULAH4:	SCF ;CY=1.
03F0 CB 18	LOOP17:	RR B ;CY para BIT 7 de B e BIT 6 de B para CY .
03F2 CB 19	RR C	;CY para BIT 7 de C .
		;Abaixo, verifico quantos BITS tem o número :
03F4 EB	EX DE,HL	

03F5 29	ADD HL,HL .
03F6 EB	EX DE,HL.
03F7 ED 6A	ADC HL, HL ;HLDE vezes 2 .
03F9 DA 04 04	JP C,LOOP18 ;Se num. de BITS /pox, pule .
03FC EB	EX DE,HL .
03FD 29	ADD HL,HL .
03FE EB	EX DE,HL .
03FF ED 6A	ADC HL,HL ;HLDE vezes 2 .
0401 E2 F0 03	JP NC,LOOP17 ;Se num. de BITS nõe en- contrado, volta .
0404 E1	LOOP18: POP HL
0405 D1	POP DE ;Recupero num. em HL em BC, tenho a primeira aproximação do resultado .
0406 D5	PUSH DE
0407 E5	PUSH HL ;Salvo num. para próxima iteração .
0408 CD 59 03	CALL DIVIDE ;Divido num. pela aproxi- mação atual .
040B EB	EX DE,HL ;Coloco resultado da di- visão em HL .
040C 09	ADD HL,BG ;Soma com a aproximação atual .
040D EB	EX DE,HL ;Coloco soma em DE .
040E CB 3A	SRL D
0410 CB 1B	RR E ;Divido por 2.Tenho ago- ra a nova aproximação da raiz em DE .
0412 60	LD H,B

0413 69	LD L,C	;Coloco aproximação anterior em HL
0414 42	LD B,D	
0415 4B	LD C,E	;Coloco aproximação atual em BC.
0416 B7	OR A	;Limpo o Carry .
0417 ED 52	SBC HL,DE	;Aproximação anterior mesmos a atual .
0419 CA 23 04	JP Z,PUL15	;Se igual, pulo .
0419 19	ADD HL, DE	;Recupero aproximação anterior. ;A aproximação atual está em DS e também em BC.
041D 13	INC DE	;Incremento aproximação atual em DE .
042E ED 52	SEC HL,DE	;Aproximação anterior=aproximação atual-1 .
0420 C2 04 64	JP NZ,LOOP15	;Se aproximação atual não estiver, volta e faz nova iteração.
0423 E1	PUL15: POP HL	
0424 D1	POP DE	;Recupero posição do STACK POINTER .
0425 CB 21	SLA C	
0427 CB 10	RL B	;Multiplico a raiz por 2, já que o num. foi dividido por 4.
0429 09	RET	;Fim. A raiz está em BC.
042A	DS 6	;Não utilizado .
	MODULO:	Calcula o módulo de cada Harmônico

com 16 BITS e armazena nas páginas 6
e 7 da RAM .

0430 3E 01		LD A,L ;primeiro módulo a ser calculado é do primeiro harmônico .
0432 32 4C 10	LOOP19:	LD (GC),A ;Salvo contador .
0435 6F		LD L,A
0436 26 00		LD H,β ;Coloca contador em HL .
0438 29		ADD HL,HL ;Dobra contador porque o resultado é com 2 BYTES.
0439 11 80 11		LD DE,MT ;End. destino do BUFFER temporário do módulo.
043C 19		ADD HL,DE ;HL aponta p/ end. tempo rário do módulo.
043D 22 40 10		LD(GM),HL ;Salvo end.
0440 B7		OR A ;Limpa CARRY .
0441 ED 52		SBC HL,DE ;Recupera contador dobrado.
0443 29		ADD HL,HL ;Dobra contador novamente porque o dado fonte é de 4 BYTES.
0444 01 02 08		LD BC, A(I+2) ;BC aponta para parte in teira de A(p) .
0447 09		ADD HL,BC ;HL aponta para parte in teira de A(I) .
0448 22 3F 10		LD(GAI),HL ;Salvo end. de A(I) .
044B 01 00 04		LD BC,1024 ;B(I) está 1024 BYTES de pois de A(I) .
044E 09		ADD HL,BC ;HL aponta para parte in teira de B(I) .
044F 5E		LD E,(HL)
0450 23		INC HL
0451 2C		

