Analisador Lógico para PC

OS PEP LEGAIS

Manual Técnico

ANALISADOR LÓGICO PARA PC

OS PEP LEGAIS

Manual Técnico

Os PEP Legais
Luiz Eduardo Bueno Minioli
Marcos Paulo José
Rafael Abolafio
Robson de Sousa Martins

Liceu de Artes e Ofícios de São Paulo
4° R Eletrônica
1997

Índice:

ANALISADOR LÓGICO PARA PC

1. ESPECIFICAÇÕES TÉCNICAS

Alimentação	110 / 220V - 60 Hz
Número de canais	08
Impedância de Entrada	≥ 1 MΩ
Capacitância de Entrada	≤ 20 pF
Níveis de Tensão das Entradas	TTL (0/5V) e CMOS (0 a 12V)
Tensão Máxima suportada nas Entradas	15 V
Taxa de transferência da comunicação paralela[1]	1 KByte/s
Entrada de Sincronismo Externo	Ativada em LO / T > 100 ns
Frequência de Amostragem[2] Mínima (10000 us/div) Máxima (1 us/div)	 1 KHz 10 MHz
Frequência a ser Visualizada[3] Mínima (10000 us/div) Máxima (1 us/div)	 2 Hz 5 MHz
Faixa de Frequência Ideal[4]	10 Hz a 1 MHz

Observações:

[1] A taxa de transferência também depende do microcomputador utilizado.

[2] A frequência de amostragem é a base de tempo de cada ponto amostrado pela Interface.

[3] A faixa de frequência a ser visualizada é definida como o menor valor de frequência que produz um ciclo completo na tela do ALOGIC e a maior frequência que produz um ponto para cada semiciclo.

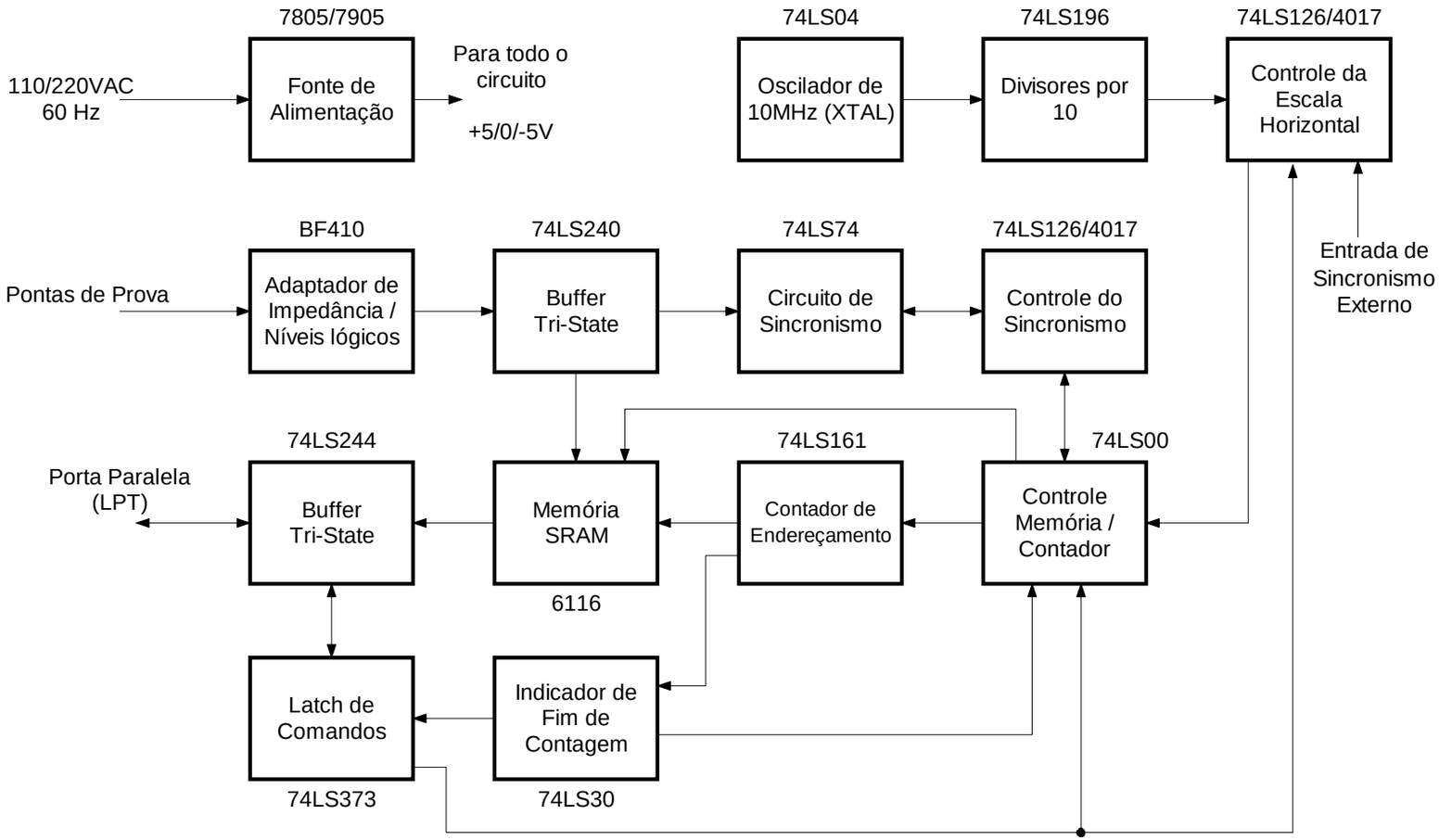
[4] A faixa de frequência ideal é definida como aquela que pode ser lida com precisão na tela do ALOGIC.

ANALISADOR LÓGICO PARA PC

2. FUNCIONAMENTO

2.1. DIAGRAMA EM BLOCOS

Analisador Lógico para PC



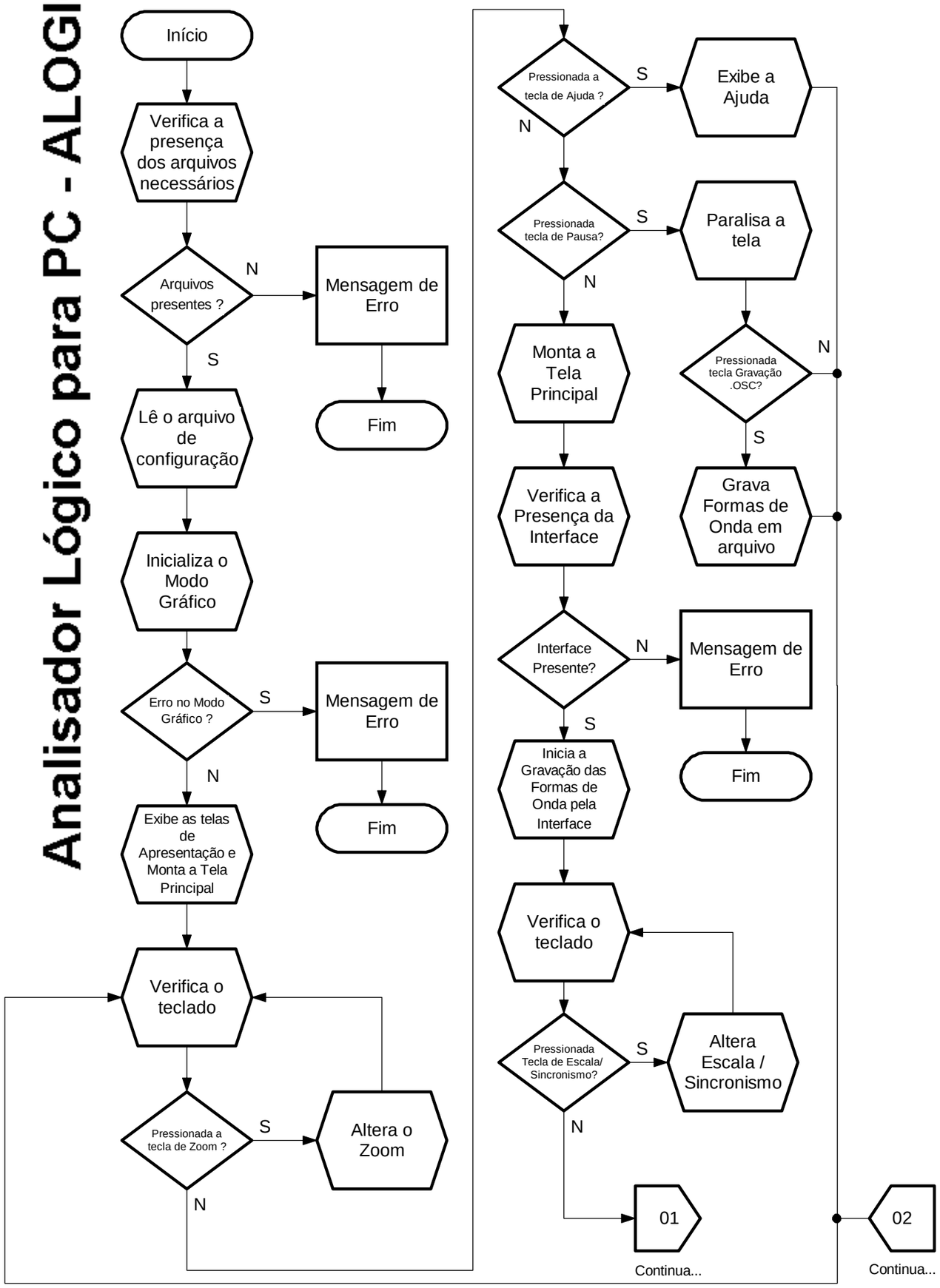
Os componentes indicados junto a cada bloco são os principais

Os PEP Legais

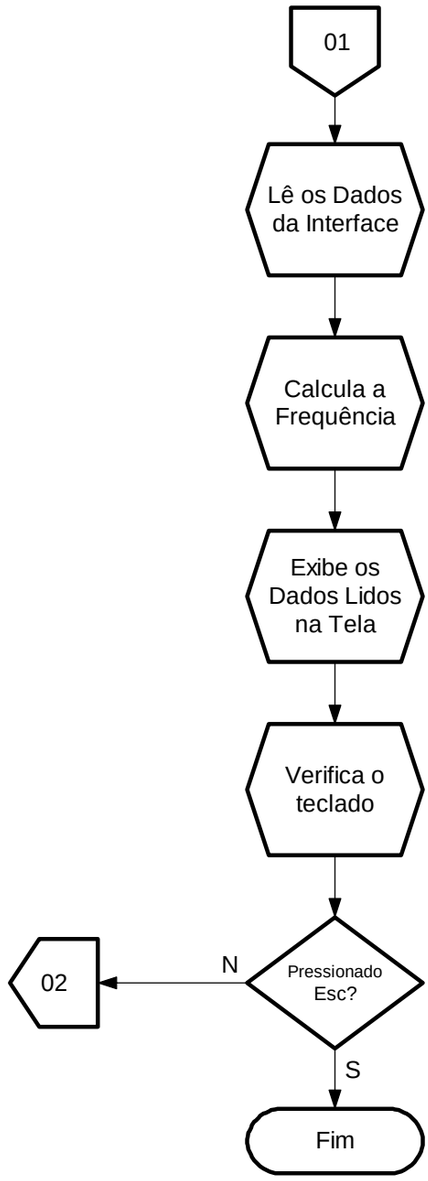
ANALISADOR LÓGICO PARA PC

2.2. FLUXOGRAMAS DO PEPSOFTWARE

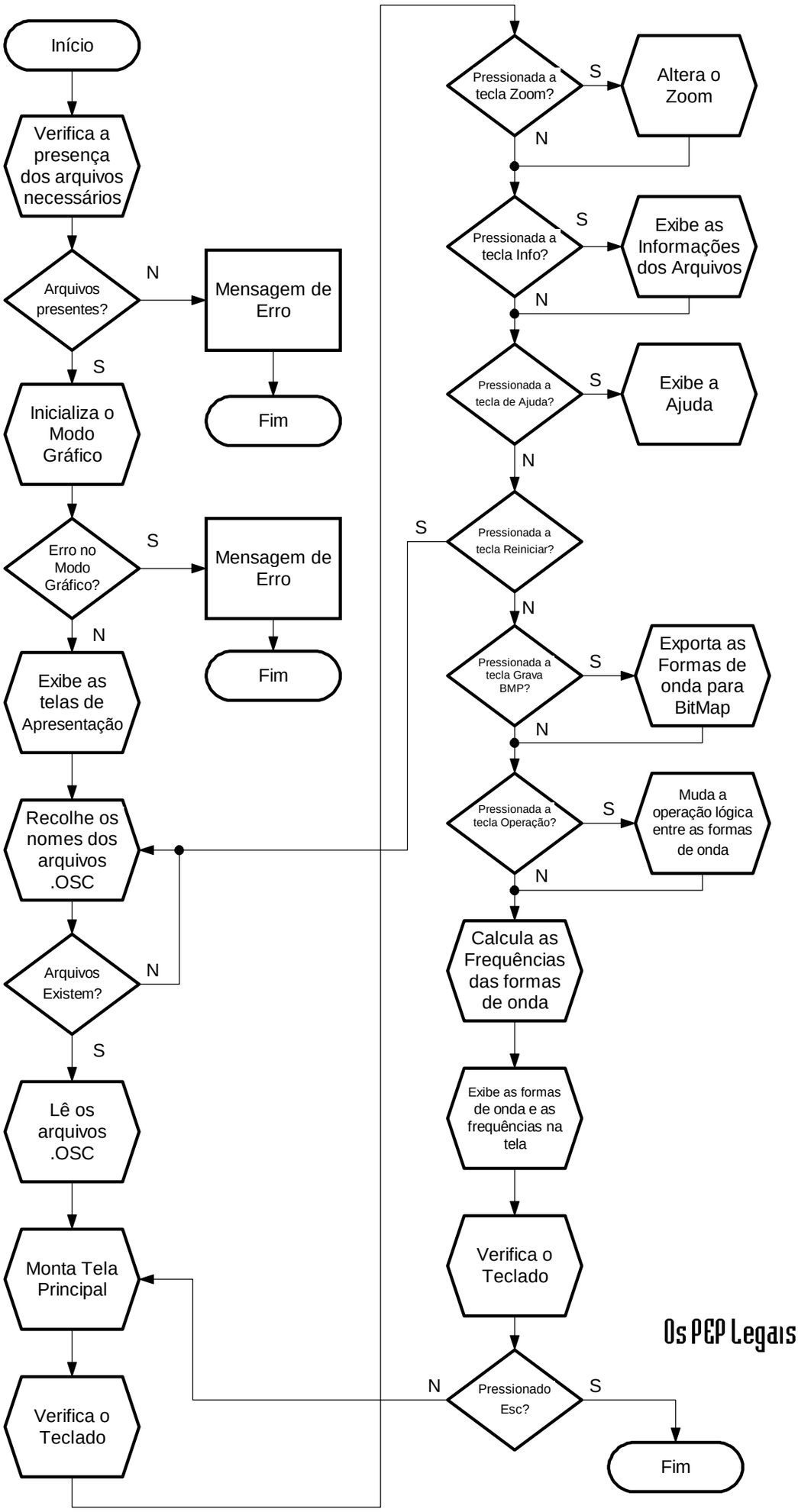
Analisador Lógico para PC - ALOGIC



Os PEP Legais

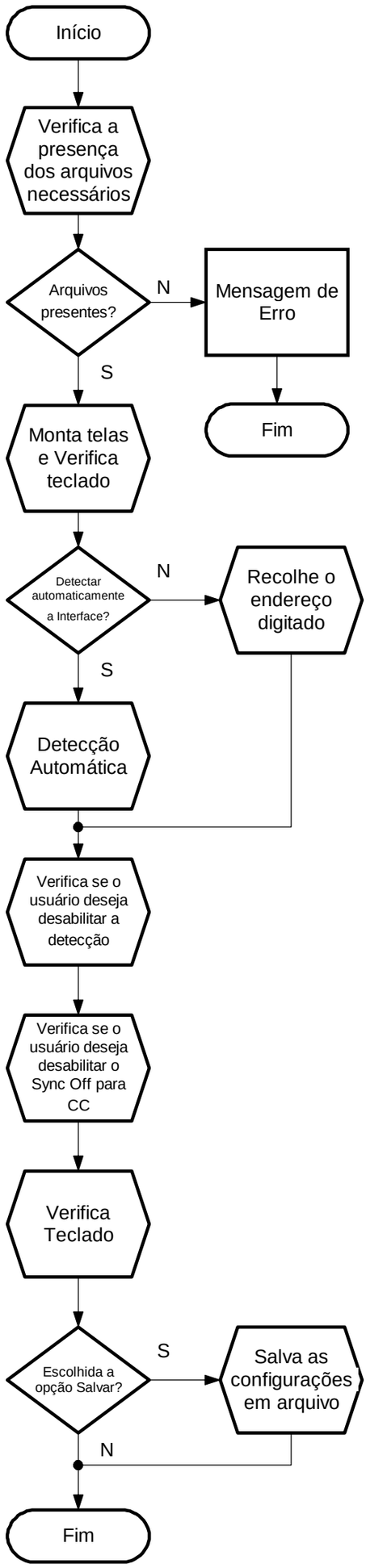

01
Lê os Dados da Interface
Calcula a Frequência
Exibe os Dados Lidos na Tela
Verifica o teclado
Pressionado Esc?
N
02
S
Fim

Analisador Lógico para PC - COMPARA



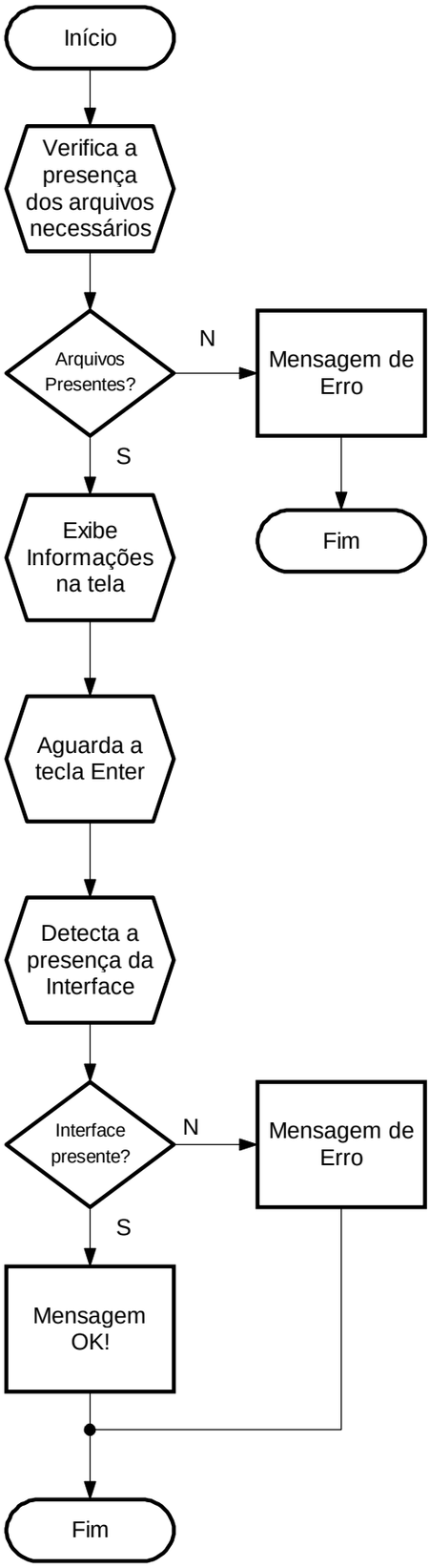
Os PEP Legais

Analisador Lógico para PC - CONFIG



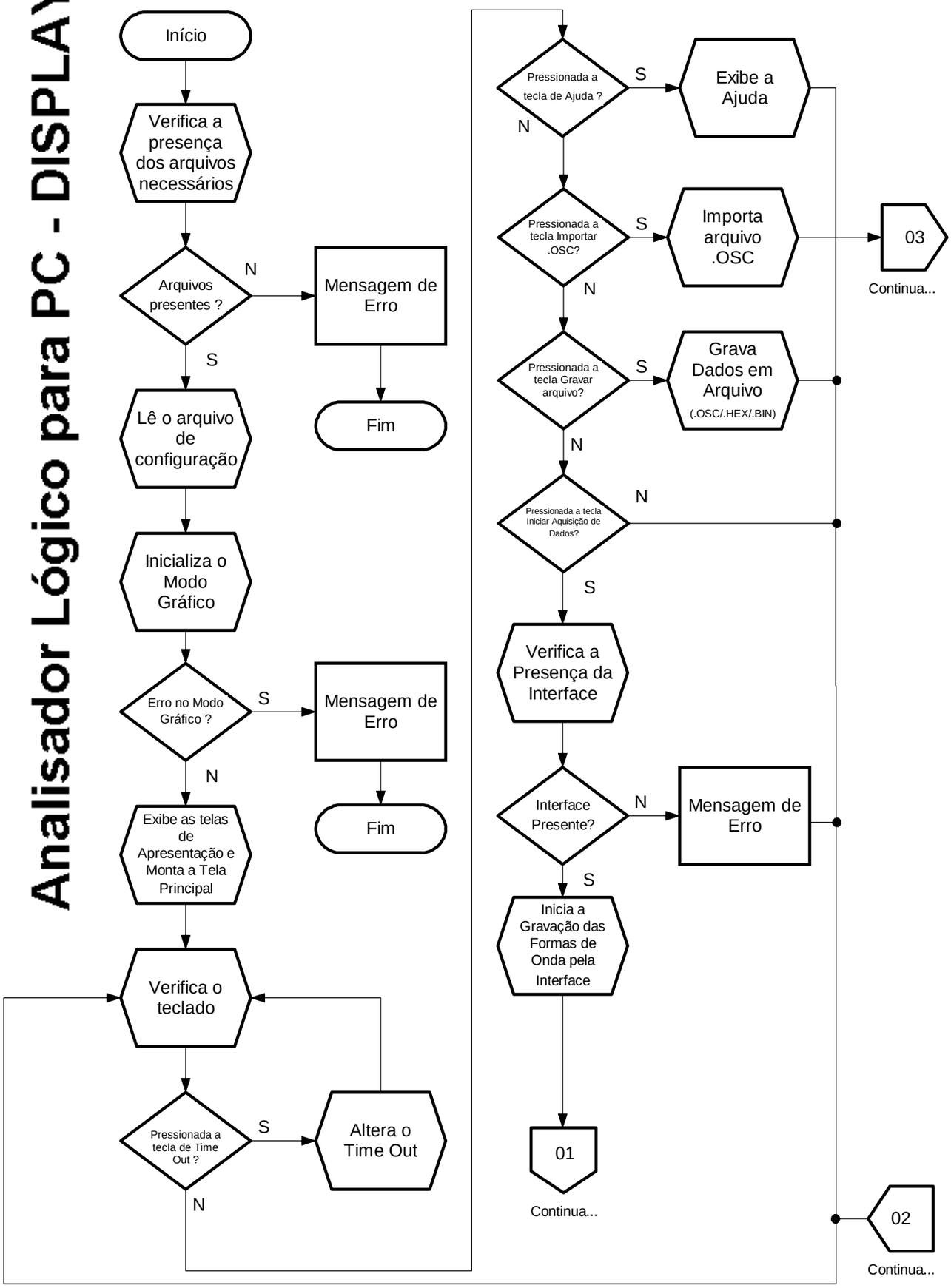
Os PEP Legais

Analisador Lógico para PC - DIAG

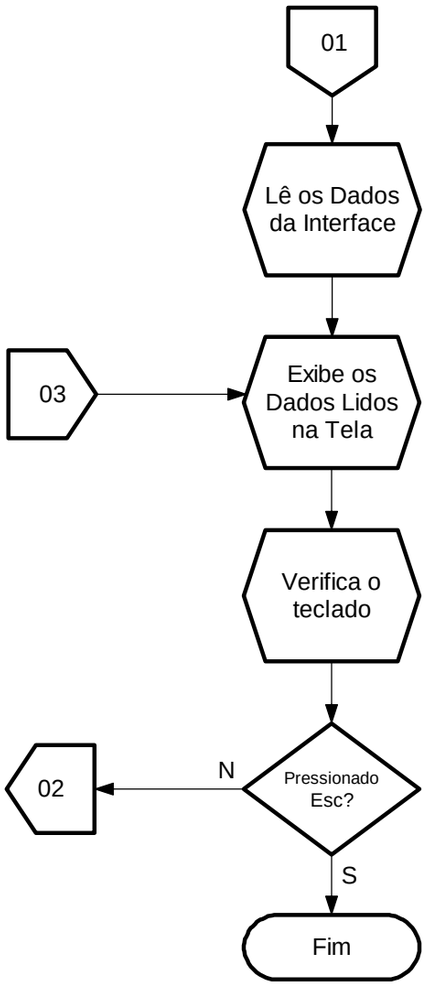


Os PEP Legais

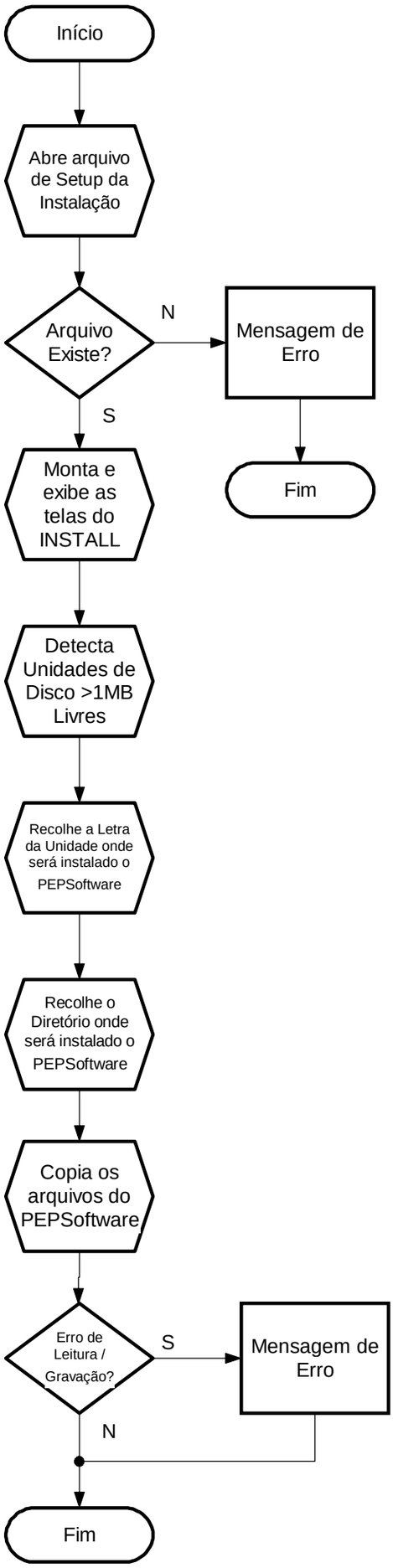
Analisador Lógico para PC - DISPLAY



Os PEP Legais

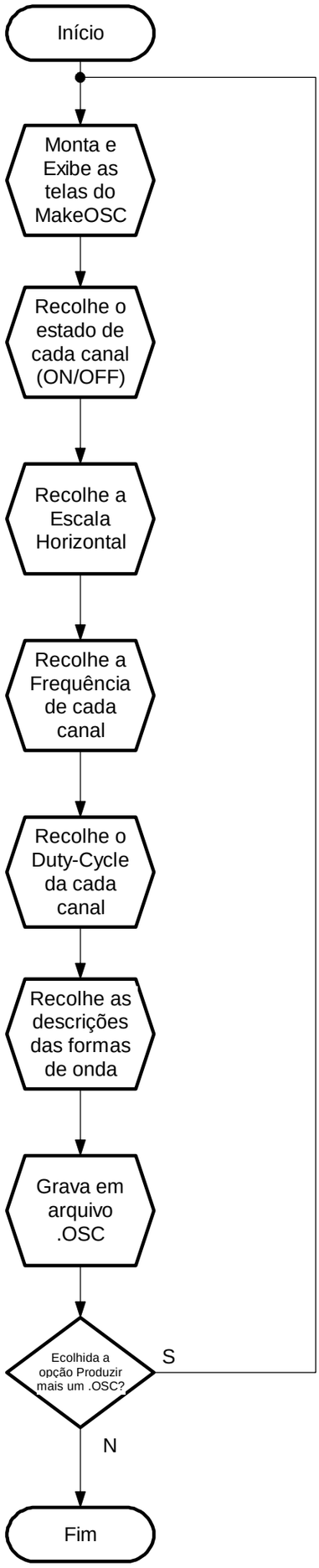

01
Lê os Dados da Interface
03
Exibe os Dados Lidos na Tela
Verifica o teclado
02
N
Pressionado Esc?
S
Fim

Analisador Lógico para PC - INSTALL



Os PEP Legais

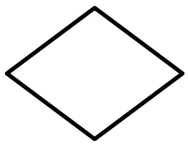
Analisador Lógico para PC - MAKEOSC



Os PEP Legais

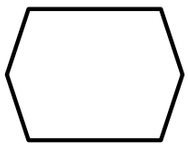
Legenda:

Terminal (Início e Fim)

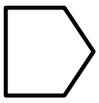


Decisão

Processo



Rotina



Mudança de Página

ANALISADOR LÓGICO PARA PC

2.3. DESCRIÇÃO DO FUNCIONAMENTO

2.3.1. Função da Interface do Analisador Lógico para PC:

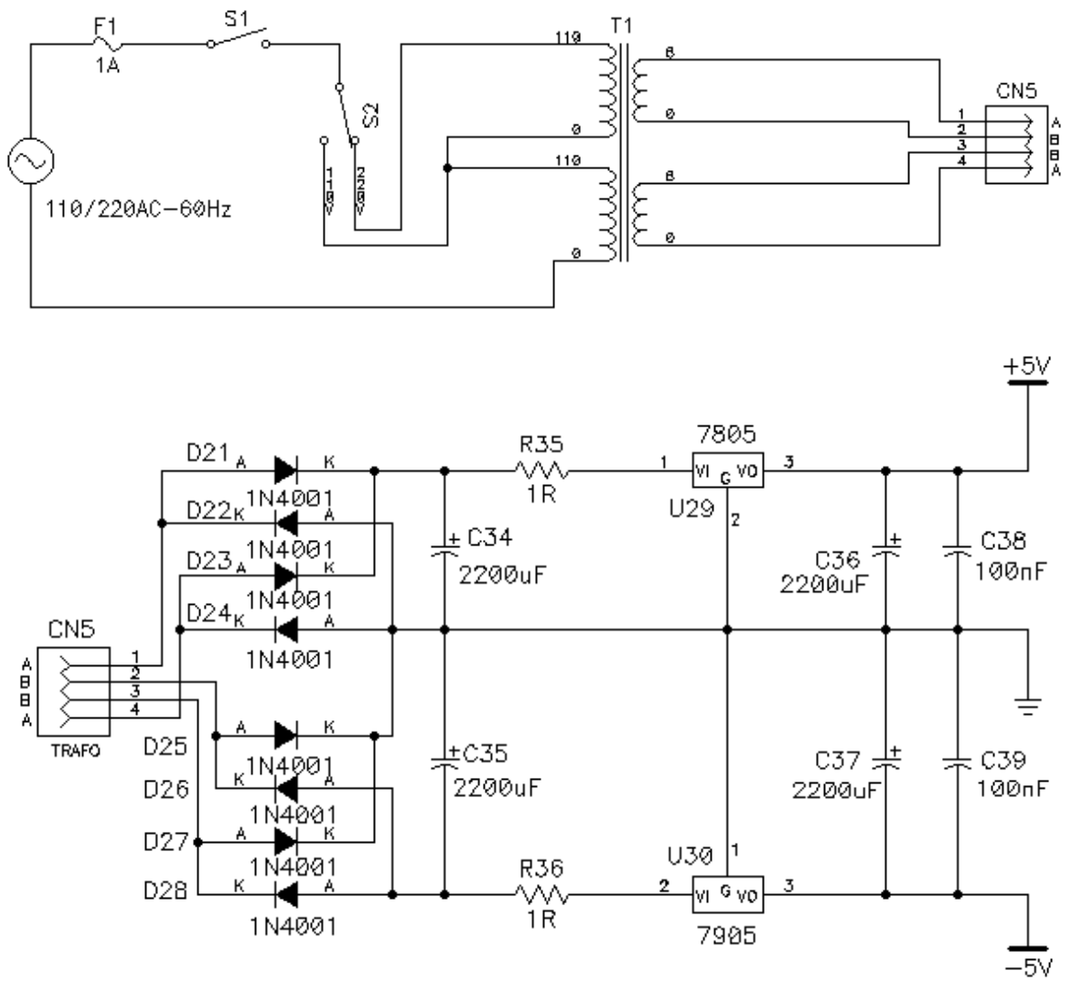
A função básica da Interface é realizar a amostragem dos sinais presentes nos oito canais, armazenar esses dados em uma memória e posteriormente transferi-los para o microcomputador através da porta paralela. O PEPSoftware então, receberá esses dados e fará o seu processamento para que possam ser visualizados na tela.

2.3.2. Descrição dos Blocos Funcionais:

A. Fonte de Alimentação:

A alimentação da Interface é realizada por uma fonte linear, composta de um transformador abaixador, um conjunto de diodos retificadores, capacitores de filtro e reguladores de voltagem. As tensões fornecidas por esta fonte são: +5V e -5V. A tensão positiva é utilizada para alimentar o circuito lógico da Interface e a negativa possibilita a adaptação de níveis lógicos pelo circuito de entrada - haverá uma explicação mais detalhada adiante.

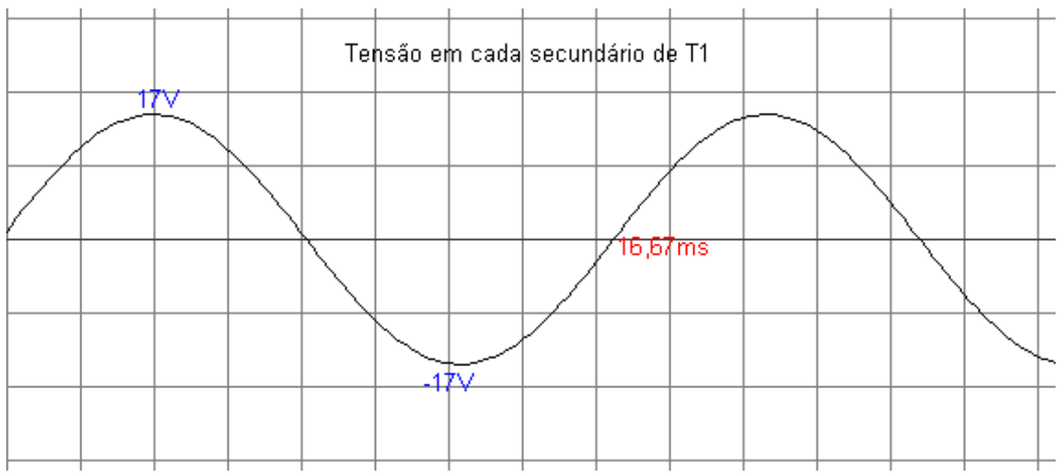
Funcionamento:

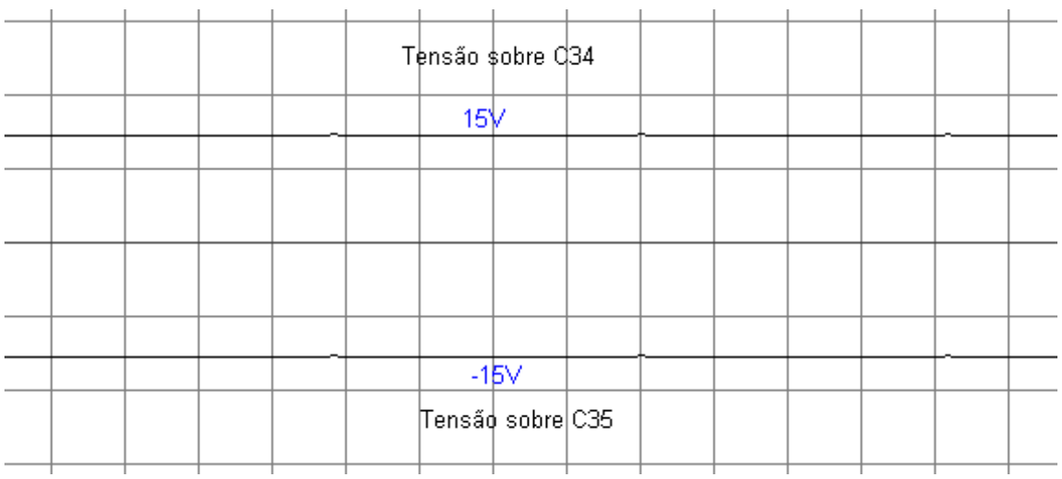


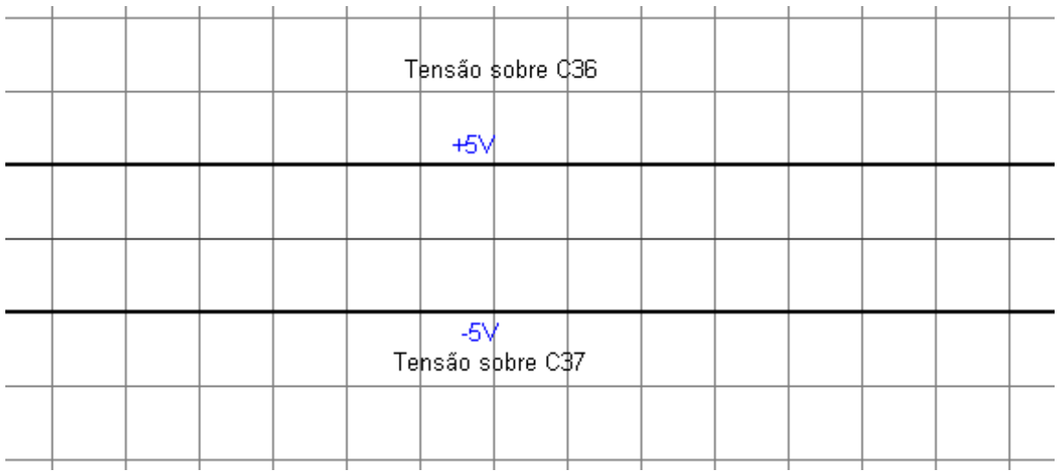
O transformador T1 reduz a tensão da rede (110 ou 220V) para 6+6V. O Fusível F1 realiza a proteção do circuito e a chave S2 seleciona a tensão de entrada (110/220V).

Por ser esta uma fonte simétrica (+5V/-5V), são utilizados dois secundários de 6V e dois retificadores em onda completa (em ponte), formados pelos diodos D21 a D28. A filtragem da tensão retificada se faz através dos capacitores C34 e C35. A tensão filtrada é regulada pelos 7805 e 7905 (U29 e U30).Os resistores R35 e R36 são limitadores de corrente para os reguladores. Os capacitores C36 e C37 colaboram com a filtragem da tensão, reduzindo muito o ripple desta fonte. Os capacitores C38 e C39 são cerâmicos, e funcionam como filtros de RF, impedindo que sinais de frequências elevadas passem para a rede de alimentação, interferindo em outros equipamentos.

Formas-de-Onda Principais:



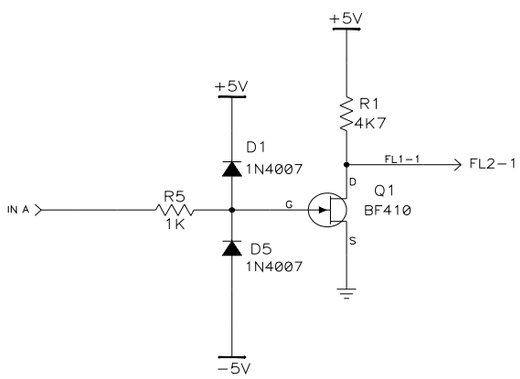




B. Adaptador de Impedância / Níveis Lógicos:

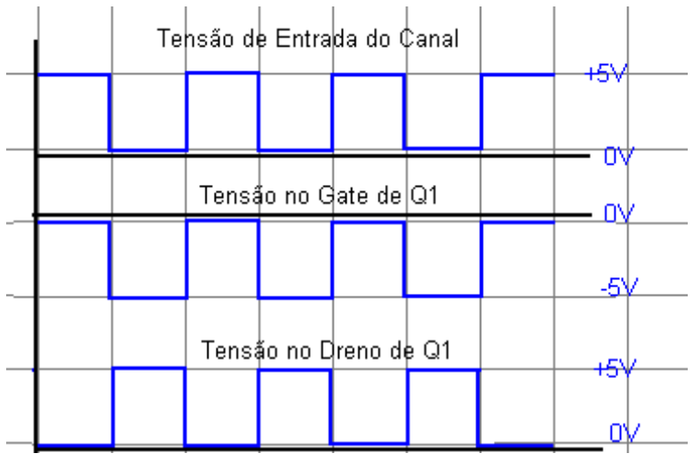
Os sinais provenientes das pontas de prova passam por uma adaptação de impedância e níveis lógicos, função esta realizada por amplificadores com JFET (Transistor de Efeito de Campo de Junção). As principais características desse circuito são a alta impedância oferecida pelos JFET e a possibilidade de se aplicar tensões maiores que 5V nas entradas, tornando o Analisador Lógico compatível também com níveis de tensão CMOS (0 a 12V).

Funcionamento:



Acima é demonstrado o circuito para apenas um dos canais (existem oito circuitos semelhantes no Analisador Lógico para PC). O transistor Q1 forma um chaveador, que opera somente em corte e saturação. O resistor R5 associado aos diodos D1 e D5, possibilita uma limitação de tensão para a entrada de cada canal, de modo que tensões acima de 5V e abaixo de 0V sejam ceifadas. Como o transistor é um JFET canal N, é utilizada uma tensão de -5V como referência na entrada, para que variações entre 0 e 5V na entrada do canal provoquem variações de -5V a 0V no gate do JFET. Deste modo, quando a tensão de entrada for 0V, haverá uma tensão de -5V no gate, o transistor entrará em corte e a tensão no dreno será de 5V. Se a tensão de entrada for de 5V, a tensão no gate será 0V, o JFET saturará e no dreno existirá uma tensão de 0V.

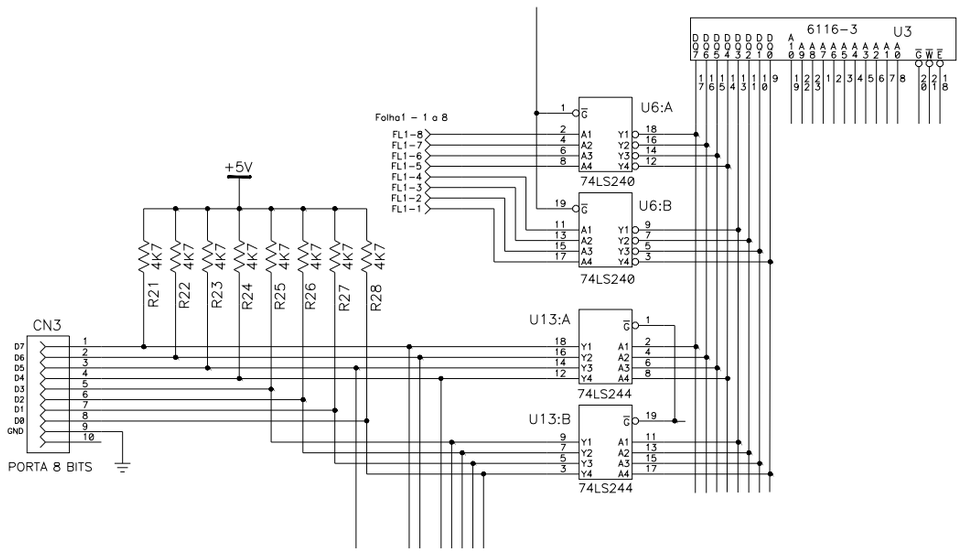
Formas-de-Onda Principais:



C. Controle de Entrada/Saída da Interface (Buffers Tri-State):

São utilizados dois Buffers Tri-State para controlar o tráfego de informações da Interface do Analisador Lógico: um deles (74LS240) é ativado quando for realizada a entrada dos dados amostrados das pontas-de-prova para gravação na memória temporária; o outro (74LS244) é acionado no momento em que os dados trafegam da memória da Interface para o microcomputador através da porta paralela.

Funcionamento:



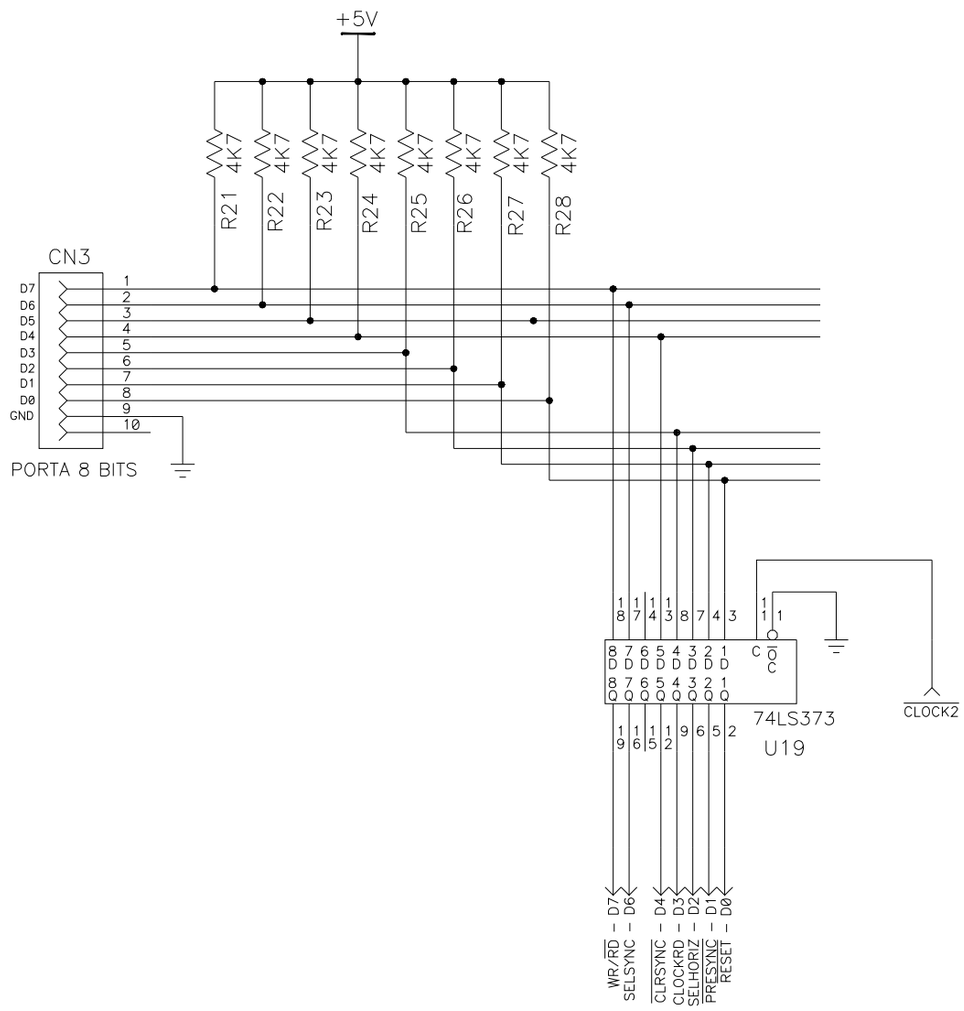
O 74LS240 (U6) é um buffer tri-state de entrada, ou seja, conecta os sinais vindos dos adaptadores de impedância / nível lógico ao barramento de dados da memória 6116 (U3) quando o microcomputador, através do PEPSoftware envia o comando de gravação de dados pela Interface. Nesse instante, a memória está sendo ativada para gravação e o buffer U13 está em tri-state. Quando o PEPSoftware requisita os dados gravados pela Interface, U6 entra em tri-state e U13 (74LS244) é acionado, fazendo com que a porta paralela seja diretamente conectada ao barramento de dados da memória U3, que está selecionada para leitura.

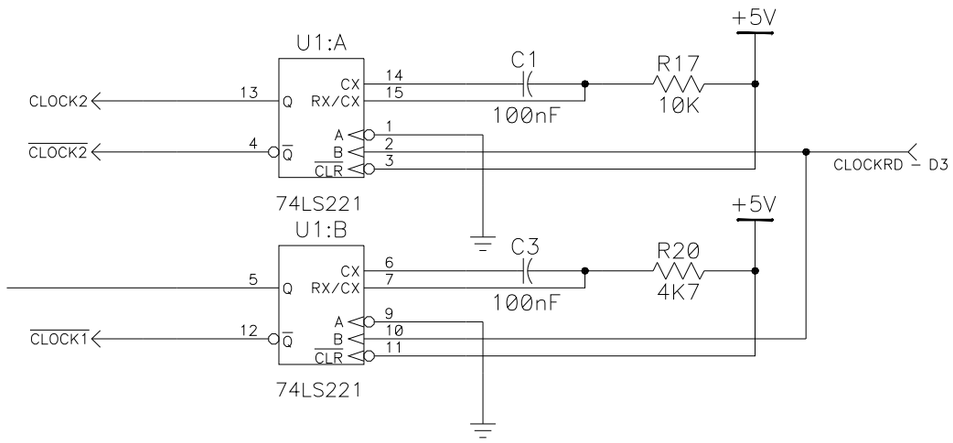
Foi utilizado um buffer inversor (74LS240) porque a etapa de entrada (formada pelos JFET Canal N) realiza uma inversão nos sinais vindos das pontas-de-prova. Sendo assim, na memória são gravados dados correspondendo exatamente às formas-de-onda presentes nas pontas-de-prova.

D. Latch de Comandos:

Para que o PEPSoftware possa controlar de maneira eficaz a Interface, existe um latch 74LS373 que armazena os comandos enviados pelo microcomputador. Com isso, a porta paralela é liberada, de modo que os dados gravados na memória (U3) trafeguem para o PC.

Funcionamento:

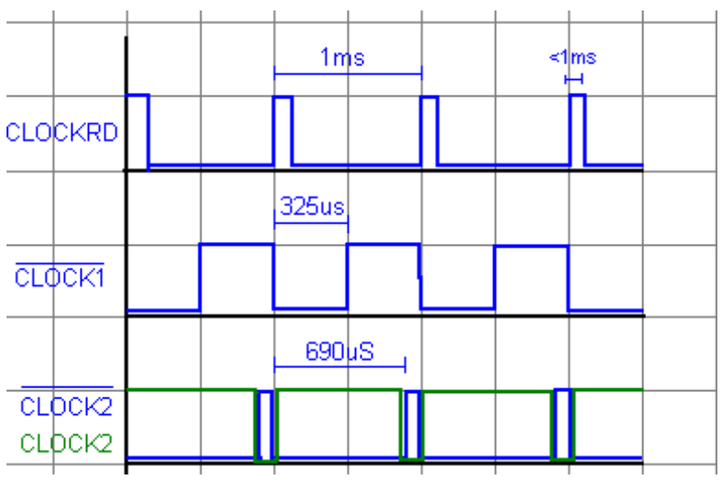




O circuito do Latch de Comandos funciona da seguinte maneira: quando o microcomputador envia um comando para a Interface, através da porta paralela, o 74LS373 (U19) recebe esse byte e o armazena. O controle do latch é feito através de dois multivibradores monoestáveis (U1). Quando não for recebido um sinal de CLOCKRD, isto é, a Interface estiver gravando em sua memória os sinais vindos das pontas-de-prova, os monoestáveis estarão em repouso. O pino 4 de U1 estará em nível alto e U19 estará habilitado de modo que sua saída corresponda a sua entrada. No momento que o microcomputador solicita a leitura da Interface, é aplicado um pulso em CLOCKRD. Com isso os monoestáveis produzirão em suas saídas pulsos com duração determinada por C1/R17 e C3/R20. Assim, o pino 4 de U1 passará a nível baixo, fazendo com que U19 armazene o comando enviado pelo microcomputador e a porta paralela esteja disponível para o envio de dados da Interface para o PC.

Os monoestáveis também atuam nos buffers tri-state para controle de entrada/saída da Interface (U6 e U13).

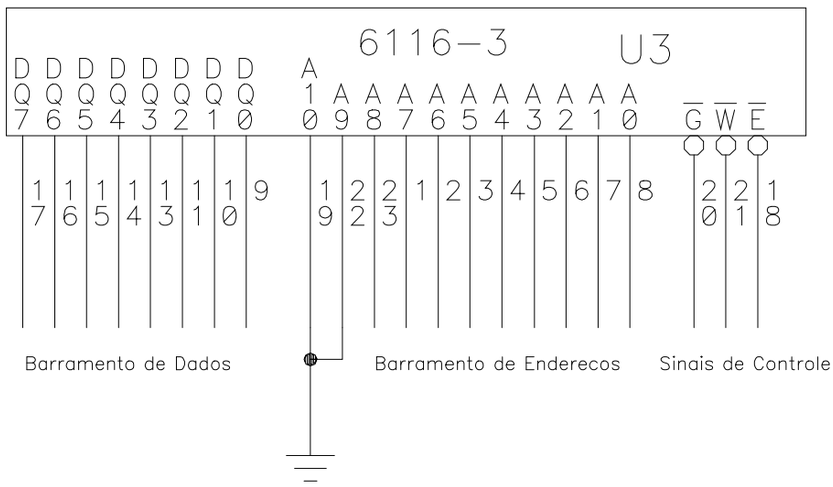
Formas-de-Onda Principais:



E. Memória SRAM:

Na memória SRAM 6116 (2K x 8) são armazenadas amostras dos sinais presentes nas pontas-de-prova, dados esses que são lidos posteriormente pelo microcomputador e processados até que sejam visualizados na tela do PC.

Funcionamento:

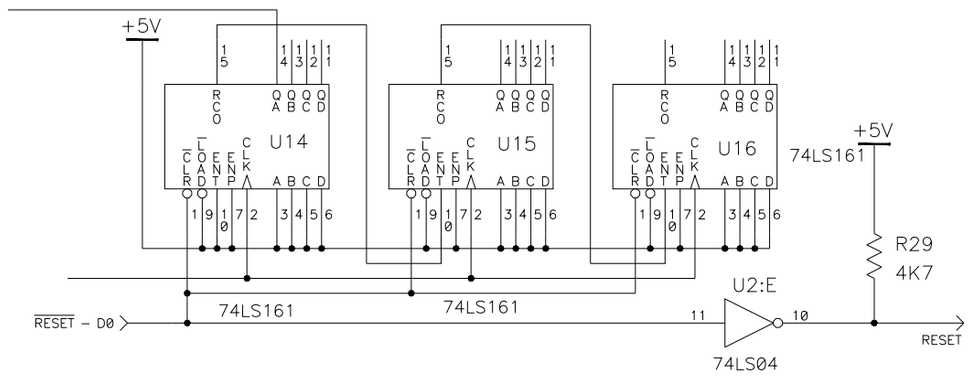


É utilizada uma memória 6116 (2K x 8 bits), porém ela é endereçada somente até 512 bytes ($2^9 = 512$). A característica mais importante dessa memória é o tempo de acesso extremamente reduzido (até 50 ns), pois a amostragem dos sinais provenientes das pontas-de-prova pode ser realizada (dependendo da escala horizontal) a uma frequência de 10MHz.

F. Contador de Endereçamento:

Para que a memória seja endereçada sequencialmente, tanto no processo de leitura como de gravação, é utilizado um contador binário síncrono de 9 bits (conta de 0 a 511).

Funcionamento:



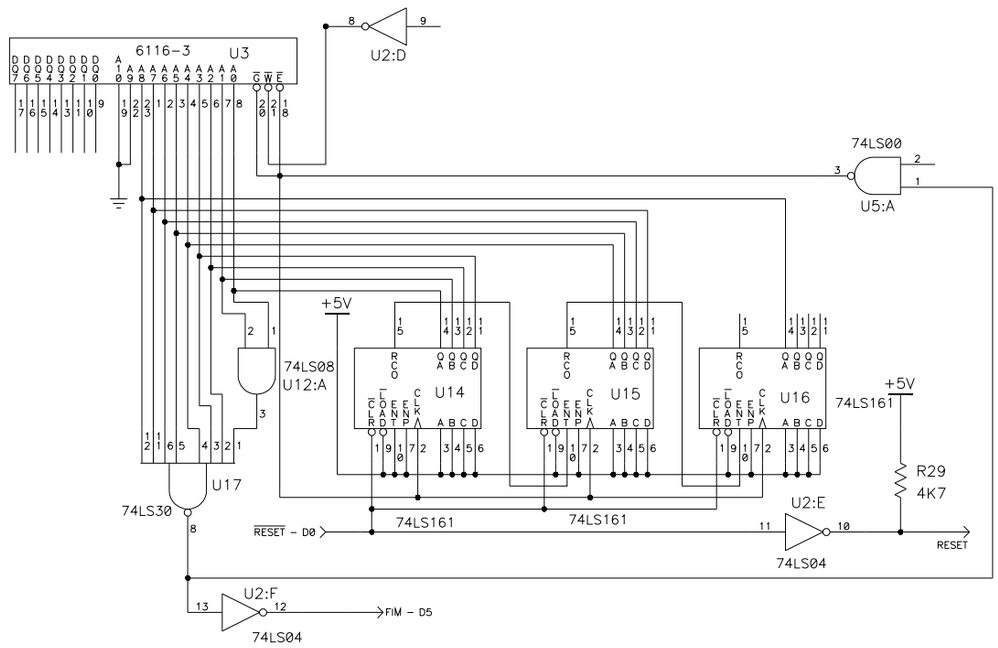
U14, U15 e U16 são contadores de 4 bits que foram associados para realizar a função de um contador de 9 bits. Os 74LS161 são contadores síncronos, o que possibilita que suas saídas mudem de nível lógico simultaneamente, desta forma, endereçando corretamente a memória 6116.

O clock aplicado no contador de endereçamento depende da operação que estiver sendo realizada na Interface. Se a memória estiver sendo habilitada para gravação (os dados amostrados das pontas-de-prova sendo gravados na 6116), o clock será gerado por um oscilador interno que produz uma base de tempo para a amostragem. Se o microcomputador estiver lendo a memória da Interface, o clock será produzido pelo próprio PC, de modo que a transferência de dados seja feita a uma velocidade adequada à porta paralela.

G. Indicador de Fim de Contagem:

O Indicador de Fim de Contagem é responsável por avisar ao microcomputador que a memória 6116 foi gravada completamente com amostras dos sinais presentes nas pontas-de-prova, de maneira que o PEPSoftware possa realizar a leitura, processamento e exibição desses dados.

Funcionamento:



O Indicador de Fim de Contagem é formado pela porta AND (U12:A), NAND (U17) e Inversora (U2:F). Enquanto os contadores estão sendo incrementados, o sinal Fim estará em nível baixo. Quando U14, U15 e U16 endereçarem a última posição de memória (1FFH ou 511) a saída de U17 apresentará nível baixo e, consequentemente, o sinal Fim estará em nível alto.

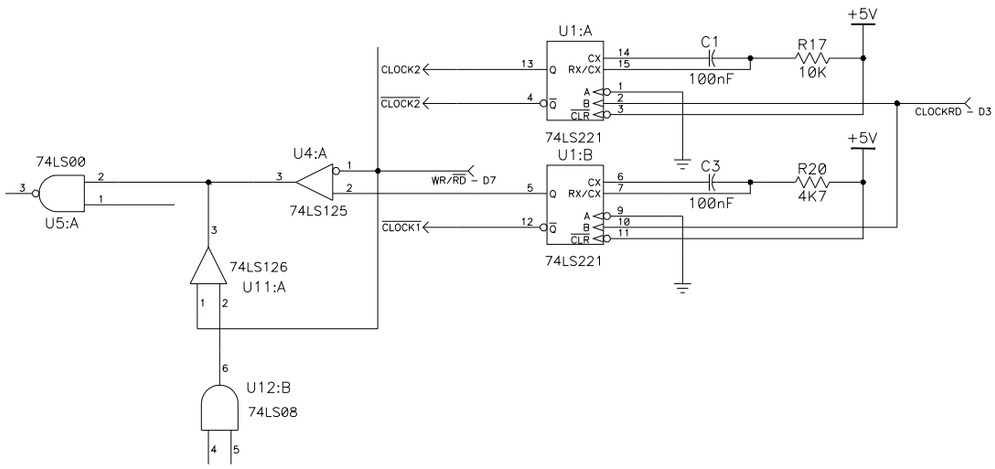
Tabela Verdade do Indicador de Fim de Contagem:

Entradas de U17	Saída de U17	Sinal Fim	Contagem em U14, U15 e U16
Qualquer uma em Lo	Hi	Lo	de 0 a 510 (000H a 1FEH)
Todas em Hi	Lo	Hi	511 (1FFH)

H. Controle de Memória / Contador de Endereçamento:

Para que a memória 6116 trabalhe em sincronismo com o Contador de Endereçamento e com o PEPSoftware, existe um circuito que realiza o controle do fluxo de dados entre a SRAM (U3) e o microcomputador e também controla o endereçamento da memória.

Funcionamento:



A porta NAND (U5:A) é utilizada para provocar o bloqueio do clock para o Contador de Endereçamento quando o sinal Fim estiver ativado. Com isso, a Interface paralisa, aguardando novo comando enviado pelo microcomputador.

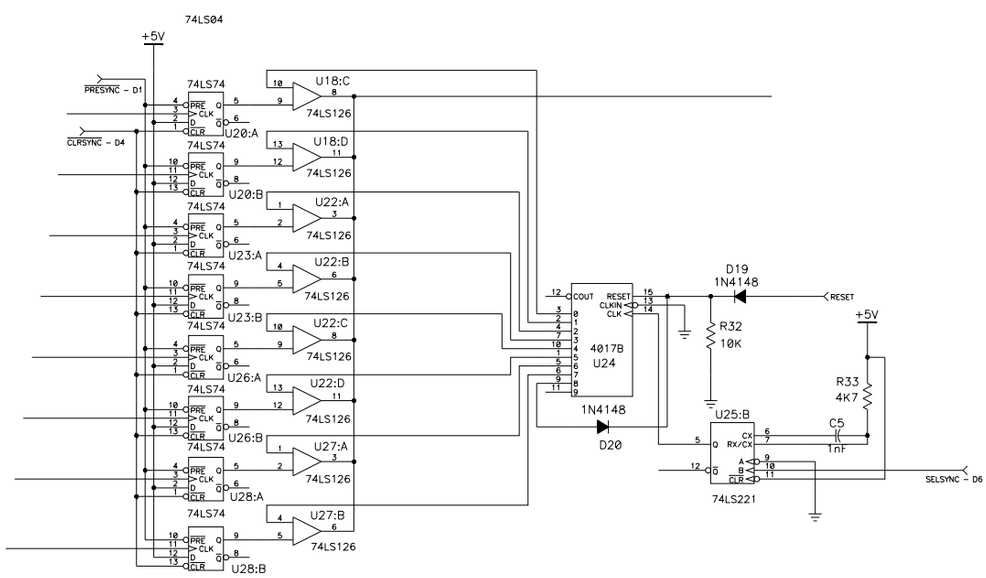
U4:A e U11:A são portas inversora e não-inversora, respectivamente, com tri-state. Realizam a seleção da origem do clock para o endereçamento da memória: interno, produzido pelo oscilador presente na Interface, quando estiver sendo realizada a amostragem dos sinais vindos das pontas-de-prova; e externo, enviado pelo PC através da porta paralela durante a leitura dos dados gravados na SRAM da Interface.