0451 56 LD D,(HL) ;Coloco B(I) em DE .
 0452 21 00 00 LD HL,0 ;Limpo HL .
 0455 B7 OR A ;Limpo CY
 0456 ED 52 SBC HL, DE ;HL=-B(I) .
 0458 FA 5C 04 JP M,PULAL6 ;Se-B(I) é negativo, o
 módulo de B(I) está em DE, então pule.
 045B EB EX DE,HL ;Sendo -B(I) positivo ,
 então o módulo que está em HL,é trans-
 ferido para DE .
 045C 42 PULAL6: LD B,D
 045D 4B LD C,E ;Coloco no multiplicando
 o multiplicador.
 045E CD 42 03 CALL MULTIP ;Faz B(I)^2 em DEHL .
 0461 D5 PUSH DE
 0462 E5 PUSH HL ;Salvo B(I)^2 no STACK .
 0463 2A 5F 10 LD HL,(GAT) ;Recupero End. da parte
 inteira de A(I) .
 0466 5E LD E,(HL) .
 0467 23 INC HL
 0468 56 LD D,(HL) ;Coloco A(I) em DE
 0469 21 00 00 LD HL,0 ;Mesmos comentários de
 B(I) para A(I) .
 046C B7 OR A
 046D ED 52 SBC HL,DE
 046F FA 73 04 JP M,PULAL7
 0472 EB EX DE,HL
 0473 42 PULAL7: LD B,D
 0474 4B LD C,E
 0475 CD 42 03 CALL MULTIP ;Faz A(I)^2 em DEHL

0478 C1 POP BC ;Recupero os 2 BYTES meno
 significativos de $B(I)^2$.
 0479 09 ADD HL,BC ;Soma os 2 LSB de $B(I)^2$
 com os 2 LSB de $A(I)^2$.
 047A EB EX DE,HL ;Salvo soma em DE e colo
 co os 2 MSB de $A(I)^2$ em HL .
 047B C1 POP BC ;Recupero os 2 MSB de
 $B(I)^2$.
 047C ED 4A ADC HL,BC ;Soma os 2 MSB de $B(I)^2$
 com os 2 MSB de $A(I)^2$ e com CX ante
 rior .
 047E CD D0 C3 CALL R4T ;Calcula SQR.
 SQR($A(I)^2+B(I)^2$) e coloca resultado
 em BC.
 0481 3A 03 10 LD A,(GM+1) ;Coloco N em A .
 0484 ED F4 NEG ;-N em A .
 0486 C6 09 ADD 9 ;9-N em A .
 ;O LOOP abaixo faz:
 $M(I)=SQR(A(I)^2+B(I)^2)/N$, assumo-se
 que o BYTE em B é a parte inteira
 e o BYTE em C é a parte fracionária.
 0488 CB 21 LOOP20: SLA C
 048A CB 10 RL B ;BC vezes 2 .
 048C 3D DEC A ;Decrementa contador .
 048D 20 F9 JR NZ,LOOP20 ;Se não terminou, volta.
 048F 2A 4D 10 LD HL,(GMI) ;Recupero end. destino d
 do BUFFER temporário .
 0492 71 LD(HL),C
 0493 23 INC HL

0494 70 LD(HL),B ;Coloco módulo M(I) no
 BUFFER temporário .
 0495 3A 4C 10 LD A,(GC) ;Recupero contador .
 0498 3C INC A ;Incremento contador .
 0499 21 00 10 LD HL,GN
 049C BE CP(HL) ;Comparo contador com N.
 049D C2 32 04 JP NZ,LOOP19 ;Se não acabou, volte .
 04A0 2A 02 08 LD HL,(AI+2) ;Apanho a parte inteira
 de A(0) .
 04A3 5A 03 10 LD A,(GN+1) ;M em A .
 04A6 ED 44 NEG ;-M em A .
 04A8 06 09 ADD 9 ;9-M em A .
 ;O trecho abaixo faz M(0)=A(0)/N ,as-
 sumindo que o BYTE em L é a parte fra-
 cionalária .
 04AA CB 3C SRL H .
 04AC CB 1D RR L ;Divido A(0) por 2 .
 04AE CB 25 LOOP21: SLA L
 04B0 CB 14 RL H ;Multiplico A(0) por 2
 04B2 3D DEC A ;Decremento contador .
 04B3 20 F9 JR NZ,LOOP21 ;Se não terminou, volte
 04B5 01 00 80 LD BC,8000H ;Nível DC que foi subtra-
 ido antes do cálculo da FFT .
 04B8 09 ADD HL,BC ;Restauro nível DC ori-
 ginal do sinal .
 04B9 22 80 11 LD (MI),HL ;Guardo M(0)
 04BC 2A 00 10 LD HL,(GN) ;Recupero N .
 04BF 01 81 11 LD BC,MI+1 ;BC aponta para BYTE
 mais significativo de M(0)