A porta AND (U12:B) permite a liberação do clock interno para o Contador de Endereçamento somente quando a etapa de sincronismo enviar um nível alto, isto é, a gravação das amostras na 6116 será sempre iniciada na borda de subida do sinal presente no canal base de sincronismo.

I. Sincronismo:

O sincronismo dos sinais presentes nas pontas-de-prova permite que as formas-de-onda exibidas na tela do PC fiquem paralisadas se o sinal for periódico. Essa função é realizada por flip-flops, que somente liberam a amostragem quando ocorrer uma borda de subida na forma-de-onda presente no canal base de sincronismo.

Funcionamento:



Os flip-flops utilizados são tipo D, sendo que na entrada D é aplicado nível 1. Quando deve ser iniciada a amostragem, o microcomputador envia um bit 0 no pino ClrSync. Com isso, todos os flip-flops apresentam 0 em suas saídas Q. Assim que o ClrSync for a nível 1, os biestáveis estarão prontos para detectar uma borda de subida. Quando isso ocorrer, a saída Q do flip-flop correspondente vai a nível alto, fazendo com que a amostragem seja iniciada. Se não ocorrer uma borda de subida, isto é, se o sinal presente no canal base de sincronismo for uma tensão constante, após um tempo limite é enviado pelo PC um bit 0 no pino PreSync. Assim, a Interface é destravada, iniciando a amostragem mesmo sem a borda de subida.

A seleção do canal base de sincronismo é feita através de um contador 4017B (U24), auxiliado por buffers tri-state e um monoestável. O estado do contador é alterado através do PEPSoftware. Toda a vez que a Interface for reiniciada, U24 é zerado e o estado do contador dependerá da escolha do canal base de sincronismo. Os buffers tri-state estarão em alta impedância, exceto o que estiver selecionado por U24.

O multivibrador monoestável (U25:B) tem por função aumentar o tempo do pulso enviado pelo PC no pino SelSync, evitando que ruídos provoquem a comutação de U24.

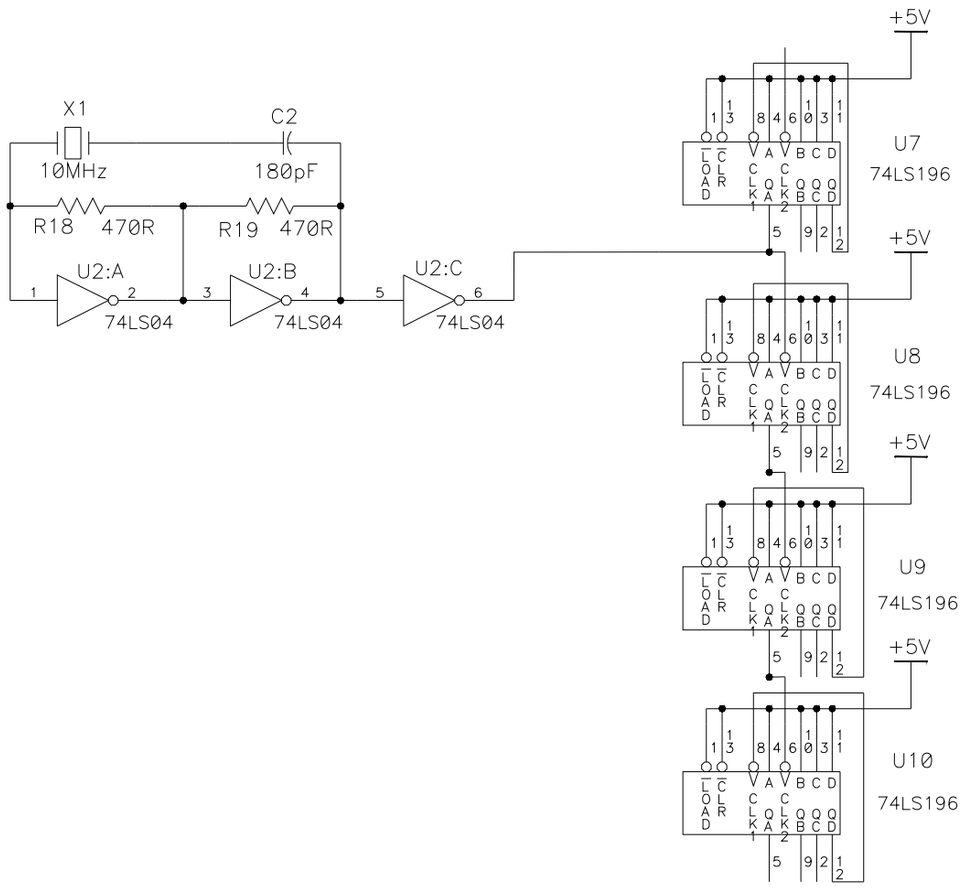
Pulsos necessários para selecionar o canal base de sincronismo:

Canal Base de Sincronismo	Pulsos em SelSync
A	0
B	1
C	2
D	3
E	4
F	5
G	6
H	7

J. Oscilador a Cristal e Divisores por 10:

Para a realização das bases de tempo necessárias para a amostragem dos sinais presentes nas pontas-de-prova, é utilizado um oscilador baseado em um cristal de 10 MHz em conjunto com divisores de frequência por 10.

Funcionamento:



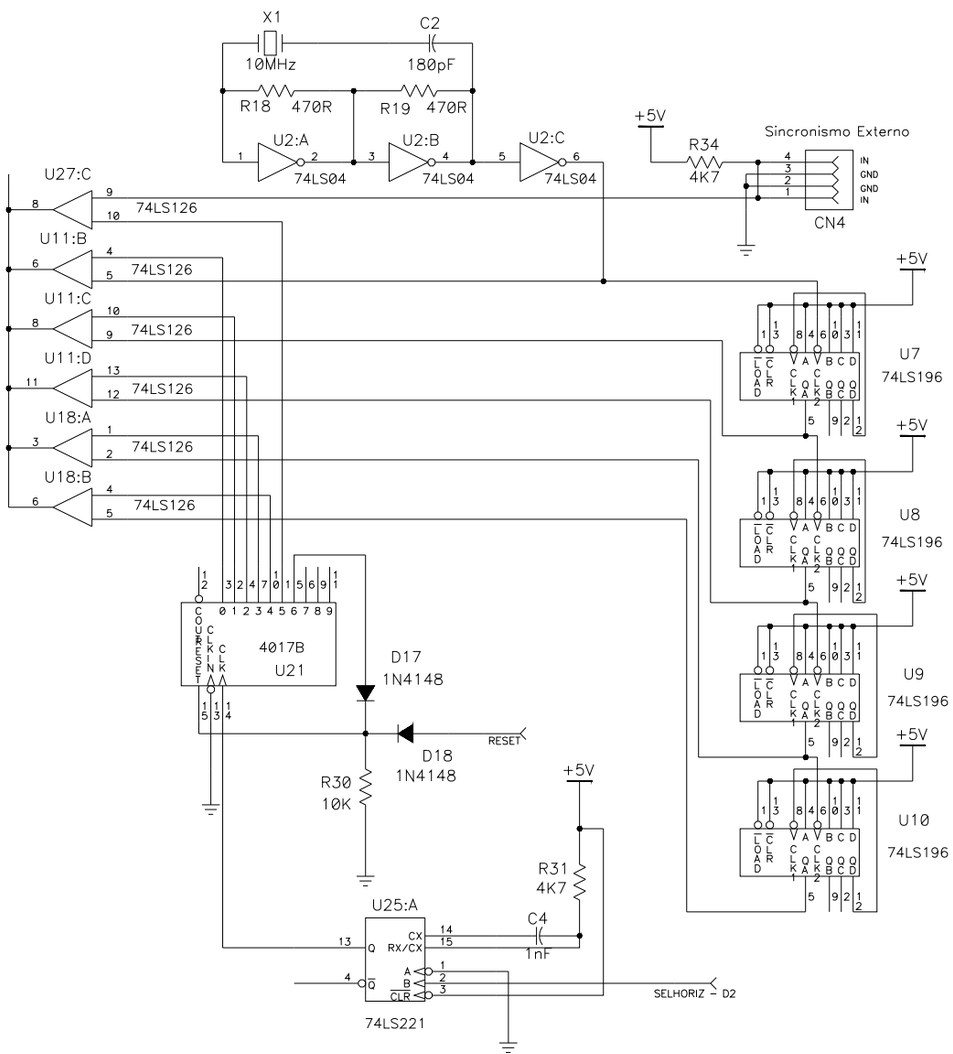
O oscilador de 10MHz é formado por três portas inversoras (U2), dois resistores (R18 e R19), um capacitor (C2) e um cristal (X1). Os componentes passivos formam uma malha de realimentação para os inversores, de forma que haja uma realimentação positiva na frequência de oscilação do cristal.

São também utilizados divisores por 10 (U7 a U10), que são contadores de década assíncronos. Assim, são obtidas as seguintes frequências para as base de tempo da amostragem: 10MHz, 1MHz, 100KHz, 10KHz e 1KHz (1us/div, 10us/div, 100us/div, 1000 us/div e 10000 us/div).

L. Controle da Escala Horizontal:

O circuito de controle da escala horizontal é semelhante ao circuito de seleção do canal base de sincronismo. A Interface do Analisador Lógico para PC permite 6 escalas diferentes, sendo que uma delas é reservada para a utilização da entrada de sincronismo externo.

Funcionamento:



Para realizar a seleção da escala horizontal, é utilizado um contador 4017B (U21) auxiliado por um conjunto de buffers tri-state e um monoestável (U25). O estado do contador é alterado através do PEPSoftware. Toda a vez que a Interface for reiniciada, U21 é zerado e o estado do contador dependerá da escolha da escala horizontal. Os buffers tri-state estarão em alta impedância, exceto o que estiver selecionado por U21.

O multivibrador monoestável (U25:A) tem por função aumentar o tempo do pulso enviado pelo PC no pino SelHoriz, evitando que ruídos provoquem a comutação de U21.

Pulsos necessários para selecionar a escala horizontal:

Escala Horizontal	Pulsos em SelHoriz
1us/div (10MHz)	0
10 us/div (1MHz)	1
100 us/div (100KHz)	2
1000 us/div (10KHz)	3
10000 us/div (1KHz)	4
Entrada de Sincronismo Externo	5

ANALISADOR LÓGICO PARA PC

2.4. DESCRIÇÃO DO SOFTWARE

2.4.1. Funcionamento do PEPSoftware (ALOGIC e DISPLAY):

Sinais de controle da Interface	Bit da Porta Paralela	Estado
1. Detecção da Interface:		
WR/$\overline{RD}$	D7	0
CLKRD	D3	0
$\overline{CLRSYNC}$	D4	0
$\overline{PRESYNC}$	D1	1
SELHORIZ	D2	0
SELSYNC	D6	0
$\overline{RESET}$	D0	0
Byte Enviado: 22H / Bit Lido: D5 = 0 se Interface Presente		
2. Inicialização da Interface:		
WR/$\overline{RD}$	D7	0
CLKRD	D3	0
$\overline{CLRSYNC}$	D4	0
$\overline{PRESYNC}$	D1	1
SELHORIZ	D2	0
SELSYNC	D6	0
$\overline{RESET}$	D0	0
Byte Enviado: 22H		
3. Atualização da Escala / Canal Base de Sincronismo:		
WR/$\overline{RD}$	D7	0
CLKRD	D3	0
$\overline{CLRSYNC}$	D4	0
$\overline{PRESYNC}$	D1	1
SELHORIZ	D2	Pulsos de acordo com a escala a ser selecionada
SELSYNC	D6	Pulsos de acordo com o canal base de sincronismo a ser selecionado
$\overline{RESET}$	D0	0
Bytes Enviados: 22H / 26H para SELHORIZ e 22H / 62H para SELSYNC		
4. Inicio da Amostragem na Interface:		
WR/$\overline{RD}$	D7	1
CLKRD	D3	0
$\overline{CLRSYNC}$	D4	1
$\overline{PRESYNC}$	D1	1
SELHORIZ	D2	0
SELSYNC	D6	0
$\overline{RESET}$	D0	1
Byte Enviado: B3H		
5. Verificação do Fim da Amostragem:		
Bit Lido: D5 = 1 se Final da Amostragem		
6. Se D5=1, Repetição dos itens 1 e 2, salta para o item 9.		
Se D5=0 e tempo limite esgotado (600ms), salta para o item 7.		
7. Desligamento do Sincronismo:		
WR/$\overline{RD}$	D7	1
CLKRD	D3	0
$\overline{CLRSYNC}$	D4	1

$\overline{\text{PRESYNC}}$	D1	0
SELHORIZ	D2	0
SELSYNC	D6	0
$\overline{\text{RESET}}$	D0	1
Byte Enviado: B1H		
8. Repetição dos itens 1 e 2, salta para o item 9.		
9. Leitura da Interface:		
WR/$\overline{\text{RD}}$	D7	0
CLKRD	D3	Pulsos, um a cada dado lido
$\overline{\text{CLRSYNC}}$	D4	0
$\overline{\text{PRESYNC}}$	D1	1
SELHORIZ	D2	0
SELSYNC	D6	0
$\overline{\text{RESET}}$	D0	1
Byte Enviado: 23H / 2BH		
10. Byte Lido: D7 a D0 = Dado lido da memória da Interface.		
11. Processamento e exibição do dado na tela do microcomputador.		
12. Verificação da mudança de escala / canal base de sincronismo.		
13. Repetem-se todos os procedimentos enquanto o PEPSoftware estiver sendo executado.		

2.4.2. Estrutura dos arquivos auxiliares do PEPSoftware:

A. Arquivos de Imagem compactados - Extensão .PEP:

Descrição:

Os arquivos .PEP são arquivos de imagem compactados, utilizados para as telas de apresentação do PEPSoftware. Têm por origem arquivos BitMap RGB 16 Colors for Windows (.BMP), sendo realizada uma compactação e eliminação de signaturas e bytes de tamanho e palette, já que o .PEP sempre possui a resolução de 640 x 350 pixels x 16 cores. Para a eliminação dos bytes de palette, foi realizada uma correção de cores durante a preparação dos arquivos. O algoritmo utilizado na compactação da imagem baseia-se na contagem do número de pixels de cada cor que serão exibidos na tela, em sequências.

Estrutura dos bytes:

Bytes 0 a 2 = Quantidade de pixels da cor indicada no próximo Byte

Exemplo: 0 1 2
01H / 02H / FFH
0102FFH (66.303) Pixels

Byte 3 = Cor dos pixels

Exemplo: 3
0FH

0FH = White (Branco)
(Repete-se essa sequência até o final do arquivo)

B. Arquivos de Forma-de-Onda - Extensão .OSC:

Descrição:

Os arquivos .OSC são produzidos pelo PEPSoftware para armazenar formas-de-onda gravadas através do ALOGIC, DISPLAY e MAKEOSC. No caso do MAKEOSC, os dados gravados no arquivo .OSC não são obtidos através das pontas-de-prova do Analisador Lógico para PC. Eles são gerados pelo Software para imitar uma forma-de-onda medida.

Estrutura dos Bytes:

Bytes 0 a 2 = Signatura: 'PEP'
Bytes 3 a 12 = Descrição do Conjunto
Bytes 13 a 22 = Descrição do Canal A
Bytes 23 a 32 = Descrição do Canal B
Bytes 33 a 42 = Descrição do Canal C
Bytes 43 a 52 = Descrição do Canal D
Bytes 53 a 62 = Descrição do Canal E
Bytes 63 a 72 = Descrição do Canal F
Bytes 73 a 82 = Descrição do Canal G
Bytes 83 a 92 = Descrição do Canal H
Byte 93 = 00H se Canal A = OFF / 01H se Canal A = ON
Byte 94 = 00H se Canal B = OFF / 01H se Canal B = ON
Byte 95 = 00H se Canal C = OFF / 01H se Canal C = ON
Byte 96 = 00H se Canal D = OFF / 01H se Canal D = ON
Byte 97 = 00H se Canal E = OFF / 01H se Canal E = ON
Byte 98 = 00H se Canal F = OFF / 01H se Canal F = ON
Byte 99 = 00H se Canal G = OFF / 01H se Canal G = ON
Byte 100 = 00H se Canal H = OFF / 01H se Canal H = ON
Byte 101 = Escala Horizontal:
- *00H = 1uS/div*
- *01H = 10uS/div*
- *02H = 100uS/div*
- *03H = 1.000uS/div*
- *04H = 10.000uS/div*
- *05H = Externo*

Bytes 102 a 613 = Dados (0 a 511):
- *Canal A = Bit 0*
- *Canal B = Bit 1*
- *Canal C = Bit 2*
- *Canal D = Bit 3*
- *Canal E = Bit 4*
- *Canal F = Bit 5*
- *Canal G = Bit 6*
- *Canal H = Bit 7*

C. Arquivos de Imagem em BitMap - Extensão .BMP:

Descrição:

Os arquivos .BMP são produzidos pelo comando Exportar para BitMap do COMPARA.EXE. São arquivos de imagem padrão do Windows, sendo utilizada pelo PEPSoftware a resolução de 640 x 350 pixels x 16 cores.

Estrutura dos Bytes:

Bytes 0 a 17 = Signatura do BitMap
Bytes 18 e 19 = Largura em Pixels
Byte 22 e 23 = Altura em Pixels
Byte 54 a 56 = B,G,R da Cor 0 - Palette
Byte 58 a 60 = B,G,R da Cor 1 - Palette
Byte 62 a 64 = B,G,R da Cor 2 - Palette
Byte 66 a 68 = B,G,R da Cor 3 - Palette
Byte 70 a 72 = B,G,R da Cor 4 - Palette
Byte 74 a 76 = B,G,R da Cor 5 - Palette
Byte 78 a 80 = B,G,R da Cor 6 - Palette
Byte 82 a 84 = B,G,R da Cor 7 - Palette
Byte 86 a 88 = B,G,R da Cor 8 - Palette
Byte 90 a 92 = B,G,R da Cor 9 - Palette
Byte 94 a 96 = B,G,R da Cor 10 - Palette
Byte 98 a 100 = B,G,R da Cor 11 - Palette
Byte 102 a 104 = B,G,R da Cor 12 - Palette
Byte 106 a 108 = B,G,R da Cor 13 - Palette
Byte 110 a 112 = B,G,R da Cor 14 - Palette
Byte 114 a 116 = B,G,R da Cor 15 - Palette
Byte 118 = Inicio do BMP (Cada 1/2 Byte é a cor de um Pixel)

O BMP inicia no canto inferior esquerdo da imagem e termina no canto superior direito.

D. Arquivo de configuração do PEPSoftware - ALOGIC.CFG:

Descrição:

ALOGIC.CFG é um arquivo texto, que armazena o endereço base da porta paralela a ser utilizada pelo Analisador Lógico para PC, a habilitação da detecção da Interface no PEPSoftware e a habilitação do desligamento do sincronismo para tensões constantes. Tudo isso é determinado através do CONFIG.EXE.

Estrutura dos Registros:

Record 0 = Endereço Base da Porta Paralela (em decimal)
Record 1:
0 se a Detecção Automática estiver desabilitada
1 se a Detecção Automática estiver habilitada
Record 2:
0 se o Desligamento do Sincronismo para CC estiver desabilitado
1 se o Desligamento do Sincronismo para CC estiver habilitado

E. Arquivo de listagem de endereços - PEPLEG.DAT:

Descrição:

O arquivo PEPLEG.DAT é um arquivo texto que armazena todos os endereços que são pesquisados na detecção automática da Interface, realizada pelos programas DIAG e CONFIG.

Estrutura das linhas:

Linhas = Endereços utilizados pela detecção da Interface (em Hexadecimal)

Exemplo:
378
3BC
278

F. Arquivos binários de dados - Extensão .BIN:

Descrição:

Os arquivos .BIN são produzidos pelo DISPLAY.EXE para armazenar os dados lidos a partir dos oito canais do Analisador Lógico para PC, em formato binário.

Estrutura dos bytes:

Bytes 0 a 511 = ASCII dos Dados

G. Arquivos texto de dados - Extensão .HEX:

Descrição:

Os arquivos .HEX são produzidos pelo DISPLAY.EXE para armazenar os dados lidos a partir dos oito canais do Analisador Lógico para PC, em formato texto e em hexadecimal.

Estrutura das Colunas:

Coluna 1 = Posição do Dado (em Hexadecimal)
Coluna 2 = Valor do Dado (em Hexadecimal)

H. Arquivo de Setup da instalação - INSTALL.SET:

Descrição:

O arquivo INSTALL.SET é um arquivo texto que determina ao programa de instalação do PEPSoftware (INSTALL.EXE) os arquivos que deverão ser copiados do disco de instalação para o disco rígido.

Estrutura das linhas:

Linhas = Nomes dos arquivos de destino da instalação

Exemplo:
ALOGIC.EXE
ALOGIC.CFG
CONFIG.EXE
...

I. Arquivos temporários do MAKEOSC - Extensão .TMP:

Descrição:

Os arquivos .TMP são produzidos pelo MAKEOSC durante a execução do programa para armazenar temporariamente as características das formas-de-onda criadas. Quando essas informações são gravadas para o .OSC, os arquivos temporários são automaticamente apagados.

Estrutura dos bytes:

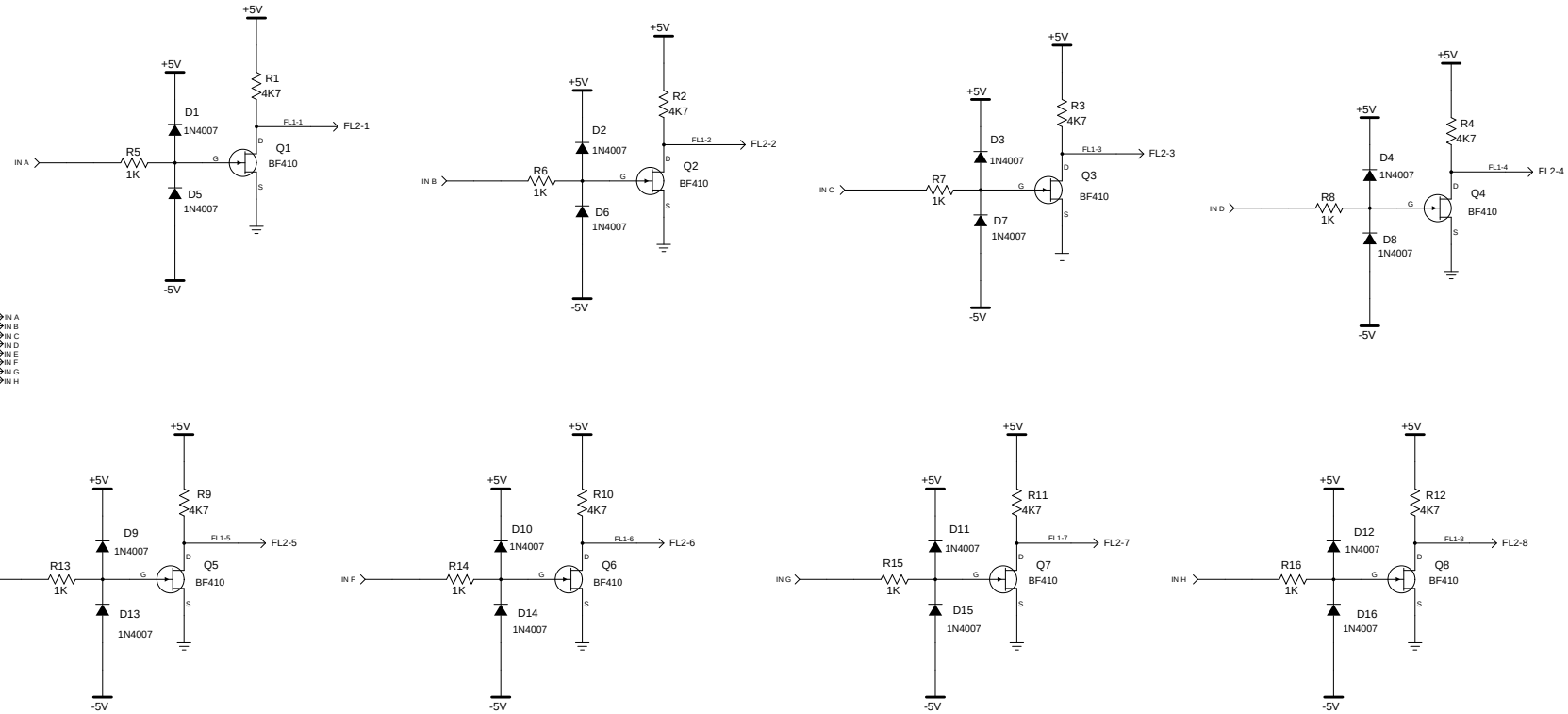
Byte 0: Descrição do Conjunto
Byte 10: Descrição do Canal
Byte 20: Estado do Canal
00H = OFF
01H = ON
Byte 21 a Byte 532: Dados (Bits)
00H = Forma-de-onda em nível baixo
01H = Forma-de-onda em nível alto

ANALISADOR LÓGICO PARA PC

APÊNDICE A

DIAGRAMA ESQUEMÁTICO

ANALISADOR LOGICO PARA PC



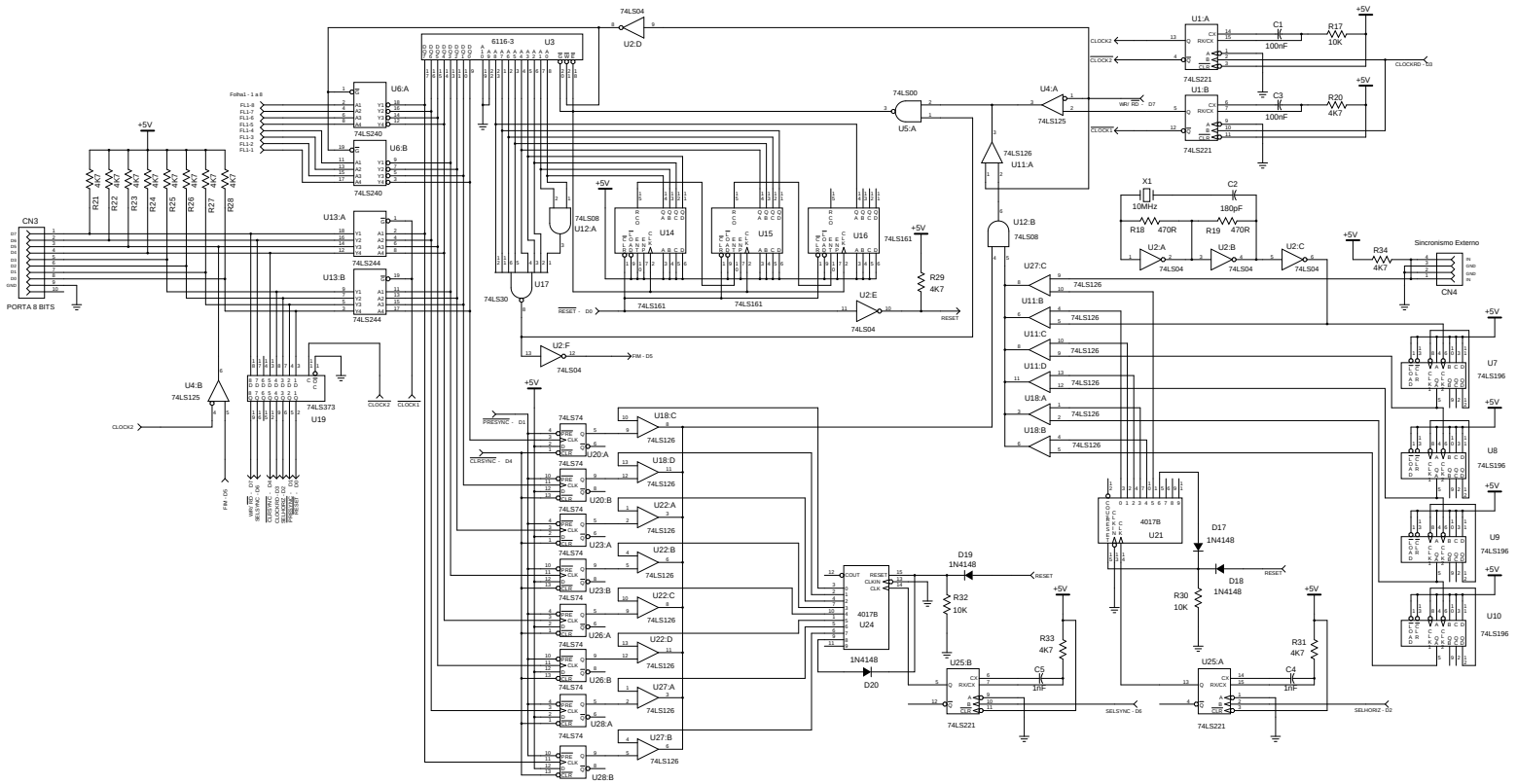
Todos os Resistores sao de 1/4W

As Entradas dos Canais sao conectores coaxiais

As malhas dos conectores sao o -5V

Title	Analisador Logico para PC		PCB-01
Size A2	Number	Luiz Eduardo Marcos Paulo Rafael Abolafio Robson S. Martins "Os PEP Legais"	Rev
Date	19,23,26,30 4oR-Elo	Drawn by	30/08/97
Filename	ALOGIC.S01	Sheet 1 of 2	

ANALISADOR LOGICO PARA PC



Alimentacao dos CI's:

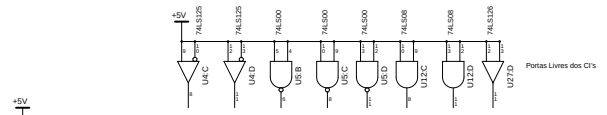
U1, U14, U15, U16, U21, U24, U25: +5V = 16 / GND = 8

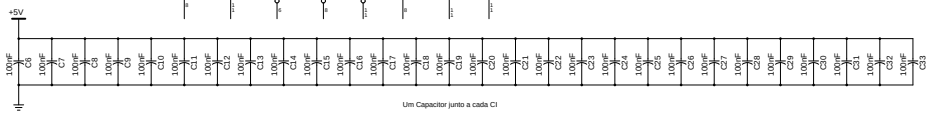
U2, U4, U5, U7, U8, U9, U10, U11, U12, U17,
U18, U20, U22, U23, U26, U27, U28: +5V = 14 / GND = 7

U3: +5V = 24 / GND = 12

U6, U13, U19: +5V = 20 / GND = 10

Todos os resistores sao de 1/4W

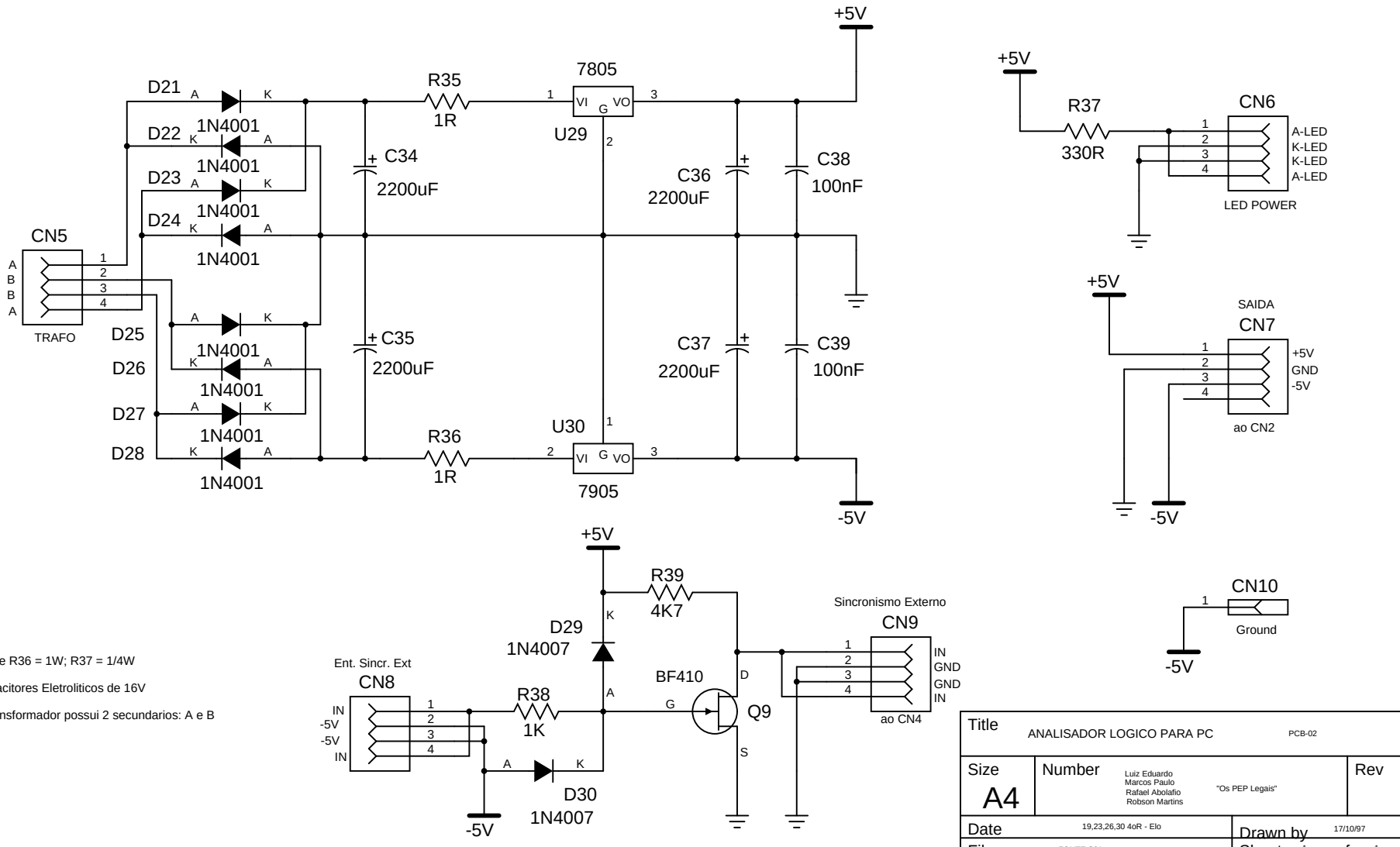




Um Capacitor junto a cada CI

Title	ANALISADOR LOGICO P/ PC		PCB-01
Size A1	Number	Luiz Eduardo Marcos Paulo Rafael Abolafio Robson Martins "Os PEP Legais"	Rev
Date	19, 23, 26, 30 Abril - Ele		Drawn by 30/08/97
Filename	ALOGIC.SCI		Sheet 2 of 2

ANALISADOR LOGICO PARA PC



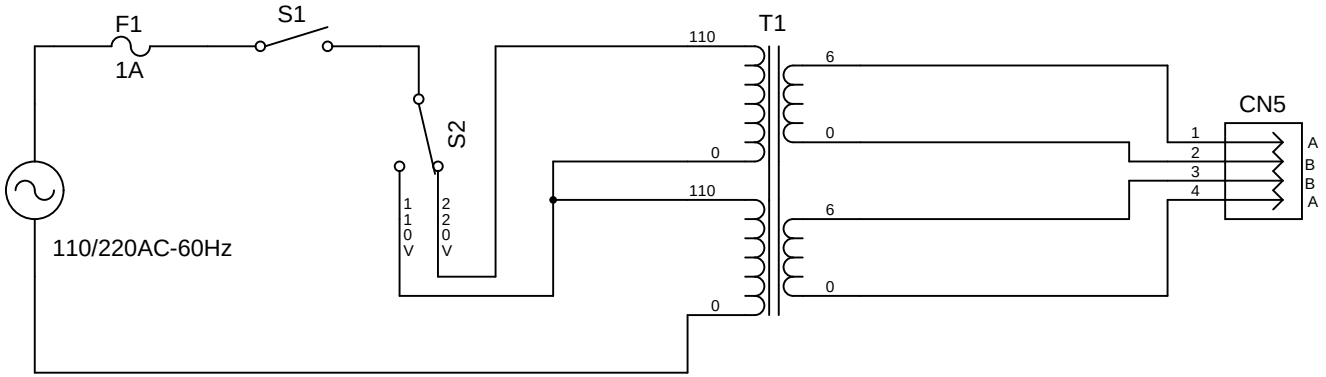
R35 e R36 = 1W; R37 = 1/4W

Capacitores Eletroliticos de 16V

O transformador possui 2 secundarios: A e B

Title	ANALISADOR LOGICO PARA PC	PCB-02	
Size A4	Number	Luiz Eduardo Marcos Paulo Rafael Abolafio Robson Martins "Os PEP Legais"	Rev
Date	19,23,26,30 4oR - Elo	Drawn by	17/10/97
Filename	FONTE.S01	Sheet 1 of 1	

ANALISADOR LOGICO PARA PC



Title	ANALISADOR LOGICO PARA PC		Fonte
Size A4	Number	Luiz Eduardo Marcos Paulo Rafael Abolafio Robson Martins	Rev
Date	19,23,26,30 - 4o R Elo	Drawn by	30/10/97
Filename	TRAFO.S01	Sheet 1 of 1	

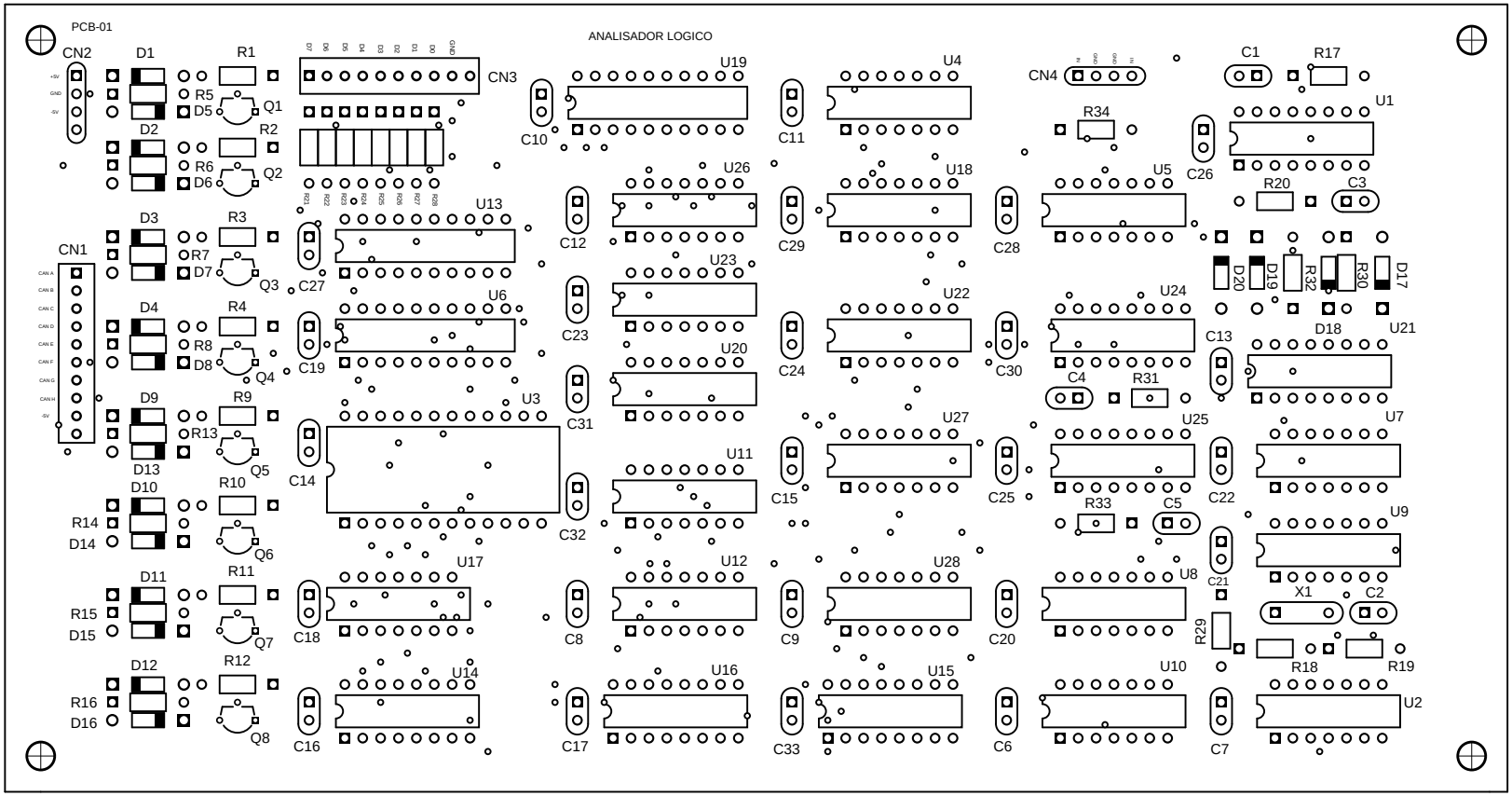
ANALISADOR LÓGICO PARA PC

APÊNDICE B

LAY-OUT DAS PLACAS DE CIRCUITO IMPRESSO

ANALISADOR LOGICO PARA PC

OS PEP LEGAIS - 4o R ELETRONICA - LAO

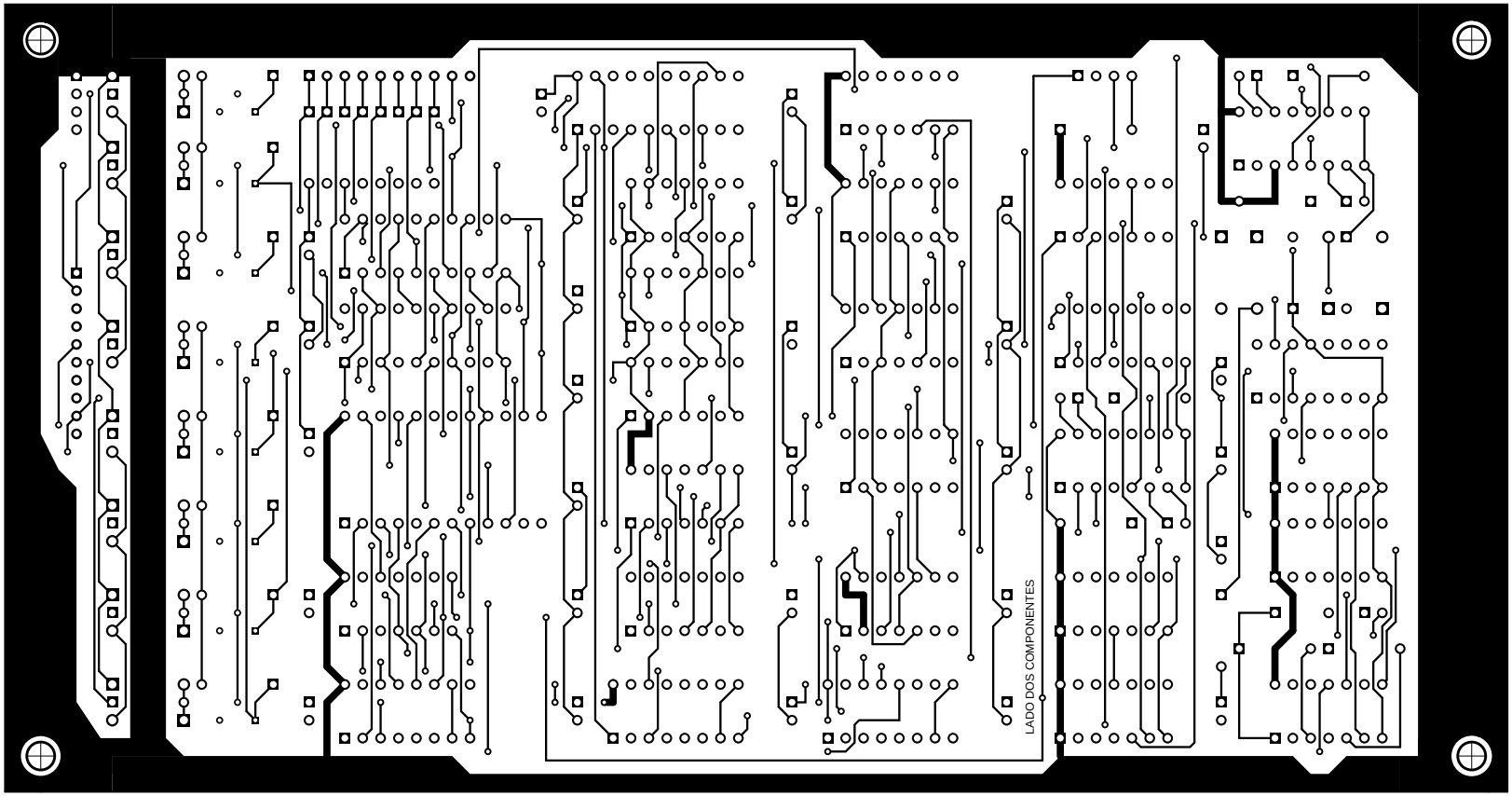


PCB-01

Nota: Q1 a Q8 sao soldados invertidos se forem BF410

ANALISADOR LOGICO PARA PC

OS PEP LEGAIS - 4o R ELETRONICA - LAO

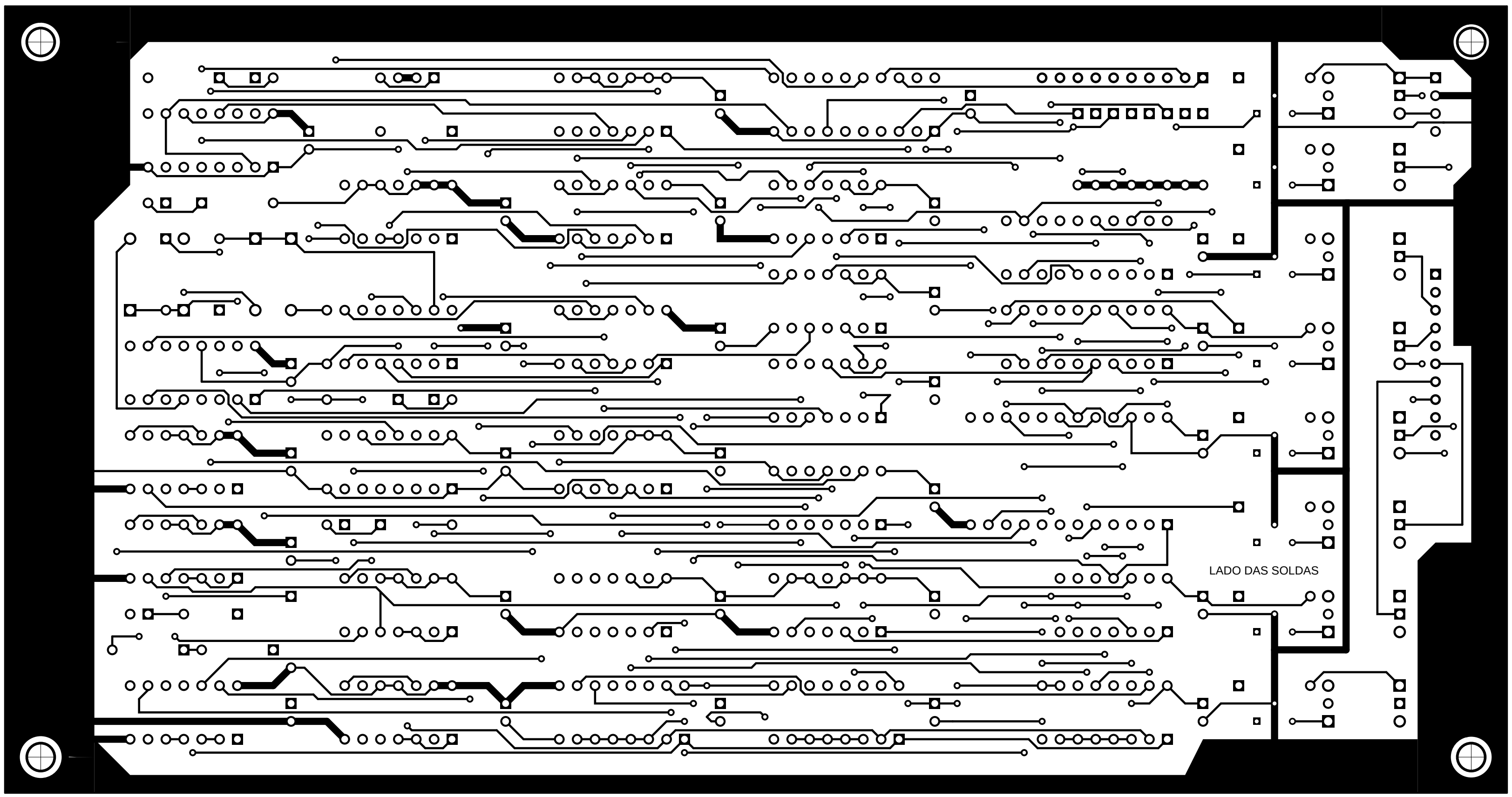


PCB-01

Nota: Q1 a Q8 sao soldados invertidos se forem BF410

ANALISADOR LOGICO PARA PC

OS PEP LEGAIS - 4o R ELETRONICA - LAO

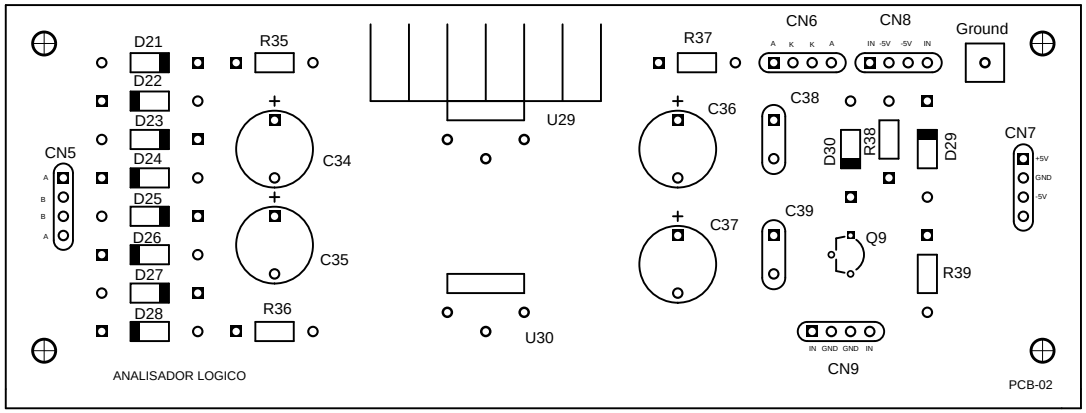


PCB-01

Nota: Q1 a Q8 sao soldados invertidos se forem BF410

ANALISADOR LOGICO PARA PC

OS PEP LEGAIS - 4o R ELETRONICA - LAO

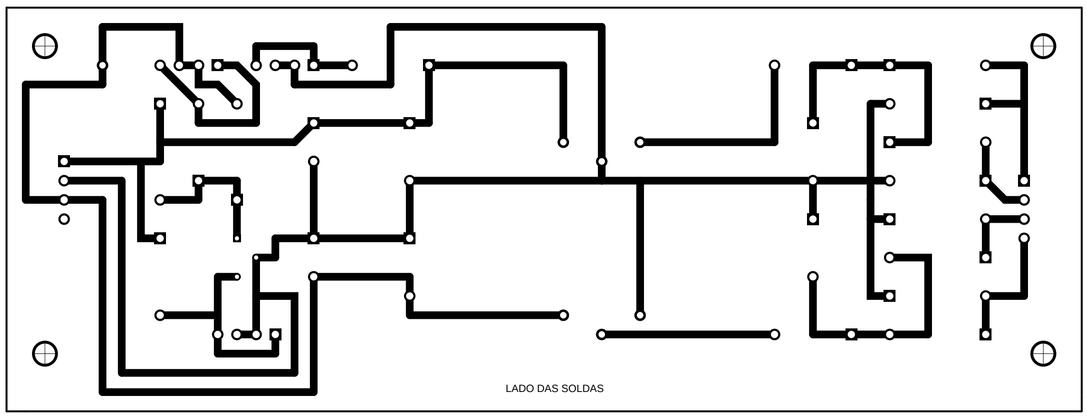


PCB-02

Nota: Q9 e soldado invertido se for BF410

ANALISADOR LOGICO PARA PC

OS PEP LEGAIS - 4o R ELETRONICA - LAO



PCB-02

Nota: Q9 e soldado invertido se for BF410

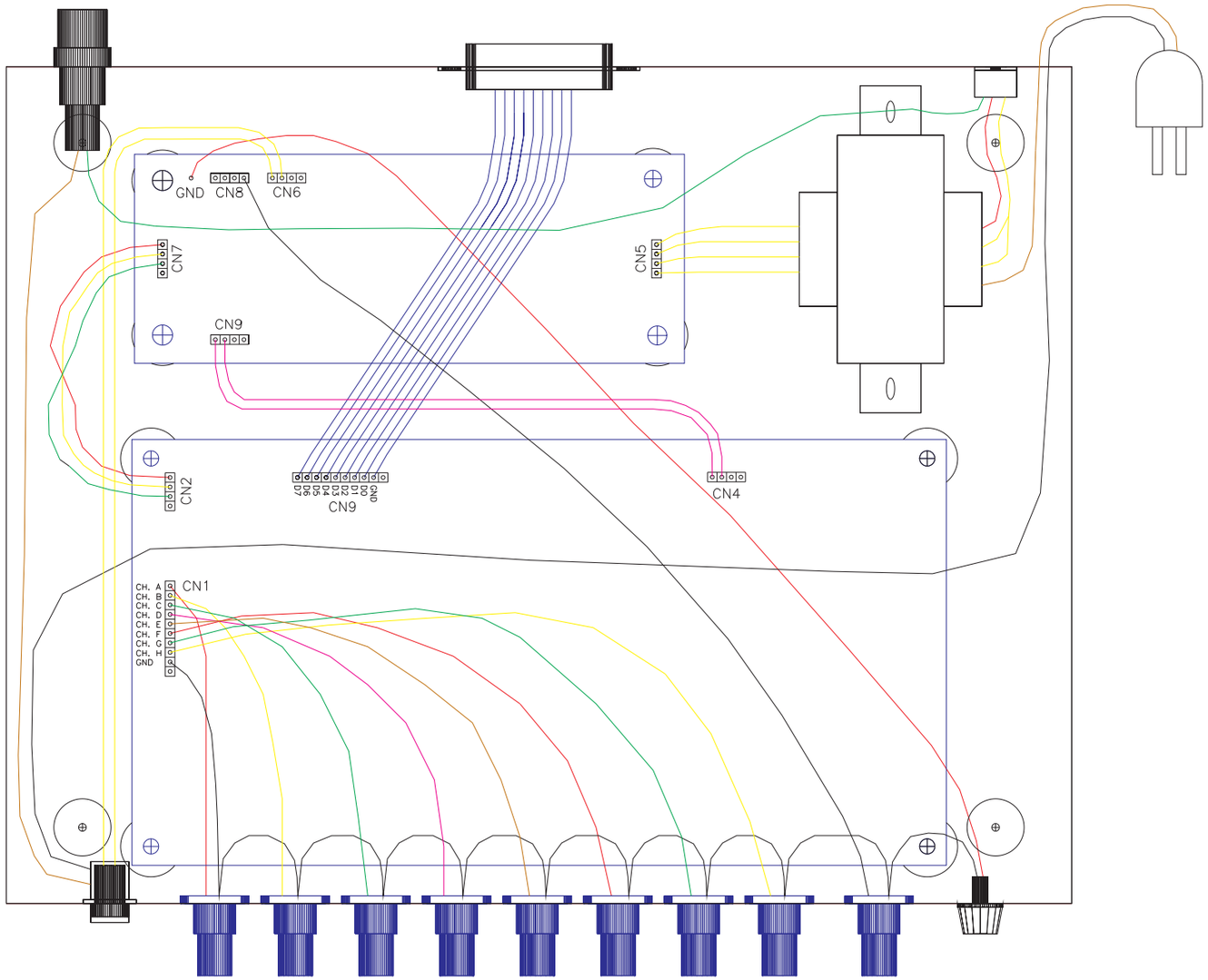
ANALISADOR LÓGICO PARA PC

APÊNDICE C

DESENHO DAS CONEXÕES INTERNAS

Analisador Lógico para PC

Conexões Internas



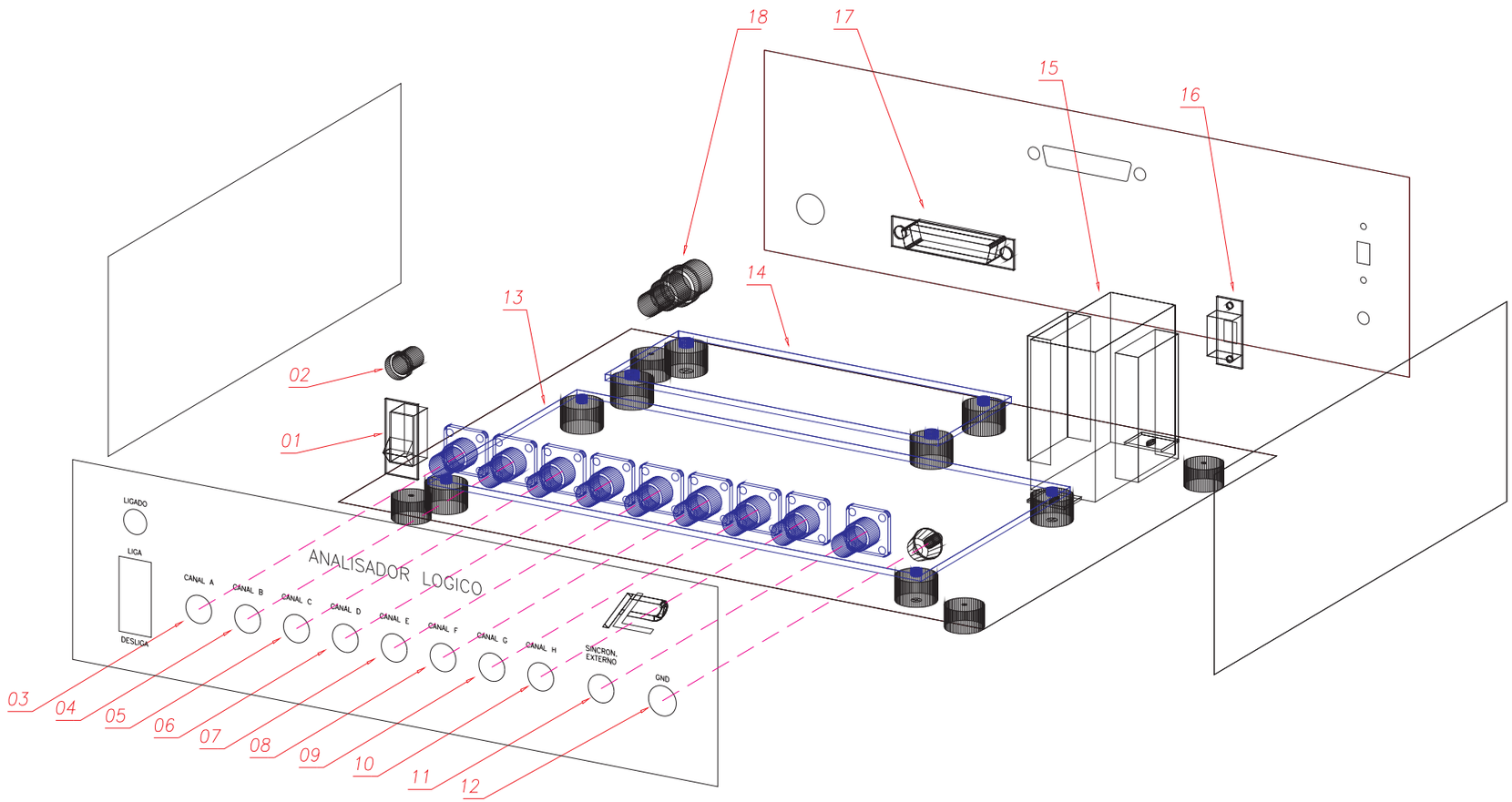
Os PEP Legais

ANALISADOR LÓGICO PARA PC

APÊNDICE D

VISTA EXPLODIDA E PROJETO DA CAIXA

Vista Explodida - Analisador Lógico para PC

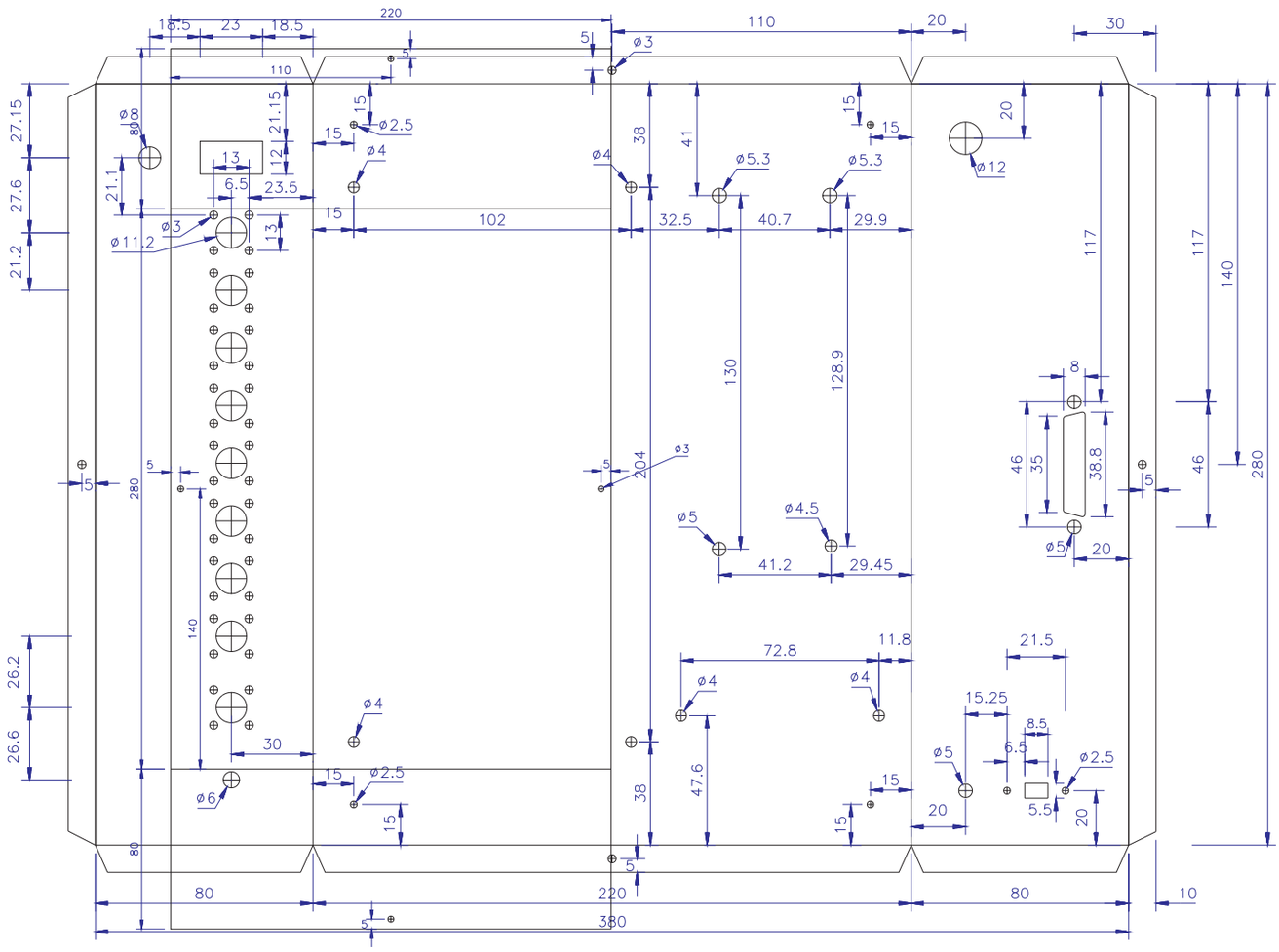


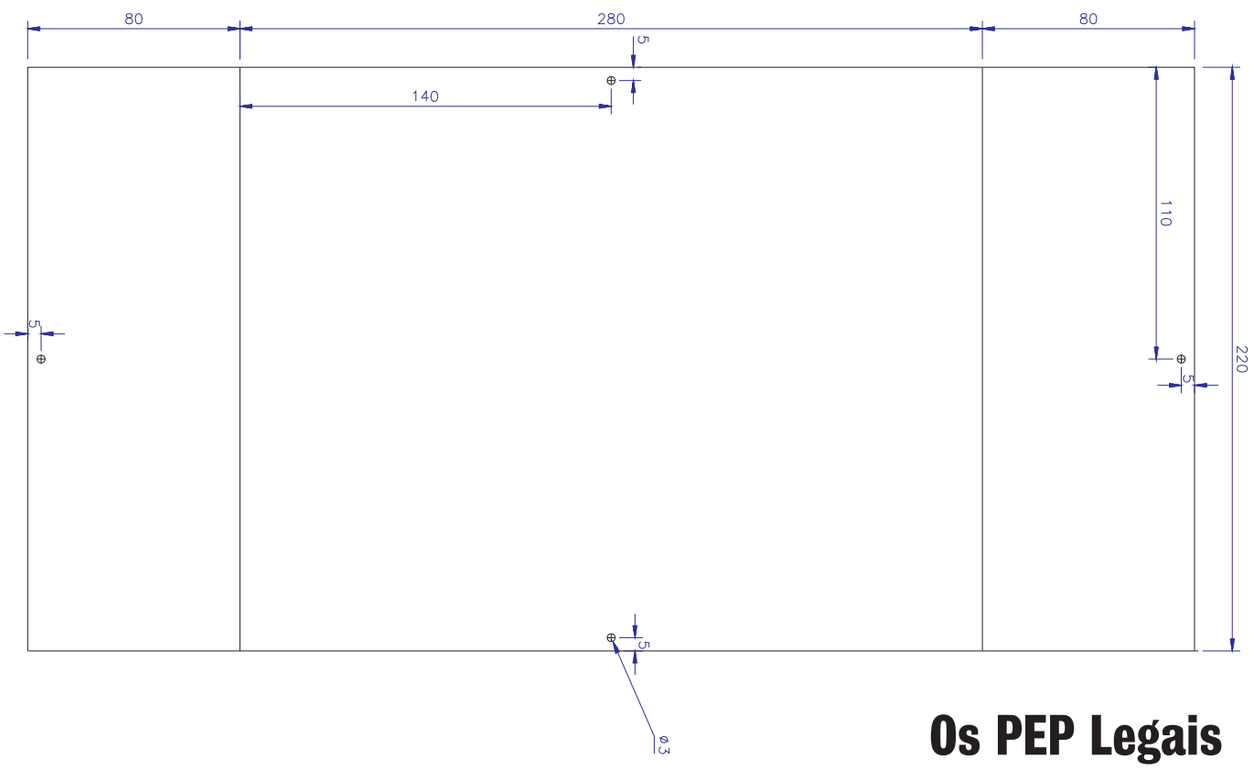
ITEM	DESCRICAO
01	CHAVE LIGA/DESLIGA
02	LED POWER
03	BNC CANAL A
04	BNC CANAL B
05	BNC CANAL C
06	BNC CANAL D
07	BNC CANAL E
08	BNC CANAL F
09	BNC CANAL G
10	BNC CANAL H
11	BNC SINCRONISMO EXTERNO
12	CONECTOR DO GND
13	PCB-01
14	PCB-02
15	TRANSFORMADOR
16	CHAVE SELETORA 110/220V
17	CONECTOR DB-25
18	PORTA FUSIVEL

Os PEP Legais

Analisador Lógico para PC

Projeto da Caixa





Os PEP Legais

ANALISADOR LÓGICO PARA PC

APÊNDICE E

LISTAGEM DO PEPSOFTWARE

Analisador Lógico para PC

Os PEP Legais

Listagem do PEPSoftware

PEPUnit (PEPUNIT.PAS)

```
{$A-,B-,D-,E+,F-,G-,I+,L-,N+,Q-,R-,S+,X+,V+}
{Diretivas de compilação}

{************************************************************************}

Unit PEPUnit;

{************************************************************************}
{
        Analisador Lógico - "Os PEP Legais"

        Créditos:
                  Luiz Eduardo    - N°19
                  Marcos Paulo    - N°23
                  Rafael Abolafio - N°26
                  Robson Martins  - N°30

                  4°R - Eletrônica - 1997

               Liceu de Artes e Ofícios de São Paulo

               Prof: Marco Antônio Togniazzolo
}
{************************************************************************}

{Unit com as Funções e Rotinas mais utilizadas pelo PEPSoftware}

{************************************************************************}

Interface
{Declaração das Rotinas e Funções desta Unit}

{Rotinas para Interface com Usuário - Modo Texto}

Procedure CenterTxt (YCoord:Integer;TextoStr:String);
Procedure RightTxt (YCoord:Integer;TextoStr:String);
Procedure ApagaWin;
Procedure CenterWin (XSize:Integer;YSize:Integer;TxtCol:Integer;BkCol:Integer);
Procedure AmpliaWin (XSizemax:Integer;YSizemax:Integer;FC:Integer;BC:Integer);
Procedure Tela (Titulo:String);

{Rotinas e Funções para Modo Gráfico (640 X 350 X 16)}

Procedure ModoGrafico;
Function ViewPEP (NameFil:String;PagAt:Byte;CorExclu:Byte;HabilTecla:Boolean):
                 Char;

{Funções Matemáticas}

Function Eleva (Base:Integer;Expoente:Integer):Integer;
Function DecToHex(W:Word;DefW:Boolean):String;
Function HexToDec(HS:String):Word;

{************************************************************************}

Implementation
{Implementação das Funções e Rotinas}

Uses Crt, Dos, Graph;
{Units a serem utilizadas}

{************************************************************************}

Procedure CenterTxt (YCoord:Integer;TextoStr:String);
{Centraliza uma String em uma linha determinada
  YCoord = Linha
  TextoStr = Texto a ser centralizado}
```

```
Begin
     Window (1,1,80,25);
     Gotoxy(Round((80-Length(TextoStr))/2),YCoord);
     Write (TextoStr);
End;

{*************************************************************************}

Procedure RightTxt (YCoord:Integer;TextoStr:String);
{Alinha a direita uma String em uma linha determinada
  YCoord = Linha
  TextoStr = Texto a ser centralizado}

Begin
     Window (1,1,80,25);
     Gotoxy(80-Length(TextoStr),YCoord);
     Write (TextoStr);
End;

{*************************************************************************}

Procedure ApagaWin;
{Apaga a janela}

Var XPos,YPos:Integer; {Posição atual do cursor (X e Y)}

Begin
     Window (1,2,80,25);
     TextColor(LightGray);
     TextBackGround(Black);
     For YPos:=1 To 22 Do Begin
         For XPos:=1 To 80 Do Begin
             Gotoxy (XPos,YPos);
             Write (#176);
         End;
     End;
End;

{*************************************************************************}

Procedure CenterWin (XSize:Integer;YSize:Integer;TxtCol:Integer;BkCol:Integer);
{Produz uma Janela no centro da tela
  XSize,YSize = Dimensões da janela
  TxtCol = Cor do texto
  BkCol = Cor do fundo}

Var X1,X2,Y1,Y2:Integer; {Coordenadas da janela}
    XP,YP:Integer; {Utilizadas pelo contador para movimentar o cursor}

Begin
     X1:=Round((80-XSize)/2);
     X2:=Round((80-XSize)/2+XSize);
     Y1:=Round((23-YSize)/2);
     Y2:=Round((23-YSize)/2+YSize);

     Window (1,1,80,25);
     Window (X1,Y1,X2,Y2);
     TextColor (TxtCol);
     TextBackground (BkCol);
     Clrscr;

     Window (1,1,80,25);
     GotoXy (X1+1,Y1); Write (#201);
     Gotoxy (X2-1,Y1); Write (#187);
     Gotoxy (X1+1,Y2); Write (#200);
     Gotoxy (X2-1,Y2); Write (#188);

     For XP:=X1+2 To X2-2 Do Begin
         Gotoxy (XP,Y1); Write (#205);
         Gotoxy (XP,Y2); Write (#205);
     End;

     For YP:=Y1+1 To Y2-1 Do Begin
         Gotoxy (X1+1,YP); Write (#186);
         Gotoxy (X2-1,YP); Write (#186);
     End;

     TextColor (Black);
     TextBackGround (Black);
     For XP:=X1+2 To X2+1 Do Begin
         Gotoxy (XP,Y2+1); Write (#223);
     End;
```

```
     For YP:=Y1+1 To Y2 Do Begin
         Gotoxy (X2+1,YP); Write (#219);
         Gotoxy (X2+2,YP); Write (#219);
     End;

     Gotoxy (X2+2,Y2+1); Write (#220);
End;

{*************************************************************************}

Procedure AmpliaWin (XSizemax:Integer;YSizemax:Integer;FC:Integer;BC:Integer);
{Produz o efeito de ampliação da janela
  XSizemax,YSizemax = Tamanho máximo da janela
  FC = Cor do texto
  BC = Cor do fundo}

Var XSz,YSz:Integer; {Utilizadas para variar o tamanho da jenela}
    SzMax:Real; {Relação de aspecto máxima}

Begin
     SzMax:=YSizemax/XSizemax;
     For XSz:=3 To XSizeMax Do Begin
         YSz:=Round(SzMax*XSz);
         CenterWin (XSz,YSz,FC,BC);
         Delay (1);
     End;
End;

{*************************************************************************}

Procedure Tela (Titulo:String);
{Tela Principal - Modo Texto
  Titulo: Texto a ser exibido na barra de título}

Const Dias : Array [0..6] Of String[9] =
  ('Domingo','Segunda','Terça',
   'Quarta','Quinta','Sexta',
   'Sábado');                      {Dias da semana}

Var Ano,Mes,Dia,Diasem:Word;    {Variáveis que armazenam a data atual}
    AnoStr,MesStr,DiaStr:String; {Aramazenam ano, mês e dia como strings}

Begin
     {Fundo}
     ApagaWin;

     {Título}
     Window (1,1,80,1);
     TextColor(White);
     TextBackGround (Blue);
     Clrscr;
     CenterTxt (1,Titulo);

     {Linha de Status}
     Window (1,24,80,24);
     Textcolor (Black);
     TextBackGround (White);
     Clrscr;

     {Data Atual}
     TextColor (Red);
     GetDate (Ano,Mes,Dia,DiaSem);
     Window (1,1,80,25);
     Str (Ano:4,AnoStr);
     Str (Mes:2,MesStr);
     Str (Dia:2,DiaStr);
     RightTxt (24,Dias[DiaSem]+', '+DiaStr+'/'+MesStr+'/'+AnoStr);

     {Janela Central}
     Delay (500);
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
End;

{*************************************************************************}

Procedure ModoGrafico;
{Inicia o modo Gráfico}

Var Gd,Gm:Integer;  {GraphicDriver e GraphicMode: usadas pelo InitGraph}

Procedure MensErro;
```

```
{Mensagem de Erro: Não foi possível iniciar Modo Gráfico}


Begin
        ClrScr;
        TextColor (Blue);
        Writeln ('Analisador Lógico - Mensagem de Erro');
        Writeln;
        TextColor (LightRed);
        Writeln ('Houve um erro na inicialização do modo Gráfico!');
        Writeln ('Verifique se os drivers gráficos estão presentes');
        Writeln ('no mesmo diretório do PEPSoftware!');
        TextColor (LightGray);
        Writeln;
        Writeln ('Caso não consiga solucionar este problema,');
        Writeln ('entre em contato com:');
        Writeln;
        Textcolor (White);
        Writeln ('"Os PEP Legais" - LAO - 4oR Eletrônica');
        Textcolor (LightGray);
        Writeln;
        Halt(2);
End;

Begin
     Gd:=Detect;
     InitGraph(Gd, Gm, '');
     If GraphResult <> GrOk Then MensErro;
     SetGraphMode (1); {640 X 350}
End;

{*************************************************************************}

Function ViewPEP(NameFil:String;PagAt:Byte;CorExclu:Byte;HabilTecla:Boolean):
                 Char;
{Visualiza um .PEP especificado (NameFil) na Pág. de Vídeo especificada
(PageAt), excluindo uma cor determinada (CorExclu)
HabilTecla = Habilita verificação de pressionamento de tecla durante a carga
             do .PEP
Retorna o valor da Tecla pressionada (Char)}

Var FPep:File; {Representa o arquivo .PEP a ser lido}
    Chs:Char; {Armazena o caracter lido do arquivo}
    ContToFor,ForMax:Longint; {Utilizadas pelo contador que movimenta o cursor}
    M1:Longint; { Armazenam                        }
    M2:Word;    { o número de pixels para cada cor }
    M3:Byte;    { lido do arquivo .PEP             }
    Xz,Yz:Word; {Armazenam a posição do cursor na tela}
    TecPep:Char; {Armazena o caracter da tecla lida durante a carga do .PEP}
    Buf:Array [1..40] Of Char; {Buffer de leitura do arquivo}
    IndBuf:Word; {Index do Buffer}
    NumRead:Word; {Retorna o número de bytes lidos do arquivo}

Procedure Pixel (XValC:Word;YValC:Word;CorPix:Byte;VPageAt:Byte);
{Produz um Pixel na tela
  XValC,YValC = coordenadas
  CorPix = cor do pixel
  VPageAt = página de vídeo}

Begin
     Asm
        Mov AH, 0CH
        Mov Al,(CorPix)
        Mov CX,(XValC)
        Mov DX,(YValC)
        Mov BH,(VPageAt)
        Int 10H
     End;
End;

Begin
     Assign (FPep,NameFil);
     Reset (FPep,1);

     Xz:=0; Yz:=349;  {640X350}
     TecPep:=#0;
     SetActivePage(PagAt);
     SetFillStyle (1,CorExclu);
     Bar (0,0,639,349);

Repeat
     BlockRead(FPep, Buf, SizeOf(Buf), NumRead);
     IndBuf:=1;
Repeat
```

```
     Chs:=Buf[IndBuf];
     M1:=Ord(Chs)*$10000;
     Chs:=Buf[IndBuf+1];
     M2:=Ord(Chs)*$100;
     Chs:=Buf[IndBuf+2];
     M3:=Ord(Chs);
     ForMax:=M1 Or M2 Or M3;
     Chs:=Buf[IndBuf+3];
     For ContToFor:=1 To ForMax Do Begin
         If Ord(Chs) <> CorExclu Then
            Pixel (Xz,Yz,Ord(Chs),PagAt);
         If Keypressed Then Begin
            TecPep:=Readkey;
            If ((TecPep=#27) Or (TecPep=#13)) And (HabilTecla) Then Begin
               ViewPEP:=#27;
               Exit;
            End;
         End;
         If Xz<639 Then Xz:=Xz+1
         Else Begin
              Xz:=0;
              Yz:=Yz-1;
         End;
     End;
     IndBuf:=IndBuf+4;
Until (IndBuf=41) Or (Eof(FPep));
Until Eof(FPep);
      ViewPEP:=TecPep;
      Close (FPep);
End;

{*************************************************************************}

Function Eleva (Base:Integer;Expoente:Integer):Integer;
{Eleva uma Base a um Expoente
Retorna o valor de Base elevado a Expoente}

Var Vezes:Integer;  {Número de vezes que a base deve ser multiplicada}
    ValorExp:Integer; {Valor da base elevada ao expoente}

Begin
     ValorExp:=1;
     If Expoente=0 Then Eleva:=1
     Else Begin
          For Vezes:=1 To Expoente Do ValorExp:=ValorExp*Base;
          Eleva:=ValorExp;
     End;
End;

{*************************************************************************}

Function DecToHex(W:Word;DefW:Boolean):String;
{Entrada: Word/Byte = Decimal de 0 a 65535}
{Saída: String = Hexadecimal de 0000 a FFFF ou 00 a FF
  DefW = Define se a saida vai a FFFF ou FF
         0 = 00H a FFH
         1 = 0000H a FFFFH }

Const
     HexChars:Array[0..15] Of Char = '0123456789ABCDEF'; {Caracteres Hexa}

Begin
     If DefW Then DecToHex:=HexChars[(Hi(W)) Shr 4]+HexChars[(Hi(W)) And $F]+
                 HexChars[(Lo(W)) Shr 4]+HexChars[(Lo(W)) And $F]
     Else DecToHex:=HexChars[(Lo(W)) Shr 4]+HexChars[(Lo(W)) And $F];
End;

{*************************************************************************}

Function HexToDec(HS:String):Word;
{Entrada: Hexadecimal = String de 0000 a FFFF}
{Saída: Decimal = Word de 0 a 65535}

Var WordHex:Word; {Valor que varia até que seu hexadecimal seja igual a HS}

Begin
     For WordHex:=0 To $FFFF Do Begin
         If DecToHex(WordHex,True)=HS Then Break;
     End;
     HexToDec:=WordHex;
End;

{*************************************************************************}
```

```
End.
```

Analisador Lógico (ALOGIC.PAS)

```
{$A-,B-,D-,E+,F-,G-,I+,L-,N+,Q-,R-,S+,X+,V+}
{Diretivas de compilação}

{*************************************************************************}

Program Analisador_Logico;

{*************************************************************************}
{
        Analisador Lógico - "Os PEP Legais"

        Créditos:
                  Luiz Eduardo    - N°19
                  Marcos Paulo    - N°23
                  Rafael Abolafio - N°26
                  Robson Martins  - N°30

                  4°R - Eletrônica - 1997

              Liceu de Artes e Ofícios de São Paulo

              Prof: Marco Antônio Togniazzolo
}
{*************************************************************************}
{
        Software Principal - Visualizador de formas de onda;
                             Frequencímetro;
                             Gravador de arquivos .OSC
        Versão 1.0 - 1997

        Software for MS-DOS, Escrito no compilador Turbo Pascal 7.0

        Requisitos Mínimos do Sistema:

                   IBM PC XT/AT ou Compatível (Recomenda-se 486)
                   640KB de RAM ou mais (Recomenda-se 4MB)
                   Monitor EGA ou superior (Recomenda-se VGA Color)
                   Disk Drive de 1,44MB
                   Hard Disk com pelo menos 1MB livres
}
{*************************************************************************}

Uses Crt,Graph,Dos,PEPUnit;
{Units a serem utilizadas}

{*************************************************************************}

Var
{Declaração das variáveis}

   X,Y:Integer; {Deteriminam as coord. X e Y}
   A,B,C,D,E,F,G,H:Boolean; {Status dos canais (On/Off)}
   Escala:Real; {Armazena o valor da escala horizontal}
   CanalSync:Char; {Armazena qual o canal base de Sync}
   StatusPort:Byte; {Armazena o valor atual da LPT de Controle}
   Dado:Array [0..511] Of Byte; {Armazena os dados da leitura}
   IndDado:Integer; {Variável para indicar a posição do Array Dado}
   Hr,M,S,S100:Word; {Variáveis para contar Tempo}
   Timea:Real; {Variável para contar Tempo}
   Ap:Integer; {Indica a página de vídeo ativa}
   Pausa:Boolean; {Indica estado de pausa}
   Zoom:Boolean; {Indica estado de zoom}
   Ajuda:Boolean; {Indica chamada de Ajuda}
   NameArq:String; {Recebe o nome do arquivo .OSC que será gravado}
   Descr:Array [1..10,0..8] Of Char; {Armazena as descrições dos canais e do
                                      conjunto de formas de onda}
   Arq:File of Char; {Representa o arquivo .OSC}
   Fator,FimPonto:Integer; {Determinam o fator de zoom}
   Cfg:Text; {Arquivo de configuração}
   P1:Word; {LPT de Dados}
   P2:Word; {LPT de Controle}
   HabilVerif:Boolean; {Habilita a verificação da presença da Interface}
   HabilSyncOff:Boolean; {Habilita o desligamento do Sync para C.C.}

{*************************************************************************}
```

```
Const HabilAtuEscSync=True;
{Habilita atualização da escala/sincronismo - p/ reset de escala/canal base
de Sync}

{*****************************************************************************}

Procedure AbreCfg;
{Abre o arquivo Oscilos.Cfg}

Var HV,HS:Byte; {Armazenam os bytes de configuração lidos de Oscilos.Cfg}

Procedure MensErro;
{Indica Erro de falta de arquivo: ALOGIC.CFG}

Begin
        ClrScr;
        TextColor (Blue);
        Writeln ('Analisador Lógico - Mensagem de Erro');
        Writeln;
        TextColor (LightRed);
        Writeln ('Não existe o arquivo "ALOGIC.CFG"');
        Writeln ('neste diretório!');
        Writeln;
        Writeln ('Verifique qual o diretório corrente!');
        Writeln ('Após isso, execute novamente este programa.');
        TextColor (LightGray);
        Writeln;
        Writeln ('Caso não consiga solucionar este problema,');
        Writeln ('entre em contato com:');
        Writeln;
        Textcolor (White);
        Writeln ('"Os PEP Legais" - LAO - 4oR Eletrônica');
        Textcolor (LightGray);
        Writeln;
        Halt(2);
End;

Begin
     Assign (Cfg,'Alogic.Cfg');
{$I-}
     Reset (Cfg);
{$I+}
     If Ioresult<>0 Then MensErro;
     Readln (Cfg,P1,HV,HS);
     P2:=P1;

     If HV=1 Then HabilVerif:=True
     Else HabilVerif:=False;

     If HS=1 Then HabilSyncOff:=True
     Else HabilSyncOff:=False;

     Close (Cfg);
End;

{*****************************************************************************}

Procedure VerifArq;
{Verifica a presença dos arquivos necessários para a execução do programa}

Function ArqPres(NomeArqP:String):Boolean;
{Verifica se o arquivo NomeArqP está presente}

Var FPres:File Of Char; {Representa o arquivo a ser testado}

Begin
     Assign (FPres,NomeArqP);
{$I-}
     Reset (FPres);
{$I+}
     If IOResult <> 0 Then ArqPres:=False
     Else Begin
          ArqPres:=True;
          Close (FPres);
     End;
End;

Procedure MensErroArq(NameArqF:String);
{Mensagem de Erro: Falta o arquivo NameArqF}

Begin
     ClrScr;
     TextColor (Blue);
     Writeln ('Analisador Lógico - Mensagem de Erro');
```

```
     Writeln;
     TextColor (LightRed);
     Writeln ('Não existe o arquivo "',NameArqF,'"');
     Writeln ('neste diretório!');
     Writeln;
     Writeln ('Verifique qual o diretório corrente!');
     Writeln ('Após isso, execute novamente este programa.');
     TextColor (LightGray);
     Writeln;
     Writeln ('Caso não consiga solucionar este problema,');
     Writeln ('entre em contato com:');
     Writeln;
     Textcolor (White);
     Writeln ('"Os PEP Legais" - LAO - 4oR Eletrônica');
     Textcolor (LightGray);
     Writeln;
     Halt(2);
End;

Begin
     If Not ArqPres('CONFIG.EXE') Then MensErroArq('CONFIG.EXE');
     If Not ArqPres('EGAVGA.BGI') Then MensErroArq('EGAVGA.BGI');
     If Not ArqPres('LITT.CHR') Then MensErroArq('LITT.CHR');
     If Not ArqPres('APRES1.PEP') Then MensErroArq('APRES1.PEP');
     If Not ArqPres('APRES2.PEP') Then MensErroArq('APRES2.PEP');
     If Not ArqPres('APRES3.PEP') Then MensErroArq('APRES3.PEP');
     If Not ArqPres('APRES4.PEP') Then MensErroArq('APRES4.PEP');
End;

{************************************************************************}

Procedure Apres;
{Exibe as telas de apresentação do programa}

Var ApresTec:Char; {Tecla lida durante a apresentação}
    TempoDelay:Longint; {Delay dado após a última tela de apresentação}

Begin
     SetActivePage(0);
     SetTextStyle (SmallFont,HorizDir,4);
     SetTextJustify (CenterText,CenterText);
     SetColor (White);
     OutTextxy (320,175,'Carregando Tela de Apresentacao. Aguarde...');
     SetActivePage(1);
     ApresTec:=ViewPEP('APRES1.PEP',1,0,True);
     If ApresTec<>#27 Then Begin
        SetVisualPage(1);
        SetActivePage(0);
        Ap:=0;
     End;
     ApresTec:=ViewPEP('APRES2.PEP',Ap,0,True);
     If ApresTec<>#27 Then Begin
        SetVisualPage(0);
        SetActivePage(1);
        Ap:=1;
     End;
     ApresTec:=ViewPEP('APRES3.PEP',Ap,0,True);
     If ApresTec<>#27 Then Begin
        SetVisualPage(1);
        SetActivePage(0);
        Ap:=0;
     End;
     ApresTec:=ViewPEP('APRES4.PEP',Ap,0,True);
     If ApresTec<>#27 Then Begin
        SetVisualPage(0);
        SetActivePage(1);
        Ap:=1;
     End;
     For TempoDelay:=0 To 500000 Do Begin
         If Keypressed Then ApresTec:=Readkey;
         If (ApresTec=#13) Or (ApresTec=#27) Then Break;
     End;
     SetActivePage(1);
     ClearDevice;
     SetActivePage(0);
     ClearDevice;
     SetTextStyle (SmallFont,HorizDir,4);
     SetTextJustify (CenterText,CenterText);
     SetColor (White);
     OutTextxy (320,175,'Carregando Tela Principal. Aguarde...');
     SetVisualPage(0);
     SetActivePage(1);
     Ap:=1;
End;
```

```
{*************************************************************************}

Procedure IndicCanal (Canal:Char;Estado:Boolean);
{Seta os canais para ON ou OFF
  Canal = Qual o canal
  Estado = 1:ON ou 0:OFF}

Begin
     Canal:=Upcase(Canal);
     SetFillStyle (0,0);
     SetTextJustify (CenterText,CenterText);
     SetTextStyle (SmallFont,HorizDir,4);

     If Canal='A' Then Begin
        If Estado=True Then Begin
           A:=True;
           Bar (4,50,62,68);
           SetColor (LightGreen);
           OutTextXy (33,58,'ON');
        End
        Else Begin
             A:=False;
             Bar (4,50,62,68);
             SetColor (LightRed);
             OutTextXy (33,58,'OFF');
        End;
     End
     Else If Canal='B' Then Begin
          If Estado=True Then Begin
             B:=True;
             Bar (4,90,62,108);
             SetColor (LightGreen);
             OutTextXy (33,98,'ON');
          End
          Else Begin
               B:=False;
               Bar (4,90,62,108);
               SetColor (LightRed);
               OutTextXy (33,98,'OFF');
          End;
     End
     Else If Canal='C' Then Begin
          If Estado=True Then Begin
             C:=True;
             Bar (4,130,62,148);
             SetColor (LightGreen);
             OutTextXy (33,138,'ON');
          End
          Else Begin
               C:=False;
               Bar (4,130,62,148);
               SetColor (LightRed);
               OutTextXy (33,138,'OFF');
          End;
     End
     Else If Canal='D' Then Begin
          If Estado=True Then Begin
             D:=True;
             Bar (4,170,62,188);
             SetColor (LightGreen);
             OutTextXy (33,178,'ON');
          End
          Else Begin
               D:=False;
               Bar (4,170,62,188);
               SetColor (LightRed);
               OutTextXy (33,178,'OFF');
          End;
     End
     Else If Canal='E' Then Begin
          If Estado=True Then Begin
             E:=True;
             Bar (4,210,62,228);
             SetColor (LightGreen);
             OutTextXy (33,218,'ON');
          End
          Else Begin
               E:=False;
               Bar (4,210,62,228);
               SetColor (LightRed);
               OutTextXy (33,218,'OFF');
          End;
     End
```

```
     Else If Canal='F' Then Begin
          If Estado=True Then Begin
             F:=True;
             Bar (4,250,62,268);
             SetColor (LightGreen);
             OutTextXy (33,258,'ON');
          End


          Else Begin
               F:=False;
               Bar (4,250,62,268);
               SetColor (LightRed);
               OutTextXy (33,258,'OFF');
          End;
     End
     Else If Canal='G' Then Begin
          If Estado=True Then Begin
             G:=True;
             Bar (4,290,62,308);
             SetColor (LightGreen);
             OutTextXy (33,298,'ON');
          End
          Else Begin
               G:=False;
               Bar (4,290,62,308);
               SetColor (LightRed);
               OutTextXy (33,298,'OFF');
          End;
     End
     Else If Canal='H' Then Begin
          If Estado=True Then Begin
             H:=True;
             Bar (4,330,62,348);
             SetColor (LightGreen);
             OutTextXy (33,338,'ON');
          End
          Else Begin
               H:=False;
               Bar (4,330,62,348);
               SetColor (LightRed);
               OutTextXy (33,338,'OFF');
          End;
     End;
End;

{*************************************************************************}

Procedure IndicaFreq (FA:Real;FB:Real;FC:Real;FD:Real;FE:Real;FF:Real;FG:Real;
                      FH:Real);
{Mostra os valores de frequência dos canais ativos
   FA a FH = Valores calculados pelo frequencímetro}

Var Valor:String; {Armazena o valor da frequência em string}

Begin
     SetTextJustify (CenterText,CenterText);
     SetTextStyle (SmallFont,HorizDir,4);
     SetColor (Cyan);
     SetFillStyle (0,0);
     Y:=40;
     Repeat
           Bar (576,Y,634,Y+20);
           Y:=Y+40;
     Until Y>320;
     If A=True Then Begin
        Str (FA:8:3,Valor);
        OutTextXy (605,50,Valor);
     End;
     If B=True Then Begin
        Str (FB:8:3,Valor);
        OutTextXy (605,90,Valor);
     End;
     If C=True Then Begin
        Str (FC:8:3,Valor);
        OutTextXy (605,130,Valor);
     End;
     If D=True Then Begin
        Str (FD:8:3,Valor);
        OutTextXy (605,170,Valor);
     End;
     If E=True Then Begin
        Str (FE:8:3,Valor);
        OutTextXy (605,210,Valor);
```

```
     End;
     If F=True Then Begin
        Str (FF:8:3,Valor);
        OutTextXy (605,250,Valor);
     End;
     If G=True Then Begin
        Str (FG:8:3,Valor);
        OutTextXy (605,290,Valor);
     End;

     If H=True Then Begin
        Str (FH:8:3,Valor);
        OutTextXy (605,330,Valor);
     End;
End;

{*************************************************************************}

Procedure IndicEscala;
{Indica a escala atual de varredura horizontal}

Var ValEscala:String; {Armazena o valor da escala em string}
    ValFator:String; {Armazena o fator de multiplicação em string}

Begin
     SetColor (Cyan);
     SetTextJustify (LeftText,CenterText);
     SetTextStyle (SmallFont,HorizDir,4);
     If Round(Escala*Fator)=100000 Then Begin
        Str (Fator:2,ValFator);
        ValEscala:='EXT/'+ValFator;
     End
     Else Str (Escala:7:1,ValEscala);
     SetFillStyle(0,0);
     Bar (TextWidth('Escala Horizontal: '),15,
          3+TextWidth('Escala Horizontal:  ')+40,25);
     OutTextXy (3+TextWidth('Escala Horizontal: '),20,ValEscala);
End;

{*************************************************************************}

Procedure IndicSync;
{Indica o Canal Base de Sync}

Begin
     SetColor (Cyan);
     SetTextJustify (LeftText,CenterText);
     SetTextStyle (SmallFont,HorizDir,4);
     SetFillStyle(0,0);
     Bar (200+TextWidth('-  Canal base de Sincronismo: '),15,
          200+TextWidth('-  Canal base de Sincronismo: ')+5,25);
     OutTextxy (200+TextWidth('-  Canal base de Sincronismo: '),20,CanalSync);
End;

{*************************************************************************}

Procedure IndicZoom;
{Indica o valor do Zoom}

Var FtStr:String; {Armazena o fator de multiplicação em string}

Begin
      SetFillStyle(0,0);
      SetTextStyle (SmallFont,HorizDir,4);
      SetTextJustify (RightText,CenterText);
      SetColor (White);
      Bar (639-TextWidth('              Pressione F1 para Ajuda '),15,
           639-TextWidth('  Pressione F1 para Ajuda '),25);
      Str (Fator:1,FtStr);
      OutTextxy (639-TextWidth('         Pressione F1 para Ajuda '),20,
                 '-  Zoom: ');
      SetColor (Cyan);
      OutTextxy (639-TextWidth('       Pressione F1 para Ajuda '),
                 20,FtStr);
End;

{*************************************************************************}

Procedure MudaSync;
{Altera o canal base de Sync}

Begin
     Port[P2]:=StatusPort Xor $40;
     Port[P2]:=StatusPort;
```

```
     If CanalSync='H' Then CanalSync:='A'
     Else If CanalSync='A' Then CanalSync:='B'
     Else If CanalSync='B' Then CanalSync:='C'
     Else If CanalSync='C' Then CanalSync:='D'
     Else If CanalSync='D' Then CanalSync:='E'
     Else If CanalSync='E' Then CanalSync:='F'
     Else If CanalSync='F' Then CanalSync:='G'
     Else If CanalSync='G' Then CanalSync:='H';
     IndicSync;
End;

{*************************************************************************}

Procedure MudaEscala;
{Muda a escala de uS/div}

Var FatEsc:Real; {Armazena a escala * fator}

Begin
     Port[P2]:=StatusPort Xor $4;
     Port[P2]:=StatusPort;
     FatEsc:=Escala*Fator;
     If FatEsc=100e3 Then Escala:=Escala/100e3
     Else Escala:=Escala*10;
     IndicEscala;
End;

{*************************************************************************}

Procedure DivEscala;
{Zoom X1, X2, X5 ou X10}

Begin
     If Fator=1 Then Begin
        Fator:=2;
        Escala:=(Escala/2);
        FimPonto:=Round(FimPonto/2);
     End
     Else If Fator=2 Then Begin
          Fator:=5;
          Escala:=((Escala*2)/5);
          FimPonto:=Round((FimPonto*2)/5);
     End
     Else If Fator=5 Then Begin
          Fator:=10;
          Escala:=((Escala*5)/10);
          FimPonto:=Round((FimPonto*5)/10);
     End
     Else Begin
          Fator:=1;
          Escala:=Escala*10;
          FimPonto:=510;
     End;
     IndicZoom;
     IndicEscala;
     Zoom:=False;
End;

{*************************************************************************}

Procedure FimProg;
{Finaliza a execução do Programa}

Begin
     CloseGraph;
     Textcolor (LightGray);
     TextBackGround (Black);
     Writeln;
     Port[P1]:=$00; {Desliga a Porta Paralela}
     Halt(0);
End;

{*************************************************************************}

Procedure GravaArq;
{Grava as formas de onda em um arquivo .OSC}

Var SobreGrava:Boolean; {Verifica se sobreescreve o arquivo}
    Io:Boolean; {Retorna o resultado da operação de I/O}
    OffCan,OnCan, {Status dos canais (On/Off)}
    DadoLido:Char; {Dado lido da Interface em formato Char}
    DadoEsc:Byte; {Representa o valor da escala}
    Tecdo:Char; {Armazena caracter lido do teclado}
    PP:Integer; {Utilizada pelo contador For para apagar as linhas da tela}
```

```
Label DigitNovam; {Digitar novamente o nome-de-arquivo}
Label EscOutNome; {Escolher outro nome-de-arquivo}

Procedure Jaexiste;
{Já existe um arquivo .OSC com o mesmo nome}

Var Teclado:Char; {Armazena caracter lido do teclado}
    PQ:Integer;  {Utilizada pelo contador For para apagar as linhas da tela}



Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PQ:=2 To 60 Do Begin
         Gotoxy (PQ,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (9,'O arquivo já existe!');
     CenterTxt (14,'(S)im ou (N)ão');
     TextColor (White);
     CenterTxt (12,'Substituir o arquivo existente?');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Escolha a opção: S ou N');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
     Sobregrava:=False;
Repeat
     Teclado:=Readkey;
     Teclado:=Upcase(Teclado);
Until (Teclado='S') Or (Teclado='N');
     If Teclado='S' Then Sobregrava:=True
     Else Sobregrava:=False;
End;

Procedure ErroIO;
{Mensagem: Erro de I/O}

Var PQ:Integer;  {Utilizada pelo contador For para apagar as linhas da tela}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PQ:=2 To 60 Do Begin
         Gotoxy (PQ,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (9,'Houve um erro na Gravação');
     CenterTxt (10,'do Arquivo!');
     TextColor (White);
     CenterTxt (12,'Pressione Enter para Continuar');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Pressione Enter para Continuar');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
Repeat
      TecDo:=Readkey;
Until TecDo=#13;
End;

Procedure Gravacao;
```

```
{Gravação do arquivo .OSC}

Var PTec:Char; {Armazena caracter lido do teclado}
    Num,Num2:Integer; {Representam coordenadas no array de descrições}
    PQ:Integer;  {Utilizada pelo contador For para apagar as linhas da tela}

Procedure Descricao (YValue:Integer);
{Permite o recolhimento de descrição
         YValue = Numero da descrição}

Var Cont:Integer; {Número do caracter a ser escrito no array de descrição}
    X1,Y1:Integer; {Armazenam posição atual do cursor}

Begin
     TextColor (Yellow);
     TextBackGround (Cyan);
     X1:=WhereX; Y1:=WhereY;
     Cont:=1;
Repeat
         PTec:=Readkey;
         PTec:=Upcase(PTec);
         If PTec=#13 Then Break;
         If (PTec=#8) And (Cont>1) Then Begin
            Cont:=Cont-1;
            Descr[Cont,YValue]:=#0;
            X1:=X1-1;
            GotoXy (X1,Y1);
            Write (' ');
            GotoXy (X1,Y1);
          End;
          If PTec<>#8 Then Begin
              Descr[Cont,YValue]:=PTec;
              Write (PTec);
              X1:=X1+1;
              Cont:=Cont+1;
         End;
         If Cont=11 Then Begin
            Repeat
                  PTec:=Readkey;
            Until (PTec=#13) Or (PTec=#8);
            If PTec=#8 Then Begin
               Cont:=Cont-1;
               Descr[Cont,YValue]:=#0;
               X1:=X1-1;
               GotoXy (X1,Y1);
               Write (' ');
               GotoXy (X1,Y1);
            End;
         End;
Until Cont>10;
End;

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PQ:=2 To 60 Do Begin
         Gotoxy (PQ,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (60,15,White,Cyan);
     CenterWin (60,15,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (5,'Digite a Descrição para o Conjunto de Formas de Onda:');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Digite as Descrições - Máximo 10 Caracteres cada');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
     Window (1,1,80,25);
     Gotoxy (35,7);
     Descricao (0);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (9,'Digite a Descrição para cada Canal:');
     CenterTxt (11,'A = '+          ');
     CenterTxt (12,'B = '+          ');
     CenterTxt (13,'C = '+          ');
```

```
     CenterTxt (14,'D = '+          ');
     CenterTxt (15,'E = '+          ');
     CenterTxt (16,'F = '+          ');
     CenterTxt (17,'G = '+          ');
     CenterTxt (18,'H = '+          ');
     For Num:=1 To 8 Do Begin
         GotoXy (38,Num+10);
         Descricao (Num);
     End;

     {Identificador do arquivo: Bytes 0 a 2}
     OnCan:='P';
     Write (Arq,OnCan);
     OnCan:='E';
     Write (Arq,OnCan);
     OnCan:='P';
     Write (Arq,OnCan);

     {Descrições: Bytes 3 a 92}
     For Num2:=0 To 8 Do Begin
         For Num:=1 To 10 Do Write (Arq,Descr[Num,Num2]);
     End;

     {Status dos canais: Bytes 93 a 100}
     OnCan:=Chr(1);
     OffCan:=Chr(0);
     If A Then Write (Arq,OnCan)
     Else Write (Arq,OffCan);
     If B Then Write (Arq,OnCan)
     Else Write (Arq,OffCan);
     If C Then Write (Arq,OnCan)
     Else Write (Arq,OffCan);
     If D Then Write (Arq,OnCan)
     Else Write (Arq,OffCan);
     If E Then Write (Arq,OnCan)
     Else Write (Arq,OffCan);
     If F Then Write (Arq,OnCan)
     Else Write (Arq,OffCan);
     If G Then Write (Arq,OnCan)
     Else Write (Arq,OffCan);
     If H Then Write (Arq,OnCan)
     Else Write (Arq,OffCan);

     {Escala Horizontal: Byte 101}
     If Escala*Fator=1 Then DadoEsc:=0
     Else If Escala*Fator=10 Then DadoEsc:=1
     Else If Escala*Fator=100 Then DadoEsc:=2
     Else If Escala*Fator=1000 Then DadoEsc:=3
     Else If Escala*Fator=10000 Then DadoEsc:=4
     Else If Escala*Fator=100e3 Then DadoEsc:=5;
     OnCan:=Chr(DadoEsc);
     Write (Arq,OnCan);

     {Dados - 512 Bytes: Bytes 102 a 614}
     For Num:=0 To 511 Do Begin
         DadoLido:=Chr(Dado[Num]);
         Write (Arq,DadoLido);
     End;
End;

Begin
     CloseGraph;
     Tela('Analisador Lógico');
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackground (Cyan);
     CenterTxt (9,'Utilitário de Gravação');
     CenterTxt (10,'de Arquivos do');
     CenterTxt (11,'Analisador Lógico');
     TextColor (White);
     CenterTxt (14,'"Os PEP Legais"');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Pressione Enter para Continuar');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
Repeat
      TecDo:=Readkey;
Until TecDo=#13;

Repeat

EscOutNome:
```

```
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (8,'Digite o nome do arquivo a ser gravado');
     CenterTxt (9,'(com caminho completo, se necessário e');
     CenterTxt (10,'sem extensão - Extensão padrão: .OSC):');
     CenterTxt (15,'Pressione somente Enter para Cancelar');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Digite o nome do arquivo sem extensão / Enter: Cancela');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

DigitNovam:
     Io:=False;
     SobreGrava:=True;
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (12,'                                                  ');
     Gotoxy (25,12);
     Readln (NameArq);
     If NameArq = '' Then Break;
     Assign (Arq,NameArq+'.OSC');
{$I-}
     Reset(Arq);
{$I+}
     If IoResult = 0 Then Jaexiste;
     If Not Sobregrava Then Goto EscOutNome;
{$I-}
     Rewrite (Arq);
{$I+}
     If IoResult <> 0 Then ErroIO
     Else Io:=True;
Until Io;

     If Io Then Begin
        Gravacao;
        Close (Arq);
     End;
     ModoGrafico;
     SetColor(White);
     SetTextStyle (SmallFont,HorizDir,5);
     SetTextJustify (CenterText,CenterText);
     OutTextxy(319,160,'Pressione F5 para retornar a Tela Principal...');
     OutTextxy(319,190,
     'Pressione F6 para retornar ao Utilitário de Gravacao de Arquivos...');
End;

{*************************************************************************}

Procedure Pausar;
{Paralisa a tela em uso}

Var Tec:Char; {Armazena caracter lido do teclado}

Begin
      If Ap=1 Then Begin
         Ap:=0;
         SetActivepage(0);
      End
      Else Begin
           Ap:=1;
           SetActivePage(1);
      End;
      Bar (610-TextWidth('              Pressione F1 para Ajuda '),15,
           639-TextWidth('  Pressione F1 para Ajuda '),25);
      SetFillStyle(0,0);
      SetTextStyle (SmallFont,HorizDir,4);
      SetTextJustify (RightText,CenterText);
      SetColor (LightRed);
      OutTextxy (639-TextWidth('      Pressione F1 para Ajuda '),20,'Pausado');
```

```
Repeat
      If Keypressed Then Begin
         Tec:=Readkey;
         If Tec=#0 Then Begin
            Tec:=Readkey;
            If Tec=#63 Then Break {F5}
            Else If Tec=#64 Then GravaArq; {F6}
         End;
      End;
Until Pausa=False; {Nunca acontecerá}
      Pausa:=False;
      SetFillStyle (0,0);
      Bar (639-TextWidth('  Pressione F1 para Ajuda ')-80,15,
           639-TextWidth('  Pressione F1 para Ajuda '),25);


      If Ap=1 Then Begin
         Ap:=0;
         SetActivepage(0);
         SetVisualPage(1);
      End
      Else Begin
           Ap:=1;
           SetActivePage(1);
           SetVisualPage(0);
      End;
End;

{*************************************************************************}

Procedure Help;
{Mostra a tela de ajuda}

Var TecHelp:Char; {Armazena caracter lido do teclado}

Begin
     If Ap=1 Then Begin
        SetActivePage(0);
        Ap:=0;
     End
     Else Begin
          SetActivePage(1);
          Ap:=1;
     End;
     SetTextJustify(CenterText,CenterText);
     SetTextStyle(SmallFont,HorizDir,4);
     SetFillStyle(1,Black);
     Bar (130,105,530,265);
     SetFillStyle(1,White);
     Bar (120,95,520,255);
     SetColor(LightGray);
     SetLineStyle(0,1,3);
     Line (120,95,120,255);
     Line (520,95,520,255);
     Line (120,95,520,95);
     Line (120,255,520,255);
     SetLineStyle (0,0,0);
     SetFillStyle(1,Blue);
     Bar (122,97,518,110);
     SetColor(Black);
     Line (119,94,119,256);
     Line (521,94,521,256);
     Line (119,94,521,94);
     Line (119,256,521,256);
     SetColor (White);
     OutTextXy (320,103,'Ajuda do Analisador Lógico');
     SetColor(Magenta);
     SetTextStyle (SmallFont,HorizDir,5);
     OutTextxy(320,120,'Os PEP Legais');
     SetTextStyle (SmallFont,HorizDir,4);
     SetColor(Black);
     OutTextxy(320,140,'Esc = Sai do Programa');
     OutTextxy(320,152,'A,B,C,D,E,F,G,H = Liga/Desliga Canal');
     OutTextxy(320,164,'F1 = Ajuda');
     OutTextxy(320,176,'F2 = Seleciona a Escala Horizontal (uS/Div)');
     OutTextxy(320,188,'F3 = Zoom Horizontal (1, 2, 5, 10)');
     OutTextxy(320,200,'F4 = Seleciona o Canal Base de Sincronismo');
     OutTextxy(320,212,'F5 = Paralisa a Tela Atual');
     OutTextxy(320,224,'F6 = Salva Formas de Onda em Arquivo');
     SetColor(Red);
     OutTextxy(320,240,'Pressione Enter para Fechar esta Janela');

Repeat
      TecHelp:=Readkey;
```

```
Until TecHelp=#13;

     If Ap=1 Then Begin
        SetActivePage(0);
        Ap:=0;
     End
     Else Begin
          SetActivePage(1);
          Ap:=1;
     End;

     Ajuda:=False;
End;

{*************************************************************************}


Procedure VerifTecla;
{Verifica qual Tecla foi pressionada}

Var Tecla:Char; {Armazena caracter lido do teclado}

Procedure VerifASCII;
{Verifica teclas extendidas}

Begin
     Tecla:=Readkey;
     If Tecla=#59 Then Ajuda:=True {F1}
     Else If Tecla=#60 Then MudaEscala {F2}
     Else If Tecla=#61 Then Zoom:=True {F3}
     Else If Tecla=#62 Then MudaSync {F4}
     Else If Tecla=#63 Then Pausa:=True {F5}
                                        {GravaArq = F6}
End;

Begin
     Tecla:=Readkey;
     Tecla:=Upcase(Tecla);
     If Tecla=#27 Then FimProg {Esc}
     Else If Tecla='A' Then Begin
          If A Then IndicCanal('A',False)
          Else IndicCanal ('A',True);
     End
     Else If Tecla='B' Then Begin
          If B Then IndicCanal('B',False)
          Else IndicCanal ('B',True);
     End
     Else If Tecla='C' Then Begin
          If C Then IndicCanal('C',False)
          Else IndicCanal ('C',True);
     End
     Else If Tecla='D' Then Begin
          If D Then IndicCanal('D',False)
          Else IndicCanal ('D',True);
     End
     Else If Tecla='E' Then Begin
          If E Then IndicCanal('E',False)
          Else IndicCanal ('E',True);
     End
     Else If Tecla='F' Then Begin
          If F Then IndicCanal('F',False)
          Else IndicCanal ('F',True);
     End
     Else If Tecla='G' Then Begin
          If G Then IndicCanal('G',False)
          Else IndicCanal ('G',True);
     End
     Else If Tecla='H' Then Begin
          If H Then IndicCanal('H',False)
          Else IndicCanal ('H',True);
     End
     Else If Tecla=#0 Then VerifASCII;
End;

{*************************************************************************}

Procedure TelaPrinc;
{Desenha a tela principal do programa}

Begin
     {Título}
     SetColor (White);
     SetTextJustify (CenterText,CenterText);
     SetTextStyle (Smallfont,HorizDir,5);
```

```
SetFillStyle (1,Blue);
Bar (0,0,639,12);
OutTextxy (320,4,'Analisador Lógico');

{Grid}
SetColor(DarkGray);
X:=64;
Repeat
      Line (X,29,X,349);
      X:=X+2;
Until X>574;
Setcolor(LightGray);
X:=64;
Repeat
      Line (X,29,X,349);
      X:=X+10;
Until X>574;

{Linhas Horizontais}
SetColor(White);
Y:=29;
Repeat
      Line (3,Y,635,Y);
      Y:=Y+40;
Until Y>349;

{Linhas Verticais}
Setcolor(White);
Line (3,29,3,349);
Line (63,29,63,349);
Line (575,29,575,349);
Line (635,29,635,349);

{Nomes dos canais}
SetTextStyle (SmallFont,HorizDir,7);
SetColor (White);
SetTextJustify (CenterText,CenterText);
OutTextXy (33,40,'A');
OutTextXy (33,80,'B');
OutTextXy (33,120,'C');
OutTextXy (33,160,'D');
OutTextXy (33,200,'E');
OutTextXy (33,240,'F');
OutTextXy (33,280,'G');
OutTextXy (33,320,'H');

{Status dos Canais}
IndicCanal ('A',A);
IndicCanal ('B',B);
IndicCanal ('C',C);
IndicCanal ('D',D);
IndicCanal ('E',E);
IndicCanal ('F',F);
IndicCanal ('G',G);
IndicCanal ('H',H);

If Keypressed Then VerifTecla;

{Nomes dos Frequencímetros dos Canais}
SetColor (White);
SetTextJustify (CenterText,CenterText);
SetTextStyle (SmallFont,HorizDir,2);
Y:=35;
Repeat
      OutTextXy (605,Y,'Frequencia');
      Y:=Y+40;
Until Y>315;
SetTextStyle (SmallFont,HorizDir,4);
Y:=63;
Repeat
      OutTextXy (605,Y,'KHz');
      Y:=Y+40;
Until Y>345;

{Indicador de escala horizontal / Zoom e Canal de Sync / e Help}
SetColor (White);
SetTextJustify (LeftText,CenterText);
SetTextStyle (SmallFont,HorizDir,4);
OutTextXy (3,20,'Escala Horizontal: ');
OutTextxy (TextWidth('Escala Horizontal:   ')+40,20,'uS/div');
OutTextxy (200,20,'- Canal base de Sincronismo: ');
SetTextJustify (RightText,CenterText);
OutTextxy (639,20,'Pressione F1 para Ajuda ');
IndicEscala;
```

```
     IndicSync;
End;

{*************************************************************************}

Procedure VerifIntface;
{Verifica se a Interface está conectada corretamente ao Microcomputador}

Procedure MensErro;
{Mensagem de Erro: Interface não detectada}

Begin
        Port[P1]:=$00; {Desliga a Porta Paralela}
        CloseGraph;
        TextBackGround (Black);
        ClrScr;
        TextColor (Blue);
        Writeln ('Analisador Lógico - Mensagem de Erro');
        Writeln;
        TextColor (LightRed);
        Writeln ('A Interface não está conectada corretamente!');
        Writeln ('Verifique as conexões e a alimentação da Interface!');
        Writeln ('Após isso, execute novamente este programa.');
        TextColor (LightGray);
        Writeln;
        Writeln ('Caso não consiga solucionar este problema,');
        Writeln ('entre em contato com:');
        Writeln;
        Textcolor (White);
        Writeln ('"Os PEP Legais" - LAO - 4oR Eletrônica');
        Textcolor (LightGray);
        Writeln;
        Halt(2);
End;

Begin
     If HabilVerif Then Begin
        Port[P2]:=$22; {0010 0010 - Reset/RD/ClrSync}
        Delay(10);
        If (Port[P2] And $20) Shr 5 = 1 Then MensErro; {Fim=1 > Erro}
     End;
End;

{*************************************************************************}

Procedure AtualizaEscSync;
{Atualiza a escala/canal base de sync para reset de escala simultâneo ao}
{reset do endereçador}

Var EscAt:Integer; {Escala a ser modificada}
    Vezes:Integer; {Número de pulsos enviados para atualizar escala/sync}

Procedure AlteraPortEsc;
{Produz um pulso no SelHoriz}

Begin
     Port[P2]:=StatusPort Xor $4; {SelHoriz}
     Port[P2]:=StatusPort;
End;

Procedure AlteraPortSync;
{Produz um pulso no SelSync}

Begin
     Port[P2]:=StatusPort Xor $40; {SelSync}
     Port[P2]:=StatusPort;
End;

Begin
     EscAt:=Round(Escala*Fator);
     If EscAt=10 Then AlteraPortEsc
     Else If EscAt=100 Then Begin
          For Vezes:=1 To 2 Do AlteraPortEsc;
     End
     Else If EscAt=1000 Then Begin
          For Vezes:=1 To 3 Do AlteraPortEsc;
     End
     Else If EscAt=10000 Then Begin
          For Vezes:=1 To 4 Do AlteraPortEsc;
     End
     Else If EscAt=100e3 Then Begin
          For Vezes:=1 To 5 Do AlteraPortEsc;
     End;
```

```
     If CanalSync='B' Then AlteraPortSync
     Else If CanalSync='C' Then Begin
          For Vezes:=1 To 2 Do AlteraPortSync;
     End
     Else If CanalSync='D' Then Begin
          For Vezes:=1 To 3 Do AlteraPortSync;
     End
     Else If CanalSync='E' Then Begin
          For Vezes:=1 To 4 Do AlteraPortSync;
     End
     Else If CanalSync='F' Then Begin
          For Vezes:=1 To 5 Do AlteraPortSync;
     End
     Else If CanalSync='G' Then Begin
          For Vezes:=1 To 6 Do AlteraPortSync;
     End
     Else If CanalSync='H' Then Begin
          For Vezes:=1 To 7 Do AlteraPortSync;
     End;
End;

{*************************************************************************}

Procedure WriteDados;
{Permite a gravação das formas de onda pela Interface}

Begin
     Port[P2]:=$22; {0010 0010 - Reset/ClrSync/RD}
     VerifIntface;
     Port[P2]:=$A3; {1010 0011 - WR/ClrSync}
     StatusPort:=$A3;
     If HabilAtuEscSync Then AtualizaEscSync;
     StatusPort:=$B3;
     Port[P2]:=$B3; {1011 0011 - WR - Start}
     GetTime (Hr,M,S,S100);
     Timea:=S100+S*100+M*6E3+Hr*360E3;

Repeat
      If Keypressed Then VerifTecla;
      GetTime (Hr,M,S,S100);                    {Aguarda resposta até 600mS}
      If ((S100+S*100+M*6E3+Hr*360E3)-Timea > 60) And (HabilSyncOff) Then Begin
         Port[P2]:=$B1;  {1011 0001 - WR /PresetSync}
         StatusPort:=$B1;               {Se não há resposta, desliga Sync}
      End;
Until ((Port[P2] And $20) Shr 5 = 1) Or
      ((S100+S*100+M*6E3+Hr*360E3)-Timea > 100); {Espera o sinal Fim até 1s}

End;

{*************************************************************************}

Procedure ReadDados;
{Lê os dados a partir da Interface}

Begin
     VerifIntface;
     Port[P2]:=$32; {0011 0010 - RD/Reset}
     Port[P2]:=$33; {0011 0011 - RD}
     StatusPort:=$33;

     For IndDado:=0 To 511 Do Begin
         Port[P2]:=$3B; {0011 1011 - RD/ClkRD}
         Port[P2]:=$33; {0011 0011 - RD}
         Port[P1]:=$FF;
         Dado[IndDado]:=Port[P1];
         Port[P2]:=StatusPort;
         Delay (1);
         If Keypressed Then VerifTecla;
         End;
End;

{*************************************************************************}

Procedure MostraDados;
{Plota na Tela as Formas de Onda dos canais ativos}

Var YPlota:Integer; {Coordenada Y da forma-de-onda a ser plotada}
    CanalByte:Integer; {Identifica qual canal a ser plotado}

Procedure Plota;
{Plota as formas-de-onda na tela}

Var Ct:Integer; {Número de vezes que se repetirá a plotagem do dado}
```

```
Begin
        For IndDado:=1 To FimPonto Do Begin
            For Ct:=1 To Fator Do Begin

                PutPixel ((Ct-Fator)+(IndDado*Fator)+64,
                      (((Dado[IndDado] And Eleva(2,CanalByte))
                      Shr CanalByte) Xor 1)*32+YPlota+3,LightGreen);

                PutPixel ((Ct-Fator)+(IndDado*Fator)+64,
                      (((Dado[IndDado] And Eleva(2,CanalByte))
                      Shr CanalByte) Xor 1)*34+YPlota+2,LightGreen);

                If Keypressed Then VerifTecla;
            End;
        End;
End;

Begin
     If A Then Begin
        YPlota:=30;
        CanalByte:=0;
        Plota;
     End;
     If B Then Begin
        CanalByte:=1;
        YPlota:=70;
        Plota;
     End;
     If C Then Begin
        YPlota:=110;
        CanalByte:=2;
        Plota;
     End;
     If D Then Begin
        CanalByte:=3;
        YPlota:=150;
        Plota;
     End;
     If E Then Begin
        YPlota:=190;
        CanalByte:=4;
        Plota;
     End;
     If F Then Begin
        YPlota:=230;
        CanalByte:=5;
        Plota;
     End;
     If G Then Begin
        YPlota:=270;
        CanalByte:=6;
        Plota;
     End;
     If H Then Begin
        CanalByte:=7;
        YPlota:=310;
        Plota;
     End;
End;

{*************************************************************************}

Procedure Frequenc;
{Calcula as frequências dos canais}

Var FrA,FrB,FrC,FrD,FrE,FrF,FrG,FrH:Real; {Armazenam as frequências dos canais}
    Pontos:Integer; {Número de pontos em um estado lógico}
    Dc:Boolean; {Identifica sinal contínuo constante}

Procedure MedeFreq (CanBit:Integer);
{Medidor de Frequência
  CanBit = Indica qual o canal}

Begin
     IndDado:=0;
     Pontos:=0;
     Dc:=False;
Repeat
      IndDado:=IndDado+1;
Until ((Dado[IndDado] And Eleva(2,CanBit)) Shr CanBit =1) Or (IndDado=511);

Repeat
      IndDado:=IndDado+1;
Until ((Dado[IndDado] And Eleva(2,CanBit)) Shr CanBit =0) Or (IndDado=511);
```

```
      If IndDado>=511 Then Dc:=True;

      If Keypressed Then VerifTecla;

Repeat
      IndDado:=IndDado+1;
      Pontos:=Pontos+1;
Until ((Dado[IndDado] And Eleva(2,Canbit)) Shr CanBit =1) Or (IndDado=511);
      If IndDado>=511 Then Dc:=True;

Repeat
      IndDado:=IndDado+1;
      Pontos:=Pontos+1;
Until ((Dado[IndDado] And Eleva(2,CanBit)) Shr CanBit =0) Or (IndDado=511);
      If IndDado>=511 Then Dc:=True;

End;

Begin
     MedeFreq (0);
     If Dc Then FrA:= 0
     Else FrA:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq (1);
     If Dc Then FrB:= 0
     Else FrB:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq (2);
     If Dc Then FrC:= 0
     Else FrC:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq (3);
     If Dc Then FrD:= 0
     Else FrD:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq (4);
     If Dc Then FrE:= 0
     Else FrE:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq (5);
     If Dc Then FrF:= 0
     Else FrF:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq (6);
     If Dc Then FrG:= 0
     Else FrG:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq (7);
     If Dc Then FrH:= 0
     Else FrH:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     IndicaFreq (FrA,FrB,FrC,FrD,FrE,FrF,FrG,FrH);
End;

{*************************************************************************}

Procedure InitDefault;
{Seta as variáveis para o Default}

Begin
     {Status inicial dos canais: True}
     A:=True; B:=True; C:=True; D:=True; E:=True; F:=True; G:=True; H:=True;
     {Escala inicial: 1uS/div}
     Escala:=1;
     FimPonto:=510;
     Fator:=1;
     {Canal Base de Sync inicial: Canal A}
     CanalSync:='A';
     {Seta Página de Vídeo Ativa como 1 e visual como 0}
     SetActivePage(1);
     SetVisualPage(0);
     Ap:=1;
End;

{*************************************************************************}

Begin
     AbreCfg;
     VerifArq;
     Clrscr;
     Port[P2]:=$22; {0010 0010 - Reset/RD/ClrSync}
     ModoGrafico;
     VerifIntface;
     InitDefault;
```

```
     Apres;

Repeat
     If Zoom Then DivEscala;
     If Ajuda Then Help;
     If Pausa Then Pausar;
     Delay(10);
     SetFillStyle(0,0);
     Bar (0,30,639,349);
     TelaPrinc;
     WriteDados;
     ReadDados;
     MostraDados;
     Frequenc;
     If Ap=1 Then Begin
        SetActivePage(0);
        SetVisualPage(1);
        Ap:=0;
     End
     Else Begin
        SetActivePage(1);
        SetVisualPage(0);
        Ap:=1;
     End;
     IndicZoom;
Until IndDado=1000; {IndDado=1000 é impossível de ocorrer}
End.

{*************************************************************************}
{
        * Estrutura dos arquivos .PEP *

Bytes 0 a 2 = Quantidade de pixels da cor indicada no próximo Byte
Byte 3 = Cor do pixel

(Repete-se essa sequência até o final do arquivo)

}
{*************************************************************************}
```