04C2 09 ADD HL,BC ;HL aponta para BYTE mais significativo de M(N/2)
 04C3 CB 3E SRL (HL)
 04C5 2B DEC HL
 04C6 CB 1E RR (HL) ;Divido M(N/2) por 2.
 04C8 01 00 02 LD BC,512 ;Num. de BYTES a transferir.
 04CB 11 00 0E LD DE,MI16 ;End. destino definitivo do módulo com 16 Bits.
 04CE 21 80 11 LD HL,MI ;End. temporário do módulo em 16 BITS.
 04D1 ED B0 LDIR ;Transfere 512 BYTES .
 04D3 C9 RET ;Fim. O módulo da transformada se encontra armazenado do End. DE00H a OFFFH para modificações posteriores, como apresentação dos resultados em dB .
 04D4 21 00 0E TRANSF: ;Esta subrotina representa o módulo com 8 BITS arredondado e coloca na área de memória de 1180H a 127FH e repete na área de 0800H a 08FFH , aceitando também o DISPLAY com a letra F.
 04D4 21 00 DE LD HL,MI16 ;End. do módulo com 16 BITS.
 04D7 11 80 11 LD DE,MIS ;End. do módulo a ser armazenado com 8 BITS.
 04DA ED 4B 00 10 LD BC,(GN) ;Num. de pontos a serem transferidos.
 04DE 7E LOOP22: LD A, (HL) ;BYTE menos significativo de M(I) .

04DF 87	ADD A, A ;Se BIT mais significativo é 1 ,há CARRY .
04E0 23	INC HL ;Aponta para BYTE mais significativo de M(I) .
04E1 7E	LD A, (HL) ;Pego BYTE mais significativo.
04E2 30 01	JR NC,PUL18 ;Se BIT mais significativo do BYTE menos significativo de M(I) é zero, pule.
04E4 3C	INC A ;Incrementa BYTE mais significativo .
04E5 12	PUL18: LD (DE), A ;Salva módulo arredondado com 8 BITS no novo enderêço .
04E6 13	INC DE ;Aponta para próximo end. destino .
04E7 23	INC HL ;Aponta para BYTE menos significativo do próximo módulo em 16 BITS.
04E8 0D	DEC C ;Decremento contador.
04E9 C2 DE 04	JP NZ, LOOP22;Se não terminou, volte.
04EC 01 00 01	LD BC,256 ;Num. de BYTES a serem copiados .
04EF 11 00 08	LD DE, 0800H ;End. da pág. zero da RAM .
04F2 21 80 11	LD HL, M18 ;End. onde estão armazenado os módulos do espectro .
04F5 ED B0	LDIR ;Copio os módulos com 8 BITS na página zero da RAM.
04F7 3E 71	LD A,71H ;Código da letra F no DISPLAY .

04F9 32 00 38 LD (DISP), A ;Acendo DISPLAY com F , indicando que o que está na página zero da RAM é o espectro .

04FC C9 RET ;Eim .

04FD DS 0183H ;Não utilizado .

;A seguir está a tabela de Seno, em passos de PI/128 rad DE 0 à 191*PI/128, ou seja 3 quadrantes.Para Seno endereçamos a partir do primeiro quadrante até o final do segundo (127*PI/128). Para Cosseno endereçamos 128 BYTES depois, ou seja do segundo quadrante ao final do terceiro.

$\text{Cos}(0) = \text{Sen}(64\pi/128)$ e

$\text{Cos}(127\pi/128) = \text{Sen}(191\pi/128)$.

A tabela está representada em módulo e sinal. O BIT mais significativo é o de sinal. O módulo foi obtido multiplicando o Seno (\propto) por 32767, sendo necessário multiplicar por 2 após ser determinado o sinal para trabalhar com o módulo correto , sabendo-se que os 16 BITS estão após a vírgula .

0680 0000 2403 4806 6B09	TABSEN: DW 0000H,0324H,0648H,096BH
0688 8C0C ABOF C812 E215	DW 0C8CH,0FABH,12C8H,15E2H
0690 F918 0C1C 1A1F 2422	DW 18F9H,1C0CH,1A1FH,2224H
0698 2825 2728 1F2B 112E	DW 2528H,2827H,2B1FH,2E1EH
06A0 FC30 DF33 BA36 8D39	DW 30FCH,33DFH,36BAH,398DH
06AB 5730 173F CE41 7B44	DW 3057H,3F17H,41CEH,447BH

06B0 1D47 B449 404C C04E	DW 471DH, 49B4H, 4C40H, 4EC0H
06B8 3451 9B53 F655 4358	DW 5134H, 539BH, 55F6H, 5843H
06C0 825A B450 D75E EC60	DW 5A82H, 5CB4H, 5ED7H, 60ECN
06C8 F262 E964 D066 A768	DW 62F2H, 64E9H, 66D0H, 6847H
06D0 6E6A 2460 CA6D 5F6F	DW 6A6EH, 6C24H, 6DCAH, 6F5FH
06D8 E370 5572 B673 0575	DW 70E3H, 7255H, 73B6H, 7505H
06E0 4276 6C77 8578 8A79	DW 7642H, 776CH, 7885H, 798AH
06E8 7D7A 5D7B 2A7C E47C	DW 7A7DH, 7B5DH, 7C2AH, 7CE4H
06F0 8A7D 1E7E 9D7E 0A7F	DW 7D8AH, 7E1EH, 7E9DH, 7FOAH
06F8 627F A77F D97F F67F	DW 7F62H, 7FA7H, 7FD9H, 7FF6H
0700 FF7F E67F D97F A77F	DW 7FFTH, 7FF6H, 7FD9H, 7FA7H
0708 627F 0A7F 9D7E 1E7E	DW 7F62H, 7FOAH, 7E9DH, 7E1EH
0710 8A7D E47C 2A7C 5D7B	DW 7D8AH, 7CE4H, 7C2AH, 7B5DH
0718 7D7A 8A79 8578 6C77	DW 7A7DH, 798AH, 7885H, 776CH
0720 4276 0575 B673 5572	DW 7642H, 7505H, 73B6H, 7255H
0728 E370 5F6F CA6D 246C	DW 70E3H, 6F5FH, 6DCAH, 6C24H
0730 6E6A A768 D066 E964	DW 6A6EH, 68A7H, 66D0H, 64E9H
0738 F262 EC60 D75E B45C	DW 62F2H, 60ECN, 5ED7H, 5CB4H
0740 825A 4358 F655 9B53	DW 5A82H, 5843H, 55F6H, 539BH
0748 3451 C04E 404C B449	DW 5134H, 4EC0H, 4C40H, 49B4H
0750 1D47 7B44 CE41 173F	DW 471DH, 447BH, 41CEH, 3F17H
0758 573C 8D39 BA36 DF33	DW 3C57H, 398DH, 36BAH, 33DFH
0760 FC30 112E 1F2B 2728	DW 30FCH, 2E11H, 2B1FH, 2827H
0768 2825 2422 1A1F 0C1C	DW 2528H, 2224H, 1ALFH, 1COCH
0770 F918 E215 C812 AB0F	DW 18F9H, 15E2H, 12C8H, 0FABH
0778 SC0C 6B09 4806 2403	DW 0C8CH, 096BH, 0648H, 0324H
0780 0000 2483 4886 6B89	DW 0000H, 8324H, 8648H, 896BH
0788 8C8C AB8F C892 E295	DW 8C8CH, 8FABH, 9208H, 95E2H
0790 F998 0C9C 1A9F 24A2	DW 98F9H, 9C0CH, 9F1AH, A224H

0798 28A5 27AB 1FAB 11AE
07A0 FCBO DFB3 BAB6 8DB9
07A8 57BC 17BF CEC1 7BC4
07B0 1DC7 B4C9 40CC COCE
07B8 34D1 9BD3 F6D5 43DS
07C0 82DA B4DC D7DE ECEO
07C8 F2E2 E9E4 DOE6 A7E8
07D0 6EEA 24EC CAEB 5FEE
07D8 E3FO 55F2 B6F3 05F5
07E0 42F6 6CF7 85F8 8AF9
07E8 7DFA 5DFB 2AFC E4FC
07F0 8AFD 1EFE 9DFE 0AFF
07F8 62FF A7FF D9FF F6FF