Comparador (COMPARA.PAS)

```
{$A-,B-,D-,E+,F-,G-,I+,L-,N+,Q-,R-,S+,X+,V+}
{Diretivas de compilação}

{*************************************************************************}

Program Comparador;

{*************************************************************************}

{
                Comparador de Formas de Onda

                       "Os PEP Legais"

                          Versão 1.0
}

{*************************************************************************}

{       Créditos:
                  Luiz Eduardo    - N°19
                  Marcos Paulo    - N°23
                  Rafael Abolafio - N°26
                  Robson Martins  - N°30

                  4°R - Eletrônica - 1997

               Liceu de Artes e Ofícios de São Paulo

               Prof: Marco Antônio Togniazzolo
}
{*************************************************************************}
{
        Comparador de Formas de Onda:
                              Plota na tela um ou dois conjuntos de 8 formas
                              Realiza Operações Lógicas entre formas
                              Exporta imagem para .BMP
                              Versão 1.0 - 1997

        Software for MS-DOS, Escrito no compilador Turbo Pascal 7.0

        Requisitos Mínimos do Sistema:
```

```
                    IBM PC XT/AT ou Compatível (Recomenda-se 486)
                    640KB de RAM ou mais (Recomenda-se 4MB)
                    Monitor EGA ou superior (Recomenda-se VGA Color)
                    Disk Drive de 1,44MB
                    Hard Disk com pelo menos 1MB livres
}
{*************************************************************************}

Uses Crt,Dos,Graph,PEPUnit;
{Units a serem usadas}

{*************************************************************************}

Var
   Hr,M,S,S100:Word; {Variáveis para contar Tempo}
   Timea:Real; {Variável para contar Tempo}
   Ap:Integer; {Indica a página de vídeo ativa}
   NameArq1,NameArq2:String; {Recebem os nomes dos arquivos que serão lidos}
   Arq1,Arq2:File of Char; {Representam os arquivos .OSC}
   Arq2Habil:Boolean; {Indica se haverá um segundo arquivo}
   Desc1,DescA1,DescB1,DescC1,DescD1,DescE1,DescF1,DescG1,DescH1,
   Desc2,DescA2,DescB2,DescC2,DescD2,DescE2,DescF2,DescG2,DescH2:String;
   {Armazenam as descrições}
   A1,B1,C1,D1,E1,F1,G1,H1,
   A2,B2,C2,D2,E2,F2,G2,H2:Boolean; {Indicam se o canal está On ou Off}
   Escala1,Escala2:String[5]; {Indica a Escala}
   Dado1,Dado2,DadoSalva: Array [0..511] Of Char;
   {Armazenam os dados dos arquivos}
   X,Y:Integer; {Coordenadas X e Y}
   Tecla:Char; {Armazena a tecla pressionada}
   PutBMP:Boolean; {Determina OK para grav. do BMP}
   BMPWr:Boolean; {Determina se será Gravado um BMP}
   NameBMP:String; {Nome do arquivo BMP}
   ArqBMP:File of Char; {Variável que representa BMP}
   FilTog:Boolean; {Determina se TogFileName foi pressionado}
   DT1,DT2:DateTime; {Armazena a data e hora dos arquivos .OSC}
   Iniciar:Boolean; {Verifica se o usuário deseja escolher novos .OSC}
   Oper:String; {Define a operação a ser realizada com as formas de onda}
   Help:Boolean; {Indica se pressionado Help}
   ZoomHabil:Boolean; {Determina que foi pressionado Zoom}
   Fator:Integer; {Fator de Zoom (1,2,5 ou 10)}

{*************************************************************************}

Label Inicio; {Permite que o usuário escolha novos arquivos .OSC}

{*************************************************************************}

Procedure VerifArq;
{Verifica a presença dos arquivos necessários para a execução do programa}

Function ArqPres(NomeArqP:String):Boolean;
{Verifica a presença do arquivo NomeArqP}

Var FPres:File Of Char; {Representa o arquivo a ser testado}

Begin
     Assign (FPres,NomeArqP);
{$I-}
     Reset (FPres);
{$I+}
     If IOResult <> 0 Then ArqPres:=False
     Else Begin
          ArqPres:=True;
          Close (FPres);
     End;
End;

Procedure MensErroArq(NameArqF:String);
{Mensagem de Erro: Não existe o arquivo NameArqF}

Begin
     ClrScr;
     TextColor (Blue);
     Writeln ('Comparador do Analisador Lógico - Mensagem de Erro');
     Writeln;
     TextColor (LightRed);
     Writeln ('Não existe o arquivo "',NameArqF,'"');
     Writeln ('neste diretório!');
     Writeln;
     Writeln ('Verifique qual o diretório corrente!');
     Writeln ('Após isso, execute novamente este programa.');
     TextColor (LightGray);
```

```
     Writeln;
     Writeln ('Caso não consiga solucionar este problema,');
     Writeln ('entre em contato com:');
     Writeln;
     Textcolor (White);
     Writeln ('"Os PEP Legais" - LAO - 4oR Eletrônica');
     Textcolor (LightGray);
     Writeln;
     Halt(2);
End;

Begin
     If Not ArqPres('CONFIG.EXE') Then MensErroArq('CONFIG.EXE');
     If Not ArqPres('ALOGIC.EXE') Then MensErroArq('ALOGIC.EXE');
     If Not ArqPres('ALOGIC.CFG') Then MensErroArq('ALOGIC.CFG');
     If Not ArqPres('EGAVGA.BGI') Then MensErroArq('EGAVGA.BGI');
     If Not ArqPres('LITT.CHR') Then MensErroArq('LITT.CHR');
     If Not ArqPres('APRES2.PEP') Then MensErroArq('APRES2.PEP');
     If Not ArqPres('APRES4.PEP') Then MensErroArq('APRES4.PEP');
     If Not ArqPres('APRES5.PEP') Then MensErroArq('APRES5.PEP');
     If Not ArqPres('BITMAP.PAL') Then MensErroArq('BITMAP.PAL');
End;

{*************************************************************************}

Procedure Apres;
{Exibe as telas de apresentação do programa}

Var ApresTec:Char; {Armazena o valor da tecla press. durante apresentação}
    TempoDelay:Longint; {Delay para a última apresentação}

Begin
     Ap:=1;
     SetActivePage(0);
     SetTextStyle (SmallFont,HorizDir,4);
     SetTextJustify (CenterText,CenterText);
     SetColor (White);
     OutTextxy (320,175,'Carregando Tela de Apresentacao. Aguarde...');
     SetActivePage(1);
     ApresTec:=ViewPEP('APRES5.PEP',1,0,True);
     If ApresTec<>#27 Then Begin
        SetVisualPage(1);
        SetActivePage(0);
        Ap:=0;
        SetFillStyle(0,0);
        Bar(0,0,639,349);
     End;
     ApresTec:=ViewPEP('APRES2.PEP',Ap,0,True);
     If ApresTec<>#27 Then Begin
        SetVisualPage(0);
        SetActivePage(1);
        Ap:=1;
        SetFillStyle(0,0);
        Bar(0,0,639,349);
     End;
     ApresTec:=ViewPEP('APRES4.PEP',Ap,0,True);
     If ApresTec<>#27 Then Begin
        SetVisualPage(1);
        SetActivePage(0);
        Ap:=0;
        SetFillStyle(0,0);
        Bar(0,0,639,349);
     End;
     For TempoDelay:=0 To 500000 Do Begin
         If Keypressed Then ApresTec:=Readkey;
         If (ApresTec=#13) Or (ApresTec=#27) Then Break;
     End;
     CloseGraph;
End;

{*************************************************************************}

Procedure FimProg;
{Termina a execução do programa}

Begin
     Window(1,1,80,25);
     TextColor(LightGray);
     TextBackGround(Black);
     ClrScr;
     Writeln;
     Halt(0);
End;
```

```
{*************************************************************************}

Procedure Win1;
{Primeira tela do software}
{Apresentação}

Var TecDo:Char; {Armazena o valor da tecla pressionada}

Begin
     Tela('Comparador de Formas-de-Onda');
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackground (Cyan);
     CenterTxt (8,'Comparador');
     CenterTxt (9,'de');
     CenterTxt (10,'Formas de Onda');
     TextColor (White);
     CenterTxt (13,'"Os PEP Legais"');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Pressione Enter para Continuar');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

Repeat
      TecDo:=Readkey;
Until TecDo=#13;
End;

{*************************************************************************}

Procedure Win2;
{segunda tela do software}
{Lê 1o Arquivo .OSC}

Var PP:Integer; {Coordenada X = utilizada para apagar as linhas da tela}
    IoErr:Integer; {Armazena o último resultado da operação de I/O}
    FileOk:Boolean; {Arquivo Correto se True}
    Chj:Char; {Utilizado para armazenar o caracter lido do arquivo .OSC}
    PEPStr:String; {String 'PEP'}
    PEPOk:Boolean; {Arquivo correto se houver 'PEP' nos 3 primeiros bytes}

Begin
     FileOk:=False;
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;
     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (7,'Digite o nome do primeiro arquivo');
     CenterTxt (8,'a ser visualizado');
     CenterTxt (9,'(com caminho completo, se necessário e');
     CenterTxt (10,'sem extensão - Extensão padrão: .OSC):');
     CenterTxt (14,'Pressione somente Enter para Cancelar');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Digite o nome do arquivo sem extensão / Enter: Cancela');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
     TextColor (Yellow);
     TextBackGround(Cyan);
     Window (1,1,80,25);
Repeat
     CenterTxt (12,'                                              ');
     Gotoxy (30,12);
     Readln (NameArq1);
     If NameArq1='' Then FimProg;
     Assign (Arq1,NameArq1+'.OSC');
{$I-}
     Reset (Arq1);
{$I+}
     IoErr:=IoResult;
     If (IoErr <>0) Then Begin
```

```
        CenterTxt (12,'Houve um erro na abertura do arquivo!');
        Sound (1000);
        Delay (100);
        Nosound;
        Delay (1000);
        CenterTxt (12,'                                          ');
     End
     Else Begin
          Read (Arq1,Chj);
          PEPStr:=Chj;
          Read (Arq1,Chj);
          PEPStr:=PEPStr+Chj;
          Read (Arq1,Chj);
          PEPStr:=PEPStr+Chj;
          If PEPStr='PEP' Then PEPOk:=True
          Else PEPOk:=False;
          Reset(Arq1);
          If (FileSize(Arq1)=614) And (PEPOk) Then FileOk:=True
          Else Begin
                CenterTxt (12,'O formato do arquivo não é válido!');
                Sound (1000);
                Delay (100);
                Nosound;
                Delay (1000);
                CenterTxt (12,'                                                ');
          End;
     End;
Until FileOk;
End;

{*************************************************************************}

Procedure Win3;
{Terceira tela do programa}
{Lê 2o Arquivo .OSC}

Var PP:Integer; {Coordenada X = utilizada para apagar as linhas da tela}
    IoErr:Integer; {Armazena o último resultado da operação de I/O}
    FileOk:Boolean; {Arquivo Correto se True}
    Chj:Char; {Utilizado para armazenar o caracter lido do arquivo .OSC}
    PEPStr:String; {String 'PEP'}
    PEPOk:Boolean; {Arquivo correto se houver 'PEP' nos 3 primeiros bytes}

Begin
     FileOk:=False;
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;
     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (7,'Digite o nome do segundo arquivo');
     CenterTxt (8,'a ser visualizado');
     CenterTxt (9,'(com caminho completo, se necessário e');
     CenterTxt (10,'sem extensão - Extensão padrão: .OSC):');
     CenterTxt (14,'Pressione somente Enter');
     CenterTxt (15,'para visualizar um só arquivo');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write('Digite o nome do arquivo sem extensão / Enter: Cancela');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
     TextColor (Yellow);
     TextBackGround(Cyan);
     Window (1,1,80,25);
Repeat
     CenterTxt (12,'                                              ');
     Gotoxy (30,12);
     Readln (NameArq2);
     If NameArq2='' Then Begin
        Arq2Habil:=False;
        Break;
     End;
     Assign (Arq2,NameArq2+'.OSC');
{$I-}
```

```
     Reset (Arq2);
{$I+}
     IoErr:=IoResult;
     If (IoErr <>0) Then Begin
        CenterTxt (12,'Houve um erro na abertura do arquivo!');
        Sound (1000);
        Delay (100);
        Nosound;
        Delay (1000);
        CenterTxt (12,'                                                ');
     End
     Else Begin
          Read (Arq2,Chj);
          PEPStr:=Chj;
          Read (Arq2,Chj);
          PEPStr:=PEPStr+Chj;
          Read (Arq2,Chj);
          PEPStr:=PEPStr+Chj;
          If PEPStr='PEP' Then PEPOk:=True
          Else PEPOk:=False;
          Reset(Arq2);
          If (FileSize(Arq2)=614) And (PEPOk) Then FileOk:=True
          Else Begin
                CenterTxt (12,'O formato do arquivo não é válido!');
                Sound (1000);
                Delay (100);
                Nosound;
                Delay (1000);
                CenterTxt (12,'                                                ');
          End;
     End;
     Arq2Habil:=True;
Until FileOk;
End;

{*************************************************************************}

Procedure LeArqs;
{Lê os arquivos .OSC}

Var Cht:Char; {Armazena o caracter lido do arquivos}
    PP:Integer; {Cursor de leitura dos arquivos}
    TimeF1,TimeF2:Longint; {Armazenam Data/Hora dos arquivos}

Begin
     GetFTime(Arq1,TimeF1);
     UnpackTime(TimeF1,DT1);
     Desc1:=''; DescA1:=''; DescB1:=''; DescC1:=''; DescD1:=''; DescE1:='';
     DescF1:=''; DescG1:=''; DescH1:='';
     For PP:=0 To 2 Do Read(Arq1,Cht);
     For PP:=3 To 12 Do Begin
         Read(Arq1,Cht);
         Desc1:=Desc1+Cht;
     End;
     For PP:=13 To 22 Do Begin
         Read(Arq1,Cht);
         DescA1:=DescA1+Cht;
     End;
     For PP:=23 To 32 Do Begin
         Read(Arq1,Cht);
         DescB1:=DescB1+Cht;
     End;
     For PP:=33 To 42 Do Begin
         Read(Arq1,Cht);
         DescC1:=DescC1+Cht;
     End;
     For PP:=43 To 52 Do Begin
         Read(Arq1,Cht);
         DescD1:=DescD1+Cht;
     End;
     For PP:=53 To 62 Do Begin
         Read(Arq1,Cht);
         DescE1:=DescE1+Cht;
     End;
     For PP:=63 To 72 Do Begin
         Read(Arq1,Cht);
         DescF1:=DescF1+Cht;
     End;
     For PP:=73 To 82 Do Begin
         Read(Arq1,Cht);
         DescG1:=DescG1+Cht;
     End;
     For PP:=83 To 92 Do Begin
         Read(Arq1,Cht);
```

```
      DescH1:=DescH1+Cht;
  End;

  Read(Arq1,Cht);
  If Ord(Cht)=0 Then A1:=False
  Else A1:=True;
  Read(Arq1,Cht);
  If Ord(Cht)=0 Then B1:=False
  Else B1:=True;
  Read(Arq1,Cht);
  If Ord(Cht)=0 Then C1:=False
  Else C1:=True;
  Read(Arq1,Cht);
  If Ord(Cht)=0 Then D1:=False
  Else D1:=True;
  Read(Arq1,Cht);
  If Ord(Cht)=0 Then E1:=False
  Else E1:=True;
  Read(Arq1,Cht);
  If Ord(Cht)=0 Then F1:=False
  Else F1:=True;
  Read(Arq1,Cht);
  If Ord(Cht)=0 Then G1:=False
  Else G1:=True;
  Read(Arq1,Cht);
  If Ord(Cht)=0 Then H1:=False
  Else H1:=True;

  Read(Arq1,Cht);
  If Ord(Cht)=0 Then Escala1:='    1'
  Else If Ord(Cht)=1 Then Escala1:='   10'
  Else If Ord(Cht)=2 Then Escala1:='  100'
  Else If Ord(Cht)=3 Then Escala1:=' 1000'
  Else If Ord(Cht)=4 Then Escala1:='10000'
  Else If Ord(Cht)=5 Then Escala1:='  EXT';

  For PP:=102 To 613 Do Begin
      Read(Arq1,Cht);
      Dado1[PP-102]:=Cht;
  End;
  Close(Arq1);

If Arq2Habil Then Begin
  GetFTime(Arq2,TimeF2);
  UnpackTime(TimeF2,DT2);
  Desc2:=''; DescA2:=''; DescB2:=''; DescC2:=''; DescD2:=''; DescE2:='';
  DescF2:=''; DescG2:=''; DescH2:='';
  For PP:=0 To 2 Do Read(Arq2,Cht);
  For PP:=3 To 12 Do Begin
      Read(Arq2,Cht);
      Desc2:=Desc2+Cht;
  End;
  For PP:=13 To 22 Do Begin
      Read(Arq2,Cht);
      DescA2:=DescA2+Cht;
  End;
  For PP:=23 To 32 Do Begin
      Read(Arq2,Cht);
      DescB2:=DescB2+Cht;
  End;
  For PP:=33 To 42 Do Begin
      Read(Arq2,Cht);
      DescC2:=DescC2+Cht;
  End;
  For PP:=43 To 52 Do Begin
      Read(Arq2,Cht);
      DescD2:=DescD2+Cht;
  End;
  For PP:=53 To 62 Do Begin
      Read(Arq2,Cht);
      DescE2:=DescE2+Cht;
  End;
  For PP:=63 To 72 Do Begin
      Read(Arq2,Cht);
      DescF2:=DescF2+Cht;
  End;
  For PP:=73 To 82 Do Begin
      Read(Arq2,Cht);
      DescG2:=DescG2+Cht;
  End;
  For PP:=83 To 92 Do Begin
      Read(Arq2,Cht);
      DescH2:=DescH2+Cht;
  End;
```

```
     Read(Arq2,Cht);
     If Ord(Cht)=0 Then A2:=False
     Else A2:=True;
     Read(Arq2,Cht);
     If Ord(Cht)=0 Then B2:=False
     Else B2:=True;
     Read(Arq2,Cht);
     If Ord(Cht)=0 Then C2:=False
     Else C2:=True;
     Read(Arq2,Cht);
     If Ord(Cht)=0 Then D2:=False
     Else D2:=True;
     Read(Arq2,Cht);
     If Ord(Cht)=0 Then E2:=False
     Else E2:=True;
     Read(Arq2,Cht);
     If Ord(Cht)=0 Then F2:=False
     Else F2:=True;
     Read(Arq2,Cht);
     If Ord(Cht)=0 Then G2:=False
     Else G2:=True;
     Read(Arq2,Cht);
     If Ord(Cht)=0 Then H2:=False
     Else H2:=True;



     Read(Arq2,Cht);
     If Ord(Cht)=0 Then Escala2:='    1'
     Else If Ord(Cht)=1 Then Escala2:='   10'
     Else If Ord(Cht)=2 Then Escala2:='  100'
     Else If Ord(Cht)=3 Then Escala2:=' 1000'
     Else If Ord(Cht)=4 Then Escala2:='10000'
     Else If Ord(Cht)=5 Then Escala2:='  EXT';

     For PP:=102 To 613 Do Begin
         Read(Arq2,Cht);
         Dado2[PP-102]:=Cht;
     End;
     Close(Arq2);
   End;
End;

{*************************************************************************}

Procedure GravaBMP;
{Grava a tela como um BMP}

Var BMPSim,BMPOk:Boolean; {Verificam se é desejado gravar em BMP}
    ArqPAL:File of Char; {Representa o arquivo de Palette BitMap}
    BitDado:Char; {Armazena o caracter lido do arquivo de Palette}
    ContBy:Integer; {Cursor no arquivo BMP}

Procedure WinBMP1;
{Tela de apresentação - Utilitário de Exportação para BMP}

Var Tc:Char; {Armazena o caracter lido do teclado}

Begin
     BMPSim:=False;
     Tela('Comparador de Formas-de-Onda');
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (9,'Utilitário de Exportação');
     CenterTxt (10,'para BitMap');
     Textcolor(White);
     CenterTxt (14,'"Os PEP Legais"');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write('Pressione Enter para continuar ou Esc para cancelar');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
Repeat
     Tc:=Readkey;
     If Tc=#13 Then BMPSim:=True;
Until (Tc=#27) Or (Tc=#13);

End;
```

```
Procedure Win2BMP;
{ Utilitário de Exportação para BMP - recolhe o nome de arquivo}

Var PP:Integer; {Coordenada X: utilizada para apagar linhas na tela}
    IoErr:Integer; {Armazena o resultado da última operação de I/O}
    NameOk:Boolean; {Informa se o nome-de-arquivo está correto}
    TpT:Char; {Armazena o caracter lido do teclado}

Procedure Jaexiste;
{Já existe um arquivo .BMP com o mesmo nome}

Var Pl:Integer; {Coordenada X: utilizada para apagar linhas na tela}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For Pl:=2 To 60 Do Begin
         Gotoxy (Pl,24);
         Write (' ');
     End;
     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (9,'O arquivo já existe!');
     CenterTxt (12,'Substituir? (S,N)');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write('Escolha a opção: S ou N');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
End;

Begin
     BMPOk:=False;
     PutBMP:=False;
Repeat
Repeat
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;
     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (7,'Digite o nome do arquivo BitMap');
     CenterTxt (8,'a ser gravado');
     CenterTxt (9,'(com caminho completo, se necessário e');
     CenterTxt (10,'sem extensão - Extensão padrão: .BMP):');
     CenterTxt (14,'Pressione somente Enter');
     CenterTxt (15,'para cancelar');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write('Digite o nome do arquivo sem extensão / Enter: Cancela');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
     TextColor (Yellow);
     TextBackGround(Cyan);
     Window (1,1,80,25);

     NameOk:=False;
     CenterTxt (12,'                                          ');
     Gotoxy (30,12);
     Readln (NameBMP);
     If NameBMP='' Then Begin
        Break;
     End;
     Assign (ArqBMP,NameBMP+'.BMP');
{$I-}
     Reset(ArqBMP);
{$I+}
```

```
     IoErr:=IoResult;
     If IoErr=0 Then Begin
        Jaexiste;
Repeat
      TpT:=Readkey;
      Tpt:=Upcase(TpT);
Until (Tpt='S') Or (Tpt='N');
        If Tpt='S' Then NameOk:=True;
     End
     Else NameOk:=True;
Until NameOk;

     If Not NameOk Then Begin
        Break;
     End;

{$I-}
     Rewrite (ArqBMP);
{$I+}
     IoErr:=IoResult;
     If IoErr <>0 Then Begin
        CenterTxt (12,'Houve um erro na gravação do arquivo!');
        Sound (1000);
        Delay (100);
        Nosound;
        Delay (1000);
        CenterTxt (12,'                                                ');
     End
     Else BMPOk:=True;
Until BMPOk;
End;

Begin
     CloseGraph;
     WinBMP1;
     If BMPSim Then Begin
          Win2BMP;
          If BMPOk Then Begin
             Assign (ArqPAL,'BITMAP.PAL');
             Reset (ArqPAL);
             For ContBy:=0 To 117 Do Begin
                 Read (ArqPAL,BitDado);
                 Write (ArqBMP,BitDado);
             End;
             Close (ArqPAL);
             PutBMP:=True;
          End;
     End;
     BMPWr:=False;
     ModoGrafico;
     SetActivePage(1);
     SetVisualPage(0);
     Ap:=1;
End;

{*************************************************************************}

Procedure TogFName;
{Mostra uma janela com os nomes de arquivo e data/hora dos arquivos}

Var DiaStr,MesStr,AnoStr,HoraStr,MinStr,SegStr:String; {Armazenam Data/Hora
                                                        dos arquivos}
    TEnt:Char; {Armazena o caracter lido do teclado}

Function UpStr(Strg:String):String;
{Converte uma String em caracteres maiúsculos}

Var ChSt:String; {String de 1 caracter lida de Strg}
    CSint:Integer; {Conta de 1 até o número da caracteres de Strg}
    StrUp:String; {String de letras maiúsculas}

Function Upc(SCh:String):Char;
{Converte String em Char}

Var SetChar:Char; {Valor Char}
    SetByte:Byte; {Valor Byte}
    SetStr:String; {Valor String}

Begin
     For SetByte:=0 To 255 Do Begin
         SetStr:=Chr(SetByte);
         If SCh=SetStr Then Break;
     End;
     Upc:=Chr(SetByte);
```

```
End;

Begin
     StrUp:='';
     For CSint:=1 To Length(Strg) Do Begin
         ChSt:=Copy (Strg,CSint,1);
         StrUp:=StrUp+Upcase(Upc(ChSt));
     End;
     UpStr:=StrUp;
End;

Procedure DataHora(DV1:DateTime);
{Obtém data/hora dos arquivos .OSC}

Begin
     Str(DV1.Day,DiaStr);
     If Length(DiaStr)=1 Then DiaStr:='0'+DiaStr;
     Str(DV1.Month,MesStr);
     If Length(MesStr)=1 Then MesStr:='0'+MesStr;
     Str(DV1.Year,AnoStr);
     If Length(AnoStr)=1 Then AnoStr:='0'+AnoStr;
     Str(DV1.Hour,HoraStr);
     If Length(HoraStr)=1 Then HoraStr:='0'+HoraStr;
     Str(DV1.Min,MinStr);
     If Length(MinStr)=1 Then MinStr:='0'+MinStr;
     Str(DV1.Sec,SegStr);
     If Length(SegStr)=1 Then SegStr:='0'+SegStr;
End;



Begin
     If Ap=1 Then Begin
        SetActivePage(0);
        SetVisualPage(0);
     End
     Else Begin
          SetActivePage(1);
          SetVisualPage(1);
     End;
     SetTextJustify(CenterText,CenterText);
     SetTextStyle(SmallFont,HorizDir,4);
     SetFillStyle(1,Black);
     Bar (130,105,530,265);
     SetFillStyle(1,White);
     Bar (120,95,520,255);
     SetColor(LightGray);
     SetLineStyle(0,1,3);
     Line (120,95,120,255);
     Line (520,95,520,255);
     Line (120,95,520,95);
     Line (120,255,520,255);
     SetLineStyle (0,0,0);
     SetFillStyle(1,Blue);
     Bar (122,97,518,110);
     SetColor(Black);
     Line (119,94,119,256);
     Line (521,94,521,256);
     Line (119,94,521,94);
     Line (119,256,521,256);
     SetColor (White);
     OutTextXy (320,103,'Informacao de Arquivos');
     SetFillStyle(1,LightRed);
     Bar (135,125,155,135);
     SetColor(Black);
     SetTextJustify(LeftText,CenterText);
     OutTextxy(165,130,UpStr(NameArq1)+'.OSC');
     DataHora(DT1);
     OutTextxy(165,150,'Data e Hora da ultima modificacao: '+DiaStr+'/'+
               MesStr+'/'+AnoStr+'  '+HoraStr+':'+MinStr+':'+SegStr);

     If Arq2Habil Then Begin
        SetFillStyle(1,LightGreen);
        Bar (135,177,155,187);
        OutTextxy(165,182,UpStr(NameArq2)+'.OSC ');
        DataHora(DT2);
        OutTextxy(165,202,'Data e Hora da ultima modificacao: '+DiaStr+'/'+
                  MesStr+'/'+AnoStr+'  '+HoraStr+':'+MinStr+':'+SegStr);
     End;

     SetTextJustify (CenterText,CenterText);
     OutTextxy(320,240,'Pressione Enter para fechar esta Janela');
```

```
Repeat
      TEnt:=Readkey;
Until TEnt=#13;

     If Ap=1 Then Begin
        SetActivePage(1);
        SetVisualPage(0);
     End
     Else Begin
          SetActivePage(0);
          SetVisualPage(1);
     End;
     FilTog:=False;
End;

{*************************************************************************}

Procedure ModoPlot;
{Alterna o modo normal e operações lógicas}

Begin
     If Oper='NORMAL' Then Oper:='AND'
     Else If Oper='AND' Then Oper:='OR'
     Else If Oper='OR' Then Oper:='XOR'
     Else If Oper='XOR' Then Oper:='NORMAL';
End;

{*************************************************************************}

Procedure VerifTecla;
{Verifica que tecla foi pressionada}


Procedure VerifASCII;
{Verifica teclas extendidas}

Var TecASC:Char; {Caracter lido do teclado - teclas extendidas}

Begin
     TecASC:=Readkey;
     If TecASC=#59 Then Help:=True {F1}
     Else If TecASC=#60 Then ZoomHabil:=True {F2}
     Else If TecASC=#61 Then FilTog:=True {F3}
     Else If TecASC=#62 Then Iniciar:=True {F4}
     Else If (TecASC=#63) And (Arq2Habil)
             And (Escala1=Escala2) Then ModoPlot {F5}
     Else If TecASC=#64 Then BMPWr:=True; {F6}
End;

Begin
     Tecla:=Readkey;
     If Tecla=#27 Then Begin {Esc}
        CloseGraph;
        FimProg;
     End
     Else If Tecla=#0 Then VerifASCII;
End;

{*************************************************************************}

Procedure IndicEscala;
{Indica a escala horizontal}

Var FatorStr:String; {Fator de zoom horizontal}

Begin
     SetFillStyle(0,0);
     Bar(110,15,310,25);
     SetColor(LightRed);
     SetTextJustify (LeftText,CenterText);
     SetTextStyle (SmallFont,HorizDir,4);
     OutTextXy (3+TextWidth('Escala Horizontal: '),20, Escala1+' uS/div');
     SetColor(White);
     Str(Fator:2,FatorStr);
     OutTextxy (265,20,'Zoom:'+FatorStr);

     If Arq2Habil Then Begin
        SetColor(LightGreen);
        OutTextXy (3+TextWidth('Escala Horizontal: '+Escala1+' uS/div'),20,
                   ' '+Escala2+' uS/div');
     End;
End;

{*************************************************************************}
```

```
Procedure MostraDesc;
{Mostra as descrições do conj. e das formas de onda}

Begin
     SetColor(LightRed);
     SetTextJustify (CenterText,CenterText);
     SetTextStyle (SmallFont,HorizDir,4);
     OutTextXy (385,20,Desc1);
     OutTextXy (33,50,DescA1);
     OutTextXy (33,90,DescB1);
     OutTextXy (33,130,DescC1);
     OutTextXy (33,170,DescD1);
     OutTextXy (33,210,DescE1);
     OutTextXy (33,250,DescF1);
     OutTextXy (33,290,DescG1);
     OutTextXy (33,330,DescH1);

     If Arq2Habil Then Begin
        SetColor(LightGreen);
        OutTextXy (455,20,Desc2);
        OutTextXy (33,60,DescA2);
        OutTextXy (33,100,DescB2);
        OutTextXy (33,140,DescC2);
        OutTextXy (33,180,DescD2);
        OutTextXy (33,220,DescE2);
        OutTextXy (33,260,DescF2);
        OutTextXy (33,300,DescG2);
        OutTextXy (33,340,DescH2);
     End;
End;

{*************************************************************************}

Procedure TelaPrinc;
{Desenha a tela principal do programa}

Begin
     {Título}
     SetColor (White);
     SetTextJustify (CenterText,CenterText);
     SetTextStyle (Smallfont,HorizDir,5);
     SetFillStyle (1,Blue);
     Bar (0,0,639,12);
     OutTextxy (320,4,'Comparador de Formas de Onda');

     {Grid}
     SetColor(DarkGray);
     X:=64;
     Repeat
           Line (X,29,X,349);
           X:=X+2;
     Until X>574;
     Setcolor(LightGray);
     X:=64;
     Repeat
           Line (X,29,X,349);
           X:=X+10;
     Until X>574;

     {Linhas Horizontais}
     SetColor(White);
     Y:=29;
     Repeat
           Line (3,Y,635,Y);
           Y:=Y+40;
     Until Y>349;

     {Linhas Verticais}
     Setcolor(White);
     Line (3,29,3,349);
     Line (63,29,63,349);
     Line (575,29,575,349);
     Line (635,29,635,349);

     If Keypressed Then VerifTecla;

     {Nomes dos canais}
     SetTextStyle (SmallFont,HorizDir,6);
     SetColor (White);
     SetTextJustify (CenterText,CenterText);
     OutTextXy (33,36,'A');
     OutTextXy (33,76,'B');
     OutTextXy (33,116,'C');
```

```
     OutTextXy (33,156,'D');
     OutTextXy (33,196,'E');
     OutTextXy (33,236,'F');
     OutTextXy (33,276,'G');
     OutTextXy (33,316,'H');

     {Indicador de escala horizontal , Help e Descrições}
     SetColor (White);
     SetTextJustify (LeftText,CenterText);
     SetTextStyle (SmallFont,HorizDir,4);
     OutTextXy (3,20,'Escala Horizontal: ');
     SetTextJustify (RightText,CenterText);
     OutTextxy (639,20,'Pressione F1 para Ajuda ');
     IndicEscala;
     MostraDesc;

     If Keypressed Then VerifTecla;

     {Nomes dos Frequencímetros dos Canais}
     SetColor (White);
     SetTextJustify (CenterText,CenterText);
     SetTextStyle (SmallFont,HorizDir,2);
     Y:=35;
     Repeat
           OutTextXy (605,Y,'Frequencia');
           Y:=Y+40;
     Until Y>315;
     SetTextStyle (SmallFont,HorizDir,4);
     Y:=63;
     Repeat
           OutTextXy (605,Y,'KHz');
           Y:=Y+40;
     Until Y>345;
End;

{*************************************************************************}

Procedure Plotar;
{Plota as formas de onda na tela}

Var CorP:Integer; {Cor do Pixel}
    YPlota:Integer; {Coordenada Y para plotagem}
    CanalByte:Integer; {Canal a ser plotado}

Procedure Plota1;
{Plota a 1a Forma-de-onda}

Var IndDado:Integer; {Index do Array Dado}
    NVezes:Integer;  {Número de vezes a se repetir o mesmo dado (no Zoom)}

Begin
        For IndDado:=1 To Round(510/Fator) Do Begin
            For NVezes:=1 To Fator Do Begin

                PutPixel ((NVezes-Fator)+(IndDado*Fator)+64,
                          (((Ord(Dado1[IndDado]) And Eleva(2,CanalByte))
                          Shr CanalByte) Xor 1)*32+YPlota+3,CorP);

                PutPixel ((NVezes-Fator)+(IndDado*Fator)+64,
                          (((Ord(Dado1[IndDado]) And Eleva(2,CanalByte))
                          Shr CanalByte) Xor 1)*34+YPlota+2,CorP);

                If Keypressed Then VerifTecla;
            End;
        End;
End;

Procedure Plota2;
{Plota a 2a Forma-de-onda}

Var IndDado:Integer; {Index do Array Dado}
    NVezes:Integer; {Número de vezes a se repetir o mesmo dado (no Zoom)}

Begin
        For IndDado:=1 To Round(510/Fator) Do Begin
            For NVezes:=1 To Fator Do Begin

                PutPixel ((NVezes-Fator)+(IndDado*Fator)+64,
                          (((Ord(Dado2[IndDado]) And Eleva(2,CanalByte))
                          Shr CanalByte) Xor 1)*26+YPlota+6,CorP);

                PutPixel ((NVezes-Fator)+(IndDado*Fator)+64,
                          (((Ord(Dado2[IndDado]) And Eleva(2,CanalByte))
                          Shr CanalByte) Xor 1)*28+YPlota+5,CorP);
```

```
                If Keypressed Then VerifTecla;
            End;
        End;
End;

Begin
     If Oper='NORMAL' Then CorP:=LightRed
     Else CorP:=Yellow;
     If A1 Then Begin
        YPlota:=30;
        CanalByte:=0;
        Plota1;
     End;
     If B1 Then Begin
        CanalByte:=1;
        YPlota:=70;
        Plota1;
     End;
     If C1 Then Begin
        YPlota:=110;
        CanalByte:=2;
        Plota1;
     End;
     If D1 Then Begin
        CanalByte:=3;
        YPlota:=150;
        Plota1;
     End;
     If E1 Then Begin
        YPlota:=190;
        CanalByte:=4;
        Plota1;
     End;


     If F1 Then Begin
        YPlota:=230;
        CanalByte:=5;
        Plota1;
     End;
     If G1 Then Begin
        YPlota:=270;
        CanalByte:=6;
        Plota1;
     End;
     If H1 Then Begin
        CanalByte:=7;
        YPlota:=310;
        Plota1;
     End;

   If (Arq2Habil) And (Oper='NORMAL') Then Begin
     CorP:=LightGreen;
     If A2 Then Begin
        YPlota:=30;
        CanalByte:=0;
        Plota2;
     End;
     If B2 Then Begin
        CanalByte:=1;
        YPlota:=70;
        Plota2;
     End;
     If C2 Then Begin
        YPlota:=110;
        CanalByte:=2;
        Plota2;
     End;
     If D2 Then Begin
        CanalByte:=3;
        YPlota:=150;
        Plota2;
     End;
     If E2 Then Begin
        YPlota:=190;
        CanalByte:=4;
        Plota2;
     End;
     If F2 Then Begin
        YPlota:=230;
        CanalByte:=5;
        Plota2;
     End;
```

```
     If G2 Then Begin
        YPlota:=270;
        CanalByte:=6;
        Plota2;
     End;
     If H2 Then Begin
        CanalByte:=7;
        YPlota:=310;
        Plota2;
     End;
   End;
End;

{*************************************************************************}

Procedure IndicaFreq (NumForm:Integer;
                      FA:Real;FB:Real;FC:Real;FD:Real;FE:Real;FF:Real;FG:Real;
                      FH:Real);
{Mostra os valores de frequência dos canais ativos
  FA a FH = Valores medidos pelo frequencímetro
  NumForm = 1 ou 2 forma(s)-de-onda}

Var Valor:String; {Frequência convertida em String}
    CorP:Integer; {Cor do frequencímetro}

Begin
   If Oper='NORMAL' Then CorP:=LightRed
   Else CorP:=Yellow;
   If NumForm=1 Then Begin
     SetTextJustify (CenterText,CenterText);
     SetTextStyle (SmallFont,HorizDir,4);
     SetColor (CorP);
     If A1=True Then Begin
        Str (FA:8:3,Valor);
        OutTextXy (605,45,Valor);
     End;

     If B1=True Then Begin
        Str (FB:8:3,Valor);
        OutTextXy (605,85,Valor);
     End;
     If C1=True Then Begin
        Str (FC:8:3,Valor);
        OutTextXy (605,125,Valor);
     End;
     If D1=True Then Begin
        Str (FD:8:3,Valor);
        OutTextXy (605,165,Valor);
     End;
     If E1=True Then Begin
        Str (FE:8:3,Valor);
        OutTextXy (605,205,Valor);
     End;
     If F1=True Then Begin
        Str (FF:8:3,Valor);
        OutTextXy (605,245,Valor);
     End;
     If G1=True Then Begin
        Str (FG:8:3,Valor);
        OutTextXy (605,285,Valor);
     End;
     If H1=True Then Begin
        Str (FH:8:3,Valor);
        OutTextXy (605,325,Valor);
     End;

     If Keypressed Then VerifTecla;

   End
   Else If (NumForm=2) And (Oper='NORMAL') Then Begin
     SetTextJustify (CenterText,CenterText);
     SetTextStyle (SmallFont,HorizDir,4);
     SetColor (LightGreen);
     If A2=True Then Begin
        Str (FA:8:3,Valor);
        OutTextXy (605,54,Valor);
     End;
     If B2=True Then Begin
        Str (FB:8:3,Valor);
        OutTextXy (605,94,Valor);
     End;
     If C2=True Then Begin
        Str (FC:8:3,Valor);
        OutTextXy (605,134,Valor);
```

```
     End;
     If D2=True Then Begin
        Str (FD:8:3,Valor);
        OutTextXy (605,174,Valor);
     End;
     If E2=True Then Begin
        Str (FE:8:3,Valor);
        OutTextXy (605,214,Valor);
     End;
     If F2=True Then Begin
        Str (FF:8:3,Valor);
        OutTextXy (605,254,Valor);
     End;
     If G2=True Then Begin
        Str (FG:8:3,Valor);
        OutTextXy (605,294,Valor);
     End;
     If H2=True Then Begin
        Str (FH:8:3,Valor);
        OutTextXy (605,334,Valor);
     End;
   End;
End;

{*************************************************************************}

Procedure Frequenc;
{Calcula as frequências dos canais}

Var FrA,FrB,FrC,FrD,FrE,FrF,FrG,FrH:Real; {Armazenam os valores de frequência}
    Pontos:Integer; {Pontos em mesmo estado lógico (Hi ou Lo)}
    Dc:Boolean; {Indica Tensão Contínua Constante}
    IndDado:Integer; {Index do Array Dado}
    Escala:Real; {Escala Horizontal}
    Cod:Integer; {Indica o resultado da conversão Freq => String}



Function Eleva2 (Base:Integer;Expoente:Integer):Integer;
{Eleva uma Base a um Expoente
 Verifica se alguma tecla foi pressionada}

Var Vezes:Integer; {Número de vezes que a base deve ser multiplicada}
    ValorExp:Integer; {Resultado da operação}

Begin
     ValorExp:=1;
     If Expoente=0 Then Eleva2:=1
     Else Begin
          For Vezes:=1 To Expoente Do ValorExp:=ValorExp*Base;
          Eleva2:=ValorExp;
     End;

     If Keypressed Then VerifTecla;

End;

Procedure MedeFreq1 (CanBit:Integer);
{Mede a frequência dos canais da 1a forma-de-onda
  CanBit = Indica qual o canal}

Begin
     IndDado:=0;
     Pontos:=0;
     Dc:=False;
Repeat
      IndDado:=IndDado+1;
Until ((Ord(Dado1[IndDado]) And Eleva2(2,CanBit)) Shr CanBit =1)
       Or (IndDado=511);

Repeat
      IndDado:=IndDado+1;
Until ((Ord(Dado1[IndDado]) And Eleva2(2,CanBit)) Shr CanBit =0)
       Or (IndDado=511);
      If IndDado>=511 Then Dc:=True;

Repeat
      IndDado:=IndDado+1;
      Pontos:=Pontos+1;
Until ((Ord(Dado1[IndDado]) And Eleva2(2,Canbit)) Shr CanBit =1)
       Or (IndDado=511);
      If IndDado>=511 Then Dc:=True;

Repeat
```

```
      IndDado:=IndDado+1;
      Pontos:=Pontos+1;
Until ((Ord(Dado1[IndDado]) And Eleva2(2,CanBit)) Shr CanBit =0)
       Or (IndDado=511);
      If IndDado>=511 Then Dc:=True;

End;

Procedure MedeFreq2 (CanBit:Integer);
{Mede a frequência dos canais da 2a forma-de-onda
  CanBit = Indica qual o canal}

Begin
     IndDado:=0;
     Pontos:=0;
     Dc:=False;
Repeat
      IndDado:=IndDado+1;
Until ((Ord(Dado2[IndDado]) And Eleva2(2,CanBit)) Shr CanBit =1)
       Or (IndDado=511);

Repeat
      IndDado:=IndDado+1;
Until ((Ord(Dado2[IndDado]) And Eleva2(2,CanBit)) Shr CanBit =0)
       Or (IndDado=511);
      If IndDado>=511 Then Dc:=True;

Repeat
      IndDado:=IndDado+1;
      Pontos:=Pontos+1;
Until ((Ord(Dado2[IndDado]) And Eleva2(2,Canbit)) Shr CanBit =1)
       Or (IndDado=511);
      If IndDado>=511 Then Dc:=True;

Repeat
      IndDado:=IndDado+1;
      Pontos:=Pontos+1;
Until ((Ord(Dado2[IndDado]) And Eleva2(2,CanBit)) Shr CanBit =0)
       Or (IndDado=511);
      If IndDado>=511 Then Dc:=True;

End;

Begin
     Val(Escala1,Escala,Cod);
     MedeFreq1 (0);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrA:= 0
     Else FrA:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq1 (1);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrB:= 0
     Else FrB:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq1 (2);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrC:= 0
     Else FrC:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq1 (3);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrD:= 0
     Else FrD:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq1 (4);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrE:= 0
     Else FrE:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq1 (5);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrF:= 0
     Else FrF:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq1 (6);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrG:= 0
     Else FrG:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq1 (7);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrH:= 0
     Else FrH:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;
```

```
     IndicaFreq (1,FrA,FrB,FrC,FrD,FrE,FrF,FrG,FrH);

     If Keypressed Then VerifTecla;

  If Arq2Habil Then Begin
     Val(Escala2,Escala,Cod);
     MedeFreq2 (0);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrA:= 0
     Else FrA:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq2 (1);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrB:= 0
     Else FrB:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq2 (2);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrC:= 0
     Else FrC:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq2 (3);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrD:= 0
     Else FrD:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq2 (4);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrE:= 0
     Else FrE:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq2 (5);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrF:= 0
     Else FrF:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;



     MedeFreq2 (6);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrG:= 0
     Else FrG:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;

     MedeFreq2 (7);
     If Cod <> 0 Then Dc:=True;
     If Dc Then FrH:= 0
     Else FrH:=(1/(Pontos*(Escala*1e-7))/1e3)/Fator;
  End;

     IndicaFreq (2,FrA,FrB,FrC,FrD,FrE,FrF,FrG,FrH);
End;

{*************************************************************************}

Procedure GetTela;
{Lê a tela e grava no BMP}

Var CorBy,Meio1,Meio2:Word; {Cor do pixel lido da tela e os nibbles que o
                             representam}
    PixByte:Char; {Caracter a ser gravado no arquivo BMP}
    BlueDt:Boolean; {Indica a cor azul: utilizada na correção de cores}

Procedure CorCorrect;
{Correção de cores para a palette do BMP}

Begin
     If CorBy=7 Then CorBy:=0
     Else If CorBy=12 Then CorBy:=8
     Else If CorBy=0 Then CorBy:=7
     Else If (CorBy=15) And (BlueDt=False) Then CorBy:=0
     Else If CorBy=10 Then CorBy:=6
     Else If CorBy=8 Then CorBy:=3
     Else If CorBy=14 Then CorBy:=10;
End;

Procedure Porcento;
{Indica o progresso da gravação em %}

Var Porc,Pa,Pb,Pc,Pd:Extended; {Utilizadas no cálculo do percentual}
    PorcStr:String; {Percentual em formato String}

Begin
```

```
     Pa:=Y-349;
     Pb:=Pa*(-640);
     Pc:=Pb+1+X;
     Pd:=640*350;
     Porc:=(Pc/Pd)*100;
     If Porc-Trunc(Porc)<0.001 Then Begin
        Str(Porc:3:0,PorcStr);
        SetActivePage(0);
        Bar(220,160,270,180);
        OutTextXy(320,170,PorcStr+' % Concluido ');
        If PorcStr='100' Then Begin
           OutTextxy(320,220,'Operacao concluida com sucesso!');
           Delay(1500);
        End;
        SetActivePage(1);
     End;
End;

Begin
     X:=0; Y:=349;
     BlueDt:=False;
     SetColor (White);
     SetTextJustify (CenterText,CenterText);
     SetActivePage(0);
     SetTextStyle (Smallfont,HorizDir,5);
     SetFillStyle (1,Cyan);
     Bar (0,0,639,349);
     SetFillStyle (1,Blue);
     Bar (0,0,639,12);
     OutTextxy (320,4,'Comparador de Formas de Onda');
     SetColor(Yellow);
     SetTextStyle(SmallFont,HorizDir,7);
     OutTextXy(320,100,'Exportando para BitMap. Aguarde...');
     SetActivePage(1);
     SetFillStyle(1,Cyan);

Repeat
     PorCento;
     CorBy:=GetPixel(X,Y);
     If CorBy=1 Then BlueDt:=True;
     CorCorrect;
     Meio1:=CorBy;
     X:=X+1;
     CorBy:=GetPixel(X,Y);
     If CorBy=1 Then BlueDt:=True;
     CorCorrect;
     Meio2:=CorBy;
     X:=X+1;
     PixByte:=Chr((Meio1 Shl 4) Or (Meio2));
     Write (ArqBMP,PixByte);
     If X=640 Then Begin
        X:=0;
        Y:=Y-1;
     End;
Until (X=638) And (Y=0);
     CorBy:=GetPixel(X,Y);
     CorCorrect;
     Meio1:=CorBy;
     X:=X+1;
     CorBy:=GetPixel(X,Y);
     CorCorrect;
     Meio2:=CorBy;
     X:=X+1;
     PorCento;
     PixByte:=Chr((Meio1 Shl 4) Or (Meio2));
     Write (ArqBMP,PixByte);
     Close(ArqBMP);
     PutBMP:=False;
     SetActivePage(0);
     SetFillStyle(0,0);
     Bar(0,0,639,349);
     SetActivePage(1);
End;

{*************************************************************************}

Procedure AlteraPlot;
{Altera o modo de Plotagem Normal ou operação}

Var PSal:Integer; {Index do Array DadoSalva}

Begin
     If Oper<>'NORMAL' Then Begin
        For PSal:=0 To 511 Do Begin
```

```
            If Oper='AND' Then Begin
               Dado1[PSal]:=Chr((Ord(DadoSalva[PSal])) And (Ord(Dado2[PSal])));
            End
            Else If Oper='OR' Then Begin
               Dado1[PSal]:=Chr(Ord(DadoSalva[PSal]) Or Ord(Dado2[PSal]));
            End
            Else If Oper='XOR' Then Begin
               Dado1[PSal]:=Chr(Ord(DadoSalva[PSal]) Xor Ord(Dado2[PSal]));
            End;
        End;
     End
     Else Begin
        For PSal:=0 To 511 Do Begin
            Dado1[PSal]:=DadoSalva[PSal];
        End;
     End;
     SetFillStyle(0,0);
     Bar(0,30,639,349);
End;

{*************************************************************************}

Procedure Operation;
{Mostra o modo atual: Normal ou operação}

Begin
     SetFillStyle(0,0);
     SetColor(Yellow);
     SetTextJustify (CenterText,CenterText);
     SetTextStyle (SmallFont,HorizDir,4);
     Bar (310,15,350,25);
     OutTextXy (330,20,Oper);
End;

{*************************************************************************}

Procedure Ajuda;
{Mostra a Ajuda}

Var PPPT:Char; {Armazena o caracter lido do teclado}


Begin
     If Ap=1 Then Begin
        SetActivePage(0);
        Ap:=0;
     End
     Else Begin
          SetActivePage(1);
          Ap:=1;
     End;
     SetTextJustify(CenterText,CenterText);
     SetTextStyle(SmallFont,HorizDir,4);
     SetFillStyle(1,Black);
     Bar (130,105,530,265);
     SetFillStyle(1,White);
     Bar (120,95,520,255);
     SetColor(LightGray);
     SetLineStyle(0,1,3);
     Line (120,95,120,255);
     Line (520,95,520,255);
     Line (120,95,520,95);
     Line (120,255,520,255);
     SetLineStyle (0,0,0);
     SetFillStyle(1,Blue);
     Bar (122,97,518,110);
     SetColor(Black);
     Line (119,94,119,256);
     Line (521,94,521,256);
     Line (119,94,521,94);
     Line (119,256,521,256);
     SetColor (White);
     OutTextXy (320,103,'Ajuda do Comparador de Formas de Onda');
     SetColor(Magenta);
     SetTextStyle (SmallFont,HorizDir,5);
     OutTextxy(320,120,'Os PEP Legais');
     SetTextStyle (SmallFont,HorizDir,4);
     SetColor(Black);
     OutTextxy(320,140,'Esc = Sai do Programa');
     OutTextxy(320,152,'F1 = Ajuda');
     OutTextxy(320,164,'F2 = Zoom (1, 2, 5, 10)');
     OutTextxy(320,176,'F3 = Informacao de Arquivos');
     OutTextxy(320,188,'F4 = Retorna ao Menu Inicial');
     OutTextxy(320,200,'F5 = Alterna o Modo (Normal, AND, OR, XOR)');
```

```
     OutTextxy(320,212,'F6 = Exporta para BitMap');
     OutTextxy(320,240,'Pressione Enter para Fechar esta Janela');

Repeat
      PPPT:=Readkey;
Until PPPT=#13;

     Help:=False;
     If Ap=1 Then Begin
        SetActivePage(0);
        Ap:=0;
     End
     Else Begin
          SetActivePage(1);
          Ap:=1;
     End;
End;

{*************************************************************************}

Procedure Zoom;
{Realiza o zoom de 1X, 2X, 5X ou 10X}

Var EscaInt1,EscaInt2:Real; {Armazenam a escala horizontal em formato real}
    Code:Integer; {Resultado da conversão Escala => Real}
    CharUlt:String; {Último caracter da conversão Real => Escala}

Procedure Conj1;
{Zoom para 1a forma-de-onda}

Begin
     Val(Escala1,EscaInt1,Code);
     If Code <> 0 Then EscaInt1:=1;
     If Fator=1 Then Begin
        EscaInt1:=EscaInt1/2;
     End
     Else If Fator=2 Then Begin
          EscaInt1:=(EscaInt1*2)/5;
     End
     Else If Fator=5 Then Begin
          EscaInt1:=(EscaInt1*5)/10;
     End

     Else If Fator=10 Then Begin
          EscaInt1:=EscaInt1*10;
     End;

     If Escala1='  EXT' Then Escala1:='  EXT'
     Else Str(EscaInt1:5:1,Escala1);

     CharUlt:=Copy(Escala1,Length(Escala1),1);
     If CharUlt='.' Then Delete(Escala1,Length(Escala1),1);
     CharUlt:=Copy(Escala1,Length(Escala1)-1,2);
     If CharUlt='.0' Then Delete(Escala1,Length(Escala1)-1,2);
End;

Procedure Conj2;
{Zoom para a 2a forma-de-onda}

Begin
     Val(Escala2,EscaInt2,Code);
     If Code <> 0 Then EscaInt2:=1;
     If Fator=1 Then Begin
        EscaInt2:=EscaInt2/2;
     End
     Else If Fator=2 Then Begin
          EscaInt2:=(EscaInt2*2)/5;
     End
     Else If Fator=5 Then Begin
          EscaInt2:=(EscaInt2*5)/10;
     End
     Else If Fator=10 Then Begin
          EscaInt2:=EscaInt2*10;
     End;

     If Escala2='  EXT' Then Escala2:='  EXT'
     Else Str(EscaInt2:5:1,Escala2);

     CharUlt:=Copy(Escala2,Length(Escala2),1);
     If CharUlt='.' Then Delete(Escala2,Length(Escala2),1);
     CharUlt:=Copy(Escala2,Length(Escala2)-1,2);
     If CharUlt='.0' Then Delete(Escala2,Length(Escala2)-1,2);
End;
```

```
Begin
     Conj1;
     If Arq2Habil Then Conj2;

     If Fator=1 Then Fator:=2
     Else If Fator=2 Then Fator:=5
     Else If Fator=5 Then Fator:=10
     Else If Fator=10 Then Fator:=1;

     ZoomHabil:=False;
End;

{*************************************************************************}

Procedure InitDefault;
{Inicia as variáveis}

Begin
     Iniciar:=False;
     Help:=False;
     BMPWr:=False;
     FilTog:=False;
     PutBMP:=False;
     Oper:='NORMAL';
     Fator:=1;
End;

{*************************************************************************}

Begin
     VerifArq;
     ModoGrafico;
     Apres;
Inicio:
     Win1;
     Win2;
     Win3;
     LeArqs;
     For X:=0 To 511 Do Begin
         DadoSalva[X]:=Dado1[X];
     End;
     InitDefault;
     ModoGrafico;
     SetActivePage(1);
     SetVisualPage(0);
     Ap:=1;
Repeat
     If ZoomHabil Then Zoom;
     If BMPWr Then GravaBMP;
     If FilTog Then TogFName;
     If Help Then Ajuda;
     If Iniciar Then Begin
        CloseGraph;
        Goto Inicio;
     End;
     Delay(10);
     AlteraPlot;
     TelaPrinc;
     Plotar;
     Frequenc;
     Operation;
     If PutBMP Then GetTela;
     If Ap=1 Then Begin
        SetActivePage(0);
        SetVisualPage(1);
        Ap:=0;
     End
     Else Begin
        SetActivePage(1);
        SetVisualPage(0);
        Ap:=1;
     End;
     Delay(10);
     If Keypressed Then VerifTecla;
Until Tecla=#27;
     CloseGraph;
     FimProg;
End.

{*************************************************************************}
{
        * Estrutura do arquivo .OSC *

Bytes 0 a 2 = 'PEP'
```

```
Bytes 3 a 12 = Descrição do Conjunto
Bytes 13 a 22 = Descrição do Canal A
Bytes 23 a 32 = Descrição do Canal B
Bytes 33 a 42 = Descrição do Canal C
Bytes 43 a 52 = Descrição do Canal D
Bytes 53 a 62 = Descrição do Canal E
Bytes 63 a 72 = Descrição do Canal F
Bytes 73 a 82 = Descrição do Canal G
Bytes 83 a 92 = Descrição do Canal H
Byte 93 = 00H se Canal A = OFF / 01H se Canal A = ON
Byte 94 = 00H se Canal B = OFF / 01H se Canal B = ON
Byte 95 = 00H se Canal C = OFF / 01H se Canal C = ON
Byte 96 = 00H se Canal D = OFF / 01H se Canal D = ON
Byte 97 = 00H se Canal E = OFF / 01H se Canal E = ON
Byte 98 = 00H se Canal F = OFF / 01H se Canal F = ON
Byte 99 = 00H se Canal G = OFF / 01H se Canal G = ON
Byte 100 = 00H se Canal H = OFF / 01H se Canal H = ON
Byte 101 = Escala Horizontal:
           00H = 1uS/div
           01H = 10uS/div
           02H = 100uS/div
           03H = 1.000uS/div
           04H = 10.000uS/div
           05H = Externo
Bytes 102 a 613 = Dados (0 a 511)

}
{******************************************************************************}
{
        * Estrutura do arquivo .BMP *

Bytes 0 a 17 = Identificadores do BitMap
Bytes 18 e 19 = Largura em Pixels
Byte 22 e 23 = Altura em Pixels
Byte 54 a 56 = B,G,R da Cor 0 - Palette
Byte 58 a 60 = B,G,R da Cor 1 - Palette
Byte 62 a 64 = B,G,R da Cor 2 - Palette
Byte 66 a 68 = B,G,R da Cor 3 - Palette
Byte 70 a 72 = B,G,R da Cor 4 - Palette
Byte 74 a 76 = B,G,R da Cor 5 - Palette
Byte 78 a 80 = B,G,R da Cor 6 - Palette
Byte 82 a 84 = B,G,R da Cor 7 - Palette
Byte 86 a 88 = B,G,R da Cor 8 - Palette
Byte 90 a 92 = B,G,R da Cor 9 - Palette
Byte 94 a 96 = B,G,R da Cor 10 - Palette
Byte 98 a 100 = B,G,R da Cor 11 - Palette
Byte 102 a 104 = B,G,R da Cor 12 - Palette
Byte 106 a 108 = B,G,R da Cor 13 - Palette
Byte 110 a 112 = B,G,R da Cor 14 - Palette
Byte 114 a 116 = B,G,R da Cor 15 - Palette
Byte 118 = Inicio do BMP (Cada 1/2 Byte é a cor de um Pixel)

}
{******************************************************************************}
```