DW A528H, AB27H, AB1FH, AE11H
DW BOFCH, B3DFH, B6BAH, B98DH
DW BC57H, BF17H, C1CEH, C47BK
DW C7LDH, C9B4H, CC4OH, CECOH
DW D134H, D39BH, D5F6H, D843H
DW DAB2H, DCB4H, DED7H, EOECH
DW E2F2H, E4E9H, E6DOH, E8A7H
DW EA6EH, EC24H, EDCAH, EF5FH
DW F0E3H, F255H, F3B6H, F505H
DW F642H, F76CH, F885H, F98AH
DW FA7DH, FB5DH, FC2AH, FCE4H
DW FD8AH, FE1EH, FE9DH, FF0AH
DW FF62H, FFA7H, FFD9H, FFF6H

APÊNDICE 7

Comparação dos resultados obtidos para uma onda quadrada com os primeiros 128 pontos com amplitude 0 e os últimos 128 pontos com amplitude 255.

A 1^a coluna indica qual é o harmônico. Na 2^a coluna está o valor esperado. Na 3^a coluna está o resultado dado pelo programa da FFT em BASIC. Na 4^a coluna está o resultado dado pelo programa da FFT em ASSEMBLY.

Harmônico	Valor esperado	FFT em BASIC	FFT em ASSEMBLY
0	127,5000	127,5000	127,4922
1	162,3380	162,3421	162,3281
2	0	0	0
3	54,1127	54,1249	54,1094
4	0	0	0
5	32,4676	32,4880	32,4688
6	0	0	0
7	23,1911	23,2197	23,2031
8	0	0	0
9	18,0376	18,0743	18,0625
10	0	0	0
11	14,7580	14,8030	14,7969
12	0	0	0
13	12,4875	12,5407	12,5313
14	0	0	0
15	10,8225	10,8839	10,8750
16	0	0	0
17	9,5493	9,6189	9,6094
18	0	0	0
19	8,5441	8,6220	8,6094
20	0	0	0

Harmônico	Valor esperado	FFT em BASIC	FFT em ASSEMBLY
21	7,7304	7,8166	7,8125
22	0	0	0
23	7,0582	7,1528	7,1406
24	0	0	0
25	6,4935	6,5965	6,5781
26	0	0	0
27	6,0125	6,1240	6,1094
28	0	0	0
29	5,5979	5,7178	5,7031
30	0	0	0
31	5,2367	5,3652	5,3594
32	0	0	0
33	4,9193	5,0564	5,0469
34	0	0	0
35	4,6392	4,7840	4,7656
36	0	0	0
37	4,3875	4,5420	4,5313
38	0	0	0
39	4,1625	4,3258	4,3125
40	0	0	0
41	3,9595	4,1316	4,1250
42	0	0	0
43	3,7753	3,9564	3,9531
44	0	0	0
45	3,6075	3,7976	3,7813
46	0	0	0
47	3,4540	3,6532	3,6406
48	0	0	0
49	3,3130	3,5214	3,5156

Harmônico	Valor esperado	FFT em BASIC	FFT em ASSEMBLY
50	0	0	0
51	3,1831	3,4008	3,3906
52	0	0	0
53	3,0630	3,2900	3,2813
54	0	0	0
55	2,9516	3,1882	3,1719
56	0	0	0
57	2,8480	3,0943	3,0938
58	0	0	0
59	2,7515	3,0075	3,0000
60	0	0	0
61	2,6613	2,9271	2,9219
62	0	0	0
63	2,5768	2,8526	2,8438
64	0	0	0
65	2,4975	2,7834	2,7813
66	0	0	0
67	2,4230	2,7191	2,7031
68	0	0	0
69	2,3527	2,6593	2,6563
70	0	0	0
71	2,2865	2,6036	2,5938
72	0	0	0
73	2,2238	2,5517	2,5469
74	0	0	0
75	2,1645	2,5033	2,5000
76	0	0	0
77	2,1083	2,4581	2,4531
78	0	0	0

Harmônico	Valor esperado	FFT em BASIC	FFT em ASSEMBLY
79	2,0549	2,4160	2,4063
80	0	0	0
81	2,0042	2,3767	2,3594
82	0	0	0
83	1,9559	2,3400	2,3281
84	0	0	0
85	1,9099	2,3058	2,2969
86	0	0	0
87	1,8660	2,2740	2,2656
88	0	0	0
89	1,8240	2,2443	2,2500
90	0	0	0
91	1,7839	2,2168	2,2031
92	0	0	0
93	1,7456	2,1912	2,1875
94	0	0	0
95	1,7088	2,1675	2,1719
96	0	0	0
97	1,6736	2,1456	2,1406
98	0	0	0
99	1,6398	2,1254	2,1094
100	0	0	0
101	1,6073	2,1068	2,0938
102	0	0	0
103	1,5761	2,0898	2,0781
104	0	0	0
105	1,5461	2,0743	2,0625
106	0	0	0
107	1,5172	2,0602	2,0469

Harmonico	Valor esperado	FFT em BASIC	FFT em ASSEMBLY
108	0	0	0
109	1,4893	2,0476	2,0313
110	0	0	0
111	1,4625	2,0363	2,0313
112	0	0	0
113	1,4366	2,0264	2,0313
114	0	0	0
115	1,4116	2,0178	2,0156
116	0	0	0
117	1,3875	2,0105	2,0156
118	0	0	0
119	1,3642	2,0044	2,0000
120	0	0	0
121	1,3416	1,9996	1,9844
122	0	0	0
123	1,3198	1,9959	1,9844
124	0	0	0
125	1,2987	1,9935	2,0000
126	0	0	0
127	1,2783	1,9923	1,9844
128	0	0	0

APÊNDICE 8

OPERAÇÃO DO ANALISADOR DE ESPECTRO.

O analisador possui 5 teclas de controle: PÁG., FFT, RESET, IA, FA, cujas funções são :

PÁG. - 1º caso: A FFT não foi calculada .

-faz a rotação de páginas da RAM indicando no DISPLAY qual a página da RAM que foi movida para a página 0 da RAM. O DISPLAY acende com: 0,1,2,3,4,5,6 ou 7 .

- 2º caso: A FFT foi calculada:

-Coloca na página 0 da RAM o espetro ou o sinal amostrado do qual foi calculada a FFT. Acende o DISPLAY com F ou A, respectivamente.

FFT - Executa a rotina da FFT .

RESET - Dá o " RESET " na CPU.

IA - Dá o " RESET " na CPU e inicia a amostragem .

FA - Finaliza a amostragem quando for escrito o dado do último endereço selecionado pela chave rotativa .

Possui ainda : uma chave rotativa que seleciona quantos Bytes serão utilizados na amostragem (256, 512, 1024 ou 2048) ; uma chave rotativa que seleciona a faixa de frequência de amostragem (de 1Hz a 100kHz) ; 2 potenciômetros de ajuste de frequência, sendo um deles de ajuste fino .

EXEMPLO DE UTILIZAÇÃO :

Digamos que desejamos obter o espetro de um sinal periódico cuja frequência fundamental é de 100Hz. A chave que seleciona o número de Bytes deverá ser colocada obrigatoriamente na posição de 256 Bytes. Para termos 256 amostragens de um período

da fundamental então a frequência do oscilador deverá ser ajustada para 25.600Hz . O sinal de sincronismo para o osciloscópio ("TRIGGER") tem frequência 256 vezes menor que a do oscilador. Assim , se a frequência de amostragens for 256 vezes a frequência do sinal do qual estamos amostrando , este ficará estático na tela. Com o sinal estático ou aproximadamente estático na tela pressionamos a tecla FA e o sinal estará guardado na memória. Para calcular a FFT , apertamos a tecla FFT e esperamos 4 segundos, após o que o espectro aparecerá na tela. Não devemos esquecer que o espectro é válido até 128º harmônico(Teorema da amostragem). Pela visualização do sinal de sincronismo podemos saber qual é o harmônico, sabendo-se que num período de sinal de sincronismo temos do nível D.C. até o 255º harmônico.

OBSERVAÇÕES :

1- O osciloscópio deve ser sincronizado com a transição positiva do sinal de sincronismo .

2- O analisador gera sinais de amplitude máxima e mínima para ajuste da escala de osciloscópio, bastando para isto pressionar FA e após RESET.