Configurador (CONFIG.PAS)

```
{$A-,B-,D-,E+,F-,G-,I+,L-,N+,Q-,R-,S+,X+,V+}
{Diretivas de compilação}

{******************************************************************************}

Program Configurador;

{******************************************************************************}
{

        Programa para configurar o Software Principal do Analisador Lógico

                "Os PEP Legais" - LAO - 4oR -Eletrônica


}
{******************************************************************************}

{       Créditos:
                Luiz Eduardo    - N°19
                Marcos Paulo    - N°23
                Rafael Abolafio - N°26
                Robson Martins  - N°30

                4°R - Eletrônica - 1997

             Liceu de Artes e Ofícios de São Paulo
```

```
              Prof: Marco Antônio Togniazzolo
}
{*************************************************************************}

Uses Crt,Dos,PEPUnit;
{Units a serem usadas}

{*************************************************************************}

Function Hex2Dec(H:String):Word;
{Entrada: String = Hexadecimal de 000 a FFF}
{Saida: Word = Decimal de 0 a 4095}

Var WordValue:Word; {Word que varia de 0 a 4095}
    HexV:String; {Valor em hexadecimal}

Begin
     WordValue:=0;
Repeat
     HexV:=DecToHex(WordValue,True);
     Delete(HexV,1,1);
     If H=HexV Then Break;
     WordValue:=WordValue+1;
Until WordValue>$FFF;
     If WordValue>$FFF Then Hex2Dec:=0
     Else Hex2Dec:=WordValue;
End;

{*************************************************************************}

Procedure ApagaTela;
{Apaga a tela}

Begin
     Window (1,1,80,25);
     TextColor (LightGray);
     TextBackGround (Black);
     Clrscr;
End;

{*************************************************************************}

Procedure Fim;
{Finaliza o programa}

Begin
     ApagaTela;
     Writeln;
     Halt(0);
End;

{*************************************************************************}

Procedure Win1;
{Tela inicial do programa}

Var Tecla:Char; {Armazena o caracter lido do teclado}
    PosStat:Integer; {Utilizada para apagar linhas da tela}

Begin
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackground (Cyan);
     CenterTxt (8,'PROGRAMA DE CONFIGURAÇÃO');
     CenterTxt (9,'DO');
     CenterTxt (10,'ANALISADOR LÓGICO');
     TextColor (White);
     CenterTxt (13,'"Os PEP Legais"');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Pressione Enter para Continuar ou Esc para Sair');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

Repeat
      Tecla:=Readkey;
      If Tecla=#27 Then Fim;
Until Tecla=#13;

     Window (1,1,80,25);
     Textcolor (Black);
     TextBackground (White);
```

```
     For PosStat:=2 To 60 Do Begin
         Gotoxy (PosStat,24);
         Write (' ');
     End;
End;

{*************************************************************************}

Procedure Win2;
{Telas de Configurações}

Var Tecla:Char; {Armazena o caracter lido do teclado}
    PosStat:Integer; {Utilizada para apagar linhas da tela}
    Ender:Word; {Endereço da LPT}
    VerifIntFace:Byte; {Constante de Verificação de Interface}
    VerifDC:Byte; {Constante de SyncOff para DC}
    F:Text; {Representa o arquivo de configuração}
    Io:Integer; {Resultado da última operação de I/O}

Procedure AutoDetect;
{Realiza a detecção da porta na qual a Interface está conectada}

Var PP:Integer; {Utilizada para apagar linhas da tela}
    Tec:Char; {Armazena o caracter lido do teclado}
    EndOk:Boolean; {Indica endereço correto}
    EndLst:Text; {Representa a lista de endereços para detecção}
    EndStr:String; {Endereço em formato String}

Procedure ErroArq;
{Mensagem de Erro: Falta PEPLEG.DAT}

Begin
        ClrScr;
        TextColor (Blue);
        Writeln ('Config do Analisador Lógico - Mensagem de Erro');
        Writeln;
        TextColor (LightRed);
        Writeln ('Não existe o arquivo "PEPLEG.DAT"');
        Writeln ('neste diretório!');
        Writeln;
        Writeln ('Verifique qual o diretório corrente!');
        Writeln ('Após isso, execute novamente este programa.');
        TextColor (LightGray);
        Writeln;
        Writeln ('Caso não consiga solucionar este problema,');
        Writeln ('entre em contato com:');
        Writeln;
        Textcolor (White);
        Writeln ('"Os PEP Legais" - LAO - 4oR Eletrônica');
        Textcolor (LightGray);
        Writeln;
        Halt(2);
End;

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (8,'Conecte a Interface na porta a ser');
     CenterTxt (9,'utilizada e acione sua alimentação.');
     CenterTxt (11,'Desconecte agora todos os periféricos');
     CenterTxt (12,'normalmente conectados nas Portas LPT');
     CenterTxt (13,'(Como Impressoras, ZipDrive ou outros).');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Pressione Enter quando estiver pronto');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
Repeat
      Tec:=Readkey;
Until Tec=#13;
```

```
     Assign (EndLst,'PEPLEG.DAT');
{$I-}
     Reset (EndLst);
{$I+}
     If IoResult<>0 Then ErroArq;

     Ender:=$378;
     EndOk:=False;

Repeat
     Readln(EndLst,EndStr);
     Port[Hex2Dec(EndStr)]:=$22; {0010 0010 - Reset/RD/ClrSync}
     Delay (600);

     Port[Hex2Dec(EndStr)]:=$22; {0010 0010 - Reset/RD/ClrSync}
     If (Port[Hex2Dec(EndStr)] And $20) Shr 5 = 0 Then Begin
        Ender:=Hex2Dec(EndStr); EndOk:=True;
     End;
     Port[Hex2Dec(EndStr)]:=$00; {Desliga a Porta Paralela}
Until (Eof(EndLst)) Or (EndOk);

     Close(EndLst);
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     If EndOk Then CenterTxt (10,'Interface encontrada em '+
                                 DecToHex(Ender,True)+'H')
     Else  Begin
           CenterTxt (9,'A Interface não foi encontrada!');
           CenterTxt (11,'Utilizando o endereço Default: 378H');
     End;
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Pressione Enter para continuar');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
Repeat
      Tec:=Readkey;
Until Tec=#13;
End;

Procedure Espec;
{O Usuário deve especificar o endereço da porta}

Var PQ:Integer; {Utilizada para apagar linhas da tela}
    EndHex:String; {Endereço em Hexadecimal}

Function UpStr(StrIn:String):String;
{Converte String [a..f] em maiúsculas}

Var Copint:Integer; {Varia de 1 até o número de caracteres de StrIn}
    US:String; {Resultado de UpString}

Begin
     US:='';
     For Copint:=1 To Length(StrIn) Do Begin
         If Copy(StrIn,Copint,1)='a' Then US:=US+'A'
         Else If Copy(StrIn,Copint,1)='b' Then US:=US+'B'
         Else If Copy(StrIn,Copint,1)='c' Then US:=US+'C'
         Else If Copy(StrIn,Copint,1)='d' Then US:=US+'D'
         Else If Copy(StrIn,Copint,1)='e' Then US:=US+'E'
         Else If Copy(StrIn,Copint,1)='f' Then US:=US+'F'
         Else US:=US+Copy(StrIn,Copint,1);
     End;
     UpStr:=US;
End;

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PQ:=2 To 60 Do Begin
```

```
         Gotoxy (PQ,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (9,'Digite o endereço da porta a ser utilizada');
     CenterTxt (10,'com a Interface do Analisador Lógico:');
     CenterTxt (11,'(Em Hexadecimal)');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Endereços Válidos: 200H a 3FFH');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
     TextColor (Yellow);
     TextBackGround (Cyan);
Repeat
     CenterTxt (13,'                                                ');
     CenterTxt (13,'');
     Readln (EndHex);
     Ender:=Hex2Dec(UpStr(EndHex));
Until (Ender>=$200) And (Ender<=$3FF);
End;

Begin
     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (9,'Deseja que o Config detecte automaticamente');
     CenterTxt (10,'a porta na qual a Interface está');
     CenterTxt (11,'conectada?');
     TextColor (Yellow);
     CenterTxt (13,'(S)im ou (N)ão');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Escolha a opção: S ou N');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

Repeat
      Tecla:=Readkey;
      Tecla:=Upcase(Tecla);
      If Tecla='S' Then AutoDetect
      Else If Tecla='N' Then Espec;
Until (Tecla='S') Or (Tecla='N');

     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PosStat:=2 To 60 Do Begin
         Gotoxy (PosStat,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (8,'Deseja desabilitar a Detecção da Interface?');
     TextColor (Yellow);
     CenterTxt (10,'(S)im ou (N)ão');
     Textcolor (White);
     CenterTxt (12,'(Esta opção só é recomendada para realização');
     CenterTxt (13,'de diagnósticos na Interface,');
     CenterTxt (14,'por usuários avançados)');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Escolha a opção: S ou N');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
     VerifIntFace:=1;
Repeat
```

```
      Tecla:=Readkey;
      Tecla:=Upcase(Tecla);
Until (Tecla='S') Or (Tecla='N');
     If Tecla='S' Then VerifIntFace:=0;

     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PosStat:=2 To 60 Do Begin
         Gotoxy (PosStat,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (7,'Deseja desabilitar o desligamento');
     CenterTxt (8,'do Sincronismo para CC?');
     TextColor (Yellow);
     CenterTxt (10,'(S)im ou (N)ão');
     Textcolor (White);
     CenterTxt (12,'(Esta opção só é recomendada para realização');
     CenterTxt (13,'de diagnósticos na Interface,');
     CenterTxt (14,'por usuários avançados)');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Escolha a opção: S ou N');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
     VerifDC:=1;
Repeat
      Tecla:=Readkey;
      Tecla:=Upcase(Tecla);
Until (Tecla='S') Or (Tecla='N');
     If Tecla='S' Then VerifDC:=0;

     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PosStat:=2 To 60 Do Begin
         Gotoxy (PosStat,24);
         Write (' ');
     End;


     ApagaWin;
     AmpliaWin (60,10,White,Cyan);
     CenterWin (60,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (8,'As configurações foram completadas com sucesso!');
     CenterTxt (10,'Pressione Esc para sair sem salvar as alterações');
     CenterTxt (11,'ou Pressione Enter para sair e salvar as alterações');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Escolha a opção: Esc ou Enter');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
Repeat
      Tecla:=Readkey;
Until (Tecla=#13) Or (Tecla=#27);
     If Tecla=#27 Then Fim;
     Assign (F,'ALogic.Cfg');

Repeat
{$I-}
     Rewrite (F);
{$I+}
     Io:=IoResult;
     If Io<>0 Then Begin
        Window(1,1,80,25);
        TextColor (Yellow);
        TextBackGround(Cyan);
        CenterTxt (13,'Disco está protegido contra gravação!');
        CenterTxt (14,'Retire a proteção e pressione Enter');
        CenterTxt (15,'ou pressione Esc para Cancelar');
        Window (1,25,1,25);
        TextBackground(Black);
```

```
        TextColor(Black);
        Clrscr;

Repeat
      Tecla:=Readkey;
Until (Tecla=#13) Or (Tecla=#27);
     End;
Until (Io=0) Or (Tecla=#27);
     If Tecla<>#27 Then Begin
        Writeln (F,Ender,' ',VerifIntFace,' ',VerifDC);
        Close (F);
     End;
     Fim;
End;

{*****************************************************************************}

Procedure VerifArq;
{Verifica a existência de ALogic.Cfg}

Var G:File Of Char; {Representa o arquivo ALOGIC.CFG}

Procedure MensErro;
{Mensagem de Erro: Não existe ALOGIC.CFG}

Begin
        ClrScr;
        TextColor (Blue);
        Writeln ('Config do Analisador Lógico - Mensagem de Erro');
        Writeln;
        TextColor (LightRed);
        Writeln ('Não existe o arquivo "ALOGIC.CFG"');
        Writeln ('neste diretório!');
        Writeln;
        Writeln ('Verifique qual o diretório corrente!');
        Writeln ('Após isso, execute novamente este programa.');
        TextColor (LightGray);
        Writeln;
        Writeln ('Caso não consiga solucionar este problema,');
        Writeln ('entre em contato com:');
        Writeln;
        Textcolor (White);
        Writeln ('"Os PEP Legais" - LAO - 4oR Eletrônica');
        Textcolor (LightGray);
        Writeln;
        Halt(2);
End;



Begin
     Assign (G,'ALogic.Cfg');
{$I-}
     Reset (G);
{$I+}
     If IoResult <> 0 Then MensErro;
     Close (G);
End;

{*****************************************************************************}

Begin
     VerifArq;
     Tela('Analisador Lógico');
     Win1;
     Win2;
End.

{*****************************************************************************}
{
        * Formato do arquivo ALOGIC.CFG *

Record 0 = Endereço Base da Porta Paralela (em decimal)
Record 1:
         0 se a Detecção Automática estiver desabilitada
         1 se a Detecção Automática estiver habilitada
Record 2:
         0 se o Desligamento do Sincronismo para CC estiver desabilitado
         1 se o Desligamento do Sincronismo para CC estiver habilitado

}
{*****************************************************************************}
```

Diagnóstico (DIAG.PAS)

```
{$A-,B-,D-,E+,F-,G-,I+,L-,N+,Q-,R-,S+,X+,V+}
{Diretivas de compilação}

{*************************************************************************}

Program Diagnostico;

{*************************************************************************}
{

  Verifica a presença da Interface nos endereços especificados em PEPLEG.DAT

                "Os PEP Legais" - LAO - 4oR -Eletrônica


}
{*************************************************************************}

{       Créditos:
                Luiz Eduardo    - N°19
                Marcos Paulo    - N°23
                Rafael Abolafio - N°26
                Robson Martins  - N°30

                4°R - Eletrônica - 1997

             Liceu de Artes e Ofícios de São Paulo

             Prof: Marco Antônio Togniazzolo
}
{*************************************************************************}

Uses Crt,PEPUnit;
{Units a serem utilizadas}

{*************************************************************************}

Var Porta:Word; {Endereço da porta LPT}
    Dat:Text; {Arquivo DAT}
    Ender:String; {Endereço em Hexadecimal (String)}

{Declaração das variáveis globais}

{*************************************************************************}

Function LeProx:Word;
{Lê o próximo endereço contido em PEPLEG.DAT}

Var DadStr:String; {Endereço lido do arquivo PEPLEG.DAT}

Begin
     If Not Eof(Dat) Then Begin
        Readln (Dat,DadStr);
        LeProx:=HexToDec('0'+DadStr);
        Ender:=DadStr;
     End
     Else LeProx:=$378;
End;

{*************************************************************************}

Function Testa(P:Word):Boolean;
{Testa se a Interface está presente na Porta P}

Var Ts:Boolean; {True se a Interface está presente}

Begin
     Ts:=False;
     Port[P]:=$22; {0010 0010 - Reset/RD/ClrSync}
     Delay (600);

     Port[P]:=$22; {0010 0010 - Reset/RD/ClrSync}
     If (Port[P] And $20) Shr 5 = 0 Then Ts:=True;
     Testa:=Ts;
     Port[P]:=$00; {Desliga a Porta Paralela}
End;

{*************************************************************************}

Procedure Fim;
{Finaliza o Programa}

Begin
```

```
     Writeln ('A Interface está conectada corretamente!');
     Writeln ('Encontrada no endereço: ',Ender,' H');
     Writeln;
     Close (Dat);
     Halt(0);
End;

{*************************************************************************}

Procedure MensErro;
{Indica Erro se não existir PEPLEG.DAT}

Begin
     Writeln;
     Writeln ('Arquivo PEPLEG.DAT não encontrado!');
     Writeln;
     Writeln ('Verifique se o arquivo existe no diretório corrente.');
     Writeln ('Caso não consiga solucionar o problema, entre em contato com:');
     Writeln;
     Writeln ('"OS PEP LEGAIS" 4o R - Eletrônica - LAO - 1997');
     Writeln;
     Halt(2);
End;

{*************************************************************************}

Begin
     Assign (Dat,'PEPLEG.DAT');
{$I-}
     Reset (Dat);
{$I+}
     If IoResult<>0 Then MensErro;
     Writeln;
     Writeln ('"OS PEP LEGAIS"');
     Writeln;
     Writeln ('Certifique-se de que a Interface está conectada e alimentada!');
     Writeln;
     Writeln ('Pressione Enter quando estiver pronto...');
Repeat
Until Readkey=#13;
     Writeln;
     Writeln ('Verificando a Interface...');
     Writeln;
Repeat
     If Testa(LeProx) Then Fim;
Until Eof(Dat);
     Writeln ('A Interface não foi detectada pelo DIAG!');
     Writeln;
     Writeln ('Verifique as conexões e a alimentação da Interface.');
     Writeln ('Caso não consiga solucionar o problema, entre em contato com:');
     Writeln;
     Writeln ('"OS PEP LEGAIS" 4o R - Eletrônica - LAO - 1997');
     Writeln;
     Close (Dat);
End.

{*************************************************************************}
{
        * Formato do arquivo PEPLEG.DAT *

Linhas = Endereços utilizados pela detecção da Interface (em Hexadecimal)

}
{*************************************************************************}
```

Display (DISPLAY.PAS)

```
{$A-,B-,D-,E+,F-,G-,I+,L-,N+,Q-,R-,S+,X+,V+}
{Diretivas de compilação}

{*************************************************************************}

Program Display;

{*************************************************************************}
{
        Analisador Lógico - "Os PEP Legais"

        Créditos:
                  Luiz Eduardo    - N°19
                  Marcos Paulo    - N°23
                  Rafael Abolafio - N°26
                  Robson Martins  - N°30
```

```
                    4°R - Eletrônica - 1997

               Liceu de Artes e Ofícios de São Paulo

               Prof: Marco Antônio Togniazzolo
}
{*************************************************************************}
{
                    Display - Visualizador de dados BIN/HEX/DEC;
                              Para sincronismo externo

          Versão 1.0 - 1997

          Software for MS-DOS, Escrito no compilador Turbo Pascal 7.0

          Requisitos Mínimos do Sistema:

                    IBM PC XT/AT ou Compatível (Recomenda-se 486)
                    640KB de RAM ou mais (Recomenda-se 4MB)
                    Monitor EGA ou superior (Recomenda-se VGA Color)
                    Disk Drive de 1,44MB
                    Hard Disk com pelo menos 1MB livres
}
{*************************************************************************}

Uses Crt,Graph,Dos,PEPUnit;
{Units a serem utilizadas}

{*************************************************************************}

Var
    Tecla:Char; {Armazena Tecla lida}
    Ap:Integer; {Página de vídeo ativa}
    Ga,Dw,St,Help,TOut:Boolean; {Definem as funções chamadas pelo teclado}
    TimeOut:Integer; {Tempo de espera pela Interface}
    TOt: Array [1..4] Of Char; {Recolhe o valor de TimeOut}
    HabilVerif:Boolean; {Habilitação da verificação da Presença da Interface}
    P1,P2:Word; {Endereço da porta (Dados/Controle)}
    IndDado:Integer; {Indica a posição do dado}
    Dado:Array [0..511] Of Byte; {Armazena os dados lidos}
    Cur:Integer; {Define o cursor}

{*************************************************************************}

Procedure VerifArq;
{Verifica a presença dos arquivos necessários para a execução do programa}

Function ArqPres(NomeArqP:String):Boolean;
{Verifica a presença do arquivo NomeArqP}

Var FPres:File Of Char; {Representa o arquivo a ser testado}

Begin
     Assign (FPres,NomeArqP);
{$I-}
     Reset (FPres);
{$I+}
     If IOResult <> 0 Then ArqPres:=False
     Else Begin
          ArqPres:=True;
          Close (FPres);
     End;
End;

Procedure MensErroArq(NameArqF:String);
{Mensagem de Erro: Não existe o arquivo NameArqF}

Begin
     ClrScr;
     TextColor (Blue);
     Writeln ('PEPDisplay - Mensagem de Erro');
     Writeln;
     TextColor (LightRed);
     Writeln ('Não existe o arquivo "',NameArqF,'"');
     Writeln ('neste diretório!');
     Writeln;
     Writeln ('Verifique qual o diretório corrente!');
     Writeln ('Após isso, execute novamente este programa.');
     TextColor (LightGray);
     Writeln;
     Writeln ('Caso não consiga solucionar este problema,');
     Writeln ('entre em contato com:');
     Writeln;
     Textcolor (White);
```

```
     Writeln ('"Os PEP Legais" - LAO - 4oR Eletrônica');
     Textcolor (LightGray);
     Writeln;
     Halt(2);
End;

Begin
     If Not ArqPres('ALOGIC.CFG') Then MensErroArq('ALOGIC.CFG');
     If Not ArqPres('CONFIG.EXE') Then MensErroArq('CONFIG.EXE');
     If Not ArqPres('EGAVGA.BGI') Then MensErroArq('EGAVGA.BGI');
     If Not ArqPres('LITT.CHR') Then MensErroArq('LITT.CHR');
     If Not ArqPres('APRES2.PEP') Then MensErroArq('APRES2.PEP');
     If Not ArqPres('APRES4.PEP') Then MensErroArq('APRES2.PEP');
     If Not ArqPres('APRES6.PEP') Then MensErroArq('APRES2.PEP');
End;

{*************************************************************************}

Procedure AbreCfg;
{Abre o arquivo ALogic.Cfg}

Var HV,HS:Byte; {Habilita Verificação e Habilita desligamento de Sync}
    Cfg:Text; {Representa o arquivo de configuração}

Begin
     Assign (Cfg,'ALogic.Cfg');
     Reset (Cfg);
     Readln (Cfg,P1,HV,HS);
     P2:=P1;

     If HV=1 Then HabilVerif:=True
     Else HabilVerif:=False;
     Close (Cfg);
End;

{*************************************************************************}

Procedure Apres;
{Exibe as telas de apresentação do programa}

Var ApresTec:Char; {Valor da tecla pressionada durante a apresentação}
    TempoDelay:Longint; {Delay dado após a última apresentação}

Begin
     SetActivePage(0);
     SetVisualPage(0);
     SetTextStyle (SmallFont,HorizDir,4);
     SetTextJustify (CenterText,CenterText);
     SetColor (White);
     OutTextxy (320,175,'Carregando Tela de Apresentacao. Aguarde...');
     SetActivePage(1);
     Ap:=1;
     ApresTec:=ViewPEP('APRES6.PEP',1,0,True);
     If ApresTec<>#27 Then Begin
        SetVisualPage(1);
        SetActivePage(0);
        Ap:=0;
     End;
     ApresTec:=ViewPEP('APRES2.PEP',Ap,0,True);
     If ApresTec<>#27 Then Begin
        SetVisualPage(0);
        SetActivePage(1);
        Ap:=1;
     End;
     ApresTec:=ViewPEP('APRES4.PEP',Ap,0,True);
     If ApresTec<>#27 Then Begin
        SetVisualPage(1);
        SetActivePage(0);
        Ap:=0;
     End;
     For TempoDelay:=0 To 500000 Do Begin
         If Keypressed Then ApresTec:=Readkey;
         If (ApresTec=#13) Or (ApresTec=#27) Then Break;
     End;
     SetActivePage(1);
     ClearDevice;
     SetActivePage(0);
     ClearDevice;
     SetVisualPage(0);
     SetActivePage(1);
     Ap:=1;
End;

{*************************************************************************}
```

```
Procedure FimProg;
{Finaliza o programa}

Begin
     TextColor (LightGray);
     TextBackGround (Black);
     Clrscr;
     Writeln;
     Port[P1]:=$00; {Desliga a Porta Paralela}
     Halt(0);
End;

{*****************************************************************************}

Procedure Ajuda;
{Tela de ajuda dos comandos}

Begin
     SetFillStyle(1,0);
     Bar (130,110,530,260);
     SetFillStyle(1,15);
     Bar (120,100,520,250);
     SetLineStyle(0,1,0);
     SetColor(8);
     Rectangle (120,100,520,250);
     SetLineStyle(0,1,0);
     SetFillStyle (1,1);
     Bar (121,101,519,114);
     SetTextStyle (SmallFont,HorizDir,4);
     SetColor (White);
     SetTextJustify (CenterText,CenterText);
     OutTextxy (320,107,'Ajuda do PEPDisplay');
     SetColor (Black);
     OutTextxy (320,130,'ESC = Sai do Programa');
     OutTextxy (320,142,'F1 = Ajuda');
     OutTextxy (320,154,'F2 = Altera o valor do Time Out');
     OutTextxy (320,166,'F3 = Inicia a Aquisicao de Dados');
     OutTextxy (320,178,'F4 = Importa Dados de Arquivo .OSC');
     OutTextxy (320,190,'F5 = Grava Dados em Arquivo');
     OutTextxy (320,202,'Setas de Direcao = Movem o Cursor');
     OutTextxy (320,230,'Pressione ENTER para fechar esta janela');

Repeat
Until Readkey=#13;
     Help:=False;
End;

{*****************************************************************************}

Procedure AltTO;
{Altera o Time Out}

Var CTout:Integer; {Valor do Time Out}
    Tec:Char; {Caracter lido do teclado}
    Code:Integer; {Resultado da conversão String => Integer}
    TOStr:String; {Time Out em formato String}

Begin
     SetFillStyle(1,0);
     Bar (130,110,530,260);
     SetFillStyle(1,15);
     Bar (120,100,520,250);
     SetLineStyle(0,1,0);
     SetColor(8);
     Rectangle (120,100,520,250);
     SetLineStyle(0,1,0);
     SetFillStyle (1,1);
     Bar (121,101,519,114);
     SetTextStyle (SmallFont,HorizDir,4);
     SetColor (White);
     SetTextJustify (CenterText,CenterText);
     OutTextxy (320,107,'Alteracao do valor do TIME OUT');
     SetColor (Black);
     OutTextxy (320,150,'Digite o valor do Time Out (10 a 9999ms):');
     SetTextStyle (SmallFont,HorizDir,6);
     Str(TimeOut:0,TOStr);
     If Length(TOStr)=2 Then TOStr:='00'+TOStr
     Else If Length(TOStr)=3 Then TOStr:='0'+TOStr;
     OutTextxy (310,200,ToStr+' mS');
     SetFillStyle (1,15);
     CTout:=1;

Repeat
```

```
Repeat
Repeat
       Tec:=Readkey;
Until (Tec='0') Or (Tec='1') Or (Tec='2') Or (Tec='3') Or (Tec='4') Or
      (Tec='5') Or (Tec='6') Or (Tec='7') Or (Tec='8') Or (Tec='9') Or
      (Tec=#13);
      If Tec<>#13 Then Begin
         TOt[CTout]:=Tec;
         If CTout=1 Then Begin
            Bar (275,195,283,210);
            OutTextxy (310,200,Tec+'      ');
         End
         Else If CTout=2 Then Begin
            Bar (283,195,292,210);
            OutTextxy (310,200,' '+Tec+'     ');
         End
         Else If CTout=3 Then Begin
            Bar (293,195,303,210);
            OutTextxy (310,200,'  '+Tec+'    ');
         End
         Else If CTout=4 Then Begin
            Bar (303,195,313,210);
            OutTextxy (310,200,'   '+Tec+'   ');
         End;
         CTout:=CTout+1;
         If CTout=5 Then CTout:=1;
       End;
Until (Tec=#13);
     Val(Tot[1]+Tot[2]+Tot[3]+Tot[4],TimeOut,Code);
Until (TimeOut>=10) And (TimeOut<=9999);
     TOut:=False;
End;

{*************************************************************************}

Procedure VerifIntface;
{Verifica se a Interface está conectada corretamente ao Microcomputador}

Procedure MensErro;
{Mensagem de Erro: A Interface não foi detectada}

Begin
     SetFillStyle(1,0);
     Bar (130,110,530,260);
     SetFillStyle(1,15);
     Bar (120,100,520,250);
     SetLineStyle(0,1,0);
     SetColor(8);
     Rectangle (120,100,520,250);
     SetLineStyle(0,1,0);
     SetFillStyle (1,1);
     Bar (121,101,519,114);
     SetTextStyle (SmallFont,HorizDir,4);
     SetColor (White);
     SetTextJustify (CenterText,CenterText);
     OutTextxy (320,107,'Erro ao comunicar com Interface');
     SetColor (Red);
     SetTextStyle (SmallFont,HorizDir,5);
     OutTextxy (320,140,'A Interface nao foi encontrada pelo PEPDisplay!');
     SetColor (Black);
     SetTextStyle (SmallFont,HorizDir,4);
     OutTextxy (320,170,'Verifique as conexoes e a alimentacao do Interface.');
     OutTextxy (320,190,'Caso nao consiga solucionar o problema,');
     OutTextxy (320,200,'entre em contato com:');
     SetColor (Blue);
     OutTextxy (320,220,'OS PEP LEGAIS - 4o R - Eletronica - LAO - 1997');
     SetColor (Black);
     OutTextxy (320,240,'Pressione ENTER para fechar esta janela');
     SetColor (White);
     SetTextJustify (LeftText,CenterText);
     SetFillStyle (1,8);
     Bar (15,320,200,335);
     OutTextxy (20,325,'Erro na comunicacao!');
Repeat
Until Readkey=#13;
End;

Begin
     If HabilVerif Then Begin
        Port[P2]:=$22; {0010 0010 - Reset/RD/ClrSync}
        Delay(10);
        If (Port[P2] And $20) Shr 5 = 1 Then MensErro;
     End;
End;
```

```
{*************************************************************************}

Procedure CursorRight;
{Move o cursor para Direita}

Begin
     If Cur<511 Then Cur:=Cur+1
     Else Cur:=0;
End;

{*************************************************************************}

Procedure CursorLeft;
{Move o cursor para Esquerda}

Begin
     If Cur>0 Then Cur:=Cur-1
     Else Cur:=511;
End;

{*************************************************************************}

Procedure CursorDown;
{Move o cursor para Baixo}

Begin
     If Cur<480 Then Cur:=Cur+32
     Else Cur:=Cur+32-512;
End;

{*************************************************************************}

Procedure CursorUp;
{Move o cursor para Cima}

Begin
     If Cur>31 Then Cur:=Cur-32
     Else Cur:=Cur-32+512;
End;

{*************************************************************************}

Procedure VerifTecla;
{Verifica se alguma tecla foi pressionada}

Procedure VerifASCII;
{Verifica teclas extendidas}

Var TecASC:Char; {Armazena o caracter lido do teclado - teclas extendidas}

Begin
     TecASC:=Readkey;
     If TecASC=#59 Then Help:=True {F1}
     Else If TecASC=#60 Then TOut:=True {F2}
     Else If TecASC=#61 Then St:=True {F3}
     Else If TecASC=#77 Then CursorRight {Right Arrow}
     Else If TecASC=#75 Then CursorLeft {Left Arrow}
     Else If TecASC=#80 Then CursorDown {Down Arrow}
     Else If TecASC=#72 Then CursorUp {Up Arrow}
     Else If TecASC=#62 Then Dw:=True {F4}
     Else If TecASC=#63 Then Ga:=True; {F5}
End;

Begin
     Tecla:=Readkey;
     If Tecla=#27 Then Begin {Esc}
        CloseGraph;
        FimProg;
     End
     Else If Tecla=#0 Then VerifASCII;
End;

{*************************************************************************}

Procedure WriteDados;
{Permite a gravação dos dados pela Interface}

Var Hr,M,S,S100:Word; {Armazenam hora/minuto/segundo/centésimo atuais}
    Timea:Real; {Tempo em segundos}
    Vezes:Integer; {Número de pulsos para atualização da escala p/ Ext}

Begin
     Port[P2]:=$22; {0010 0010 - Reset/ClrSync/RD}
```

```
     VerifIntface;
     Port[P2]:=$A3; {1010 0011 - WR/ClrSync}
     For Vezes:=1 To 5 Do Begin
         Port[P2]:=$A3 Xor $4;
         Port[P2]:=$A3;
     End;
     Port[P2]:=$B3; {1011 0011 - WR - Start}
     Port[P2]:=$B1;  {1011 0001 - WR /PresetSync}
     GetTime (Hr,M,S,S100);
     Timea:=(S100+S*100+M*6E3+Hr*360E3)*10;

Repeat
      If Keypressed Then VerifTecla;  {Aguarda resposta até Time Out}
      GetTime (Hr,M,S,S100);
Until ((Port[P2] And $20) Shr 5 = 1) Or
      ((S100+S*100+M*6E3+Hr*360E3)*10-Timea >= TimeOut);
End;

{*************************************************************************}

Procedure ReadDados;
{Lê os dados a partir da Interface}

Begin
     Port[P2]:=$32; {0011 0010 - RD/Reset}
     Port[P2]:=$33; {0011 0011 - RD}

     For IndDado:=0 To 511 Do Begin
         Port[P2]:=$3B; {0011 1011 - RD/ClkRD}
         Port[P2]:=$33; {0011 0011 - RD}
         Port[P1]:=$FF;
         Dado[IndDado]:=Port[P1];
         Port[P2]:=$33;
         Delay (1);
         If Keypressed Then VerifTecla;
         End;
End;

{*************************************************************************}

Procedure Start;
{Inicia a aquisição de dados}

Begin
     SetColor (White);
     SetTextJustify (LeftText,CenterText);
     SetFillStyle (1,8);
     Bar (15,320,200,335);
     OutTextxy (20,325,'Comunicando com a Interface...');
     WriteDados;
     ReadDados;
     St:=False;
End;

{*************************************************************************}

Procedure DownLoad;
{Carrega um arquivo .OSC}

Procedure Win1;
{Tela de Apresentação - Utilitário de Importação de .OSC}

Begin
     Tela('Display - Importação de .OSC');
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackground (Cyan);
     CenterTxt (8,'Utilitário de Importação');
     CenterTxt (9,'de');
     CenterTxt (10,'Arquivos de Forma-de-Onda');
     TextColor (White);
     CenterTxt (13,'"Os PEP Legais"');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Pressione Enter para Continuar');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
Repeat
Until Readkey=#13;
End;


Procedure Win2;
```

```
{Utilitário de Importação de .OSC - Recolhe o nome de arquivo}

Var PP:Integer; {Utilizado para apagar linhas da tela}
    IoErr:Integer; {Resultado da última operação de I/O}
    FileOk:Boolean; {Indica Arquivo correto}
    Chj:Char; {Caracter lido do arquivo}
    PEPStr:String; {String 'PEP'}
    PEPOk:Boolean; {String 'PEP' está presente}
    Arq:File Of Char; {Representa o arquivo .OSC}
    NameArq:String; {Nome do arquivo .OSC}

Begin
     FileOk:=False;
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;
     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (7,'Digite o nome do arquivo');
     CenterTxt (8,'a ser importado');
     CenterTxt (9,'(com caminho completo, se necessário e');
     CenterTxt (10,'sem extensão - Extensão padrão: .OSC):');
     CenterTxt (14,'Pressione somente Enter para Cancelar');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Digite o nome do arquivo sem extensão / Enter: Cancela');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
     TextColor (Yellow);
     TextBackGround(Cyan);
     Window (1,1,80,25);
Repeat
     CenterTxt (12,'                                                ');
     Gotoxy (30,12);
     Readln (NameArq);
     If NameArq='' Then Exit;
     Assign (Arq,NameArq+'.OSC');
{$I-}
     Reset (Arq);
{$I+}
     IoErr:=IoResult;
     If (IoErr <>0) Then Begin
        CenterTxt (12,'Houve um erro na abertura do arquivo!');
        Sound (1000);
        Delay (100);
        Nosound;
        Delay (1000);
        CenterTxt (12,'                                                ');
     End
     Else Begin
          Read (Arq,Chj);
          PEPStr:=Chj;
          Read (Arq,Chj);
          PEPStr:=PEPStr+Chj;
          Read (Arq,Chj);
          PEPStr:=PEPStr+Chj;
          If PEPStr='PEP' Then PEPOk:=True
          Else PEPOk:=False;
          Reset(Arq);
          If (FileSize(Arq)=614) And (PEPOk) Then FileOk:=True
          Else Begin
                CenterTxt (12,'O formato do arquivo não é válido!');
                Sound (1000);
                Delay (100);
                Nosound;
                Delay (1000);
                CenterTxt (12,'                                                ');
          End;
     End;
Until FileOk;
     For PP:=0 To 101 Do Read(Arq,Chj);

     For PP:=102 To 613 Do Begin
         Read(Arq,Chj);
```

```
          Dado[PP-102]:=Ord(Chj);
     End;
     Close(Arq);
End;

Begin
     CloseGraph;
     Win1;
     Win2;
     ModoGrafico;
     Dw:=False;
End;

{*****************************************************************************}

Procedure GravaArq;
{Salva dados como arquivo .OSC , .BIN ou .HEX}

Procedure GravaOSC;
{Grava .OSC}

Var SobreGrava:Boolean; {Indica se o arquivo deve ser substituído}
    Io:Boolean; {Resultado da última operação de I/O}
    OffCan,OnCan,DadoLido:Char; {Canal Off/On; Dado lido do array Dado}
    DadoEsc:Byte; {Escala horizontal}
    Tecdo:Char; {Caracter lido do teclado}
    PP,RR:Integer; {Utilizados para apagar as descrições}
    Arq:File Of Char; {Representa o arquivo a ser gravado}
    NameArq:String; {Nome do arquivo a ser gravado}
    Descr:Array [1..10,0..8] Of Char; {Descrições}

Label DigitNovam; {O usuário deverá digitar novamente}
Label EscOutNome; {Escolher outro nome-de-arquivo}

Procedure Jaexiste;
{Já existe um arquivo com o mesmo nome}

Var Teclado:Char; {Caracter lido do teclado}
    PQ:Integer; {Utilizado para apagar linhas da tela}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PQ:=2 To 60 Do Begin
         Gotoxy (PQ,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (9,'O arquivo já existe!');
     CenterTxt (14,'(S)im ou (N)ão');
     TextColor (White);
     CenterTxt (12,'Substituir o arquivo existente?');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Escolha a opção: S ou N');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
     Sobregrava:=False;
Repeat
     Teclado:=Readkey;
     Teclado:=Upcase(Teclado);
Until (Teclado='S') Or (Teclado='N');
     If Teclado='S' Then Sobregrava:=True
     Else Sobregrava:=False;
End;

Procedure ErroIO;
{Erro de I/O}

Var PQ:Integer; {Utilizado para apagar linhas da tela}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
```

```
     For PQ:=2 To 60 Do Begin
         Gotoxy (PQ,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (9,'Houve um erro na Gravação');
     CenterTxt (10,'do Arquivo!');
     TextColor (White);
     CenterTxt (12,'Pressione Enter para Continuar');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Pressione Enter para Continuar');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
Repeat
Until Readkey=#13;
End;

Procedure Gravacao;
{Gravação do Arquivo .OSC}

Var PTec:Char; {Caracter lido do teclado}
    Num,Num2:Integer; {Endereçam as descrições}
    PQ:Integer;  {Utilizado para apagar linhas da tela}

Procedure Descricao (YValue:Integer);
{Recolhe as descrições para o .OSC
  YValue = Qual Descrição}

Var Cont:Integer; {Número do caracter da descrição}
    X1,Y1:Integer; {Coordenadas X e Y atuais}

Begin
     TextColor (Yellow);
     TextBackGround (Cyan);
     X1:=WhereX; Y1:=WhereY;
     Cont:=1;
Repeat
         PTec:=Readkey;
         PTec:=Upcase(PTec);
         If PTec=#13 Then Break;
         If (PTec=#8) And (Cont>1) Then Begin
            Cont:=Cont-1;
            Descr[Cont,YValue]:=#0;
            X1:=X1-1;
            GotoXy (X1,Y1);
            Write (' ');
            GotoXy (X1,Y1);
          End;


          If PTec<>#8 Then Begin
              Descr[Cont,YValue]:=PTec;
              Write (PTec);
              X1:=X1+1;
              Cont:=Cont+1;
         End;
         If Cont=11 Then Begin
            Repeat
                  PTec:=Readkey;
            Until (PTec=#13) Or (PTec=#8);
            If PTec=#8 Then Begin
               Cont:=Cont-1;
               Descr[Cont,YValue]:=#0;
               X1:=X1-1;
               GotoXy (X1,Y1);
               Write (' ');
               GotoXy (X1,Y1);
            End;
         End;
Until Cont>10;
End;

Begin
     Window (1,1,80,25);
     TextBackground (White);
```

```
     TextColor (Black);
     For PQ:=2 To 60 Do Begin
         Gotoxy (PQ,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (60,15,White,Cyan);
     CenterWin (60,15,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (5,'Digite a Descrição para o Conjunto de Formas de Onda:');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Digite as Descrições - Máximo 10 Caracteres cada');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
     Window (1,1,80,25);
     Gotoxy (35,7);
     Descricao (0);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (9,'Digite a Descrição para cada Canal:');
     CenterTxt (11,'A = '+'          ');
     CenterTxt (12,'B = '+'          ');
     CenterTxt (13,'C = '+'          ');
     CenterTxt (14,'D = '+'          ');
     CenterTxt (15,'E = '+'          ');
     CenterTxt (16,'F = '+'          ');
     CenterTxt (17,'G = '+'          ');
     CenterTxt (18,'H = '+'          ');
     For Num:=1 To 8 Do Begin
         GotoXy (38,Num+10);
         Descricao (Num);
     End;

     {Identificador do arquivo: Bytes 0 a 2}
     OnCan:='P';
     Write (Arq,OnCan);
     OnCan:='E';
     Write (Arq,OnCan);
     OnCan:='P';
     Write (Arq,OnCan);

     {Descrições: Bytes 3 a 92}
     For Num2:=0 To 8 Do Begin
         For Num:=1 To 10 Do Write (Arq,Descr[Num,Num2]);
     End;

     {Status dos canais: Bytes 93 a 100}
     OnCan:=Chr(1);
     OffCan:=Chr(0);
     Write (Arq,OnCan);
     Write (Arq,OnCan);
     Write (Arq,OnCan);
     Write (Arq,OnCan);
     Write (Arq,OnCan);
     Write (Arq,OnCan);
     Write (Arq,OnCan);
     Write (Arq,OnCan);

     {Escala Horizontal: Byte 101}
     DadoEsc:=5;
     OnCan:=Chr(DadoEsc);
     Write (Arq,OnCan);

     {Dados 512 Bytes: Bytes 102 a 614}
     For Num:=0 To 511 Do Begin
         DadoLido:=Chr(Dado[Num]);
         Write (Arq,DadoLido);
     End;
End;

Begin
     For RR:=0 To 8 Do Begin
         For PP:=1 To 10 Do Descr[PP,RR]:=#0;
     End;

Repeat

EscOutNome:
```

```
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (8,'Digite o nome do arquivo a ser gravado');
     CenterTxt (9,'(com caminho completo, se necessário e');
     CenterTxt (10,'sem extensão - Extensão padrão: .OSC):');
     CenterTxt (15,'Pressione somente Enter para Cancelar');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Digite o nome do arquivo sem extensão / Enter: Cancela');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

DigitNovam:
     Io:=False;
     SobreGrava:=True;
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (12,'                                                  ');
     Gotoxy (25,12);
     Readln (NameArq);
     If NameArq = '' Then Break;
     Assign (Arq,NameArq+'.OSC');
{$I-}
     Reset(Arq);
{$I+}
     If IoResult = 0 Then Jaexiste;
     If Not Sobregrava Then Goto EscOutNome;
{$I-}
     Rewrite (Arq);
{$I+}
     If IoResult <> 0 Then ErroIO
     Else Io:=True;
Until Io;

     If Io Then Begin
        Gravacao;
        Close (Arq);
     End;
End;

Procedure GravaBIN;
{Grava .BIN}

Var SobreGrava:Boolean; {Indica se o arquivo deve ser substituído}
    Io:Boolean; {Resultado da última operação de I/O}
    DadoLido:Char; {Dado lido do array Dado}
    DadoEsc:Byte; {Escala horizontal}
    Tecdo:Char; {Caracter lido do teclado}
    PP:Integer; {Utilizado para apagar linhas da tela}
    Arq:File Of Char; {Representa o arquivo a ser gravado}
    NameArq:String; {Nome do arquivo a ser gravado}

Label DigitNovam; {O usuário deverá digitar novamente}
Label EscOutNome; {Escolher outro nome-de-arquivo}

Procedure Jaexiste;
{Já existe um arquivo com o mesmo nome}

Var Teclado:Char; {Caracter lido do teclado}
    PQ:Integer; {Utilizado para apagar linhas da tela}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PQ:=2 To 60 Do Begin
         Gotoxy (PQ,24);
         Write (' ');
     End;
```

```
     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (9,'O arquivo já existe!');
     CenterTxt (14,'(S)im ou (N)ão');
     TextColor (White);
     CenterTxt (12,'Substituir o arquivo existente?');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Escolha a opção: S ou N');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
     Sobregrava:=False;
Repeat
     Teclado:=Readkey;
     Teclado:=Upcase(Teclado);
Until (Teclado='S') Or (Teclado='N');
     If Teclado='S' Then Sobregrava:=True
     Else Sobregrava:=False;
End;

Procedure ErroIO;
{Erro de I/O}

Var PQ:Integer; {Utilizado para apagar linhas da tela}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PQ:=2 To 60 Do Begin
         Gotoxy (PQ,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (9,'Houve um erro na Gravação');
     CenterTxt (10,'do Arquivo!');
     TextColor (White);
     CenterTxt (12,'Pressione Enter para Continuar');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Pressione Enter para Continuar');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
Repeat
Until Readkey=#13;
End;


Procedure Gravacao;
{Gravação do Arquivo .BIN}

Var PTec:Char; {Caracter lido do teclado}
    Num,Num2:Integer; {Fazem a contagem dos records gravados}
    PQ:Integer;  {Utilizado para apagar linhas da tela}

Begin
     {Dados 512 Bytes: Bytes 0 a 511}
     For Num:=0 To 511 Do Begin
         DadoLido:=Chr(Dado[Num]);
         Write (Arq,DadoLido);
     End;
End;

Begin
Repeat
EscOutNome:
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
```

```
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (8,'Digite o nome do arquivo a ser gravado');
     CenterTxt (9,'(com caminho completo, se necessário e');
     CenterTxt (10,'sem extensão - Extensão padrão: .BIN):');
     CenterTxt (15,'Pressione somente Enter para Cancelar');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Digite o nome do arquivo sem extensão / Enter: Cancela');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

DigitNovam:
     Io:=False;
     SobreGrava:=True;
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (12,'                                                ');
     Gotoxy (25,12);
     Readln (NameArq);
     If NameArq = '' Then Break;
     Assign (Arq,NameArq+'.BIN');
{$I-}
     Reset(Arq);
{$I+}
     If IoResult = 0 Then Jaexiste;
     If Not Sobregrava Then Goto EscOutNome;
{$I-}
     Rewrite (Arq);
{$I+}
     If IoResult <> 0 Then ErroIO
     Else Io:=True;
Until Io;

     If Io Then Begin
        Gravacao;
        Close (Arq);
     End;
End;

Procedure GravaHEX;
{Grava .HEX}

Var SobreGrava:Boolean; {Indica se o arquivo deve ser substituído}
    Io:Boolean; {Resultado da última operação de I/O}
    NumHex,DadoHex,DadoLido:String; {Valor lido do array dado em hex}
    DadoEsc:Byte; {Escala horizontal}
    Tecdo:Char; {Caracter lido do teclado}
    PP:Integer; {Utilizado para apagar linhas da tela}
    Arq:Text; {Representa o arquivo a ser gravado}
    NameArq:String; {Nome do arquivo a ser gravado}


Label DigitNovam; {O usuário deverá digitar novamente}
Label EscOutNome; {Escolher outro nome-de-arquivo}

Procedure Jaexiste;
{Já existe um arquivo com o mesmo nome}

Var Teclado:Char; {Caracter lido do teclado}
    PQ:Integer; {Utilizado para apagar linhas da tela}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PQ:=2 To 60 Do Begin
         Gotoxy (PQ,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
```

```
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (9,'O arquivo já existe!');
     CenterTxt (14,'(S)im ou (N)ão');
     TextColor (White);
     CenterTxt (12,'Substituir o arquivo existente?');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Escolha a opção: S ou N');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
     Sobregrava:=False;
Repeat
     Teclado:=Readkey;
     Teclado:=Upcase(Teclado);
Until (Teclado='S') Or (Teclado='N');
     If Teclado='S' Then Sobregrava:=True
     Else Sobregrava:=False;
End;

Procedure ErroIO;
{Erro de I/O}

Var PQ:Integer; {Utilizado para apagar lihas da tela}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PQ:=2 To 60 Do Begin
         Gotoxy (PQ,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (9,'Houve um erro na Gravação');
     CenterTxt (10,'do Arquivo!');
     TextColor (White);
     CenterTxt (12,'Pressione Enter para Continuar');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Pressione Enter para Continuar');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
Repeat
Until Readkey=#13;
End;

Procedure Gravacao;
{Gravação do Arquivo .HEX}

Var PTec:Char; {Caracter lido do teclado}
    Num,Num2:Integer; {Fazem a contagem dos records gravados}
    PQ:Integer;  {Utilizado para apagar linhas da tela}

Begin
     For Num:=0 To 511 Do Begin
         DadoLido:=' - ';
         DadoHex:=DecToHex(Dado[Num],False);
         NumHex:=DecToHex(Num,True);
         Writeln (Arq,NumHex,DadoLido,DadoHex);
     End;
End;

Begin
Repeat
EscOutNome:
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;
```

```
     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (8,'Digite o nome do arquivo a ser gravado');
     CenterTxt (9,'(com caminho completo, se necessário e');
     CenterTxt (10,'sem extensão - Extensão padrão: .HEX):');
     CenterTxt (15,'Pressione somente Enter para Cancelar');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Digite o nome do arquivo sem extensão / Enter: Cancela');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

DigitNovam:
     Io:=False;
     SobreGrava:=True;
     TextColor (Yellow);
     TextBackGround (Cyan);
     CenterTxt (12,'                                                  ');
     Gotoxy (25,12);
     Readln (NameArq);
     If NameArq = '' Then Break;
     Assign (Arq,NameArq+'.HEX');
{$I-}
     Reset(Arq);
{$I+}
     If IoResult = 0 Then Jaexiste;
     If Not Sobregrava Then Goto EscOutNome;
{$I-}
     Rewrite (Arq);
{$I+}
     If IoResult <> 0 Then ErroIO
     Else Io:=True;
Until Io;

     If Io Then Begin
        Gravacao;
        Close (Arq);
     End;
End;

Procedure Win1;
{Tela de apresentação - Utilitário de Gravação .OSC/.BIN/.HEX}

Begin
     Tela('Display - Gravação de Arquivos');
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackground (Cyan);
     CenterTxt (8,'Utilitário de Gravação');
     CenterTxt (9,'de');
     CenterTxt (10,'Arquivos de Forma-de-Onda');
     TextColor (White);
     CenterTxt (13,'"Os PEP Legais"');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Pressione Enter para Continuar');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
Repeat
Until Readkey=#13;
End;

Procedure Win2;
{Utilitário de Gravação .OSC/.BIN/.HEX - Escolha do formato}

Var PP:Integer; {Utilizado para apagar linhas da tela}
    TecE:Char; {Caracter lido do teclado}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;
```

```
     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (7,'Escolha o formato do arquivo');
     CenterTxt (8,'a ser gravado');
     CenterTxt (10,'1. Arquivo de forma-de-onda (.OSC)');
     CenterTxt (11,'2. Arquivo binário de dados (.BIN)');
     CenterTxt (12,'3. Arquivo texto de dados   (.HEX)');
     CenterTxt (13,'4. Cancelar a operação            ');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24);
     Write ('Pressione 1, 2, 3 ou 4');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
Repeat
      TecE:=Readkey;
Until (TecE='1') Or (TecE='2') Or (TecE='3') Or (TecE='4');
      If TecE='1' Then GravaOSC
      Else If TecE='2' Then GravaBIN
      Else If TecE='3' Then GravaHEX
      Else Exit;
End;

Begin
     CloseGraph;
     Win1;
     Win2;
     Window(1,1,80,25);
     TextColor(LightGray);
     TextBackGround(Black);
     ModoGrafico;
     Ga:=False;
End;

{*************************************************************************}

Procedure Displ (XCent:Integer; YCent:Integer; XMed:Integer; Numb:Integer;
                 CorDisp:Integer; CorBord:Integer);
{Faz um display de tamanho especificado (XMed), na posição especificada
 (XCent,YCent), mostrando um dígito especificado (Numb), com cor de LED
 (CorDisp) e cor de borda (CorBord) especificados}

{Decodificador DEC(Hex)/7Segmentos}

Var XX,YY:Integer; {Centro do display}

Procedure D;
{Segmento D}

Begin
     If Keypressed Then VerifTecla;
        Line (XCent-Round((16/25)*XX),YCent+Round((23/30)*YY),
              XCent+Round((11/25)*XX),YCent+Round((23/30)*YY));
End;

Procedure A;
{Segmento A}


Begin
     If Keypressed Then VerifTecla;
        Line (XCent-Round((13/25)*XX),YCent-Round((23/30)*YY),
              XCent+Round((14/25)*XX),YCent-Round((23/30)*YY));
End;

Procedure G;
{Segmento G}

Begin
     If Keypressed Then VerifTecla;
        Line (XCent-Round((14/25)*XX),YCent,XCent+Round((12/25)*XX),YCent);
End;

Procedure F;
{Segmento F}

Begin
     If Keypressed Then VerifTecla;
        Line (XCent-Round((16/25)*XX),YCent-Round((22/30)*YY),
```

```
              XCent-Round((17/25)*XX),YCent-Round((2/30)*YY));
End;

Procedure E;
{Segmento E}

Begin
     If Keypressed Then VerifTecla;
        Line (XCent-Round((17/25)*XX),YCent+Round((2/30)*YY),
              XCent-Round((18/25)*XX),YCent+Round((22/30)*YY));
End;

Procedure B;
{Segmento B}

Begin
     If Keypressed Then VerifTecla;
        Line (XCent+Round((17/25)*XX),YCent-Round((22/30)*YY),
              XCent+Round((16/25)*XX),YCent-Round((2/30)*YY));
End;

Procedure C;
{Segmento C}

Begin
     If Keypressed Then VerifTecla;
        Line (XCent+Round((16/25)*XX),YCent+Round((2/30)*YY),
              XCent+Round((14/25)*XX),YCent+Round((22/30)*YY));
End;

Begin
     SetColor (CorBord);
     SetLineStyle (0,1,1);
     XX:=Round(XMed/2);
     YY:=Round(XX*1.2);
     Rectangle (XCent-Round(1.4*XX),YCent-Round(1.4*YY),XCent+Round(1.4*XX),
                YCent+Round(1.4*YY));
     SetFillStyle (1,0);
     Bar (XCent-Round(1.4*XX)+1,YCent-Round(1.4*YY)+1,XCent+Round(1.4*XX)-1,
          YCent+Round(1.4*YY)-1);
     SetColor (CorDisp);
     If Numb=0 Then Begin
        A; B; C; D; E; F;
     End;
     If Numb=1 Then Begin
        B; C;
     End;
     If Numb=2 Then Begin
        A; B; D; E; G;
     End;
     If Numb=3 Then Begin
        A; B; C; D; G;
     End;
     If Numb=4 Then Begin
        B; C; F; G;
     End;
     If Numb=5 Then Begin
        A; C; D; F; G;
     End;
     If Numb=6 Then Begin
        A; C; D; E; F; G;
     End;
     If Numb=7 Then Begin
        A; B; C;
     End;


     If Numb=8 Then Begin
        A; B; C; D; E; F; G;
     End;
     If Numb=9 Then Begin
        A; B; C; D; F; G;
     End;
     If Numb=10 Then Begin
        A; B; C; E; F; G;
     End;
     If Numb=11 Then Begin
        C; D; E; F; G;
     End;
     If Numb=12 Then Begin
        A; D; E; F;
     End;
     If Numb=13 Then Begin
        B; C; D; E; G;
```

```
     End;
     If Numb=14 Then Begin
        A; D; E; F; G;
     End;
     If Numb=15 Then Begin
        A; E; F; G;
     End;
End;

{*************************************************************************}

Procedure TelaPrinc;
{Mostra a tela principal, atualizando os displays}

Var XDis,YDis:Integer; {Coordenadas dos displays}
    CorDs:Integer; {Cores dos displays}

Begin
     SetFillStyle (1,8);
     Bar (10,10,630,340);
     SetLineStyle (0,1,1);
     SetColor (White);
     Rectangle (10,10,630,340);
     Line (10,25,630,25);
     SetFillStyle (1,1);
     Bar (11,11,629,24);
     SetTextStyle (SmallFont, HorizDir,5);
     SetTextJustify (CenterText,CenterText);
     OutTextxy(320,15,'Display - Os PEP Legais');

     SetColor (White);
     SetTextStyle (SmallFont, HorizDir,4);
     SetTextJustify (LeftText,CenterText);
     OutTextxy(50,38,'TIME OUT:');
     Displ (45,55,10,Trunc(TimeOut/1000),LightGreen,0);
     Displ (59,55,10,Trunc(TimeOut/100)-Trunc(TimeOut/1000)*10,LightGreen,0);
     Displ (73,55,10,Trunc(TimeOut/10)-(Trunc(TimeOut/100)-
            Trunc(TimeOut/1000)*10)*10-Trunc(TimeOut/1000)*100,LightGreen,0);
     Displ (87,55,10,TimeOut-(Trunc(TimeOut/10)-(Trunc(TimeOut/100)-
            Trunc(TimeOut/1000)*10)*10-Trunc(TimeOut/1000)*100)*10-
            (Trunc(TimeOut/100)-Trunc(TimeOut/1000)*10)*100-
            Trunc(TimeOut/1000)*1000,LightGreen,0);
     SetColor (White);
     OutTextxy(100,55,'mS');

     OutTextxy(150,38,'POSICAO DO DADO:');
     Displ (135,55,10,Trunc(Cur/256),LightGreen,0);
     Displ (149,55,10,Trunc(((Cur/256-Trunc(Cur/256))*256)/16),LightGreen,0);
     Displ (163,55,10,Trunc(((((Cur/256-Trunc(Cur/256))*256)/16)-
            Trunc(((Cur/256-Trunc(Cur/256))*256)/16))*16),LightGreen,0);
     SetColor (White);
     OutTextxy (175,55,'HEX');
     Displ (210,55,10,Trunc(Cur/100),LightGreen,0);
     Displ (224,55,10,Trunc(Cur/10)-Trunc(Cur/100)*10,LightGreen,0);
     Displ (238,55,10,Cur-(Trunc(Cur/10)-Trunc(Cur/100)*10)*10-
            Trunc(Cur/100)*100,LightGreen,0);
     SetColor (White);
     OutTextxy (250,55,'DEC');

     OutTextxy (350,38,'VALOR DO DADO:');
     Displ (286,55,10,Trunc(Dado[Cur]/100),LightGreen,0);
     Displ (300,55,10,Trunc(Dado[Cur]/10)-Trunc(Dado[Cur]/100)*10,
            LightGreen,0);
     Displ (314,55,10,Dado[Cur]-(Trunc(Dado[Cur]/10)-Trunc(Dado[Cur]/100)*10)
            *10-Trunc(Dado[Cur]/100)*100,LightGreen,0);
     SetColor (White);
     OutTextxy (330,55,'DEC');
     Displ (365,55,10,(Dado[Cur] And $80) Shr 7,LightGreen,0);
     Displ (379,55,10,(Dado[Cur] And $40) Shr 6,LightGreen,0);
     Displ (393,55,10,(Dado[Cur] And $20) Shr 5,LightGreen,0);
     Displ (407,55,10,(Dado[Cur] And $10) Shr 4,LightGreen,0);
     Displ (421,55,10,(Dado[Cur] And $08) Shr 3,LightGreen,0);
     Displ (435,55,10,(Dado[Cur] And $04) Shr 2,LightGreen,0);
     Displ (449,55,10,(Dado[Cur] And $02) Shr 1,LightGreen,0);
     Displ (463,55,10,(Dado[Cur] And $01),LightGreen,0);
     SetColor (White);
     OutTextxy (478,55,'BIN');
     OutTextxy (20,80,'Pressione F1 para Ajuda');
     OutTextxy (520,55,'Sincr: Externo');
     Rectangle (20,30,620,70);
     Line (120,30,120,70);
     Line (270,30,270,70);
     Line (510,30,510,70);
```

```
     IndDado:=0;
     YDis:=100;
Repeat
     XDis:=22;
Repeat
      If IndDado=Cur Then CorDs:=LightBlue
      Else CorDs:=LightRed;
      Displ (XDis,YDis,6,((Dado[IndDado] And $F0) Shr 4),CorDs,0);
      Displ (XDis+8,YDis,6,(Dado[IndDado] And $0F),CorDs,0);
      XDis:=XDis+19;

      IndDado:=IndDado+1;
Until ((XDis-22)/19)>=32;
      YDis:=YDis+14;
Until ((YDis-100)/14)>=16;

     SetColor (7);
     XDis:=16;
Repeat
     If Keypressed Then VerifTecla;
      Line (XDis,93,XDis,316);
      XDis:=XDis+19;
Until XDis>630;
      YDis:=92;
Repeat
     If Keypressed Then VerifTecla;
      Line (16,YDis,624,YDis);
      YDis:=YDis+14;
Until YDis>317;

     SetColor (White);
     SetTextJustify (LeftText,CenterText);
     SetFillStyle (1,8);
     Bar (15,320,200,335);
     OutTextxy (20,325,'Em Repouso                    ');
End;

{*************************************************************************}

Begin
     VerifArq;
     AbreCfg;
     ModoGrafico;
     Ap:=0;
     TimeOut:=10;
     Tot[1]:='0'; Tot[2]:='0'; Tot[3]:='1'; Tot[4]:='0';
     Cur:=0;
     Apres;

Repeat
     If Help Then Ajuda;
     If TOut Then AltTO;
     If St Then Start;
     If Dw Then DownLoad;
     If Ga Then GravaArq;
     If Ap=0 Then Begin
        SetVisualPage(0);
        SetActivePage(1);
        Ap:=1;
     End
     Else Begin
        SetVisualPage(1);
        SetActivePage(0);
        Ap:=0;
     End;
     Delay(10);
     TelaPrinc;
     If Ap=1 Then SetVisualPage(1)
     Else SetVisualPage(0);
     If Keypressed Then VerifTecla;
Until Tecla=#27;
     CloseGraph;
     FimProg;
End.

{*************************************************************************}
{
        * Formato do arquivo .BIN *

Bytes 0 a 511 = ASCII dos Dados

}
{*************************************************************************}
{
```

```
        * Formato do arquivo .HEX *

Coluna 1 = Posição do Dado (em Hexadecimal)
Coluna 2 = Valor do Dado (em Hexadecimal)

}
{*************************************************************************}
```

Install (INSTALL.PAS)

```
{$A-,B-,D-,E+,F-,G-,I+,L-,N+,Q-,R-,S+,X+,V+}
{Diretivas de compilação}

{*************************************************************************}

Program Install;

{*************************************************************************}
{
        Analisador Lógico - "Os PEP Legais"

        Créditos:
                  Luiz Eduardo    - N°19
                  Marcos Paulo    - N°23
                  Rafael Abolafio - N°26
                  Robson Martins  - N°30

                  4°R - Eletrônica - 1997

              Liceu de Artes e Ofícios de São Paulo

              Prof: Marco Antônio Togniazzolo
}
{*************************************************************************}

{Programa de Instalação do PEP Software para Analisador Lógico}
{INSTALL.SET - Arquivo com o Setup Files do Install}

{*************************************************************************}

Uses Crt,Dos,PEPUnit;
{Units a serem utilizadas}

{*************************************************************************}

Var SetupFile: Text; {Representa o arquivo de SetUp}
    ArqOrig, ArqDest, DriveDest, DirOrig, DirDest: String;  {Nomes de arquivo,
                                                             diretório e drive}
    Drives: Array [1..26] Of Byte; {Armazena os Drives disponíveis no sistema}

{*************************************************************************}

Procedure ApagaTela;
{Apaga a tela atual}

Begin
     Window(1,1,80,25);
     TextColor(LightGray);
     TextBackGround(Black);
     Clrscr;
End;

{*************************************************************************}

Procedure OpenSetup;
{Abre o arquivo de configuração INSTALL.SET}

Procedure MensErro;
{Mensagem de Erro - Não foi encontrado INSTALL.SET}

Begin
     ApagaTela;
     TextColor (Blue);
     Writeln ('Analisador Lógico - Mensagem de Erro');
     Writeln;
     TextColor (LightRed);
     Writeln ('Não existe o arquivo "INSTALL.SET"');
     Writeln ('neste disco de instalação!');
     Writeln;
     Writeln ('Adquira uma nova cópia do Software!');
     TextColor (LightGray);
     Writeln;
     Writeln ('Entre em contato com:');
     Writeln;
```

```
     Textcolor (White);
     Writeln ('"Os PEP Legais" - LAO - 4oR Eletrônica');
     Textcolor (LightGray);
     Writeln;
     Halt(2);
End;

Begin
     Assign (SetupFile,'INSTALL.SET');
{$I-}
     Reset (SetupFile);
{$I+}
     If IoResult <> 0 Then MensErro;
End;

{*********************************************************************}

Procedure LeSetup;
{Lê uma Linha do arquivo INSTALL.SET}

Var ContSet:Integer; {Número de caracteres do nome-de-arquivo a ser copiado}

Begin
     If Eof(SetupFile) Then ArqDest:=''
     Else Begin
          Readln (SetupFile,ArqDest);
          ArqOrig:='';
          For ContSet:=1 To Length(ArqDest)-1 Do Begin
              ArqOrig:=ArqOrig+Copy(ArqDest,ContSet,1);
          End;
          ArqOrig:=ArqOrig+'_';
     End;
End;

{*********************************************************************}

Procedure MensCancela;
{Mensagem se for cancelada a Instalação}

Begin
     ApagaTela;
     Writeln;
     Writeln ('O Software para o Analisador Lógico não foi instalado!');
     Writeln ('Execute novamente o INSTALL!');
     Writeln;
     Halt(0);
End;

{*********************************************************************}

Procedure Win1;
{Tela inicial do programa - Apresentação}

Var Tecla:Char; {Caracter lido do teclado}

Begin
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackground (Cyan);
     CenterTxt (9,'PROGRAMA DE INSTALAÇÃO DO');
     CenterTxt (10,'SOFTWARE PARA O ANALISADOR LÓGICO');
     TextColor (White);
     CenterTxt (13,'"Os PEP Legais"');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Pressione Enter para Continuar ou Esc para Sair');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

Repeat
      Tecla:=Readkey;
      If Tecla=#27 Then MensCancela;
Until Tecla=#13;

End;

{*********************************************************************}

Procedure DetectDrives;
{Detecta as unidades válidas >= 1MB Livre}

Var DrCont: Integer; {Conta o número de drives do sistema}
    DriveTrue: Boolean; {Drive correto (>=1MB livre)}
```

```
Procedure MensNoDiskSpace;
{Mensagem de Erro: Não há unidades com pelo menos 1MB Livres}

Begin
     ApagaTela;
     Writeln;
     Writeln ('Não existe nenhuma Unidade de Disco com pelo Menos 1MB Livres');
     Writeln;
     Write ('Libere espaço em uma das Unidades');
     Writeln (' e execute novamente o INSTALL.');
     Writeln;
     Halt(0);
End;

Begin
     For DrCont:=1 To 2 Do Drives[DrCont]:=0;
     For DrCont:=3 To 26 Do Begin
         If DiskFree(DrCont) < 1024*1000 Then Drives[DrCont]:=0
         Else Drives[DrCont]:=1;
     End;
     DriveTrue:=False;

     For DrCont:=1 To 26 Do Begin
         If Drives[DrCont]=1 Then Begin
            DriveTrue:=True;
            Break;
         End;
     End;

     If Not DriveTrue Then MensNoDiskSpace;
End;

{*************************************************************************}

Procedure Win2;
{Tela de determinação do drive de destino}

Var Tecla:Char; {Caracter lido do teclado}
    PP:Integer; {Utilizado para apagar linhas da tela}
    DrOk:Boolean; {Indica Drive correto}
    DrCont:Integer; {Conta o número de drives do sistema}
    Tec:Char; {Caracter lido do teclado}

Function DriveVal:String;
{Verifica os Drives existentes no Sistema e Atribui uma String para cada}

Var DrStr:String; {Drive em formato String}
    DrCont:Integer; {Conta os drives existentes no sistema}

Begin
     DrStr:='';
     For DrCont:=1 To 26 Do Begin
         If Drives[DrCont]=1 Then DrStr:=DrStr+Chr(DrCont+64)+': ';
     End;
     DriveVal:=DrStr;
End;

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (9,'Digite o Drive de Destino: (Default = C:)');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Drives Válidos: ',DriveVal,' / Esc: Cancela');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

     Window (1,1,80,25);
     TextBackGround(LightCyan);
     TextColor (Yellow);
```

```
     CenterTxt(12,'C:');
     DrOk:=True;
     Tec:='C';
Repeat
Repeat
      Tecla:=Readkey;
      Tecla:=Upcase(Tecla);
      If Tecla=#27 Then MensCancela;
      If Tecla=#8 Then Tecla:=#0;
      If Tecla=#13 Then Break;
      CenterTxt (12,Tecla+':');
      For DrCont:=1 To 26 Do Begin
          If Drives[Ord(Tecla)-64]=1 Then Begin
             DrOk:=True;
             Tec:=Tecla;
             Break;
          End
          Else DrOk:=False;
      End;
Until DrOk;
      If (Tecla=#13) And Not (DrOk) Then Begin
         Sound(1000);
         Delay(200);
         Nosound;
      End;
Until (Tecla=#13) And (DrOk);
      DriveDest:=Tec;
End;

{*************************************************************************}

Procedure MensErroRead;
{Mensagem de Erro - Falha na leitura}

Begin
     ApagaTela;
     TextColor (Blue);
     Writeln ('Analisador Lógico - Mensagem de Erro');
     Writeln;
     TextColor (LightRed);
     Writeln ('Houve um erro de leitura');
     Writeln ('neste Disco de Instalação!');
     Writeln;
     Writeln ('Adquira uma nova cópia do Software!');
     TextColor (LightGray);
     Writeln;
     Writeln ('Entre em contato com:');
     Writeln;
     Textcolor (White);
     Writeln ('"Os PEP Legais" - LAO - 4oR Eletrônica');
     Textcolor (LightGray);
     Writeln;
     Halt(2);
End;

{*************************************************************************}

Procedure MensErroWrite;
{Mensagem de Erro - Falha na Gravação}

Begin
     ApagaTela;
     TextColor (Blue);
     Writeln ('Analisador Lógico - Mensagem de Erro');
     Writeln;
     TextColor (LightRed);
     Writeln ('Houve um Erro Fatal de Gravação');
     Writeln ('no Disco de Destino!');
     Writeln;
     Writeln ('Execute um Utilitário de Reparo de Disco');
     Writeln ('para solucionar este problema!');
     TextColor (LightGray);
     Writeln;
     Writeln ('Caso não resolva este problema,');
     Writeln ('entre em contato com:');
     Writeln;
     Textcolor (White);
     Writeln ('"Os PEP Legais" - LAO - 4oR Eletrônica');
     Textcolor (LightGray);
     Writeln;
     Halt(2);
End;

{*************************************************************************}
```

```
Procedure Win3;
{Tela de determinação de diretório de destino}

Var Tecla:Char; {Caracter lido do teclado}
    PP:Integer; {Utilizado para apagar linhas da tela}
    Io:Integer; {Resultado da última operação de I/O}
    DirExiste:Boolean; {Diretório existe}

Procedure Confirma;
{Confirma se o Diretório será substituido}

Begin
      Sound (1000);
      Delay(200);
      Nosound;
      TextColor (White);
      CenterTxt (13,'                                                ');
      CenterTxt (14,'                                                ');
      CenterTxt(13,'Já existe um diretório com o mesmo nome!');
      CenterTxt(14,'Sobrescreve o diretório (S,N)?');
      TextColor(Yellow);
Repeat
      Tecla:=UpCase(Readkey);
Until (Tecla='S') Or (Tecla='N');
      If Tecla='N' Then Io:=1;
End;

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (8,'Digite o Diretório de Destino:');
     CenterTxt (9,'(Default = \PEPLEG)');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Digite o nome do diretório / "!": Cancela');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

     Window (1,1,80,25);
     TextBackGround(LightCyan);
     TextColor (Yellow);
Repeat
      CenterTxt (11,'                                                ');
      CenterTxt (13,'                                                ');
      CenterTxt (14,'                                                ');
      Gotoxy (36,11);
      Write(DriveDest+':\');
      Readln (DirDest);
      If DirDest='!' Then MensCancela;
      If DirDest='' Then Begin
         GotoXy(39,11);
         Write('PEPLEG');
         TextColor (White);
         CenterTxt(13,'Confirma o diretório acima indicado (S,N)?');
         TextColor(Yellow);
Repeat
      Tecla:=UpCase(Readkey);
Until (Tecla='S') Or (Tecla='N');

         If Tecla='S' Then DirDest:='PEPLEG'
         Else DirDest:=':';
      End;
      GetDir(0,DirOrig);
      DirExiste:=True;
{$I-}
      ChDir(DriveDest+':\'+DirDest);
{$I+}
      Io:=IoResult;
      If Io<>0 Then DirExiste:=False;
      Io:=0;
```

```
      If DirExiste Then Confirma;
{$I-}
      ChDir(DirOrig);
{$I+}
      If IoResult<>0 Then MensErroRead;

   If Not DirExiste Then Begin
{$I-}
      MkDir(DriveDest+':\'+DirDest);
{$I+}
      Io:=IoResult;
      If Io<>0 Then Begin
         Sound (1000);
         Delay(200);
         Nosound;
      End;
   End;
Until Io = 0;
End;

{*************************************************************************}

Procedure Win4;
{Tela de cópia dos arquivos}

Var PP:Integer; {Utilizado para apagar linhas da tela}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (9,'Copiando Arquivos para o Destino...');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Aguarde!');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
End;

{*************************************************************************}

Procedure Copia;
{Copia um arquivo de origem para um arquivo de destino}

Procedure Copiador;
{Efetua a leitura do arquivo de origem e a gravação do arquivo de destino}

Var FromF, ToF: File; {Arquivos de origem/destino}
    NumRead, NumWritten: Word; {Número de bytes lidos/escritos}
    Buf: Array[1..4096] Of Char; {Buffer de leitura}

Begin
     Assign(FromF, ArqOrig);
{$I-}
     Reset(FromF, 1);
{$I+}
     If IoResult<>0 Then MensErroRead;
     Assign(ToF, DriveDest+':\'+DirDest+'\'+ArqDest);
{$I-}
     Rewrite(ToF, 1);
{$I+}
     If IoResult<>0 Then MensErroWrite;
Repeat
{$I-}
     BlockRead(FromF, Buf, SizeOf(Buf), NumRead);
{$I+}
     If Ioresult<>0 Then MensErroRead;
{$I-}
     BlockWrite(ToF, Buf, NumRead, NumWritten);
{$I+}
     If Ioresult<>0 Then MensErroWrite;
Until (NumRead = 0) Or (NumWritten <> NumRead);
```

```
     Close(FromF);
     Close(ToF);
End;

Begin
Repeat
     LeSetup;
     If ArqDest='' Then Break;
     Window (1,1,80,25);
     TextColor(Yellow);
     TextBackGround(LightCyan);
     CenterTxt(11,'                                                  ');
     CenterTxt(12,'                                                  ');
     CenterTxt(11,'De: '+ArqOrig);
     CenterTxt(12,'Para: '+DriveDest+':\'+DirDest+'\'+ArqDest);
     Window (1,25,1,25);
     TextColor(Black);
     TextBackground(Black);
     Clrscr;
     Copiador;
Until ArqDest='';
End;

{*************************************************************************}

Procedure Fim;
{Tela final do programa}

Begin
     ApagaTela;
     Writeln;
     Writeln ('O PEP Software para Analisador Lógico');
     Writeln ('foi instalado com sucesso!');
     Writeln;
     Writeln ('Para executar o Software, digite no aviso de comando:');
     Writeln (DriveDest+': [ENTER]');
     Writeln ('CD\'+DirDest+' [ENTER]');
     Writeln;
     Writeln ('DIAG    [ENTER] (Realiza um teste na Interface)');
     Writeln ('CONFIG  [ENTER] (Configura o Analisador Lógico)');
     Writeln ('ALOGIC  [ENTER] (Analisador Lógico)');
     Writeln ('DISPLAY [ENTER] (Mostra dados adquiridos pela Interface)');
     Writeln ('COMPARA [ENTER] (Comparador de formas-de-onda)');
     Writeln ('MAKEOSC [ENTER] (Produz formas-de-onda padrão)');
     Writeln;
     Halt(0);
End;

{*************************************************************************}

Begin
     OpenSetup;
     ApagaTela;
     Tela('Analisador Lógico');
     Win1;
     DetectDrives;
     Win2;
     Win3;
     Win4;
     Copia;
     Close (SetupFile);
     Fim;
End.

{*************************************************************************}
{
        * Estrutura do arquivo INSTALL.SET *

Linhas = Nomes dos arquivos de destino da instalação

}
{*************************************************************************}
```

Faz OSC (MAKEOSC.PAS)

```
{$A-,B-,D-,E+,F-,G-,I+,L-,N+,Q-,R-,S+,X+,V+}
{Diretivas de compilação}

{*************************************************************************}

Program Faz_Osc;
```

```
{*************************************************************************}

{
                  Make OSC - Produz .OSC padrão

                       "Os PEP Legais"

                          Versão 1.0
}

{*************************************************************************}

{       Créditos:
                  Luiz Eduardo    - N°19
                  Marcos Paulo    - N°23
                  Rafael Abolafio - N°26
                  Robson Martins  - N°30

                  4°R - Eletrônica - 1997

               Liceu de Artes e Ofícios de São Paulo

               Prof: Marco Antônio Togniazzolo
}
{*************************************************************************}
{
        Make OSC:
                             Produz arquivos .OSC, dados:
                             Frequência, Escala e Duty Cycle
                             Versão 1.0 - 1997

        Software for MS-DOS, Escrito no compilador Turbo Pascal 7.0

        Requisitos Mínimos do Sistema:

                   IBM PC XT/AT ou Compatível
                   640KB de RAM ou mais
                   Monitor EGA ou superior
                   Disk Drive de 1,44MB
                   Hard Disk com pelo menos 1MB livres
}
{*************************************************************************}

Uses Crt,Dos,PEPUnit;
{Units a serem utilizadas}

{*************************************************************************}

Var Freq: Real; {Frequência}
    Duty: Integer; {Duty Cycle}
    Escala: Integer; {Escala Horizontal}
    HiT,LoT: Integer; {Tempo dos Estados de Alto e Baixo}
    ForVar: Integer; {Variável de contador para loops For}
    CurPont: Integer; {Posição corrente da forma-de-onda}
    HiStat, LoStat: Byte; {Estados de Alto e Baixo}
    CanStat: Byte; {Status do Canal}
    Desc: String; {Descrição do conjunto de formas-de-onda}
    DescCan: String; {Descrição de cada canal}
    Tecd: Char; {Armazena a Tecla Pressionada}
    TempA,TempB,TempC,TempD,TempE,TempF,TempG,TempH: File Of Byte;
        {Representam os Arquivos temporários}

{*************************************************************************}

Procedure FatalError;
{Indica Erro de R/W}

Begin
     Window(1,1,80,25);
     TextBackGround(Black);
     Clrscr;
     TextColor (Blue);
     Writeln ('Make OSC do Analisador Lógico - Mensagem de Erro');
     Writeln;
     TextColor (LightRed);
     Writeln ('Ocorreu um Erro Fatal de Leitura/Gravação na unidade corrente');
     Writeln;
     Writeln ('Execute um utilitário de reparação de discos.');
     Writeln ('Após isso, execute novamente este programa.');
     TextColor (LightGray);
     Writeln;
     Writeln ('Caso não consiga solucionar este problema,');
     Writeln ('entre em contato com:');
     Writeln;
```

```
     Textcolor (White);
     Writeln ('"Os PEP Legais" - LAO - 4oR Eletrônica');
     Textcolor (LightGray);
     Writeln;
     Halt(2);
End;

{*************************************************************************}

Function OrdConv (OrdStr:String): Char;
{Transforma String em Char}

Var OrdInt:Byte; {Varia de 0 a 255 e é convertido em char}

Begin
     For OrdInt:=0 To 255 Do Begin
         If OrdStr=Chr(OrdInt) Then Break;
     End;

     If OrdInt=255 Then OrdInt:=0;
     OrdConv:=UpCase(Chr(OrdInt));
End;

{*************************************************************************}

Procedure Formula;
{Cálculo do número de pontos a ser plotados}

Begin
   If CanStat=1 Then Begin
     If Freq<1 Then Begin
        If Duty=100 Then Begin
           HiT:=512;
           LoT:=0;
        End
        Else Begin
             HiT:=0;
             LoT:=512;
        End;
     End
     Else Begin
          HiT:=Round (((1/Freq)*(Duty/100))/((Escala/10)/1000000));
          LoT:=Round (((1/Freq)*((100-Duty)/100))/((Escala/10)/1000000));
     End;
     CurPont:=0;
     If (HiT=0) And (LoT=0) Then Begin
        HiT:=1;
        LoT:=1;
     End;
   End;
End;

{*************************************************************************}

Procedure Corrige(Ca:Char);
{Corrige os arquivos, acrescentando ao fim um byte FFH}

Var ByteCorr:Byte; {Byte de correção}

Begin
     ByteCorr:=$FF;
     If Ca='A' Then Write(TempA,ByteCorr)
     Else If Ca='B' Then Write(TempB,ByteCorr)
     Else If Ca='C' Then Write(TempC,ByteCorr)
     Else If Ca='D' Then Write(TempD,ByteCorr)
     Else If Ca='E' Then Write(TempE,ByteCorr)
     Else If Ca='F' Then Write(TempF,ByteCorr)
     Else If Ca='G' Then Write(TempG,ByteCorr)
     Else If Ca='H' Then Write(TempH,ByteCorr);
End;

{*************************************************************************}

Procedure CanalATmp;
{Cria TMP para o canal A}

Var ByteChar:Byte; {Byte gravado no arquivo TMP}
    ForJK: Integer; {Endereça a descrição}

Begin
     Assign (TempA,'CANALA.TMP');
{$I-}
     Rewrite (TempA);
{$I+}
```

```
     If IoResult<>0 Then FatalError;

     If Length(Desc) > 10 Then Desc:=Copy(Desc,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(Desc,ForJK,1)));
         Write (TempA, ByteChar);
     End;

     If Length(DescCan) > 10 Then DescCan:=Copy(DescCan,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(DescCan,ForJK,1)));
         Write (TempA, ByteChar);
     End;

     Write (TempA, CanStat);

  If CanStat=1 Then Begin
Repeat
   If HiT <> 0 Then Begin
     For ForVar:=1 To HiT Do Begin
         Write (TempA,HiStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
     End;
   End;

     If CurPont=511 Then Break;

   If LoT <> 0 Then Begin
     For ForVar:=1 To LoT Do Begin
         Write (TempA,LoStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
     End;
   End;
Until CurPont=511;
  End
  Else Begin
       ByteChar:=0;
       For ForVar:=21 To 532 Do Begin
           Write (TempA,ByteChar);
       End;
  End;
     Corrige('A');
     Close (TempA);
End;

{*************************************************************************}

Procedure CanalBTmp;
{Cria TMP para o canal B}

Var ByteChar:Byte; {Byte gravado no arquivo TMP}
    ForJK: Integer; {Endereça a descrição}

Begin
     Assign (TempB,'CANALB.TMP');
{$I-}
     Rewrite (TempB);
{$I+}
     If IoResult<>0 Then FatalError;

     If Length(Desc) > 10 Then Desc:=Copy(Desc,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(Desc,ForJK,1)));
         Write (TempB, ByteChar);
     End;

     If Length(DescCan) > 10 Then DescCan:=Copy(DescCan,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(DescCan,ForJK,1)));
         Write (TempB, ByteChar);
     End;

     Write (TempB, CanStat);

  If CanStat=1 Then Begin
Repeat
   If HiT <> 0 Then Begin
     For ForVar:=1 To HiT Do Begin
```

```
         Write (TempB,HiStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
      End;
    End;

      If CurPont=511 Then Break;

    If LoT <> 0 Then Begin
      For ForVar:=1 To LoT Do Begin
         Write (TempB,LoStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
      End;
    End;
Until CurPont=511;
  End
  Else Begin
       ByteChar:=0;
       For ForVar:=21 To 532 Do Begin
           Write (TempB,ByteChar);
       End;
  End;
     Corrige('B');
     Close (TempB);
End;

{*****************************************************************************}

Procedure CanalCTmp;
{Cria TMP para o canal C}

Var ByteChar:Byte; {Byte gravado no arquivo TMP}
    ForJK: Integer; {Endereça a descrição}

Begin
     Assign (TempC,'CANALC.TMP');
{$I-}
     Rewrite (TempC);
{$I+}
     If IoResult<>0 Then FatalError;

     If Length(Desc) > 10 Then Desc:=Copy(Desc,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(Desc,ForJK,1)));
         Write (TempC, ByteChar);
     End;

     If Length(DescCan) > 10 Then DescCan:=Copy(DescCan,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(DescCan,ForJK,1)));
         Write (TempC, ByteChar);
     End;

     Write (TempC, CanStat);

  If CanStat=1 Then Begin
Repeat
   If HiT <> 0 Then Begin
     For ForVar:=1 To HiT Do Begin
         Write (TempC,HiStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
     End;
   End;

     If CurPont=511 Then Break;

   If LoT <> 0 Then Begin
     For ForVar:=1 To LoT Do Begin
         Write (TempC,LoStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
     End;
   End;
Until CurPont=511;
  End
  Else Begin
       ByteChar:=0;
       For ForVar:=21 To 532 Do Begin
           Write (TempC,ByteChar);
       End;
```

```
  End;
     Corrige('C');
     Close (TempC);
End;

{*************************************************************************}

Procedure CanalDTmp;
{Cria TMP para o canal D}

Var ByteChar:Byte; {Byte gravado no arquivo TMP}
    ForJK: Integer; {Endereça a descrição}

Begin
     Assign (TempD,'CANALD.TMP');
{$I-}
     Rewrite (TempD);
{$I+}
     If IoResult<>0 Then FatalError;

     If Length(Desc) > 10 Then Desc:=Copy(Desc,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(Desc,ForJK,1)));
         Write (TempD, ByteChar);
     End;

     If Length(DescCan) > 10 Then DescCan:=Copy(DescCan,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(DescCan,ForJK,1)));
         Write (TempD, ByteChar);
     End;

     Write (TempD, CanStat);

  If CanStat=1 Then Begin
Repeat
   If HiT <> 0 Then Begin
     For ForVar:=1 To HiT Do Begin
         Write (TempD,HiStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
     End;
   End;

     If CurPont=511 Then Break;

   If LoT <> 0 Then Begin
     For ForVar:=1 To LoT Do Begin
         Write (TempD,LoStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
     End;
   End;
Until CurPont=511;
  End
  Else Begin
       ByteChar:=0;
       For ForVar:=21 To 532 Do Begin
           Write (TempD,ByteChar);
       End;
  End;
     Corrige('D');
     Close (TempD);
End;

{*************************************************************************}

Procedure CanalETmp;
{Cria TMP para o canal E}

Var ByteChar:Byte; {Byte gravado no arquivo TMP}
    ForJK: Integer; {Endereça a descrição}

Begin
     Assign (TempE,'CANALE.TMP');
{$I-}
     Rewrite (TempE);
{$I+}
     If IoResult<>0 Then FatalError;

     If Length(Desc) > 10 Then Desc:=Copy(Desc,1,10);
```

```
     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(Desc,ForJK,1)));
         Write (TempE, ByteChar);
     End;

     If Length(DescCan) > 10 Then DescCan:=Copy(DescCan,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(DescCan,ForJK,1)));
         Write (TempE, ByteChar);
     End;

     Write (TempE, CanStat);

  If CanStat=1 Then Begin
Repeat
   If HiT <> 0 Then Begin
     For ForVar:=1 To HiT Do Begin
         Write (TempE,HiStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
     End;
   End;

     If CurPont=511 Then Break;

   If LoT <> 0 Then Begin
     For ForVar:=1 To LoT Do Begin
         Write (TempE,LoStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
     End;
   End;
Until CurPont=511;
  End
  Else Begin
       ByteChar:=0;
       For ForVar:=21 To 532 Do Begin
           Write (TempE,ByteChar);
       End;
  End;
     Corrige('E');
     Close (TempE);
End;

{*************************************************************************}

Procedure CanalFTmp;
{Cria TMP para o canal F}

Var ByteChar:Byte; {Byte gravado no arquivo TMP}
    ForJK: Integer; {Endereça a descrição}

Begin
     Assign (TempF,'CANALF.TMP');
{$I-}
     Rewrite (TempF);
{$I+}
     If IoResult<>0 Then FatalError;

     If Length(Desc) > 10 Then Desc:=Copy(Desc,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(Desc,ForJK,1)));
         Write (TempF, ByteChar);
     End;

     If Length(DescCan) > 10 Then DescCan:=Copy(DescCan,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(DescCan,ForJK,1)));
         Write (TempF, ByteChar);
     End;

     Write (TempF, CanStat);

  If CanStat=1 Then Begin
Repeat
   If HiT <> 0 Then Begin

     For ForVar:=1 To HiT Do Begin
         Write (TempF,HiStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
```

```
     End;
   End;

     If CurPont=511 Then Break;

   If LoT <> 0 Then Begin
     For ForVar:=1 To LoT Do Begin
         Write (TempF,LoStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
     End;
   End;
Until CurPont=511;
  End
  Else Begin
       ByteChar:=0;
       For ForVar:=21 To 532 Do Begin
           Write (TempF,ByteChar);
       End;
  End;
     Corrige('F');
     Close (TempF);
End;

{*************************************************************************}

Procedure CanalGTmp;
{Cria TMP para o canal G}

Var ByteChar:Byte; {Byte gravado no arquivo TMP}
    ForJK: Integer; {Endereça a descrição}

Begin
     Assign (TempG,'CANALG.TMP');
{$I-}
     Rewrite (TempG);
{$I+}
     If IoResult<>0 Then FatalError;

     If Length(Desc) > 10 Then Desc:=Copy(Desc,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(Desc,ForJK,1)));
         Write (TempG, ByteChar);
     End;

     If Length(DescCan) > 10 Then DescCan:=Copy(DescCan,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(DescCan,ForJK,1)));
         Write (TempG, ByteChar);
     End;

     Write (TempG, CanStat);

  If CanStat=1 Then Begin
Repeat
   If HiT <> 0 Then Begin
     For ForVar:=1 To HiT Do Begin
         Write (TempG,HiStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
     End;
   End;

     If CurPont=511 Then Break;

   If LoT <> 0 Then Begin
     For ForVar:=1 To LoT Do Begin
         Write (TempG,LoStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
     End;
   End;
Until CurPont=511;
  End
  Else Begin
       ByteChar:=0;
       For ForVar:=21 To 532 Do Begin
           Write (TempG,ByteChar);
       End;
  End;
     Corrige('G');
     Close (TempG);
```

```
End;

{*************************************************************************}

Procedure CanalHTmp;
{Cria TMP para o canal H}

Var ByteChar:Byte; {Byte gravado no arquivo TMP}
    ForJK: Integer; {Endereça a descrição}

Begin
     Assign (TempH,'CANALH.TMP');
{$I-}
     Rewrite (TempH);
{$I+}
     If IoResult<>0 Then FatalError;

     If Length(Desc) > 10 Then Desc:=Copy(Desc,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(Desc,ForJK,1)));
         Write (TempH, ByteChar);
     End;

     If Length(DescCan) > 10 Then DescCan:=Copy(DescCan,1,10);

     For ForJK:=1 To 10 Do Begin
         ByteChar:=Ord(OrdConv(Copy(DescCan,ForJK,1)));
         Write (TempH, ByteChar);
     End;

     Write (TempH, CanStat);

  If CanStat=1 Then Begin
Repeat
   If HiT <> 0 Then Begin
     For ForVar:=1 To HiT Do Begin
         Write (TempH,HiStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
     End;
   End;

     If CurPont=511 Then Break;

   If LoT <> 0 Then Begin
     For ForVar:=1 To LoT Do Begin
         Write (TempH,LoStat);
         If CurPont=511 Then Break;
         CurPont:=CurPont+1;
     End;
   End;
Until CurPont=511;
  End
  Else Begin
       ByteChar:=0;
       For ForVar:=21 To 532 Do Begin
           Write (TempH,ByteChar);
       End;
  End;
     Corrige('H');
     Close (TempH);
End;

{*************************************************************************}

Procedure ApagaTela;
{Apaga a tela corrente}

Begin
     Window(1,1,80,25);
     TextColor(LightGray);
     TextBackGround(Black);
     Clrscr;
End;

{*************************************************************************}

Procedure Fim;
{Finaliza a execução do programa}


Begin
     ApagaTela;
```

```
     Writeln;
     Halt(0);
End;

{*************************************************************************}

Procedure Win1;
{Tela Inicial do Programa - Apresentação}

Var Tecla:Char; {Caracter lido do teclado}

Begin
     Window (1,1,80,25);
     TextColor (Yellow);
     TextBackground (Cyan);
     CenterTxt (9,'Programa para produção de');
     CenterTxt (10,'Arquivos do Analisador Lógico (.OSC)');
     TextColor (White);
     CenterTxt (13,'"Os PEP Legais"');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Pressione Enter para Continuar ou Esc para Sair');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

Repeat
      Tecla:=Readkey;
      If Tecla=#27 Then Fim;
Until Tecla=#13;

End;

{*************************************************************************}

Procedure Win2;
{Recolhe Descrição do Conjunto de Formas-de-onda}

Var PP:Integer; {Utilizado para apagar linhas da tela}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (9,'Digite a Descrição do Conjunto de');
     CenterTxt (10,'Formas de Onda:');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Máximo de caracteres utilizados: 10');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

     Window (1,1,80,25);
     TextBackGround(LightCyan);
     TextColor (Yellow);
     GotoXy (35,12);
     Readln (Desc);
End;

{*************************************************************************}

Procedure Win3(Cn:Char);
{Recolhe o Estado do canal Cn: ON ou OFF}

Var PP:Integer; {Utilizado para apagar linhas da tela}
    Tecla:Char; {Caracter lido do teclado}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
```

```
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (10,'O Canal '+Cn+' estará ativado (S,N)?');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Pressione S ou N');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

Repeat
      Tecla:=UpCase(Readkey);
Until (Tecla='S') Or (Tecla='N');

     If Tecla='S' Then CanStat:=1
     Else CanStat:=0;
End;

{*************************************************************************}

Procedure Win4(Cn:Char);
{Recolhe a descrição do canal Cn}

Var PP:Integer; {Utilizado para apagar linhas da tela}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (9,'Digite a Descrição do Canal '+Cn+':');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Máximo de caracteres utilizados: 10');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

     Window (1,1,80,25);
     TextBackGround(LightCyan);
     TextColor (Yellow);
     GotoXy (35,12);
     Readln (DescCan);
End;

{*************************************************************************}

Procedure Win5(Cn:Char);
{Recolhe a frequência do canal Cn}

Var PP:Integer; {Utilizado para apagar linhas da tela}
    FrStr:String; {Frequência em formato String}
    Code:Integer; {Resultado da conversão String => Integer}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
```

```
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (9,'Digite a Frequência em Hz para o Canal '+Cn+':');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Digite um valor Real (de 0 a 10MHz)');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

     Window (1,1,80,25);
     TextBackGround(LightCyan);
     TextColor (Yellow);
Repeat
     CenterTxt (12,'                                        ');
     GotoXy (35,12);
     Readln (FrStr);
     Val(FrStr,Freq,Code);
     If (Code<>0) Or (Freq>10e6) Then Begin
        Sound(1000);
        Delay(400);
        Nosound;
     End;
Until (Code=0) And (Freq<=10e6);
End;

{*************************************************************************}

Procedure Win6(Cn:Char);
{Recolhe o Duty Cycle do canal Cn}

Var PP:Integer;  {Utilizado para apagar linhas da tela}
    DCStr:String; {Duty Cycle em formato String}
    Code:Integer; {Resultado da conversão String => Integer}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (9,'Digite o Duty Cycle em % para o Canal '+Cn+':');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Digite um valor Inteiro (de 0 a 100%)');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

     Window (1,1,80,25);
     TextBackGround(LightCyan);
     TextColor (Yellow);
Repeat
     CenterTxt (12,'                                        ');
     GotoXy (35,12);
     Readln (DCStr);
     Val(DCStr,Duty,Code);
     If (Code<>0) Or (Duty>100) Then Begin
        Sound(1000);
        Delay(400);
        Nosound;
     End;
Until (Code=0) And (Duty<=100);
End;

{*************************************************************************}

Procedure Win7;
{Recolhe a Escala Horizontal}

Var PP:Integer; {Utilizado para apagar linhas da tela}
    Tecla:Char; {Caracter lido do teclado}
```

```
Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (8,'Escolha a Escala Horizontal:');
     CenterTxt (10,'A = 1 uS/div     ');
     CenterTxt (11,'B = 10 uS/div    ');
     CenterTxt (12,'C = 100 uS/div   ');
     CenterTxt (13,'D = 1.000 uS/div ');
     CenterTxt (14,'E = 10.000 uS/div');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Pressione a tecla correspondente');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

Repeat
      Tecla:=UpCase(Readkey);
Until (Tecla='A') Or (Tecla='B') Or (Tecla='C') Or (Tecla='D') Or (Tecla='E');

     If Tecla='A' Then Escala:=1
     Else If Tecla='B' Then Escala:=10
     Else If Tecla='C' Then Escala:=100
     Else If Tecla='D' Then Escala:=1000
     Else If Tecla='E' Then Escala:=10000;
End;

{*************************************************************************}

Procedure Win9;
{Tela Final do Programa}

Var PP:Integer; {Utilizado para apagar linhas da tela}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (8,'Arquivo .OSC gravado com sucesso!');
     CenterTxt (10,'Pressione ESC para sair ou');
     CenterTxt (11,'ENTER para produzir mais um arquivo .OSC');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Pressione ESC ou ENTER');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;
End;

{*************************************************************************}

Procedure Recolhe(Can:Char);
{Recolhe os dados de cada canal Can}

Begin
     DescCan:='';
     Freq:=0;
     Duty:=0;
     If Can='A' Then Begin
        Win2;
```

```
        Win7;
     End;
     Win3(Can);
     If CanStat=1 Then Begin
        Win4(Can);
        Win5(Can);
        Win6(Can);
     End;
End;

{*************************************************************************}

Procedure FazOsc;
{Grava no formato .OSC}

Var NomeOsc: String; {Nome do arquivo .OSC}
    OscF,A,B,C,D,E,F,G,H:File Of Byte; {Arquivos .OSC e TMPs}
    Dad, DA,DB,DC,DD,DE,DF,DG,DH:Byte; {Dados lidos dos arquivos}
    CoF:Integer; {Cursor do arquivo}

Procedure Win8;
{Recolhe o nome do arquivo a ser gravado}

Var PP:Integer; {Utilizado para apagar linhas da tela}
    Tecla:Char; {Caracter lido do teclado}
    Io:Integer; {Resultado da última operação de I/O}

Begin
     Window (1,1,80,25);
     TextBackground (White);
     TextColor (Black);
     For PP:=2 To 60 Do Begin
         Gotoxy (PP,24);
         Write (' ');
     End;

     ApagaWin;
     AmpliaWin (50,10,White,Cyan);
     CenterWin (50,10,White,Cyan);
     Window (1,1,80,25);
     TextColor (White);
     TextBackGround (Cyan);
     CenterTxt (7,'Digite o Nome do Arquivo .OSC a ser gravado');
     CenterTxt (8,'(Com caminho completo, se necessário');
     CenterTxt (9,'e sem extensão):');
     Textcolor (Black);
     TextBackGround (White);
     Gotoxy (2,24); Write ('Digite o Nome do Arquivo .OSC');
     Window (1,25,1,25);
     TextBackground(Black);
     Clrscr;

     Window (1,1,80,25);
     TextBackGround(LightCyan);
     TextColor (Yellow);
Repeat
Repeat
     CenterTxt (13,'                                        ');
     CenterTxt (14,'                                        ');
     CenterTxt (11,'                                        ');
     GotoXy (25,11);
     Readln (NomeOsc);
     Assign (OscF, NomeOsc+'.OSC');
{$I-}
     Reset (OscF);
{$I+}
     Io:=IoResult;
     If Io=0 Then Begin
        CenterTxt (13,'O arquivo especificado já existe!');
        CenterTxt (14,'Substituir (S,N)?');
Repeat
        Tecla:=Upcase(Readkey);
Until (Tecla='S') Or (Tecla='N');
        If Tecla='S' Then Break;
     End;
Until Io<>0;
{$I-}
     If Io=0 Then Close (OscF);
     Rewrite (OscF);
{$I+}
     Io:=IoResult;
     If Io<>0 Then Begin
        Sound(1000);
        Delay(400);
```

```
        Nosound;
     End;
Until Io=0;
End;

Procedure OpenFiles;
{Abre os arquivos TMP}

Begin
     Assign (A,'CANALA.TMP');
{$I-}
     Reset (A);
     If IoResult<>0 Then FatalError;
     Assign (B,'CANALB.TMP');
     Reset (B);
     If IoResult<>0 Then FatalError;
     Assign (C,'CANALC.TMP');
     Reset (C);
     If IoResult<>0 Then FatalError;
     Assign (D,'CANALD.TMP');
     Reset (D);
     If IoResult<>0 Then FatalError;
     Assign (E,'CANALE.TMP');
     Reset (E);
     If IoResult<>0 Then FatalError;
     Assign (F,'CANALF.TMP');
     Reset (F);
     If IoResult<>0 Then FatalError;
     Assign (G,'CANALG.TMP');
     Reset (G);
     If IoResult<>0 Then FatalError;
     Assign (H,'CANALH.TMP');
     Reset (H);
     If IoResult<>0 Then FatalError;
{$I+}
End;

Procedure CloseFiles;
{Fecha os arquivos TMP}

Begin
{$I-}
     Close (A);
     Erase (A);
     If IoResult<>0 Then FatalError;
     Close (B);
     Erase (B);
     If IoResult<>0 Then FatalError;
     Close (C);
     Erase (C);
     If IoResult<>0 Then FatalError;
     Close (D);
     Erase (D);
     If IoResult<>0 Then FatalError;
     Close (E);
     Erase (E);
     If IoResult<>0 Then FatalError;
     Close (F);
     Erase (F);
     If IoResult<>0 Then FatalError;
     Close (G);
     Erase (G);
     If IoResult<>0 Then FatalError;
     Close (H);
     Erase (H);
     If IoResult<>0 Then FatalError;
     Close (OscF);
     If IoResult<>0 Then FatalError;
{$I+}
End;

Begin
     Win8;
     Dad:=Ord('P');
     Write (OscF,Dad);
     Dad:=Ord('E');
     Write (OscF,Dad);
     Dad:=Ord('P');
     Write (OscF,Dad);
     OpenFiles;
     For CoF:=0 To 9 Do Begin
         Read (A, Dad);
         Read (B, Dad);
         Read (C, Dad);
```

```
          Read (D, Dad);
          Read (E, Dad);
          Read (F, Dad);
          Read (G, Dad);
          Read (H, Dad);
          Write (OscF,Dad);
     End;
     For CoF:=10 To 19 Do Begin
          Read (A, Dad);
          Write (OscF,Dad);
     End;
     For CoF:=10 To 19 Do Begin
          Read (B, Dad);
          Write (OscF,Dad);
     End;
     For CoF:=10 To 19 Do Begin
          Read (C, Dad);
          Write (OscF,Dad);
     End;
     For CoF:=10 To 19 Do Begin
          Read (D, Dad);
          Write (OscF,Dad);
     End;
     For CoF:=10 To 19 Do Begin
          Read (E, Dad);
          Write (OscF,Dad);
     End;
     For CoF:=10 To 19 Do Begin
          Read (F, Dad);
          Write (OscF,Dad);
     End;
     For CoF:=10 To 19 Do Begin
          Read (G, Dad);
          Write (OscF,Dad);
     End;
     For CoF:=10 To 19 Do Begin
          Read (H, Dad);
          Write (OscF,Dad);
     End;
     Read (A, Dad);
     Write (OscF,Dad);
     Read (B, Dad);
     Write (OscF,Dad);
     Read (C, Dad);
     Write (OscF,Dad);
     Read (D, Dad);
     Write (OscF,Dad);
     Read (E, Dad);
     Write (OscF,Dad);
     Read (F, Dad);
     Write (OscF,Dad);
     Read (G, Dad);
     Write (OscF,Dad);
     Read (H, Dad);
     Write (OscF,Dad);

     If Escala=1 Then Dad:=0
     Else If Escala=10 Then Dad:=1
     Else If Escala=100 Then Dad:=2
     Else If Escala=1000 Then Dad:=3
     Else If Escala=10000 Then Dad:=4;

     Write (OscF,Dad);

     For CoF:=21 To 532 Do Begin
          Read (A, DA);
          Read (B, DB);
          Read (C, DC);
          Read (D, DD);
          Read (E, DE);
          Read (F, DF);
          Read (G, DG);
          Read (H, DH);
          Dad:=(DH*128) Or (DG*64) Or (DF*32) Or (DE*16) Or (DD*8) Or (DC*4) Or
               (DB*2) Or DA;
          Write (OscF,Dad);
     End;
     CloseFiles;
End;

{*************************************************************************}

Begin
Repeat
```

```
     ApagaTela;
     Tela('Produtor de Arquivos .OSC');
     Win1;
     HiStat:=1;
     LoStat:=0;
     Recolhe ('A');
     Formula;
     CanalATmp;
     Recolhe ('B');
     Formula;
     CanalBTmp;
     Recolhe ('C');
     Formula;
     CanalCTmp;
     Recolhe ('D');
     Formula;
     CanalDTmp;
     Recolhe ('E');
     Formula;
     CanalETmp;
     Recolhe ('F');
     Formula;
     CanalFTmp;
     Recolhe ('G');
     Formula;
     CanalGTmp;
     Recolhe ('H');
     Formula;
     CanalHTmp;
     FazOsc;
     Win9;
Repeat
      Tecd:=Readkey;
Until (Tecd=#27) Or (Tecd=#13);
Until Tecd=#27;
     Fim;
End.

{*************************************************************************}
{
    * Formato dos arquivos TMP: *

Byte 0: Descrição do Conjunto
Byte 10: Descrição do Canal
Byte 20: Estado do Canal (0 - OFF ou 1 - ON)
Byte 21 a Byte 532: Dados (Bits)

}
{*************************************************************************}
```

ANALISADOR LÓGICO PARA PC

APÊNDICE F

INFORMAÇÕES SOBRE OS COMPONENTES UTILIZADOS

National Semiconductor

March 1988

CD4017BM/CD4017BC Decade Counter/Divider with 10 Decoded Outputs CD4022BM/CD4022BC Divide-by-8 Counter/Divider with 8 Decoded Outputs

CD4017BM/CD4017BC Decade Counter/Divider with 10 Decoded Outputs
CD4022BM/CD4022BC Divide-by-8 Counter/Divider with 8 Decoded Outputs

General Description

The CD4017BM/CD4017BC is a 5-stage divide-by-10 Johnson counter with 10 decoded outputs and a carry out bit.

The CD4022BM/CD4022BC is a 4-stage divide-by-8 Johnson counter with 8 decoded outputs and a carry-out bit.

These counters are cleared to their zero count by a logical "1" on their reset line. These counters are advanced on the positive edge of the clock signal when the clock enable signal is in the logical "0" state.

The configuration of the CD4017BM/CD4017BC and CD4022BM/CD4022BC permits medium speed operation and assures a hazard free counting sequence. The 10/8 decoded outputs are normally in the logical "0" state and go to the logical "1" state only at their respective time slot. Each decoded output remains high for 1 full clock cycle. The carry-out signal completes a full cycle for every 10/8 clock input cycles and is used as a ripple carry signal to any succeeding stages.

Features

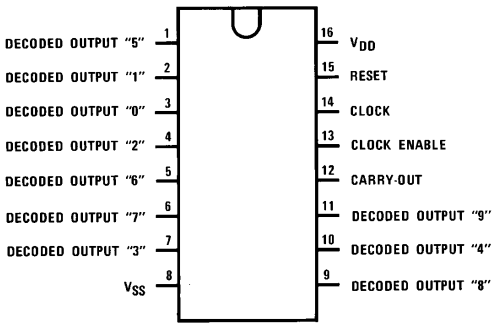
■ Wide supply voltage range	3.0V to 15V
■ High noise immunity	0.45 V_{DD} (typ.)
■ Low power TTL compatibility	Fan out of 2 driving 74L or 1 driving 74LS
■ Medium speed operation	5.0 MHz (typ.) with 10V V_{DD}
■ Low power	10 μW (typ.)
■ Fully static operation	

Applications

- Automotive
- Instrumentation
- Medical electronics
- Alarm systems
- Industrial electronics
- Remote metering

Connection Diagrams

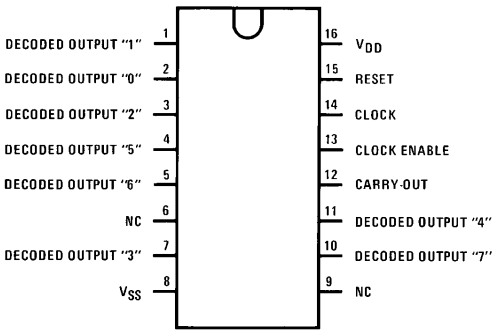
CD4017B
Dual-In-Line Package



TL/F/5950–1

Top View

CD4022B
Dual-In-Line Package



TL/F/5950–2

Top View

Order Number CD4017B or CD4022B

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings (Notes 1 & 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

DC Supply Voltage (V_{DD})	$-0.5\ V_{DC}$ to $+18\ V_{DC}$
Input Voltage (V_{IN})	$-0.5\ V_{DC}$ to $V_{DD} + 0.5\ V_{DC}$
Storage Temperature (T_S)	$-65°C$ to $+150°C$
Power Dissipation (P_D)	
Dual-In-Line	700 mW
Small Outline	500 mW
Lead Temperature (T_L)	
(Soldering, 10 seconds)	260°C

Recommended Operating Conditions (Note 2)

DC Supply Voltage (V_{DD})	$+3\ V_{DC}$ to $+15\ V_{DC}$
Input Voltage (V_{IN})	0 to $V_{DD}\ V_{DC}$
Operating Temperature Range (T_A)	
CD4017BM, CD4022BM	$-55°C$ to $+125°C$
CD4017BC, CD4022BC	$-40°C$ to $+85°C$

DC Electrical Characteristics CD4017BM, CD4022BM (Note 2)

Symbol	Parameter	Conditions	−55°C		+25°			+125°C		Units
			Min	Max	Min	Typ	Max	Min	Max	
I_{DD}	Quiescent Device Current	$V_{DD} = 5V$, $V_{IN} = V_{DD}$ or V_{SS}		5		0.3	5		150	μA
		$V_{DD} = 10V$, $V_{IN} = V_{DD}$ or V_{SS}		10		0.5	10		300	μA
		$V_{DD} = 15V$, $V_{IN} = V_{DD}$ or V_{SS}		20		1.0	20		600	μA
V_{OL}	Low Level Output Voltage	$\|I_O\| < 1.0\ \mu A$								
		$V_{DD} = 5V$		0.05		0	0.05		0.05	V
		$V_{DD} = 10V$		0.05		0	0.05		0.05	V
		$V_{DD} = 15V$		0.05		0	0.05		0.05	V
V_{OH}	High Level Output Voltage	$\|I_O\| < 1.0\ \mu A$								
		$V_{DD} = 5V$	4.95		4.95	5		4.95		V
		$V_{DD} = 10V$	9.95		9.95	10		9.95		V
		$V_{DD} = 15V$	14.95		14.95	15		14.95		V
V_{IL}	Low Level Input Voltage	$\|I_O\| < 1.0\ \mu A$								
		$V_{DD} = 5V$, $V_O = 0.5V$ or $4.5V$		1.5			1.5		1.5	V
		$V_{DD} = 10V$, $V_O = 1.0V$ or $9.0V$		3.0			3.0		3.0	V
		$V_{DD} = 15V$, $V_O = 1.5V$ or $13.5V$		4.0			4.0		4.0	V
V_{IH}	High Level Input Voltage	$\|I_O\| < 1.0\ \mu A$								
		$V_{DD} = 5V$, $V_O = 0.5V$ or $4.5V$	3.5		3.5			3.5		V
		$V_{DD} = 10V$, $V_O = 1.0V$ or $9.0V$	7.0		7.0			7.0		V
		$V_{DD} = 15V$, $V_O = 1.5V$ or $13.5V$	11.0		11.0			11.0		V
I_{OL}	Low Level Output Current (Note 3)	$V_{DD} = 5V$, $V_O = 0.4V$	0.64		0.51	0.88		0.36		mA
		$V_{DD} = 10V$, $V_O = 0.5V$	1.6		1.3	2.25		0.9		mA
		$V_{DD} = 15V$, $V_O = 1.5V$	4.2		3.4	8.8		2.4		mA
I_{OH}	High Level Output Current (Note 3)	$V_{DD} = 5V$, $V_O = 4.6V$	−0.25		−0.2	−0.36		−0.14		mA
		$V_{DD} = 10V$, $V_O = 9.5V$	−0.62		−0.5	−0.9		−0.35		mA
		$V_{DD} = 15V$, $V_O = 13.5V$	−1.8		−1.5	−3.5		−1.1		mA
I_{IN}	Input Current	$V_{DD} = 15V$, $V_{IN} = 0V$		−0.1		-10^{-5}	−0.1		−1.0	μA
		$V_{DD} = 15V$, $V_{IN} = 15V$		0.1		10^{-5}	0.1		1.0	μA

DC Electrical Characteristics CD4017BC, CD4022BC (Note 2)

Symbol	Parameter	Conditions	−40°C		+25°			+85°C		Units
			Min	Max	Min	Typ	Max	Min	Max	
I_{DD}	Quiescent Device Current	$V_{DD} = 5V$		20		0.5	20		150	μA
		$V_{DD} = 10V$		40		1.0	40		300	μA
		$V_{DD} = 15V$		80		5.0	80		600	μA
V_{OL}	Low Level Output Voltage	$\|I_O\| < 1.0\ \mu A$								
		$V_{DD} = 5V$		0.05		0	0.05		0.05	V
		$V_{DD} = 10V$		0.05		0	0.05		0.05	V
		$V_{DD} = 15V$		0.05		0	0.05		0.05	V
V_{OH}	High Level Output Voltage	$\|I_O\| < 1.0\ \mu A$								
		$V_{DD} = 5V$	4.95		4.95	5		4.95		V
		$V_{DD} = 10V$	9.95		9.95	10		9.95		V
		$V_{DD} = 15V$	14.95		14.95	15		14.95		V

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed, they are not meant to imply that the devices should be operated at these limits. The table of "Recommended Operating Conditions" and "Electrical Characteristics" provides conditions for actual device operation.

Note 2: $V_{SS} = 0V$ unless otherwise specified.

Note 3: I_{OL} and I_{OH} are tested one output at a time.

DC Electrical Characteristics CD4017BC, CD4022BC (Note 2) (Continued)

Symbol	Parameter	Conditions	−40°C		+25°			+85°C		Units
			Min	Max	Min	Typ	Max	Min	Max	
V_{IL}	Low Level Input Voltage	$\|I_O\| < 1.0\ \mu A$ $V_{DD} = 5V$, $V_O = 0.5V$ or $4.5V$ $V_{DD} = 10V$, $V_O = 1.0V$ or $9.0V$ $V_{DD} = 15V$, $V_O = 1.5V$ or $13.5V$		 1.5 3.0 4.0			 1.5 3.0 4.0		 1.5 3.0 4.0	 V V V
V_{IH}	High Level Input Voltage	$\|I_O\| < 1.0\ \mu A$ $V_{DD} = 5V$, $V_O = 0.5V$ or $4.5V$ $V_{DD} = 10V$, $V_O = 1.0V$ or $9.0V$ $V_{DD} = 15V$, $V_O = 1.5V$ or $13.5V$	 3.5 7.0 11.0		 3.5 7.0 11.0			 3.5 7.0 11.0		 V V V
I_{OL}	Low Level Output Current (Note 3)	$V_{DD} = 5V$, $V_O = 0.4V$ $V_{DD} = 10V$, $V_O = 0.5V$ $V_{DD} = 15V$, $V_O = 1.5V$	0.52 1.3 3.6		0.44 1.1 3.0	0.88 2.25 8.8		0.36 0.9 2.4		mA mA mA
I_{OH}	High Level Output Current (Note 3)	$V_{DD} = 5V$, $V_O = 4.6V$ $V_{DD} = 10V$, $V_O = 9.5V$ $V_{DD} = 15V$, $V_O = 13.5V$	−0.2 −0.5 −1.4		−0.16 −0.4 −1.2	−0.36 −0.9 −3.5		−0.12 −0.3 −1.0		mA mA mA
I_{IN}	Input Current	$V_{DD} = 15V$, $V_{IN} = 0V$ $V_{DD} = 15V$, $V_{IN} = 15V$		−0.3 0.3		-10^{-5} 10^{-5}	−0.3 0.3		−1.0 1.0	μA μA

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed, they are not meant to imply that the devices should be operated at these limits. The table of "Recommended Operating Conditions" and "Electrical Characteristics" provides condtions for actual device operation.

Note 2: $V_{SS} = 0V$ unless otherwise specified.

Note 3: I_{OL} and I_{OH} are tested one output at a time.

AC Electrical Characteristics*

$T_A = 25°C$, $C_L = 50$ pF, $R_L = 200k$, t_{rCL} and $t_{fCL} = 20$ ns, unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
CLOCK OPERATION						
t_{PHL}, t_{PLH}	Propagation Delay Time Carry Out Line	 $V_{DD} = 5V$ $V_{DD} = 10V$ $V_{DD} = 15V$		 415 160 130	 800 320 250	 ns ns ns
	Carry Out Line	$V_{DD} = 5V$ $V_{DD} = 10V$ $V_{DD} = 15V$ } $C_L = 15$ pF		240 85 70	480 170 140	ns ns ns
	Decode Out Lines	$V_{DD} = 5V$ $V_{DD} = 10V$ $V_{DD} = 15V$		500 200 160	1000 400 320	ns ns ns
t_{TLH}, t_{THL}	Transition Time Carry Out and Decode Out Lines t_{TLH}	 $V_{DD} = 5V$ $V_{DD} = 10V$ $V_{DD} = 15V$		 200 100 80	 360 180 130	 ns ns ns
	t_{THL}	$V_{DD} = 5V$ $V_{DD} = 10V$ $V_{DD} = 15V$		100 50 40	200 100 80	ns ns ns
f_{CL}	Maximum Clock Frequency	$V_{DD} = 5V$ $V_{DD} = 10V$ $V_{DD} = 15V$ } Measured with Respect to Carry Output Line	1.0 2.5 3.0	2 5 6		MHz MHz MHz
t_{WL}, t_{WH}	Minimum Clock Pulse Width	$V_{DD} = 5V$ $V_{DD} = 10V$ $V_{DD} = 15V$		125 45 35	250 90 70	ns ns ns
t_{rCL}, t_{fCL}	Clock Rise and Fall Time	$V_{DD} = 5V$ $V_{DD} = 10V$ $V_{DD} = 15V$			20 15 5	μs μs μs
t_{SU}	Minimum Clock Inhibit Data Setup Time	$V_{DD} = 5V$ $V_{DD} = 10V$ $V_{DD} = 15V$		120 40 32	240 80 65	ns ns ns
C_{IN}	Average Input Capacitance			5	7.5	pF

AC Electrical Characteristics*

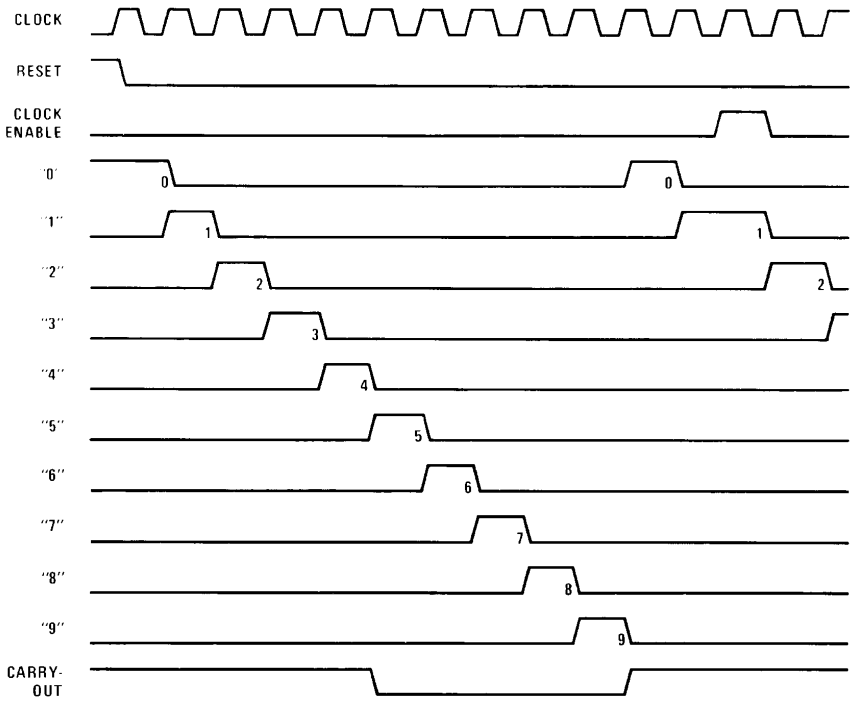
$T_A = 25°C$, $C_L = 50$ pF, $R_L = 200k$, t_{rCL} and $t_{fCL} = 20$ ns, unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
RESET OPERATION						
t_{PHL}, t_{PLH}	Propagation Delay Time Carry Out Line					
		$V_{DD} = 5V$		415	800	ns
		$V_{DD} = 10V$		160	320	ns
		$V_{DD} = 15V$		130	250	ns
	Carry Out Line	$V_{DD} = 5V$ } $C_L = 15$ pF		240	480	ns
		$V_{DD} = 10V$ } $C_L = 15$ pF		85	170	ns
		$V_{DD} = 15V$ } $C_L = 15$ pF		70	140	ns
	Decode Out Lines	$V_{DD} = 5V$		500	1000	ns
		$V_{DD} = 10V$		200	400	ns
		$V_{DD} = 15V$		160	320	ns
t_W	Minimum Reset Pulse Width	$V_{DD} = 5V$		200	400	ns
		$V_{DD} = 10V$		70	140	ns
		$V_{DD} = 15V$		55	110	ns
t_{REM}	Minimum Reset Removal Time	$V_{DD} = 5V$		75	150	ns
		$V_{DD} = 10V$		30	60	ns
		$V_{DD} = 15V$		25	50	ns

*AC Parameters are guaranteed by DC correlated testing.

Timing Diagrams

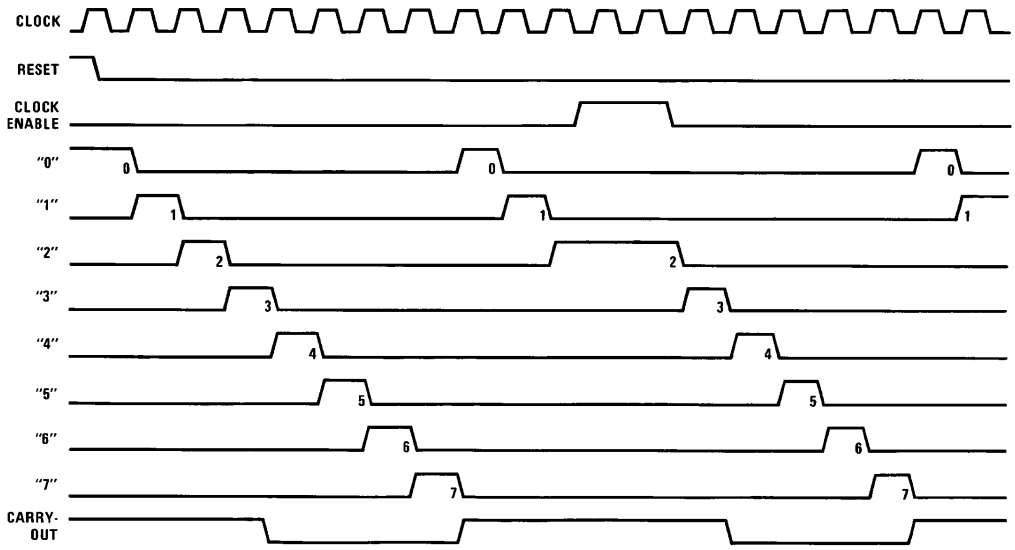
CD4017B



TL/F/5950–3

Timing Diagrams (Continued)

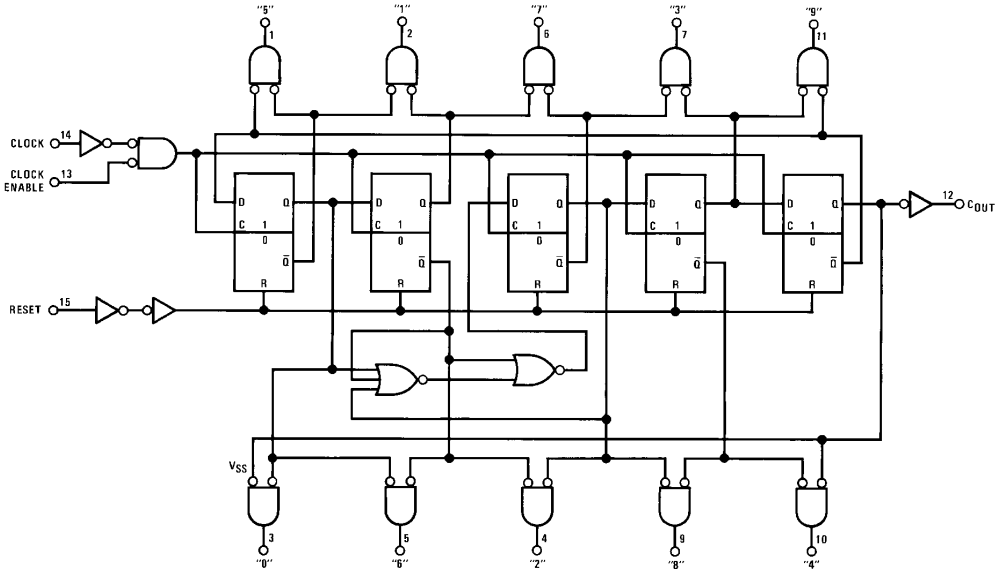
CD4022B



TL/F/5950–4

Logic Diagrams

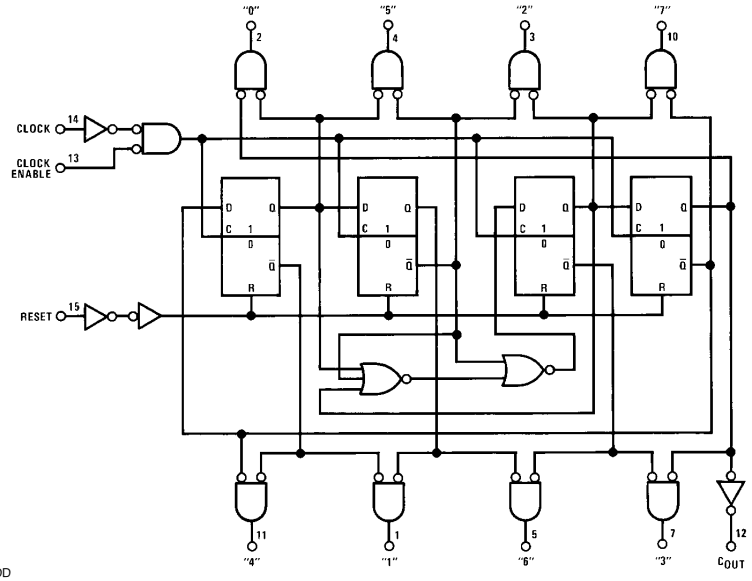
CD4017B



Terminal No. 8 = GND
Terminal No. 16 = V_{DD}

TL/F/5950–5

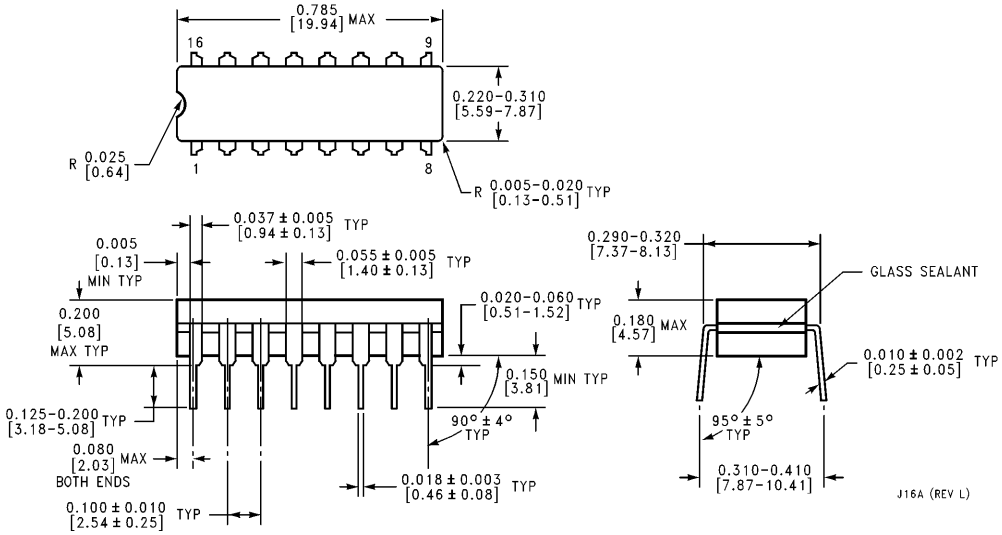
CD4022B



Terminal No. 16 = V_{DD}
Terminal No. 8 = GND

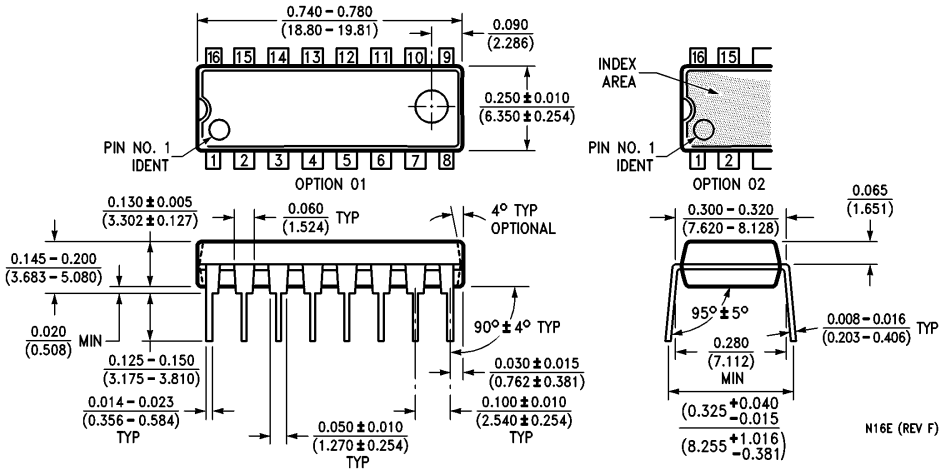
TL/F/5950–6

Physical Dimensions inches (millimeters)



Ceramic Dual-In-Line Package (J)
Order Number CD4017BMJ, CD4017BCJ, CD4022BMJ, CD4022BCJ
NS Package Number J16A

Physical Dimensions inches (millimeters) (Continued)



Molded Dual-In-Line Package (N)
Order Number CD4017BMN, CD4017BCN, CD4022BMN, CD4022BCN
NS Package Number N16E

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

June 1989

54LS04/DM54LS04/DM74LS04 Hex Inverting Gates

General Description

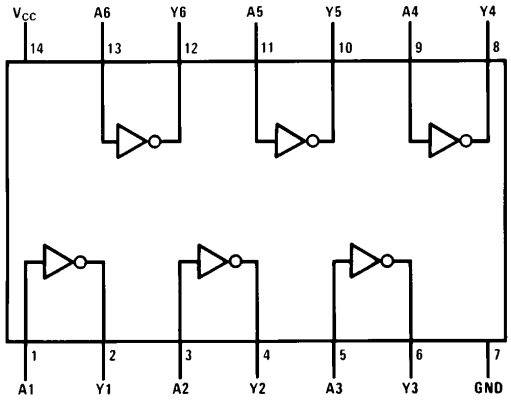
This device contains six independent gates each of which performs the logic INVERT function.