Para o uso como memória de osciloscópio, a chave que seleciona o número de Bytes poderá ser colocada em qualquer posição (256, 512, 1024, 2048 pontos). Acionando a chave IA , é iniciada a amostragem. Acionando-se a chave FA os dados são amostrados até ser escrito o dado no último endereço selecionado, após o que os dados estarão guardados na memória. Através da chave PAG pode se fazer a rotação para a esquerda dos dados na memória, de 256 em 256 Bytes.

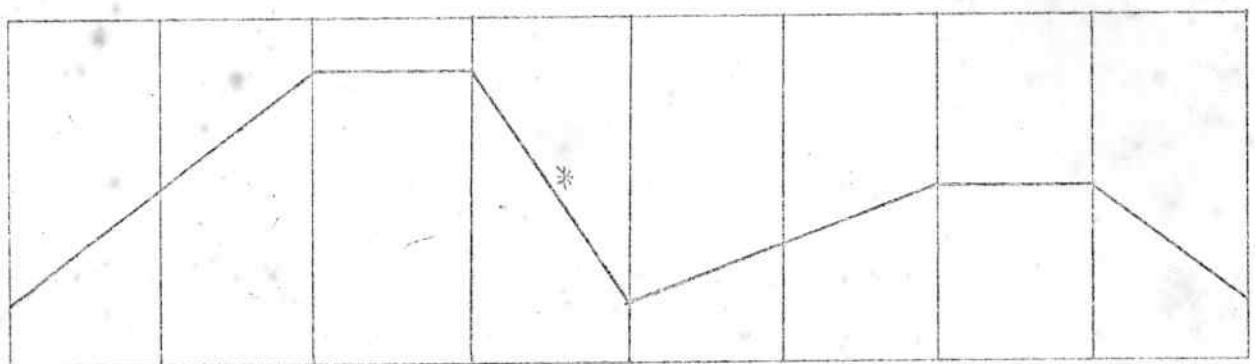
A chave PAG permite que se visualize com maior detalhe os dados de cada página de memória pela mudança da escala do osciloscópio.

EXEMPLO:

O osciloscópio e analisador ajustados para mostrar 2048 pontos(8 páginas) na tela .

DISPLAY : 0

Sinal fictício amostrado e guardado na memória :

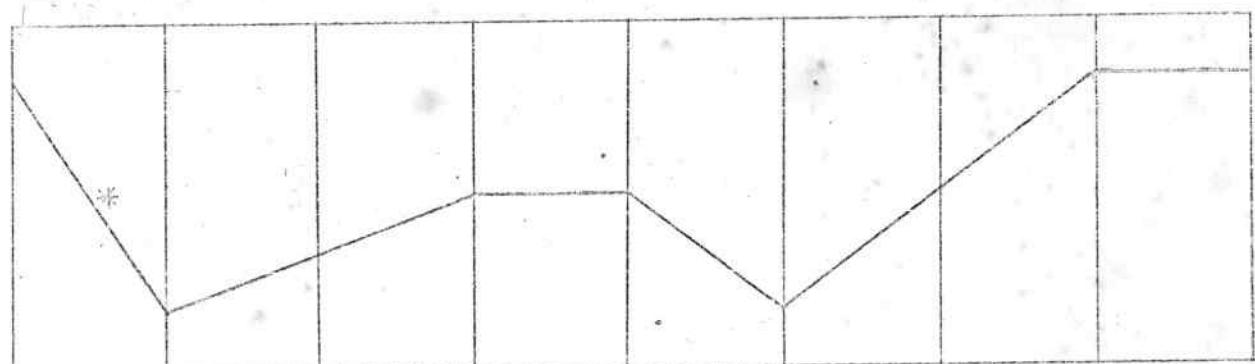


pág. 0 pág. 1 pág. 2 pág. 3 pág. 4 pág. 5 pág. 6 pág. 7

Pressionando-se PÁG 3 vêzes :

DISPLAY : 3

sinal:



pág. 0 pág. 1 pág. 2 pág. 3 pág. 4 pág. 5 pág. 6 pág. 7

Para recuperar o sinal original basta pressionar PÁG até que o DISPLAY indique 0 .

Observação : Se a tecla FFT for pressionada, obteremos o espectro do sinal que está na página 0 da memória RAM, destruindo os dados das outras páginas .