Features

- Alternate Military/Aerospace device (54LS04) is available. Contact a National Semiconductor Sales Office/ Distributor for specifications.

Connection Diagram

Dual-In-Line Package



TL/F/6345–1

Order Number 54LS04DMQB, 54LS04FMQB, 54LS04LMQB, DM54LS04J, DM54LS04W, DM74LS04M or DM74LS04N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$Y = \overline{A}$

Input	Output
A	Y
L	H
H	L

H = High Logic Level

L = Low Logic Level

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM54LS and 54LS	−55°C to +125°C
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM54LS04			DM74LS04			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.7			0.8	V
I_{OH}	High Level Output Current			−0.4			−0.4	mA
I_{OL}	Low Level Output Current			4			8	mA
T_A	Free Air Operating Temperature	−55		125	0		70	°C

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

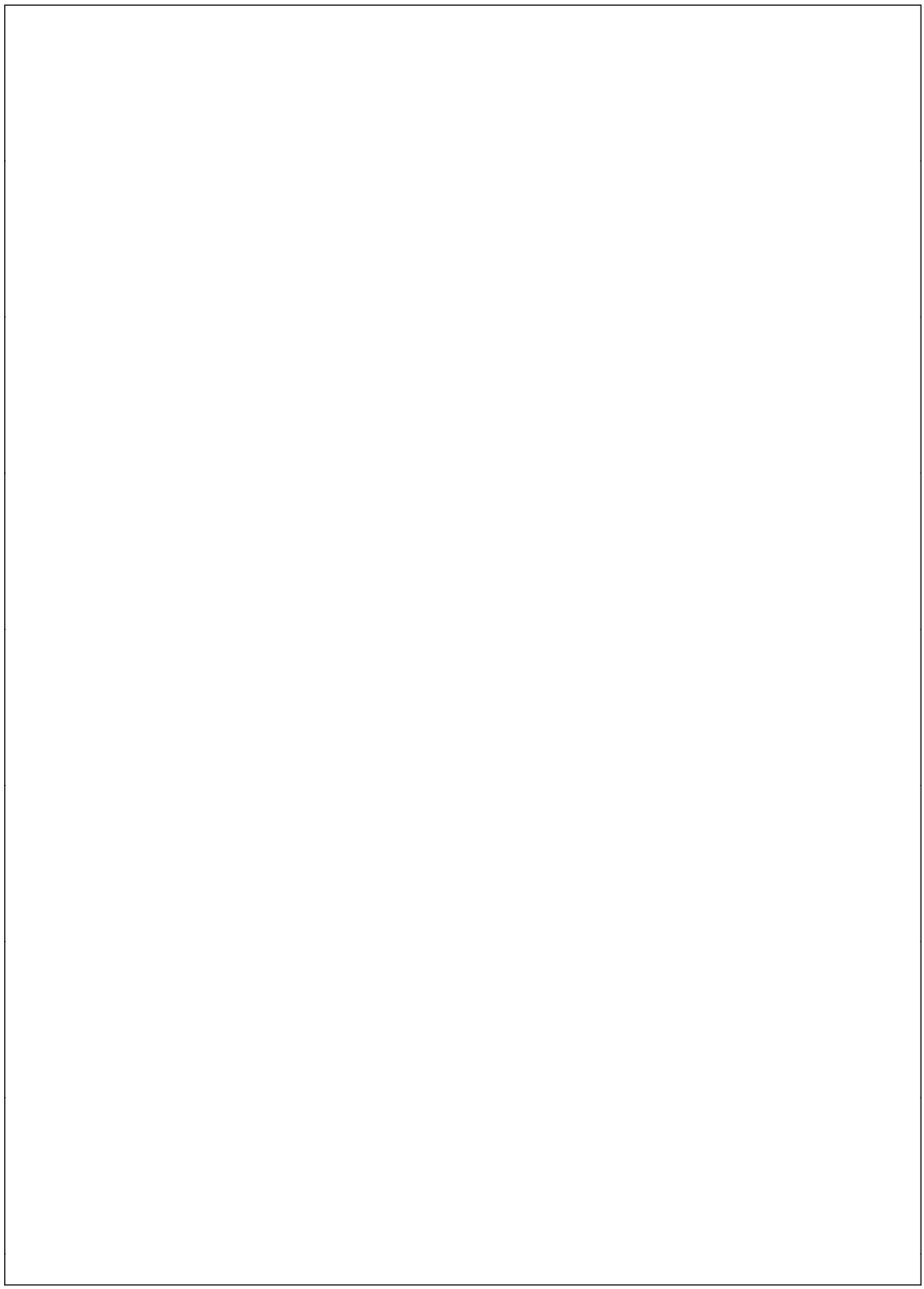
Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max	DM54	2.5	3.4		V
			DM74	2.7	3.4		
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IH} = Min	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 4 mA, V_{CC} = Min	DM74		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.36	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−20		−100	mA
			DM74	−20		−100	
I_{CCH}	Supply Current with Outputs High	V_{CC} = Max			1.2	2.4	mA
I_{CCL}	Supply Current with Outputs Low	V_{CC} = Max			3.6	6.6	mA

Switching Characteristics at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load)

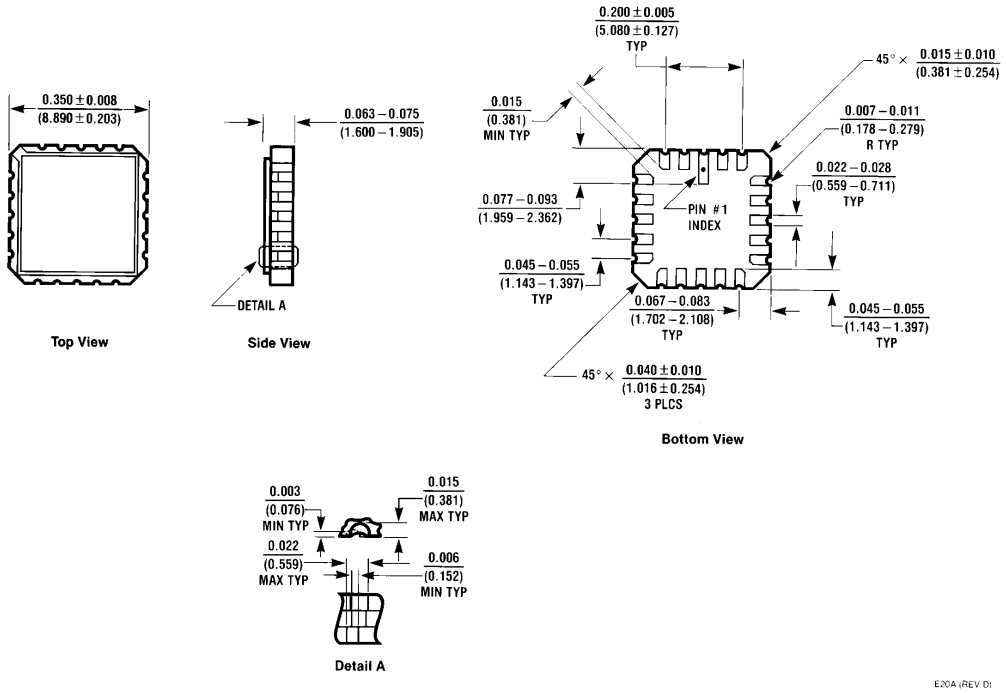
Symbol	Parameter	R_L = 2 kΩ				Units
		C_L = 15 pF		C_L = 50 pF		
		Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	3	10	4	15	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	3	10	4	15	ns

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

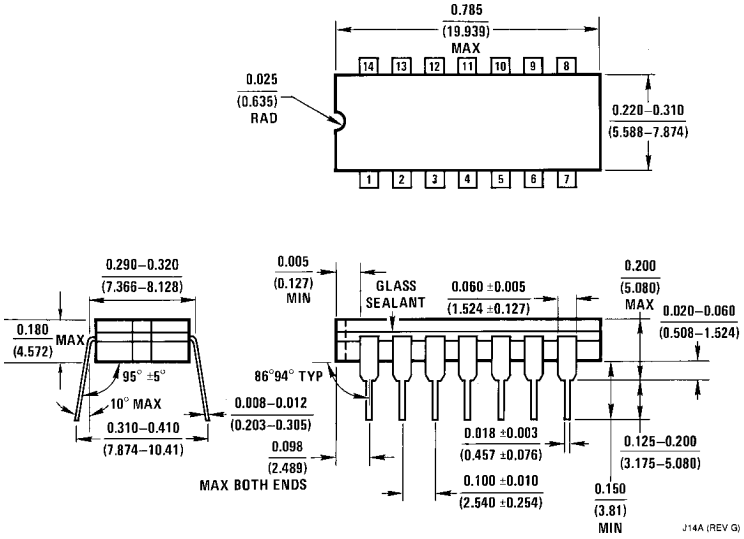
Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.



Physical Dimensions inches (millimeters)

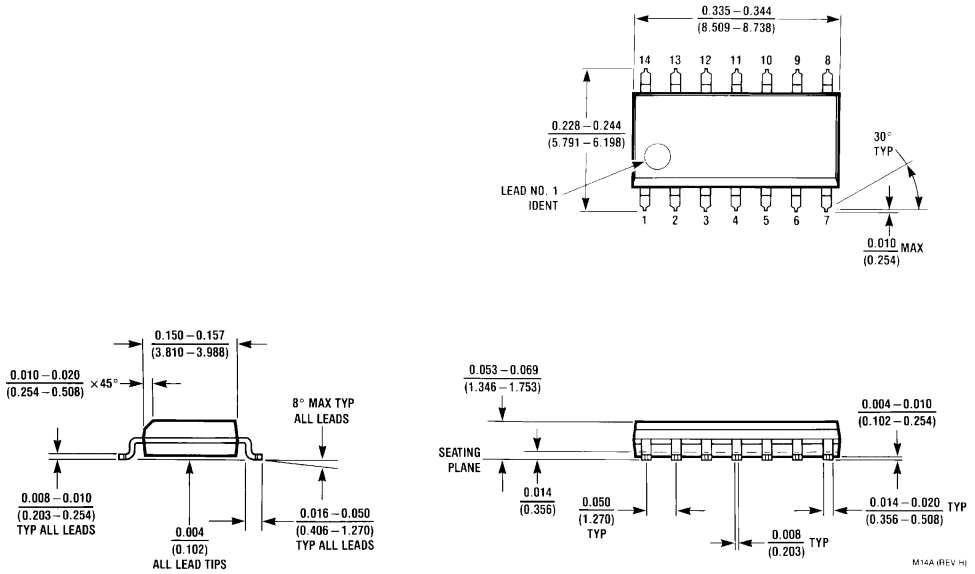


Ceramic Leadless Chip Carrier Package (E)
Order Number 54LS04LMQB
NS Package Number E20A

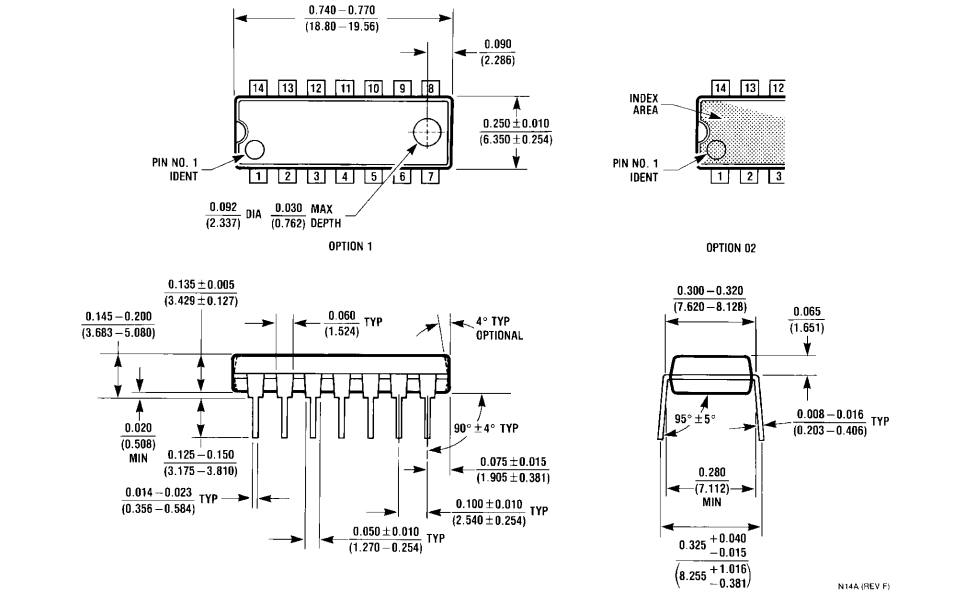


14-Lead Ceramic Dual-In-Line Package (J)
Order Number 54LS04DMQB or DM54LS04J
NS Package Number J14A

Physical Dimensions inches (millimeters) (Continued)

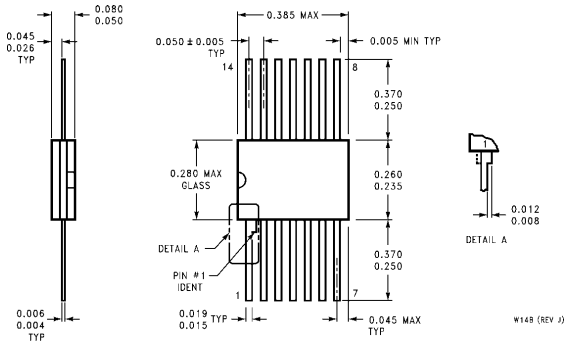


14-Lead Small Outline Molded Package (M)
Order Number DM74LS04M
NS Package Number M14A



14-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS04N
NS Package Number N14A

Physical Dimensions inches (millimeters) (Continued)



14-Lead Ceramic Flat Package (W)
Order Number 54LS04FMQB or DM54LS04W
NS Package Number W14B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

June 1989

54LS08/DM54LS08/DM74LS08 Quad 2-Input AND Gates

General Description

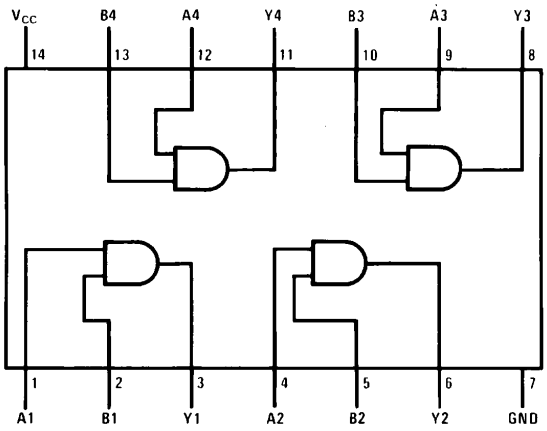
This device contains four independent gates each of which performs the logic AND function.

Features

- Alternate Military/Aerospace device (54LS08) is available. Contact a National Semiconductor Sales Office/ Distributor for specifications.

Connection Diagram

Dual-In-Line Package



TL/F/6347–1

Order Number 54LS08DMQB, 54LS08FMQB, 54LS08LMQB, DM54LS08J, DM54LS08W, DM74LS08M or DM74LS08N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$Y = AB$

Inputs		Output
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

H = High Logic Level
L = Low Logic Level

  RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM54LS and 54LS	−55°C to +125°C
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM54LS08			DM74LS08			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.7			0.8	V
I_{OH}	High Level Output Current			−0.4			−0.4	mA
I_{OL}	Low Level Output Current			4			8	mA
T_A	Free Air Operating Temperature	−55		125	0		70	°C

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

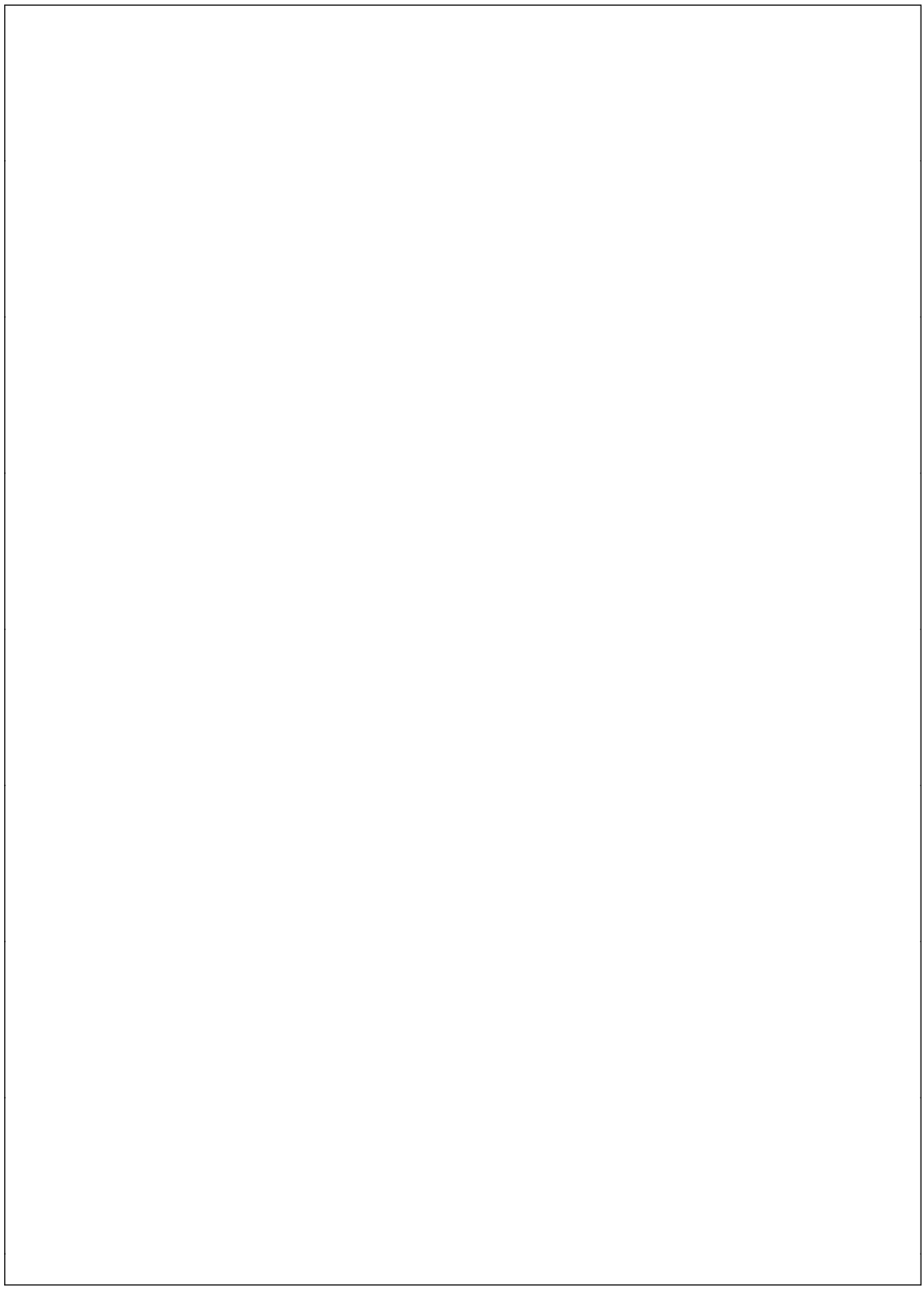
Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IH} = Min	DM54	2.5	3.4		V
			DM74	2.7	3.4		
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 4 mA, V_{CC} = Min	DM74		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.36	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−20		−100	mA
			DM74	−20		−100	
I_{CCH}	Supply Current with Outputs High	V_{CC} = Max			2.4	4.8	mA
I_{CCL}	Supply Current with Outputs Low	V_{CC} = Max			4.4	8.8	mA

Switching Characteristics at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load)

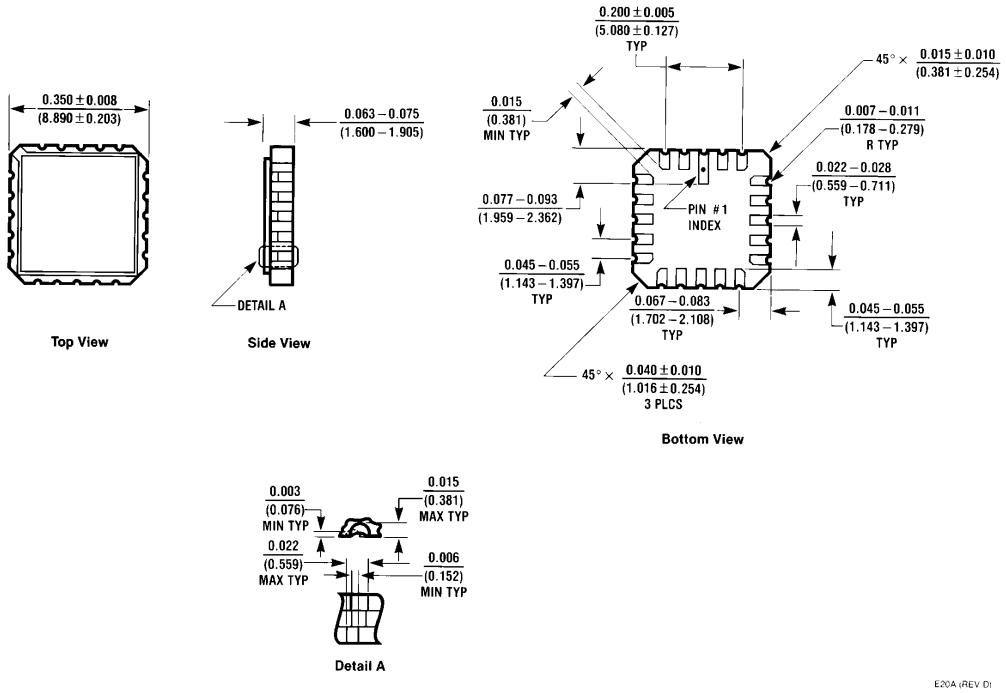
Symbol	Parameter	R_L = 2 kΩ				Units
		C_L = 15 pF		C_L = 50 pF		
		Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	4	13	6	18	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	3	11	5	18	ns

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

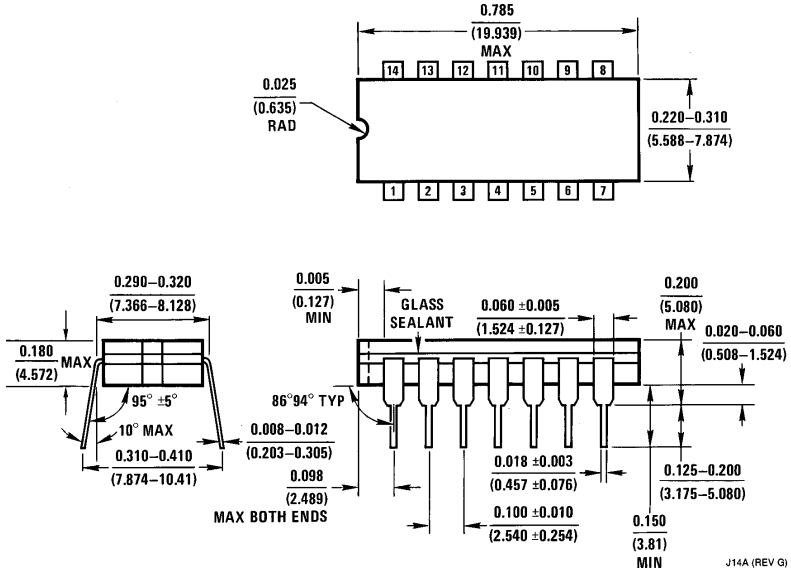
Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.



Physical Dimensions inches (millimeters)

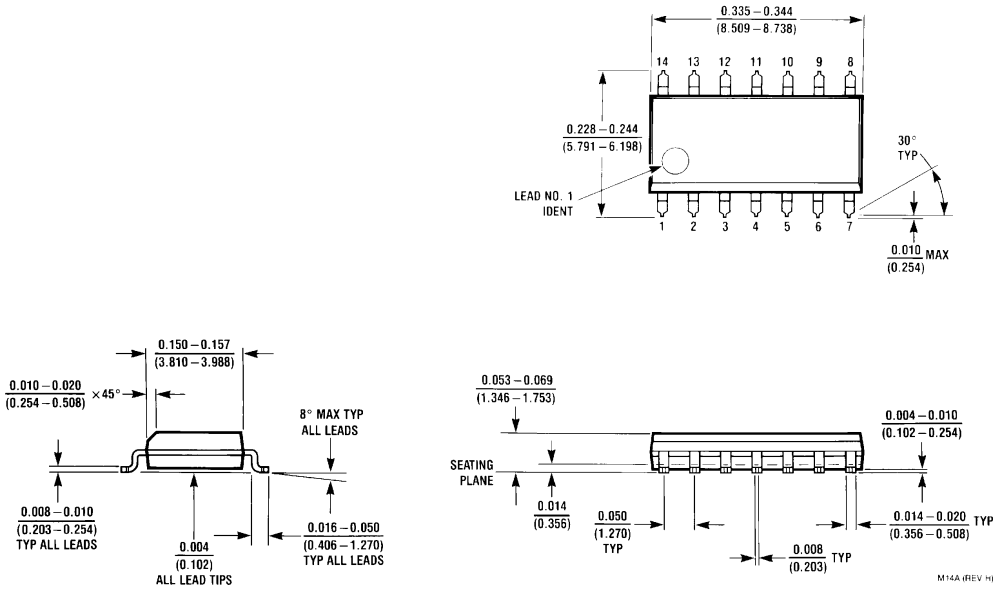


Ceramic Leadless Chip Carrier Package (E)
Order Number 54LS08LMQB
NS Package Number E20A

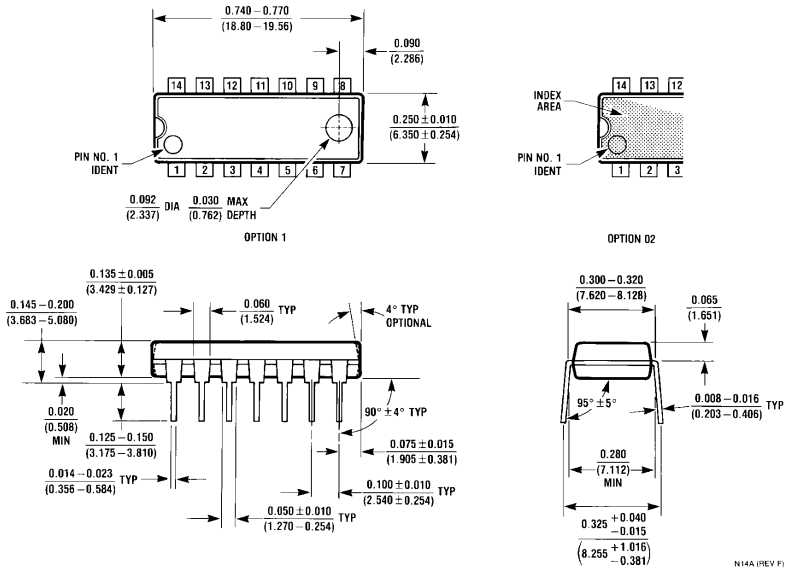


14-Lead Ceramic Dual-In-Line Package (J)
Order Number 54LS08DMQB or DM54LS08J
NS Package Number J14A

Physical Dimensions inches (millimeters) (Continued)

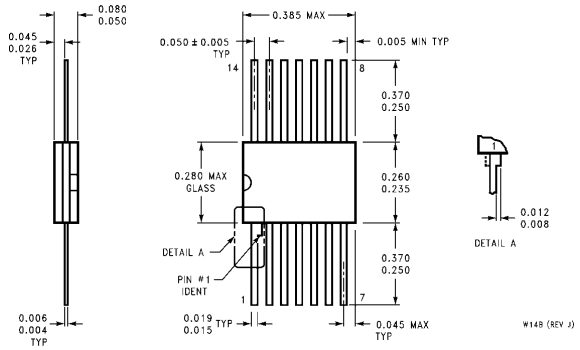


14-Lead Small Outline Molded Package (M)
Order Number DM74LS08M
NS Package Number M14A



14-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS08N
NS Package Number N14A

Physical Dimensions inches (millimeters) (Continued)



14-Lead Ceramic Flat Package (W)
Order Number 54LS08FMQB or DM54LS08W
NS Package Number W14B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

June 1989

54LS125A/DM54LS125A/DM74LS125A Quad TRI-STATE® Buffers

General Description

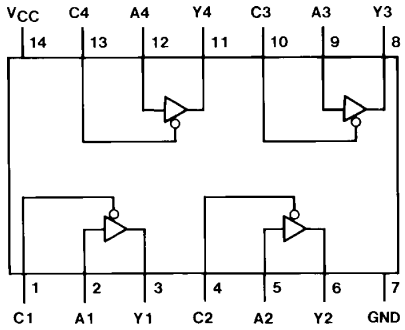
This device contains four independent gates each of which performs a non-inverting buffer function. The outputs have the TRI-STATE feature. When enabled, the outputs exhibit the low impedance characteristics of a standard LS output with additional drive capability to permit the driving of bus lines without external resistors. When disabled, both the output transistors are turned off presenting a high-impedance state to the bus line. Thus the output will act neither as a significant load nor as a driver. To minimize the possibility that two outputs will attempt to take a common bus to opposite logic levels, the disable time is shorter than the enable time of the outputs.

Features

- Alternate Military/Aerospace device (54LS125) is available. Contact a National Semiconductor Sales Office/ Distributor for specifications.

Connection Diagram

Dual-In-Line Package



TL/F/6387–1

Order Number 54LS125ADMQB, 54LS125AFMQB, 54LS125ALMQB, DM54LS125AJ, DM54LS125AW, DM74LS125AM or DM74LS125AN
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$Y = A$

Inputs		Output
A	C	Y
L	L	L
H	L	H
X	H	Hi-Z

H = High Logic Level

L = Low Logic Level

X = Either Low or High Logic Level

Hi-Z = TRI-STATE (Outputs are disabled)

TRI-STATE® is a registered trademark of National Semiconductor Corporation.

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM54LS and 54LS	−55°C to +125°C
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM54LS125A			DM74LS125A			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.7			0.8	V
I_{OH}	High Level Output Current			−1			−2.6	mA
I_{OL}	Low Level Output Current			12			24	mA
T_A	Free Air Operating Temperature	−55		125	0		70	°C

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max V_{IL} = Max, V_{IH} = Min		2.4	3.4		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max V_{IL} = Max	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 12 mA, V_{CC} = Min	DM74		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.4	mA
I_{OZH}	Off-State Output Current with High Level Output Voltage Applied	V_{CC} = Max, V_O = 2.4V V_{IH} = Min, V_{IL} = Max				20	μA
I_{OZL}	Off-State Output Current with Low Level Output Voltage Applied	V_{CC} = Max, V_O = 0.4V V_{IH} = Min, V_{IL} = Max				−20	μA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−20		−100	mA
			DM74	−20		−100	
I_{CC}	Supply Current	V_{CC} = Max (Note 3)			11	20	mA

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

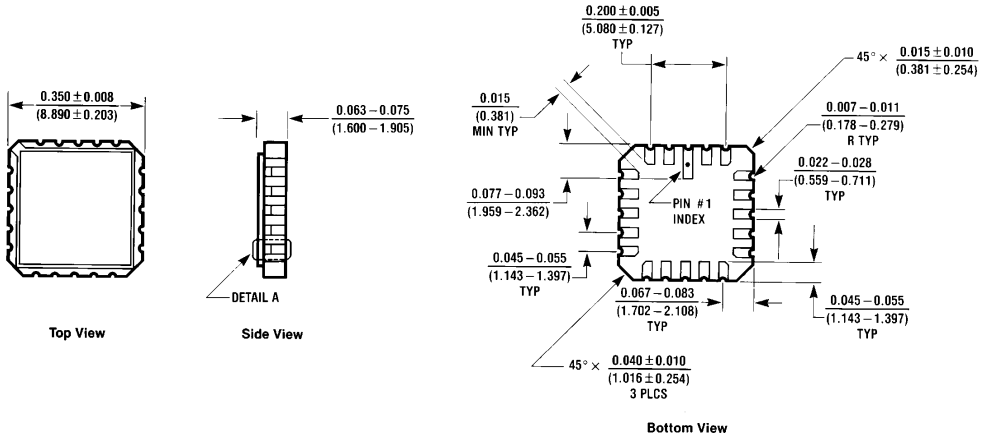
Note 3: I_{CC} is measured with the data control (C) inputs at 4.5V and the data inputs grounded.

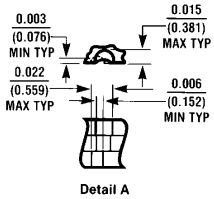
Switching Characteristics at $V_{CC} = 5V$ and $T_A = 25°C$ (See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	$R_L = 667\Omega$				Units
		$C_L = 50$ pF		$C_L = 150$ pF		
		Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output		15		21	ns
t_{PHL}	Propagation Delay Time High to Low Level Output		18		22	ns
t_{PZH}	Output Enable Time to High Level Output		25		35	ns
t_{PZL}	Output Enable Time to Low Level Output		25		40	ns
t_{PHZ}	Output Disable Time from High Level Output (Note 1)		20			ns
t_{PLZ}	Output Disable Time from Low Level Output (Note 1)		20			ns

Note 1: $C_L = 5$pF.

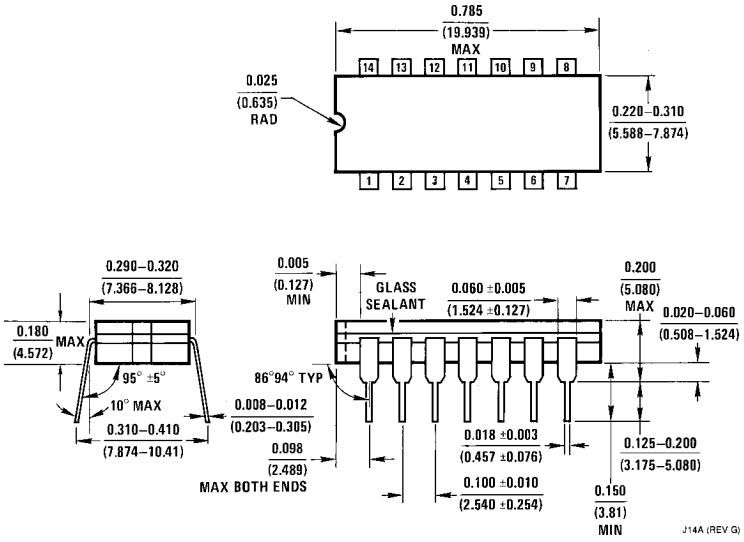
Physical Dimensions inches (millimeters)





E20A (REV D)

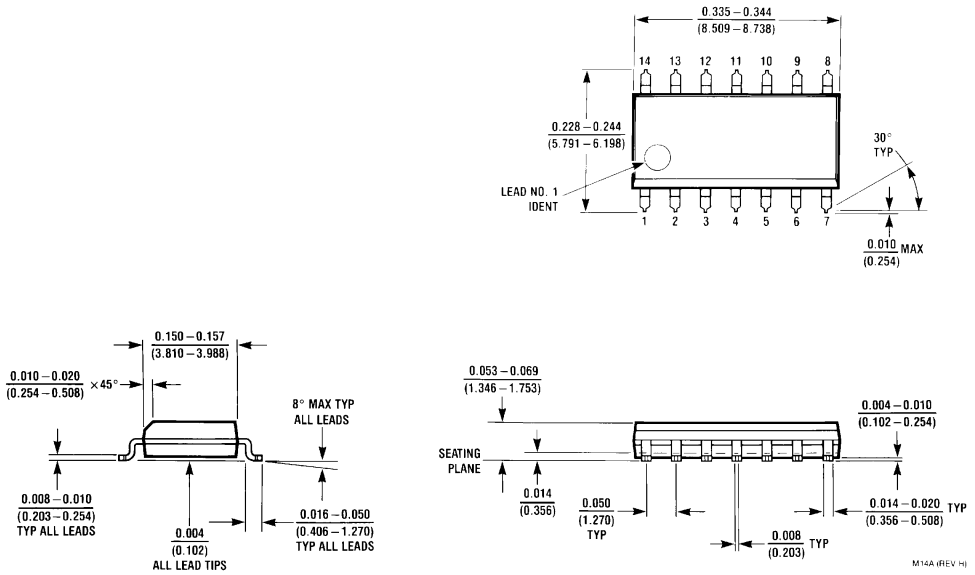
Ceramic Leadless Chip Carrier Package (E)
Order Number 54LS125ALMQB
NS Package Number E20A



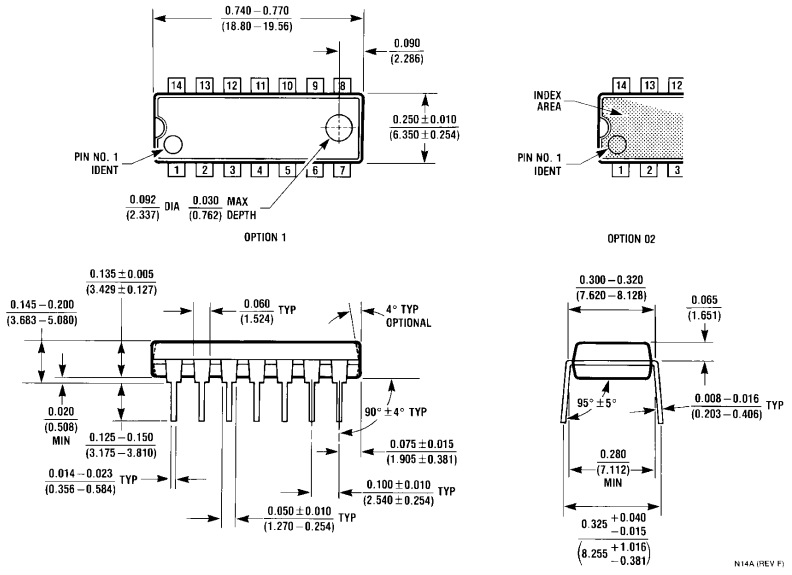
J14A (REV G)

14-Lead Ceramic Dual-In-Line Package (J)
Order Number 54LS125ADMQB or DM54LS125AJ
NS Package Number J14A

Physical Dimensions inches (millimeters) (Continued)

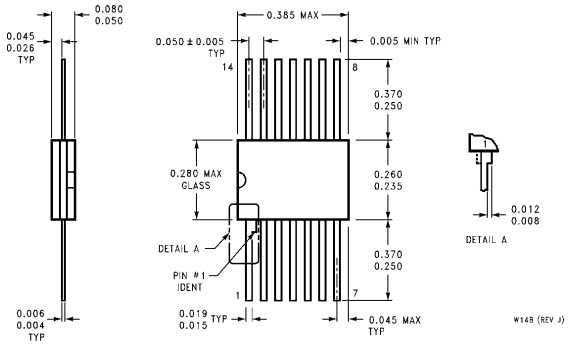


14-Lead Small Outline Molded Package (M)
Order Number DM74LS125AM
NS Package Number M14A



14-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS125AN
NS Package Number N14A

Physical Dimensions inches (millimeters) (Continued)



14-Lead Ceramic Flat Package (W)
Order Number 54LS125AFMQB or DM54LS125AW
NS Package Number W14B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

September 1991

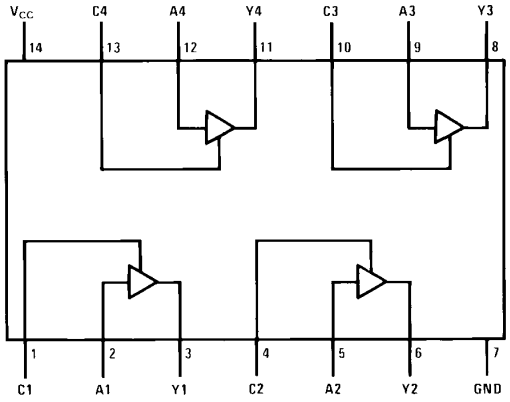
DM74LS126A Quad TRI-STATE® Buffer

General Description

This device contains four independent gates each of which performs a non-inverting buffer function. The outputs have the TRI-STATE feature. When enabled, the outputs exhibit the low impedance characteristics of a standard LS output with additional drive capability to permit the driving of bus lines without external resistors. When disabled, both the output transistors are turned off presenting a high-impedance state to the bus line. Thus the output will act neither as a significant load nor as a driver. To minimize the possibility that two outputs will attempt to take a common bus to opposite logic levels, the disable time is shorter than the enable time of the outputs.

Connection Diagram

Dual-In-Line Package



TL/F/6388–1

Order Number DM74LS126AM or DM74LS126AN
See NS Package Number M14A or N14A

Function Table

$Y = A$

Inputs		Output
A	C	Y
L	H	L
H	H	H
X	L	Hi-Z

H = High Logic Level
L = Low Logic Level
X = Either Low or High Logic Level
Hi-Z = TRI-STATE (Outputs are disabled)

TRI-STATE® is a registered trademark of National Semiconductor Corporation.

  RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings (Note)

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V_{CC}	Supply Voltage	4.75	5	5.25	V
V_{IH}	High Level Input Voltage				V
V_{IL}	Low Level Input Voltage			0.8	V
I_{OH}	High Level Output Current			−2.6	mA
I_{OL}	Low Level Output Current			24	mA
T_A	Free Air Operating Temperature	0		70	°C

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max V_{IH} = Min	2.4			V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max V_{IL} = Max, V_{IH} = Min		0.35	0.5	V
		I_{OL} = 12 mA, V_{CC} = Min		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V			0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V			20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V			−0.4	mA
I_{OZH}	Off-State Output Current with High Level Output Voltage Applied	V_{CC} = Max, V_O = 2.4V V_{IH} = Min, V_{IL} = Max			20	μA
I_{OZL}	Off-State Output Current with Low Level Output Voltage Applied	V_{CC} = Max, V_O = 0.4V V_{IH} = Min, V_{IL} = Max			−20	μA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	−20		−100	mA
I_{CC}	Supply Current	V_{CC} = Max		12	22	mA

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

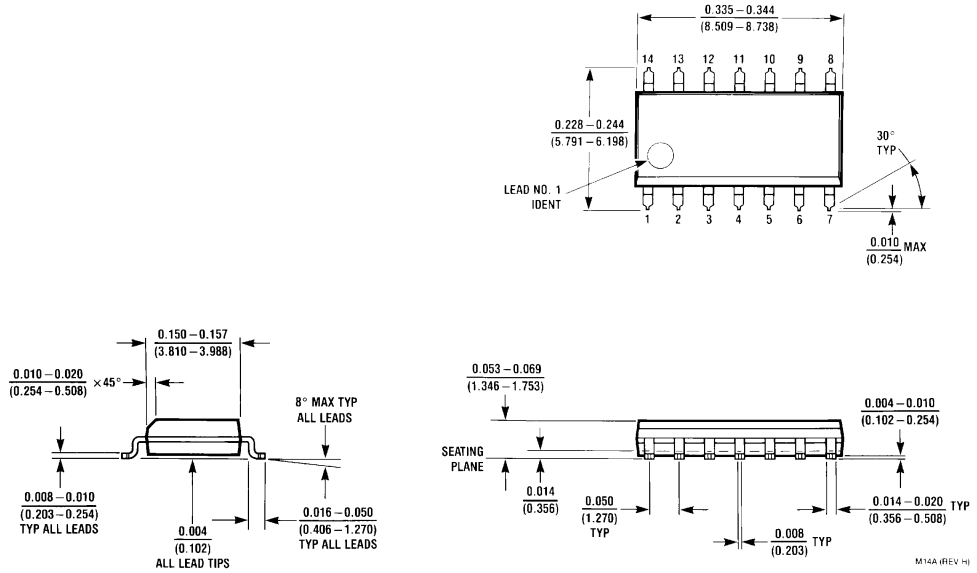
Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Switching Characteristics:

Symbol	Parameter	DM74LS				Units
		$R_L = 667\Omega$				
		$C_L = 50$ pF		$C_L = 150$ pF		
		Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output		15		21	ns
t_{PHL}	Propagation Delay Time High to Low Level Output		18		22	ns
t_{PZH}	Output Enable Time to High Level Output		30		36	ns
t_{PZL}	Output Enable Time to Low Level Output		30		42	ns
t_{PHZ}	Output Disable Time from High Level Output (Note 1)		25			ns
t_{PLZ}	Output Disable Time from Low Level Output (Note 1)		25			ns

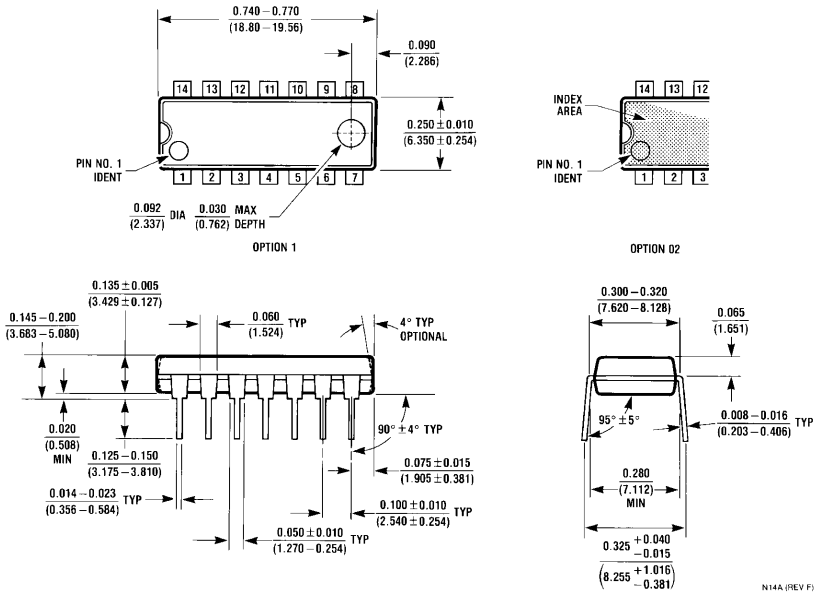
Note 1: $C_L = 5$pF.

Physical Dimensions inches (millimeters)



14-Lead Small Outline Molded Package (M)
Order Number DM74LS126AM
NS Package Number M14A

Physical Dimensions inches (millimeters) (Continued)



14-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS126AN
NS Package Number N14A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

May 1992

54LS161A/DM54LS161A/DM74LS161A, 54LS163A/DM54LS163A/DM74LS163A Synchronous 4-Bit Binary Counters

General Description

These synchronous, presettable counters feature an internal carry look-ahead for application in high-speed counting designs. The LS161A and LS163A are 4-bit binary counters. The carry output is decoded by means of a NOR gate, thus preventing spikes during the normal counting mode of operation. Synchronous operation is provided by having all flip-flops clocked simultaneously so that the outputs change coincident with each other when so instructed by the count-enable inputs and internal gating. This mode of operation eliminates the output counting spikes which are normally associated with asynchronous (ripple clock) counters. A buffered clock input triggers the four flip-flops on the rising (positive-going) edge of the clock input waveform.

These counters are fully programmable; that is, the outputs may be preset to either level. As presetting is synchronous, setting up a low level at the load input disables the counter and causes the outputs to agree with the setup data after the next clock pulse, regardless of the levels of the enable input. The clear function for the LS161A is asynchronous; and a low level at the clear input sets all four of the flip-flop outputs low, regardless of the levels of clock, load, or enable inputs. The clear function for the LS163A is synchronous; and a low level at the clear inputs sets all four of the flip-flop outputs low after the next clock pulse, regardless of the levels of the enable inputs. This synchronous clear allows the count length to be modified easily, as decoding the maximum count desired can be accomplished with one external NAND gate. The gate output is connected to the clear input to synchronously clear the counter to all low outputs.

The carry look-ahead circuitry provides for cascading counters for n-bit synchronous applications without additional gating. Instrumental in accomplishing this function are two count-enable inputs and a ripple carry output.

Both count-enable inputs (P and T) must be high to count, and input T is fed forward to enable the ripple carry output. The ripple carry output thus enabled will produce a high-level output pulse with a duration approximately equal to the high-level portion of the Q_A output. This high-level overflow ripple carry pulse can be used to enable successive cascaded stages. High-to-low level transitions at the enable P or T inputs may occur, regardless of the logic level of the clock.

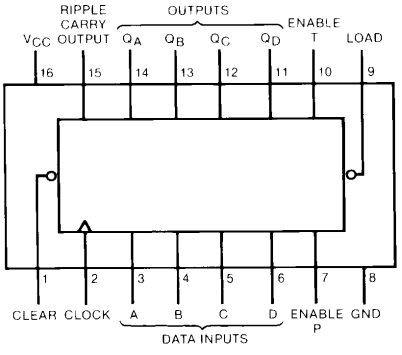
These counters feature a fully independent clock circuit. Changes made to control inputs (enable P or T or load) that will modify the operating mode have no effect until clocking occurs. The function of the counter (whether enabled, disabled, loading, or counting) will be dictated solely by the conditions meeting the stable set-up and hold times.

Features

- Synchronously programmable
- Internal look-ahead for fast counting
- Carry output for n-bit cascading
- Synchronous counting
- Load control line
- Diode-clamped inputs
- Typical propagation time, clock to Q output 14 ns
- Typical clock frequency 32 MHz
- Typical power dissipation 93 mW
- Alternate Military/Aerospace device (54LS161, 54LS163) is available. Contact a National Semiconductor Sales Office/Distributor for specificaitons.

Connection Diagram

Dual-In-Line Package



TL/F/6397–1

Order Numbers 54LS161ADMQB, 54LS161AFMQB, 54LS161ALMQB, 54LS163ADMQB, 54LS163AFMQB, 54LS163ALMQB, DM54LS161AJ, DM54LS161AW, DM54LS163AJ, DM54LS163AW, DM74LS161AM, DM74LS161AN, DM74LS163AM or DM74LS163AN
See NS Package Number E20A, J16A, M16A, N16E or W16A

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM54LS and 54LS	−55°C to +125°C
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter		DM54LS161A			DM74LS161A			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage		2			2			V
V_{IL}	Low Level Input Voltage				0.7			0.8	V
I_{OH}	High Level Output Current				−0.4			−0.4	mA
I_{OL}	Low Level Output Current				4			8	mA
f_{CLK}	Clock Frequency (Note 1)		0		25	0		25	MHz
	Clock Frequency (Note 2)		0		20	0		20	MHz
t_W	Pulse Width (Note 1)	Clock	20	6		20	6		ns
		Clear	20	9		20	9		
	Pulse Width (Note 2)	Clock	25			25			ns
		Clear	25			25			
t_{SU}	Setup Time (Note 1)	Data	20	8		20	8		ns
		Enable P	25	17		25	17		
		Load	25	15		25	15		
	Setup Time (Note 2)	Data	20			20			ns
		Enable P	30			30			
		Load	30			30			
t_H	Hold Time (Note 1)	Data	0	−3		0	−3		ns
		Others	0	−3		0	−3		
	Hold Time (Note 2)	Data	5			5			ns
		Others	5			5			
t_{REL}	Clear Release Time (Note 1)		20			20			ns
	Clear Release Time (Note 2)		25			25			ns
T_A	Free Air Operating Temperature		−55		125	0		70	°C

Note 1: $C_L = 15$ pF, $R_L = 2$ kΩ, $T_A = 25°C$ and $V_{CC} = 5.5V$.

Note 2: $C_L = 50$ pF, $R_L = 2$ kΩ, $T_A = 25°C$ and $V_{CC} = 5.5V$.

'LS161 Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	DM54	2.5	3.4		V
			DM74	2.7	3.4		
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 4 mA, V_{CC} = Min	DM74		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V	Enable T			0.2	mA
			Clock			0.2	
			Load			0.2	
			Others			0.1	
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V	Enable T			40	μA
			Clock			40	
			Load			40	
			Others			20	
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V	Enable T			−0.8	mA
			Clock			−0.8	
			Load			−0.8	
			Others			−0.4	
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−20		−100	mA
			DM74	−20		−100	
I_{CCH}	Supply Current with Outputs High	V_{CC} = Max (Note 3)			18	31	mA
I_{CCL}	Supply Current with Outputs Low	V_{CC} = Max (Note 4)			19	32	mA

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Note 3: I_{CCH} is measured with the load high, then again with the load low, with all other inputs high and all outputs open.

Note 4: I_{CCL} is measured with the clock input high, then again with the clock input low, with all other inputs low and all outputs open.

'LS161 Switching Characteristics

at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	From (Input) To (Output)	R_L = 2 kΩ				Units
			C_L = 15 pF		C_L = 50 pF		
			Min	Max	Min	Max	
f_{MAX}	Maximum Clock Frequency		25		20		MHz
t_{PLH}	Propagation Delay Time Low to High Level Output	Clock to Ripple Carry		25		30	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Clock to Ripple Carry		30		38	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Clock to Any Q (Load High)		22		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Clock to Any Q (Load High)		27		38	ns

’LS161 Switching Characteristics

at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load) (Continued)

Symbol	Parameter	From (Input) To (Output)	R_L = 2 kΩ				Units
			C_L = 15 pF		C_L = 50 pF		
			Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	Clock to Any Q (Load Low)		24		30	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Clock to Any Q (Load Low)		27		38	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Enable T to Ripple Carry		14		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Enable T to Ripple Carry		15		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Clear to Any Q		28		45	ns

Recommended Operating Conditions

Symbol	Parameter		DM54LS163A			DM74LS163A			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage		2			2			V
V_{IL}	Low Level Input Voltage				0.7			0.8	V
I_{OH}	High Level Output Current				−0.4			−0.4	mA
I_{OL}	Low Level Output Current				4			8	mA
f_{CLK}	Clock Frequency (Note 1)		0		25	0		25	MHz
	Clock Frequency (Note 2)		0		20	0		20	MHz
t_W	Pulse Width (Note 1)	Clock	20	6		20	6		ns
		Clear	20	9		20	9		
	Pulse Width (Note 2)	Clock	25			25			ns
		Clear	25			25			
t_{SU}	Setup Time (Note 1)	Data	20	8		20	8		ns
		Enable P	25	17		25	17		
		Load	25	15		25	15		
	Setup Time (Note 2)	Data	20			20			ns
		Enable P	30			30			
		Load	30			30			
t_H	Hold Time (Note 1)	Data	0	−3		0	−3		ns
		Others	0	−3		0	−3		
	Hold Time (Note 2)	Data	5			5			ns
		Others	5			5			
t_{REL}	Clear Release Time (Note 1)		20			20			ns
	Clear Release Time (Note 2)		25			25			ns
T_A	Free Air Operating Temperature		−55		125	0		70	°C

Note 1: C_L = 15 pF, R_L = 2 kΩ, T_A = 25°C and V_{CC} = 5V.

Note 2: C_L = 50 pF, R_L = 2 kΩ, T_A = 25°C and V_{CC} = 5V.

'LS163 Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	DM54	2.5	3.4		V
			DM74	2.7	3.4		
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 4 mA, V_{CC} = Min	DM74		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V	Enable T			0.2	mA
			Clock, Clear			0.2	
			Load			0.2	
			Others			0.1	
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V	Enable T			40	μA
			Load			40	
			Clock, Clear			40	
			Others			20	
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V	Enable T			−0.8	mA
			Clock, Clear			−0.8	
			Load			−0.8	
			Others			−0.4	
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−20		−100	mA
			DM74	−20		−100	
I_{CCH}	Supply Current with Outputs High	V_{CC} = Max (Note 3)			18	31	mA
I_{CCL}	Supply Current with Outputs Low	V_{CC} = Max (Note 4)			18	32	mA

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Note 3: I_{CCH} is measured with the load high, then again with the load low, with all other inputs high and all outputs open.

Note 4: I_{CCL} is measured with the clock input high, then again with the clock input low, with all other inputs low and all outputs open.

'LS163 Switching Characteristics

at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	From (Input) To (Output)	R_L = 2 kΩ				Units
			C_L = 15 pF		C_L = 50 pF		
			Min	Max	Min	Max	
f_{MAX}	Maximum Clock Frequency		25		20		MHz
t_{PLH}	Propagation Delay Time Low to High Level Output	Clock to Ripple Carry		25		30	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Clock to Ripple Carry		30		38	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Clock to Any Q (Load High)		22		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Clock to Any Q (Load High)		27		38	ns

’LS163 Switching Characteristics

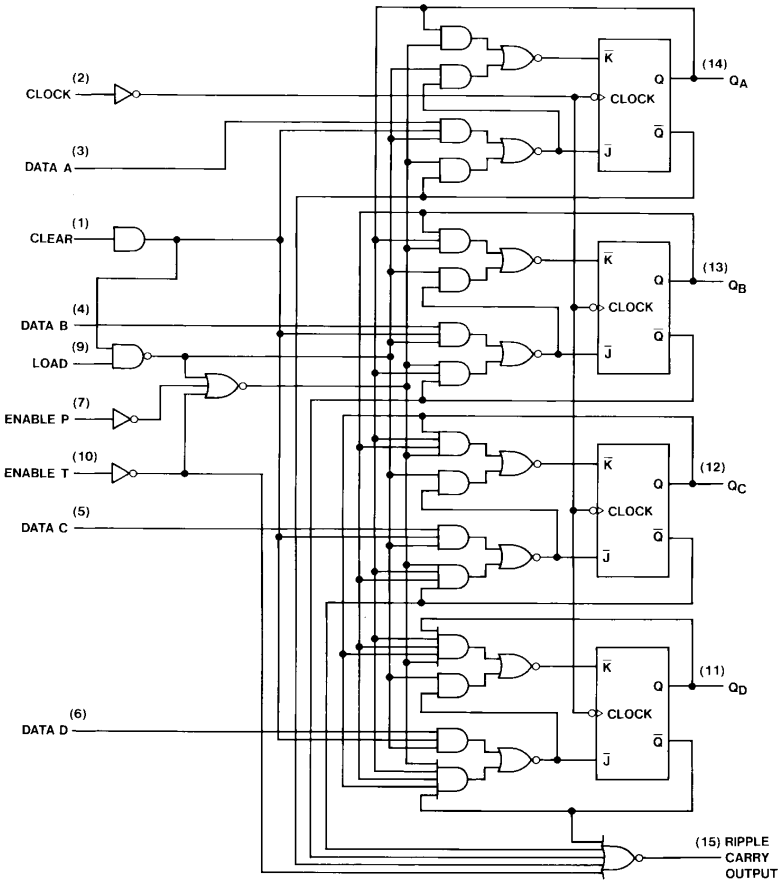
at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load) (Continued)

Symbol	Parameter	From (Input) To (Output)	R_L = 2 kΩ				Units
			C_L = 15 pF		C_L = 50 pF		
			Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	Clock to Any Q (Load Low)		24		30	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Clock to Any Q (Load Low)		27		38	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Enable T to Ripple Carry		14		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Enable T to Ripple Carry		15		27	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Clear to Any Q (Note 1)		28		45	ns

Note 1: The propagation delay clear to output is measured from the clock input transition.

Logic Diagram

LS163A

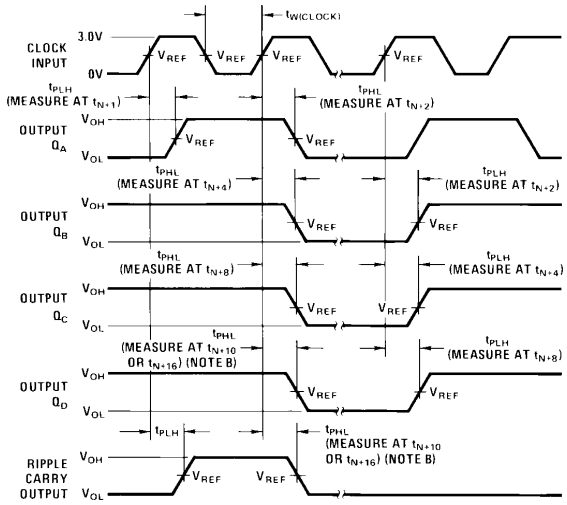


TL/F/6397–2

The LS161A is similar, however, the clear buffer is connected directly to the flip flops.

Parameter Measurement Information

Switching Time Waveforms



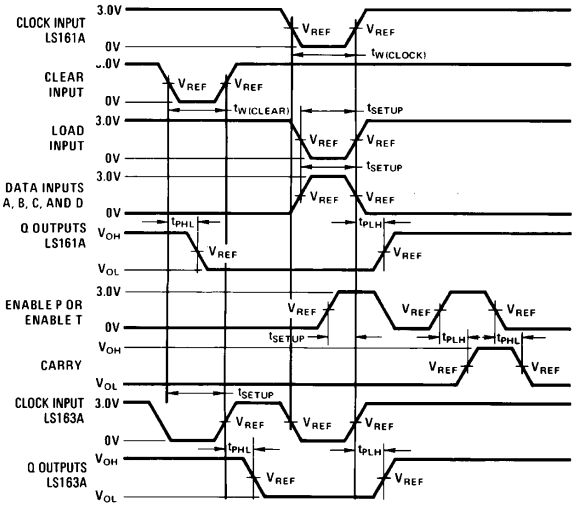
TL/F/6397–3

Note A: The input pulses are supplied by generators having the following characteristics: PRR $\leq$ 1 MHz, duty cycle $\leq$ 50%, $Z_{OUT} \approx 50\Omega$, $t_r \leq 10$ ns, $t_f \leq 10$ ns. Vary PRR to measure f_{MAX}.

Note B: Outputs Q_D and carry are tested at t_{n+16} where t_n is the bit time when all outputs are low.

Note C: $V_{REF} = 1.5V$.

Switching Time Waveforms



TL/F/6397–4

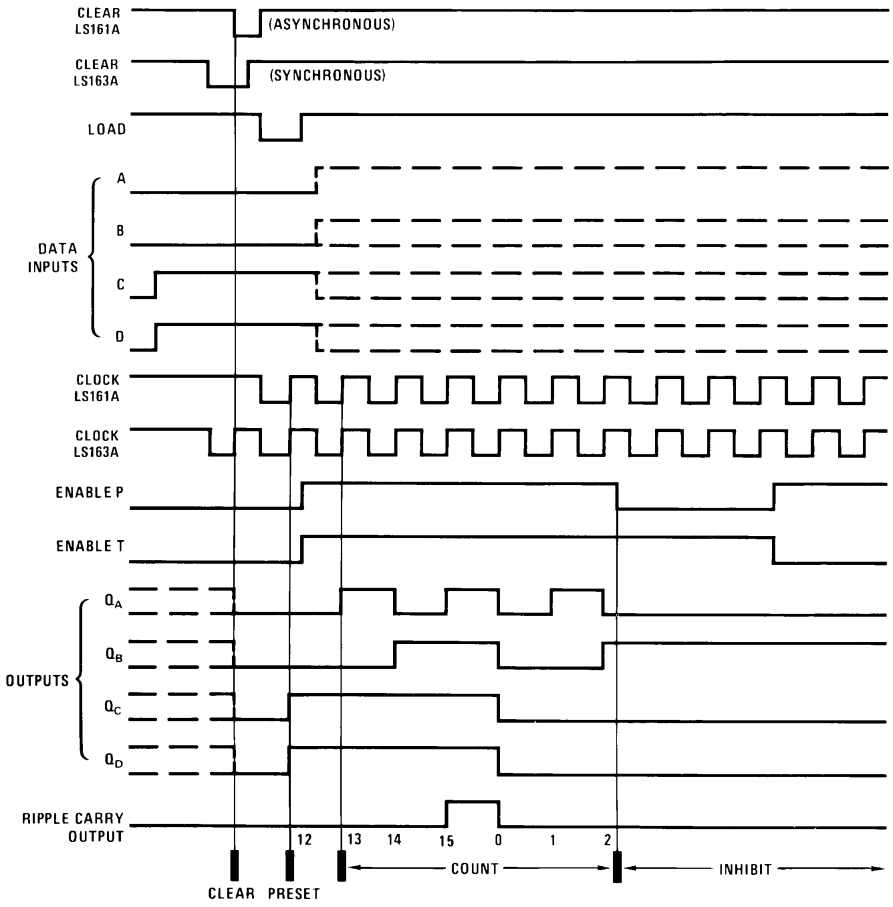
Note A: The input pulses are supplied by generators having the following characteristics: PRR $\leq$ 1 MHz, duty cycle $\leq$ 50%, $Z_{OUT} \approx 50\Omega$, $t_r \leq 6$ ns, $t_f \leq 6$ ns. Vary PRR to measure f_{MAX}.

Note B: Enable P and enable T setup times are measured at t_{n+0}.

Note C: $V_{REF} = 1.3V$.

Timing Diagram

LS161A, LS163A Synchronous Binary Counters
Typical Clear, Preset, Count and Inhibit Sequences



TL/F/6397–5

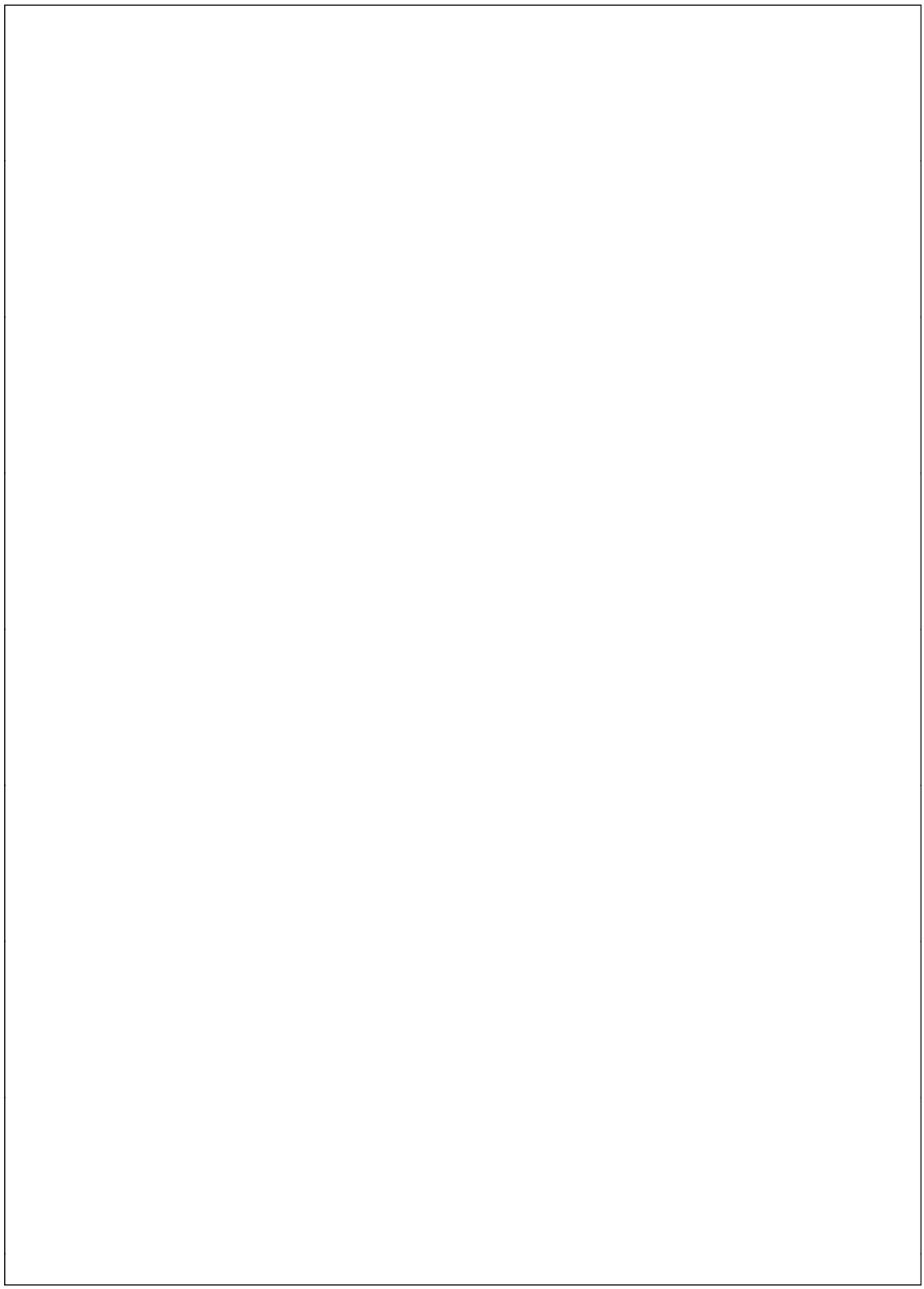
Sequence:

(1) Clear outputs to zero

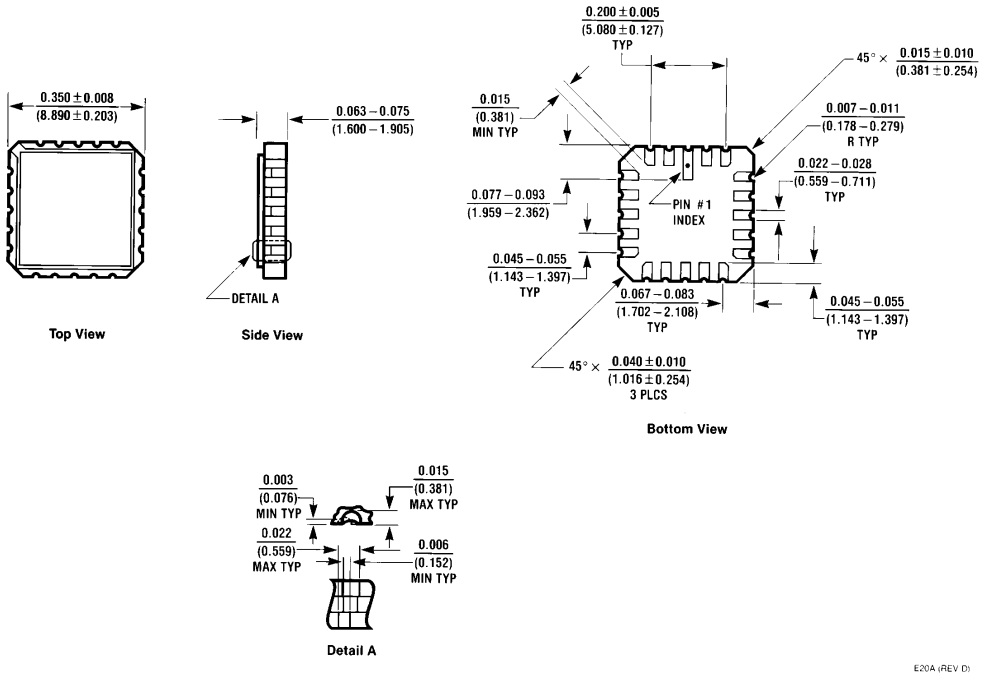
(2) Preset to binary twelve

(3) Count to thirteen, fourteen, fifteen, zero, one, and two

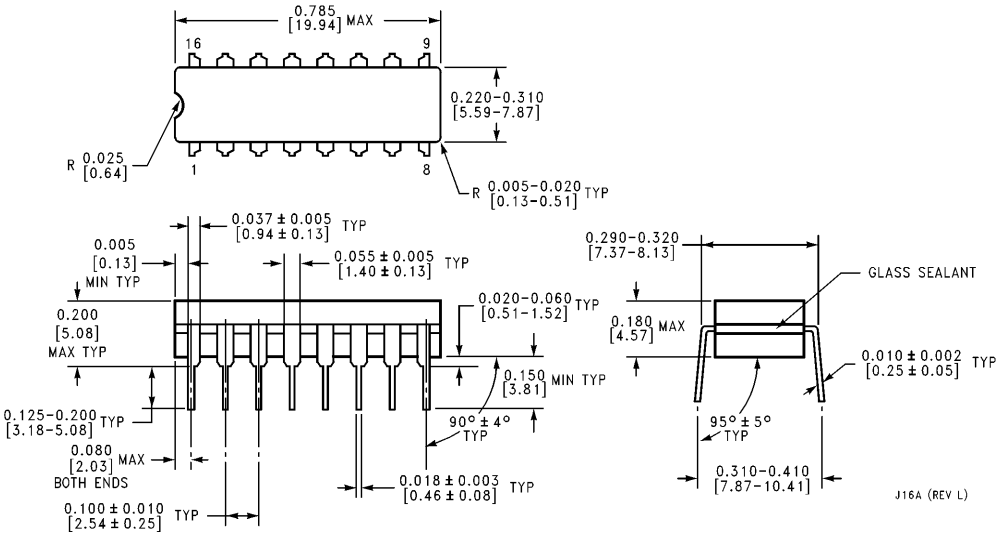
(4) Inhibit



Physical Dimensions inches (millimeters)

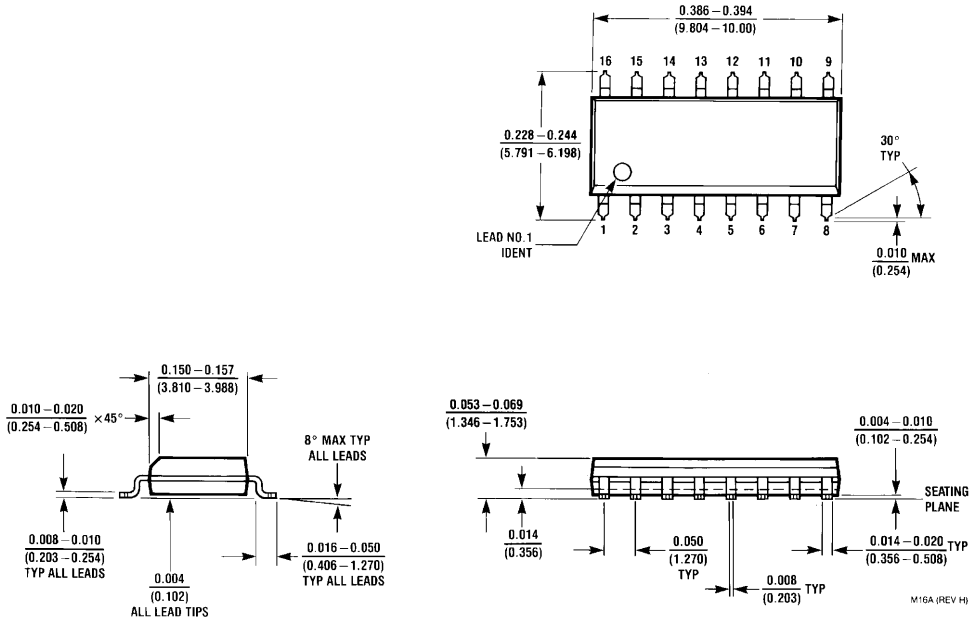


Ceramic Leadless Chip Carrier Package (E)
Order Numbers 54LS161ALMQB or 54LS163ALMQB
NS Package Number E20A

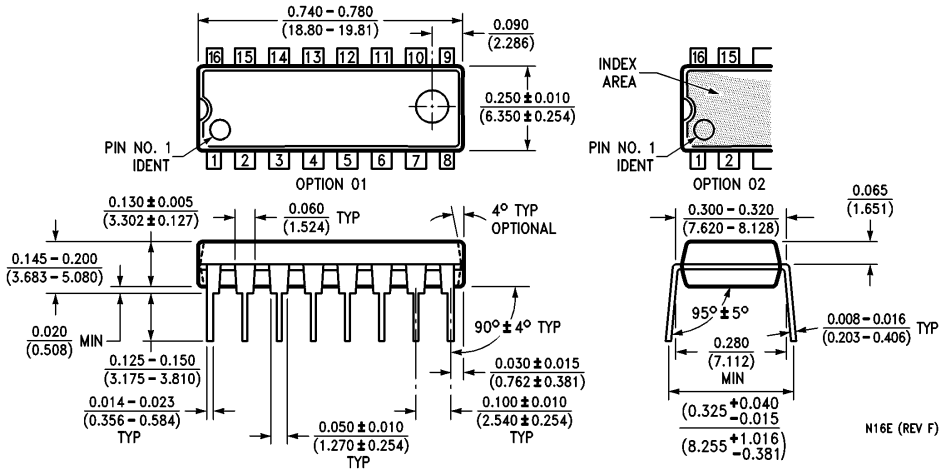


16-Lead Ceramic Dual-In-Line Package (J)
Order Numbers 54LS161ADMQB, 54LS163ADMQB, DM54LS161AJ or DM54LS163AJ
NS Package Number J16A

Physical Dimensions inches (millimeters) (Continued)

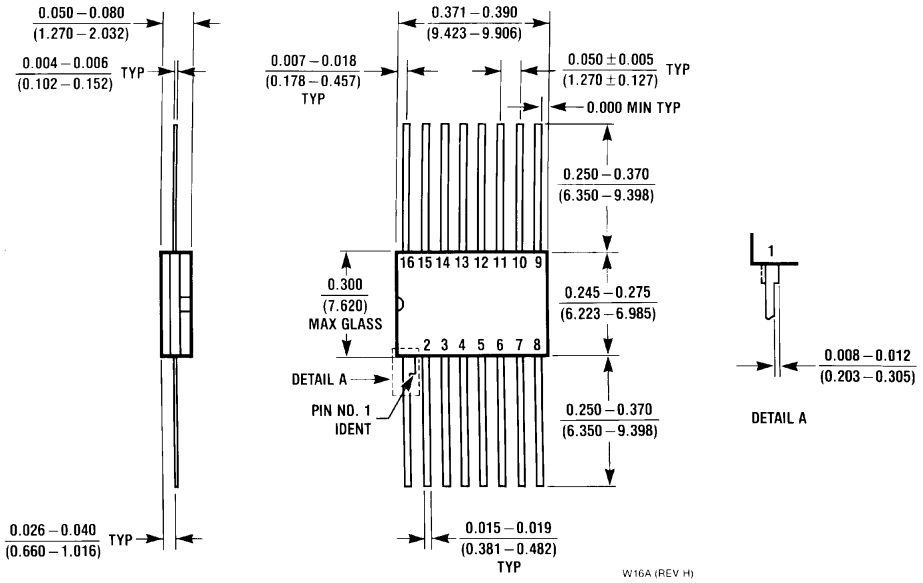


16-Lead Small Outline Molded Package (M)
Order Number DM74LS161AM or DM74LS163AM
NS Package Number M16A



16-Lead Molded Dual-In-Line Package (N)
Order Numbers DM74LS161AN, DM74LS163AN
NS Package Number N16E

Physical Dimensions inches (millimeters) (Continued)



16-Lead Ceramic Flat Package (W)
Order Numbers 54LS161AFMQB, 54LS163AFMQB, DM54LS161AN or DM54LS163AW
NS Package Number W16A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

January 1992

DM74LS196
Presettable Decade Counter

General Description

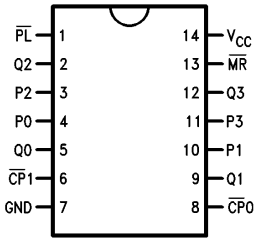
The 'LS196 decade ripple counter is partitioned into divide-by-two and divide-by-five sections which can be combined to count either in BCD (8421) sequence or in a bi-quinary mode producing a 50% duty cycle output. Both circuit types have a Master Reset ($\overline{MR}$) input which overrides all other inputs and asynchronously forces all outputs LOW. A Parallel Load input ($\overline{PL}$) overrides clocked operations and asynchronously loads the data on the Parallel Data inputs (Pn) into the flip-flops. This preset feature makes the circuits usable as programmable counters. The circuits can also be used as 4-bit latches, loading data from the Parallel Data inputs when $\overline{PL}$ is LOW and storing the data when $\overline{PL}$ is HIGH. In the counting modes, state changes are initiated by the falling edge of the clock.

Features

- High counting rates—typically 60 MHz
- Choice of counting modes—BCD, bi-quinary, binary
- Asynchronous preset and master reset

Connection Diagram

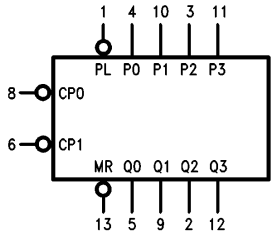
Dual-In-Line Package



TL/F/10179–1

Order Number DM74LS196M or DM74LS196N
See NS Package Number M14A or N14A

Logic Symbol



TL/F/10179–2

V_{CC} = Pin 14
GND = Pin 7

Pin Names	Description
$\overline{CP}0$	÷2 Section Clock Input (Active Falling Edge)
$\overline{CP}1$	÷5 Section Clock Input (Active Falling Edge)
$\overline{MR}$	Asynchronous Master Reset Input (Active LOW)
P0–P3	Parallel Data Inputs
$\overline{PL}$	Asynchronous Parallel Load Input (Active LOW)
Q0–Q3	Flip-Flop Outputs

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM74LS196			Units
		Min	Nom	Max	
V_{CC}	Supply Voltage	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			V
V_{IL}	Low Level Input Voltage			0.8	V
I_{OH}	High Level Output Current			−0.4	mA
I_{OL}	Low Level Output Current			8	mA
T_A	Free Air Operating Temperature	0		70	°C
t_s (H) t_s (L)	Setup Time HIGH or LOW Pn to $\overline{PL}$	8 12			ns
t_h (H) t_h (L)	Hold Time HIGH or LOW Pn to $\overline{PL}$	0 6			ns
t_w (H)	$\overline{CP}$0 Pulse Width HIGH	12			ns
t_w (H)	$\overline{CP}$1 Pulse Width HIGH	24			ns
t_w (L)	$\overline{PL}$ Pulse Width LOW	18			ns
t_w (L)	$\overline{MR}$ Pulse Width LOW	12			ns
t_{rec}	Recovery Time $\overline{PL}$ to $\overline{CP}$n	16			ns
t_{rec}	Recovery Time $\overline{MR}$ to $\overline{CP}$n	18			ns

Electrical Characteristics Over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max	2.7	3.4		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IH} = Min		0.35	0.5	V
		I_{OL} = 4 mA, V_{CC} = Min		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V			0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 5.5V, $\overline{CP}$1			40	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V			−0.4	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	−20		−100	mA
I_{CC}	Supply Current	V_{CC} = Max, V_{IN} = GND			20	mA

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Switching Characteristics

$V_{CC} = +5.0V$, $T_A = +25°C$

Symbol	Parameter	$R_L = 2k$, $C_L = 15$ pF		Units
		Min	Max	
f_{max}	Maximum Count Frequency at $\overline{CP}0$	45		MHz
f_{max}	Maximum Count Frequency at $\overline{CP}1$	22.5		MHz
t_{PLH} t_{PHL}	Propagation Delay $\overline{CP}0$ to Q0		15 15	ns
t_{PLH} t_{PHL}	Propagation Delay $\overline{CP}1$ to Q1		15 15	ns
t_{PLH} t_{PHL}	Propagation Delay $\overline{CP}1$ to Q2		34 34	ns
t_{PLH} t_{PHL}	Propagation Delay $\overline{CP}1$ to Q3		15 21	ns
t_{PLH} t_{PHL}	Propagation Delay Pn to Qn		25 35	ns
t_{PLH} t_{PHL}	Propagation Delay $\overline{PL}$ to Qn		31 37	ns
t_{PHL}	Propagation Delay $\overline{MR}$ to Qn		42	ns

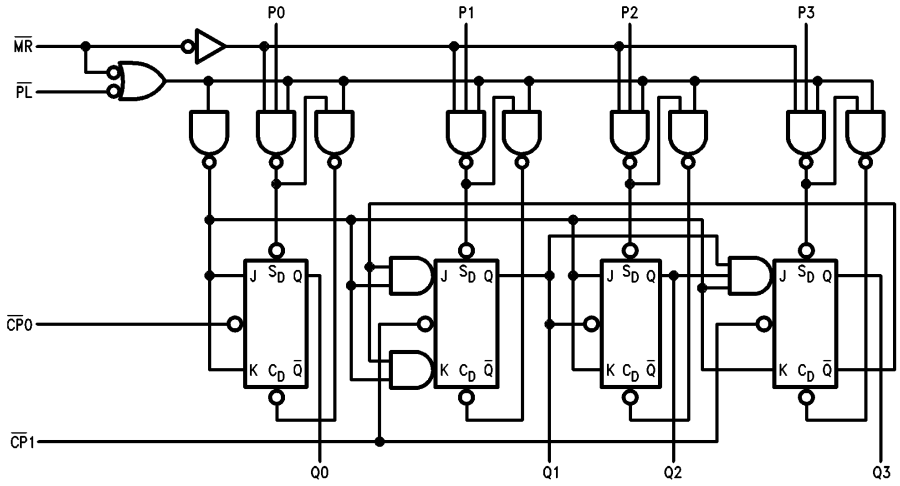
Functional Description

The ’196 and ’197 are asynchronous presettable decade and binary ripple counters. The ’196 decade counter is partitioned into divide-by-two and divide-by-five sections while the ’197 is partitioned into divide-by-two and divide-by-eight sections, with all sections having a separate Clock input. In the counting modes, state changes are initiated by the HIGH-to-LOW transition of the clock signals. State changes of the Q outputs, however, do not occur simultaneously because of the internal ripple delays. When using external logic to decode the Q outputs, designers should bear in mind that the unequal delays can lead to decoding spikes and thus a decoded signal should not be used as a clock or strobe. The $\overline{CP}0$ input serves the Q0 flip-flop in both circuit types while the $\overline{CP}1$ input serves the divide-by-five or divide-by-eight section. The Q0 output is designed and specified to drive the rated fan-out plus the $\overline{CP}1$ input. With the input frequency connected to $\overline{CP}0$ and with Q0 driving $\overline{CP}1$, the ’197 forms a straight forward modulo-16 counter, with Q0 the least significant output and Q3 the most significant output.

The ’196 decade counter can be connected up to operate in two different count sequences. With the input frequency connected to $\overline{CP}0$ and with Q0 driving $\overline{CP}1$, the circuit counts in the BCD (8421) sequence. With the input frequency connected to $\overline{CP}1$ and Q3 driving $\overline{CP}0$, Q0 becomes the low frequency output and has a 50% duty cycle waveform. Note that the maximum counting rate is reduced in the latter (bi-quinary) configuration because of the interstage gating delay within the divide-by-five section.

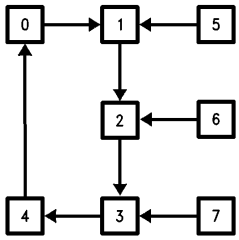
The ’196 and ’197 have an asynchronous active LOW Master Reset input ($\overline{MR}$) which overrides all other inputs and forces all outputs LOW. The counters are also asynchronously presettable. A LOW on the Parallel Load input ($\overline{PL}$) overrides the clock inputs and loads the data from Parallel Data (P0–P3) inputs into the flip-flops. While $\overline{PL}$ is LOW, the counters act as transparent latches and any change in the Pn inputs will be reflected in the outputs. In order for the intended parallel data to be entered and stored, the recommended setup and hold times with respect to the rising edge of $\overline{PL}$ should be observed.

Logic Diagram



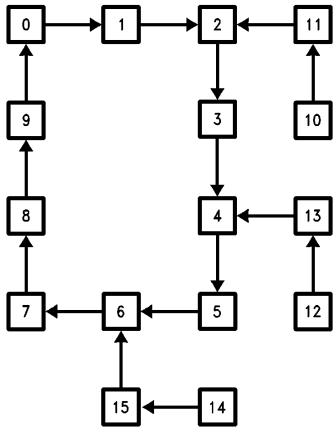
TL/F/10179–3

÷5 State Diagram



TL/F/10179–4

BCD State Diagram



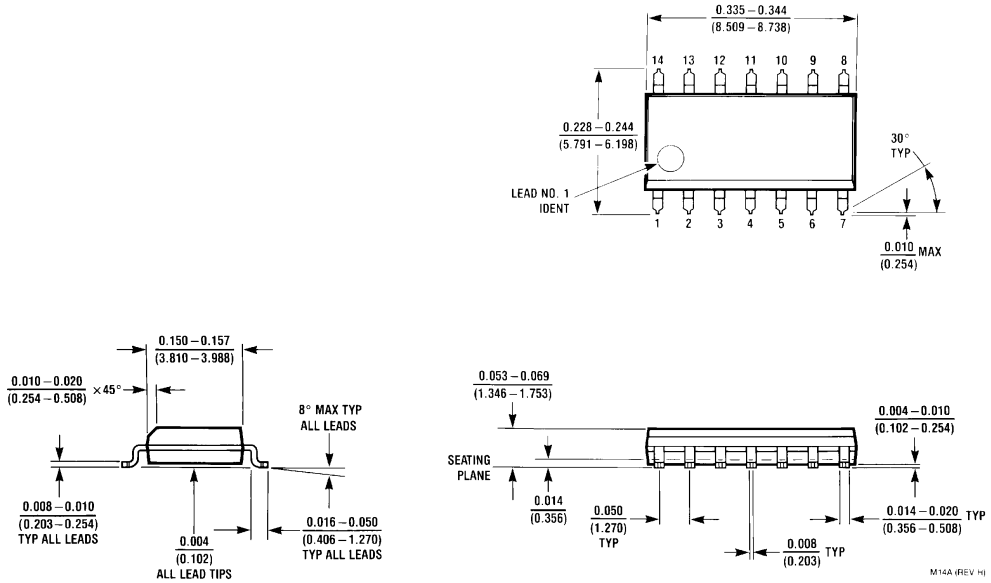
TL/F/10179–5

Mode Select Table

Inputs			Response
$\overline{MR}$	$\overline{PL}$	$\overline{CP}$	
L	X	X	Qn forced LOW
H	L	X	Pn → Qn
H	H	⌐	Count Up

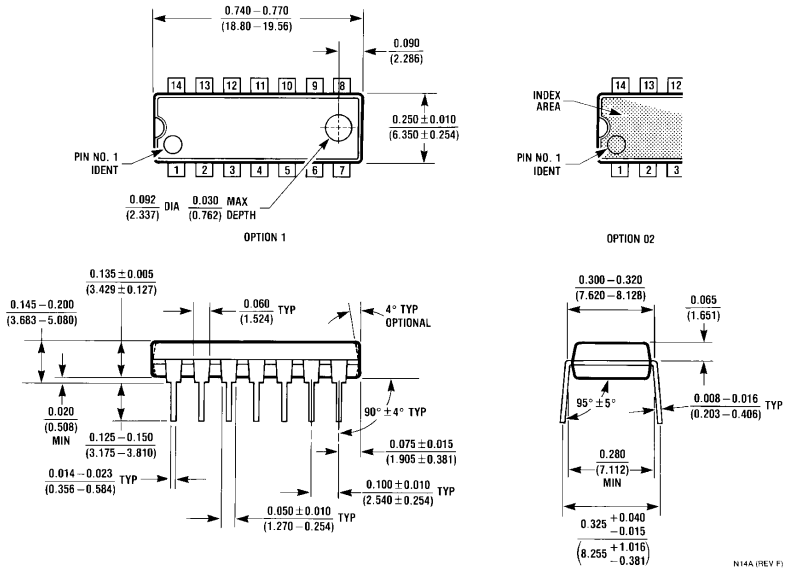
H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

Physical Dimensions inches (millimeters)



14-Lead Small Outline Molded Package (M)
Order Number DM74LS196M
NS Package Number M14A

Physical Dimensions inches (millimeters) (Continued)



14-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS196N
NS Package Number N14A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

February 1992

DM74LS221 Dual Non-Retriggerable One-Shot with Clear and Complementary Outputs

General Description

The DM74LS221 is a dual monostable multivibrator with Schmitt-trigger input. Each device has three inputs permitting the choice of either leading-edge or trailing-edge triggering. Pin (A) is an active-low trigger transition input and pin (B) is an active-high transition Schmitt-trigger input that allows jitter free triggering for inputs with transition rates as slow as 1 volt/second. This provides the input with excellent noise immunity. Additionally an internal latching circuit at the input stage also provides a high immunity to V_{CC} noise. The clear (CLR) input can terminate the output pulse at a predetermined time independent of the timing components. This (CLR) input also serves as a trigger input when it is pulsed with a low level pulse transition (⌐¦¬). To obtain the best and trouble free operation from this device please read operating rules as well as the NSC one-shot application notes carefully and observe recommendations.

Features

- A dual, highly stable one-shot
- Compensated for V_{CC} and temperature variations
- Pin-out identical to 'LS123 (Note 1)
- Output pulse width range from 30 ns to 70 seconds
- Hysteresis provided at (B) input for added noise immunity
- Direct reset terminates output pulse
- Triggerable from CLEAR input
- DTL, TTL compatible
- Input clamp diodes

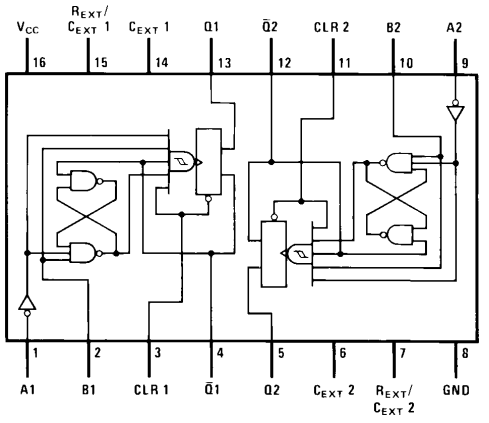
Note 1: The pin-out is identical to 'LS123 but, functionally it is not; refer to Operating Rules #10 in this datasheet.

Functional Description

The basic output pulse width is determined by selection of an external resistor (R_X) and capacitor (C_X). Once triggered, the basic pulse width is independent of further input transitions and is a function of the timing components, or it may be reduced or terminated by use of the active low CLEAR input. Stable output pulse width ranging from 30 ns to 70 seconds is readily obtainable.

Connection Diagram

Dual-In-Line Package



TL/F/6409–1

Order Number DM74LS221M or DM74LS221N
See NS Package Number M16A or N16A

Function Table

Inputs			Outputs	
CLEAR	A	B	Q	$\overline{Q}$
L	X	X	L	H
X	H	X	L	H
X	X	L	L	H
H	L	↑	⌐¦¬	⌐¦¬
H	↓	H	⌐¦¬	⌐¦¬
* ↑	L	H	⌐¦¬	⌐¦¬

H = High Logic Level

L = Low Logic Level

X = Can Be Either Low or High

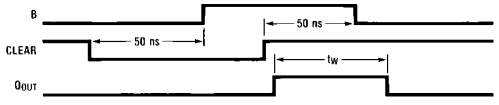
↑ = Positive Going Transition

↓ = Negative Going Transition

⌐¦¬ = A Positive Pulse

⌐¦¬ = A Negative Pulse

*This mode of triggering requires first the B input be set from a low to high level while the CLEAR input is maintained at logic low level. Then with the B input at logic high level, the CLEAR input whose positive transition from low to high will trigger an output pulse.



TL/F/6409–2

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter		DM74LS221			Units
			Min	Nom	Max	
V_{CC}	Supply Voltage		4.75	5	5.25	V
V_{T+}	Positive-Going Input Threshold Voltage at the A Input (V_{CC} = Min)			1	2	V
V_{T-}	Negative-Going Input Threshold Voltage at the A Input (V_{CC} = Min)		0.8	1		V
V_{T+}	Positive-Going Input Threshold Voltage at the B Input (V_{CC} = Min)			1	2	V
V_{T-}	Negative-Going Input Threshold Voltage at the B Input (V_{CC} = Min)		0.8	0.9		V
I_{OH}	High Level Output Current				−0.4	mA
I_{OL}	Low Level Output Current				8	mA
t_W	Pulse Width (Note 1)	Data	40			ns
		Clear	40			
t_{REL}	Clear Release Time (Note 1)		15			ns
$\frac{dV}{dt}$	Rate of Rise or Fall of Schmitt Input (B) (Note 1)				1	$\frac{V}{s}$
$\frac{dV}{dt}$	Rate of Rise or Fall of Logic Input (A) (Note 1)				1	$\frac{V}{\mu s}$
R_{EXT}	External Timing Resistor (Note 1)		1.4		100	kΩ
C_{EXT}	External Timing Capacitance (Note 1)		0		1000	μF
DC	Duty Cycle (Note 1)	R_T = 2 kΩ			50	%
		R_T = R_{EXT} (Max)			60	
T_A	Free Air Operating Temperature		0		70	°C

Note 1: T_A = 25°C and V_{CC} = 5V.

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max V_{IL} = Max, V_{IH} = Min	2.7	3.4		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max V_{IL} = Max, V_{IH} = Min		0.35	0.5	V
		V_{CC} = Min, I_{OL} = 4 mA			0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V			0.1	mA

Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted) (Continued)

Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V	A1, A2			−0.4	mA
			B			−0.8	
			Clear			−0.8	
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)		−20		−100	mA
I_{CC}	Supply Current	V_{CC} = Max	Quiescent		4.7	11	mA
			Triggered		19	27	

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Switching Characteristics

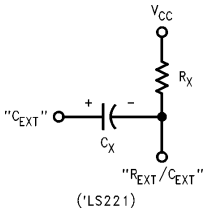
at V_{CC} = 5V and T_A = 25°C

Symbol	Parameter	From (Input) To (Output)	Conditions	Min	Max	Units
t_{PLH}	Propagation Delay Time Low to High Level Output	A1, A2 to Q	C_{EXT} = 80 pF, R_{EXT} = 2 kΩ, C_L = 15 pF, R_L = 2 kΩ		70	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	B to Q			55	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	A1, A2 to Q			80	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	B to $\overline{Q}$			65	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Clear to $\overline{Q}$			65	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Clear to Q			55	ns
$t_{W(out)}$	Output Pulse Width Using Zero Timing Capacitance	A1, A2 to Q, $\overline{Q}$	C_{EXT} = 0, R_{EXT} = 2 kΩ, R_L = 2 kΩ, C_L = 15 pF	20	70	ns
$t_{W(out)}$	Output Pulse Width Using External Timing Resistor	A1, A2 to Q, $\overline{Q}$	C_{EXT} = 100 pF, R_{EXT} = 10 kΩ, R_L = 2 kΩ, C_L = 15 pF	600	750	ns
			C_{EXT} = 1 μF, R_{EXT} = 10 kΩ, R_L = 2 kΩ, C_L = 15 pF	6	7.5	ms
			C_{EXT} = 80 pF, R_{EXT} = 2 kΩ, R_L = 2 kΩ, C_L = 15 pF	70	150	ns

Operating Rules

1. An external resistor (R_X) and an external capacitor (C_X) are required for proper operation. The value of C_X may vary from 0 to approximately 1000 μF. For small time constants high-grade mica, glass, polypropylene, polycarbonate, or polystyrene material capacitor may be used. For large time constants use tantalum or special aluminum capacitors. If timing capacitor has leakages approaching 100 nA or if stray capacitance from either terminal to ground is greater than 50 pF the timing equations may not represent the pulse width the device generates.
2. When an electrolytic capacitor is used for C_X a switching diode is often required for standard TTL one-shots to prevent high inverse leakage current. This switching diode is not needed for the 'LS221 one-shot and should not be used.

   Furthermore, if a polarized timing capacitor is used on the 'LS221, the positive side of the capacitor should be connected to the "C_{EXT}" pin *(Figure 1)*.



TL/F/6409–8

FIGURE 1

3. For $C_X >> 1000$ pF, the output pulse width (T_W) is defined as follows:

   $T_W = KR_X C_X$

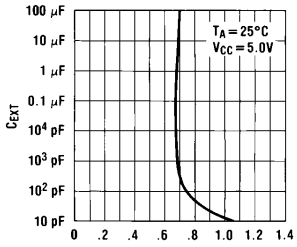
   where [R_X is in kΩ]

   [C_X is in pF]

   [T_W is in ns]

   $K \approx Ln2 = 0.70$

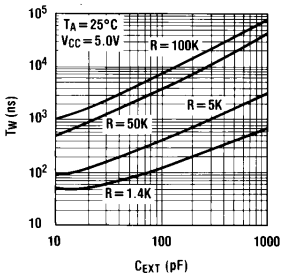
4. The multiplicative factor K is plotted as a function of C_X below for design considerations:



TL/F/6409–3

FIGURE 2

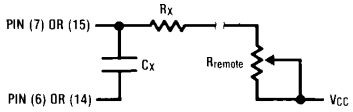
5. For $C_X < 1000$ pF see *Figure 3* for T_W vs C_X family curves with R_X as a parameter:



TL/F/6409–4

FIGURE 3

6. To obtain variable pulse widths by remote trimming, the following circuit is recommended:

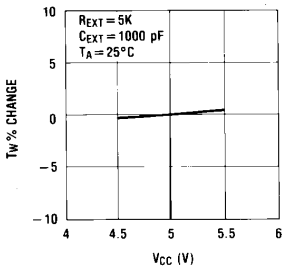


TL/F/6409–5

Note: "R_{remote}" should be as close to the one-shot as possible.

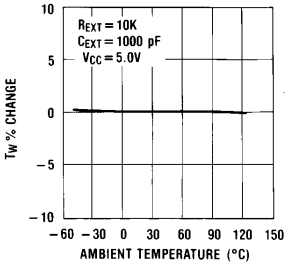
FIGURE 4

7. Output pulse width versus V_{CC} and temperatures: *Figure 5* depicts the relationship between pulse width variation versus V_{CC}. *Figure 6* depicts pulse width variation versus temperatures.



TL/F/6409–6

FIGURE 5



TL/F/6409–7

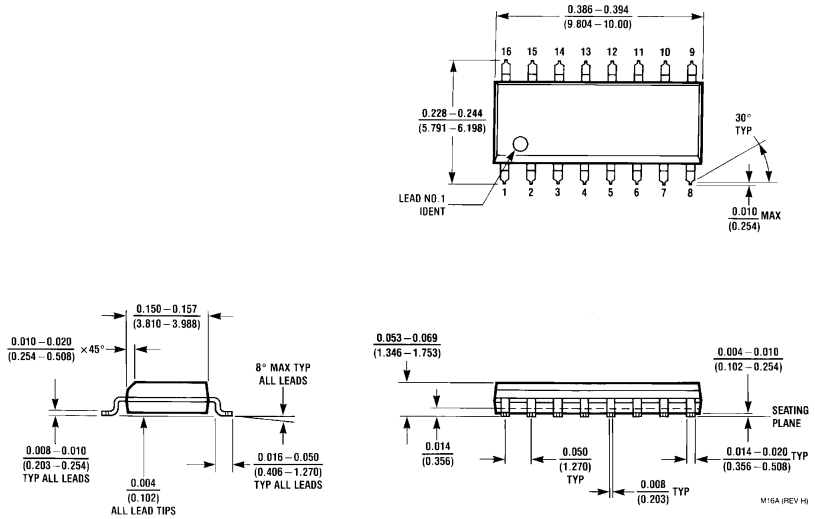
FIGURE 6

Operating Rules (Continued)

8. Duty cycle is defined as $T_W/T \times 100$ in percentage, if it goes above 50% the output pulse width will become shorter. If the duty cycle varies between low and high values, this causes output pulse width to vary, or jitter (a function of the R_{EXT} only). To reduce jitter , R_{EXT} should be as large as possible, for example, with $R_{EXT} = 100k$ jitter is not appreciable until the duty cycle approaches 90%.
9. Under any operating condition C_X and R_X must be kept as close to the one-shot device pins as possible to minimize stray capacitance, to reduce noise pick-up, and to reduce I-R and Ldi/dt voltage developed along their connecting paths. If the lead length from C_X to pins (6) and (7) or pins (14) and (15) is greater than 3 cm, for example, the output pulse width might be quite different from values predicted from the appropriate equations. A non-inductive and low capacitive path is necessary to ensure complete discharge of C_X in each cycle of its operation so that the output pulse width will be accurate.
10. Although the 'LS221's pin-out is identical to the 'LS123 it should be remembered that they are not functionally identical. The 'LS123 is a retriggerable device such that the output is dependent upon the input transitions when its output "Q" is at the "High" state. Furthermore, it is recommended for the 'LS123 to externally ground the C_{EXT} pin for improved system performance. However, this pin on the 'LS221 is not an internal connection to the device ground. Hence, if substitution of an 'LS221 onto an 'LS123 design layout where the C_{EXT} pin is wired to the ground, the device will not function.
11. V_{CC} and ground wiring should conform to good high-frequency standards and practices so that switching transients on the V_{CC} and ground return leads do not cause interaction between one-shots. A 0.01 μF to 0.10 μF bypass capacitor (disk ceramic or monolithic type) from V_{CC} to ground is necessary on each device. Furthermore, the bypass capacitor should be located as close to the V_{CC}-pin as space permits.

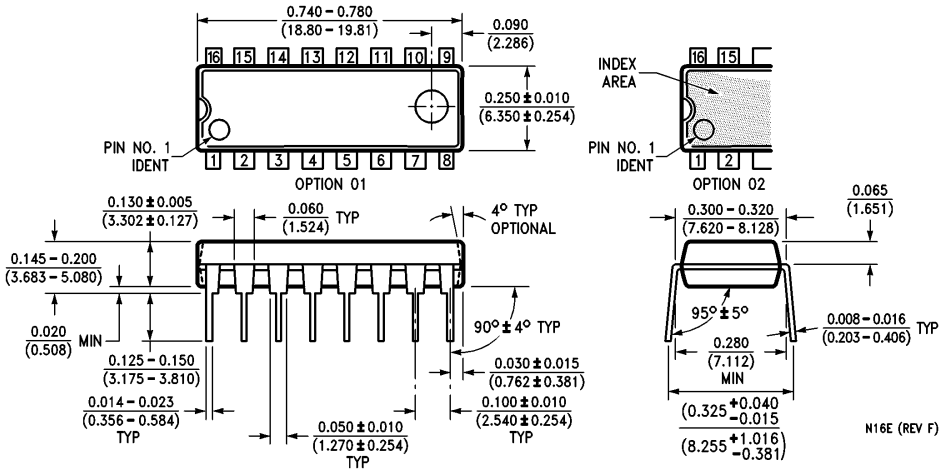
For further detailed device characteristics and output performance, please refer to the NSC one-shot application note AN-372.

Physical Dimensions inches (millimeters)



16-Lead Small Outline Molded Package (M)
Order Number DM74LS221M
NS Package Number M16A

Physical Dimensions inches (millimeters) (Continued)



16-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS221N
NS Package Number N16E

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

April 1992

DM54LS240/DM74LS240, DM54LS241/DM74LS241 Octal TRI-STATE® Buffers/Line Drivers/Line Receivers

General Description

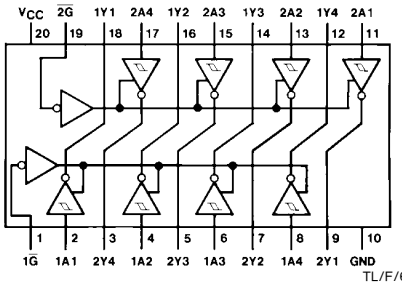
These buffers/line drivers are designed to improve both the performance and PC board density of TRI-STATE buffers/drivers employed as memory-address drivers, clock drivers, and bus-oriented transmitters/receivers. Featuring 400 mV of hysteresis at each low current PNP data line input, they provide improved noise rejection and high fanout outputs and can be used to drive terminated lines down to 133Ω.

Features

- TRI-STATE outputs drive bus lines directly
- PNP inputs reduce DC loading on bus lines
- Hysteresis at data inputs improves noise margins
- Typical I_{OL} (sink current)
  - 54LS 12 mA
  - 74LS 24 mA
- Typical I_{OH} (source current)
  - 54LS −12 mA
  - 74LS −15 mA
- Typical propagation delay times
  - Inverting 10.5 ns
  - Noninverting 12 ns
- Typical enable/disable time 18 ns
- Typical power dissipation (enabled)
  - Inverting 130 mW
  - Noninverting 135 mW

Connection Diagrams

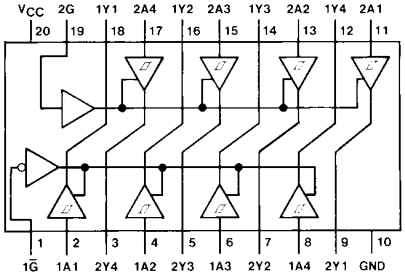
Dual-In-Line Package



TL/F/6411–1

Order Number DM54LS240J, DM54LS240W, DM54LS240E, DM74LS240WM or DM74LS240N
See NS Package Number E20A, J20A, M20B, N20A or W20A

Dual-In-Line Package



TL/F/6411–2

Order Number DM54LS241J, DM54LS241W, DM54LS241E, DM74LS241WM or DM74LS241N
See NS Package Number E20A, J20A, M20B, N20A or W20A

Function Tables

LS240

Inputs		Output
$\overline{G}$	A	Y
L	L	H
L	H	L
H	X	Z

LS241

Inputs				Outputs	
G	$\overline{G}$	1A	2A	1Y	2Y
X	L	L	X	L	
X	L	H	X	H	
X	H	X	X	Z	
H	X	X	L		L
H	X	X	H		H
L	X	X	X		Z

L = Low Logic Level
H = High Logic Level
X = Either Low or High Logic Level
Z = High Impedance

TRI-STATE® is a registered trademark of National Semiconductor Corporation.

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM54LS, 54LS	−55°C to +125°C
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM54LS240, 241			DM74LS240, 241			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.7			0.8	V
I_{OH}	High Level Output Current			−12			−15	mA
I_{OL}	Low Level Output Current			12			24	mA
T_A	Free Air Operating Temperature	−55		125	0		70	°C

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions			Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA					−1.5	V
HYS	Hysteresis ($V_{T+} - V_{T-}$) Data Inputs Only	V_{CC} = Min			0.2	0.4		V
V_{OH}	High Level Output Voltage	V_{CC} = Min, V_{IH} = Min V_{IL} = Max, I_{OH} = −1 mA		DM74	2.7			V
		V_{CC} = Min, V_{IH} = Min V_{IL} = Max, I_{OH} = −3 mA		DM54/DM74	2.4	3.4		
		V_{CC} = Min, V_{IH} = Min V_{IL} = 0.5V, I_{OH} = Max		DM54/DM74	2			
V_{OL}	Low Level Output Voltage	V_{CC} = Min V_{IL} = Max V_{IH} = Min	I_{OL} = 12 mA	DM74			0.4	V
			I_{OL} = Max	DM54			0.4	
				DM74			0.5	
I_{OZH}	Off-State Output Current, High Level Voltage Applied	V_{CC} = Max V_{IL} = Max V_{IH} = Min	V_O = 2.7V				20	μA
I_{OZL}	Off-State Output Current, Low Level Voltage Applied		V_O = 0.4V				−20	μA
I_I	Input Current at Maximum Input Voltage	V_{CC} = Max, V_I = 7V (DM74) V_I = 10V (DM54)					0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V					20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V					−0.2	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)			−40		−225	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open	Outputs High	LS240, LS241		13	23	mA
			Outputs Low	LS240		26	44	
				LS241		27	46	
			Outputs Disabled	LS240		29	50	
				LS241		32	54	

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

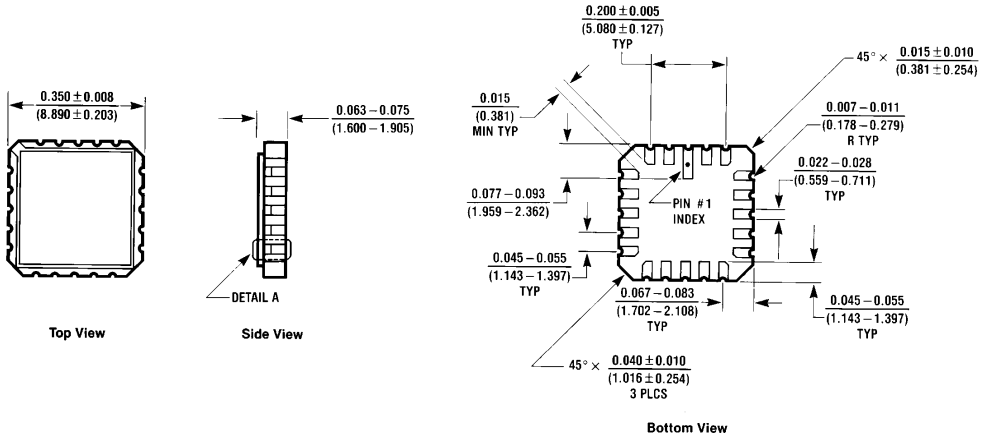
Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

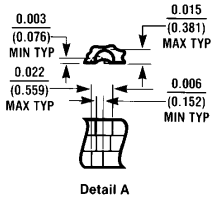
Switching Characteristics at $V_{CC} = 5V$ and $T_A = 25°C$

Symbol	Parameter	Conditions		DM54LS	DM74LS	Units
				Max	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	$C_L = 45$ pF $R_L = 667\Omega$	LS240	18	14	ns
			LS241	18	18	
t_{PHL}	Propagation Delay Time High to Low Level Output	$C_L = 45$ pF $R_L = 667\Omega$	LS240	18	18	ns
			LS241	18	18	
t_{PZL}	Output Enable Time to Low Level	$C_L = 45$ pF $R_L = 667\Omega$	LS240	30	30	ns
			LS241	30	30	
t_{PZH}	Output Enable Time to High Level	$C_L = 45$ pF $R_L = 667\Omega$	LS240	23	23	ns
			LS241	23	23	
t_{PLZ}	Output Disable Time from Low Level	$C_L = 5$ pF $R_L = 667\Omega$	LS240	25	25	ns
			LS241	25	25	
t_{PHZ}	Output Disable Time from High Level	$C_L = 5$ pF $R_L = 667\Omega$	LS240	18	18	ns
			LS241	18	18	
t_{PLH}	Propagation Delay Time Low to High Level Output	$C_L = 150$ pF $R_L = 667\Omega$	LS240		18	ns
			LS241		21	
t_{PHL}	Propagation Delay Time High to Low Level Output	$C_L = 150$ pF $R_L = 667\Omega$	LS240		22	ns
			LS241		22	
t_{PZL}	Output Enable Time to Low Level	$C_L = 150$ pF $R_L = 667\Omega$	LS240		33	ns
			LS241		33	
t_{PZH}	Output Enable Time to High Level	$C_L = 150$ pF $R_L = 667\Omega$	LS240		26	ns
			LS241		26	

Note: 54LS Output load is $C_L = 50$ pF for t_{PLH}, t_{PHL}, t_{PZL} and t_{PZH}.

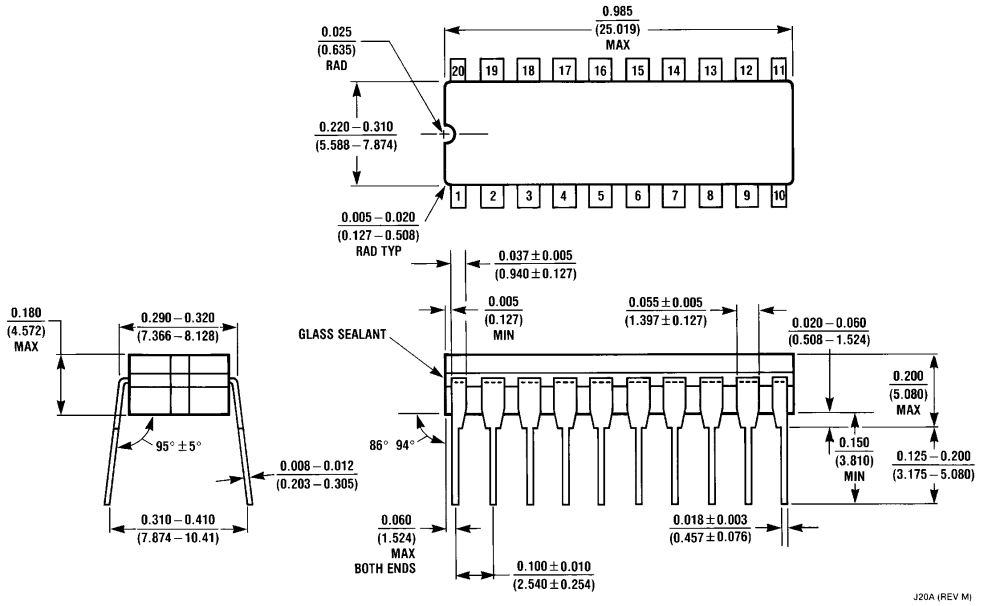
Physical Dimensions inches (millimeters)





E20A (REV D)

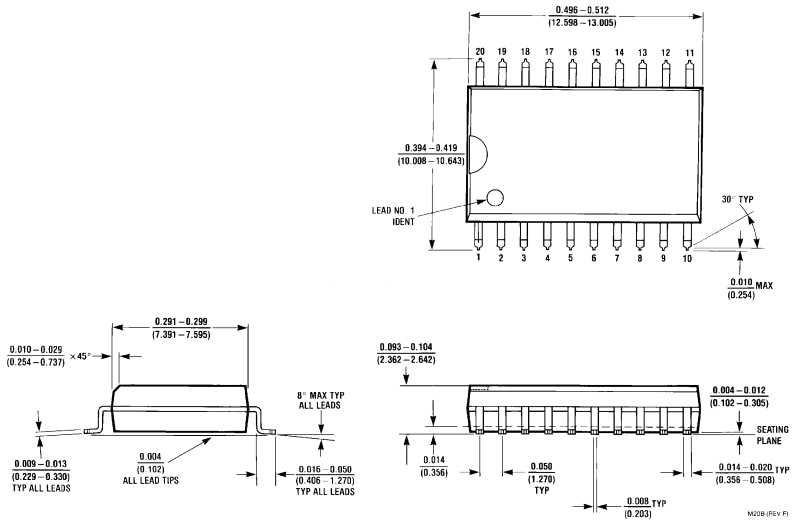
Ceramic Leadless Chip Carrier Package (E)
Order Number DM54LS240E or DM54LS241E
NS Package Number E20A



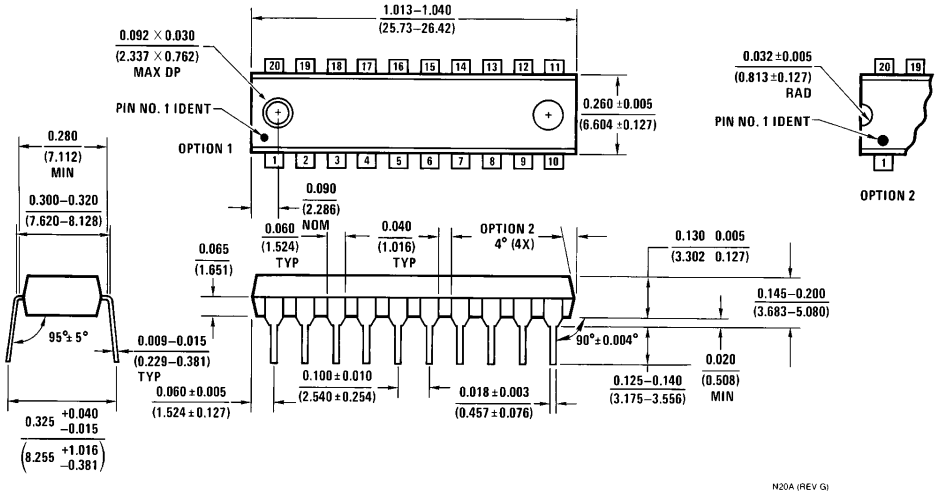
J20A (REV M)

20-Lead Ceramic Dual-In-Line Package (J)
Order Number DM54LS240J or DM54LS241J
NS Package Number J20A

Physical Dimensions inches (millimeters) (Continued)

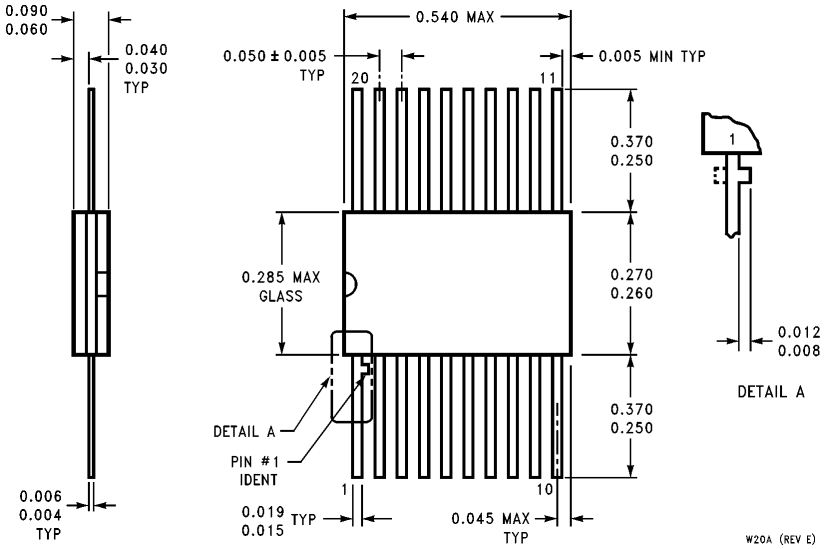


20-Lead Wide Small Outline Molded Package (M)
Order Number DM74LS240WM or DM74LS241WM
NS Package Number M20B



20-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS240N or DM74LS241N
NS Package Number N20A

Physical Dimensions inches (millimeters) (Continued)



20-Lead Ceramic Flat Package (W)
Order Number DM54LS240W or DM54LS241W
NS Package Number W20A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

August 1989

54LS244/DM74LS244 Octal TRI-STATE® Buffers/Line Drivers/Line Receivers

General Description

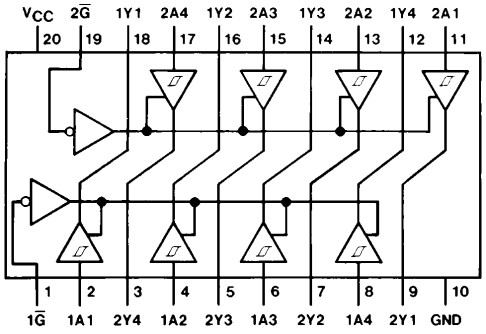
These buffers/line drivers are designed to improve both the performance and PC board density of TRI-STATE buffers/drivers employed as memory-address drivers, clock drivers, and bus-oriented transmitters/receivers. Featuring 400 mV of hysteresis at each low current PNP data line input, they provide improved noise rejection and high fanout outputs and can be used to drive terminated lines down to 133Ω.

Features

- TRI-STATE outputs drive bus lines directly
- PNP inputs reduce DC loading on bus lines
- Hysteresis at data inputs improves noise margins
- Typical I_{OL} (sink current)
  - 54LS 12 mA
  - 74LS 24 mA
- Typical I_{OH} (source current)
  - 54LS −12 mA
  - 74LS −15 mA
- Typical propagation delay times
  - Inverting 10.5 ns
  - Noninverting 12 ns
- Typical enable/disable time 18 ns
- Typical power dissipation (enabled)
  - Inverting 130 mW
  - Noninverting 135 mW

Connection Diagram

Dual-In-Line Package



TL/F/8442–1

Order Number 54LS244DMQB, 54LS244FMQB, 54LS244LMQB, DM74LS244WM or DM74LS244N
See NS Package Number E20A, J20A, M20B, N20A or W20A

Function Table

Inputs		Output
$\overline{G}$	A	Y
L	L	L
L	H	H
H	X	Z

L = Low Logic Level
H = High Logic Level
X = Either Low or High Logic Level
Z = High Impedance

TRI-STATE® is a registered trademark of National Semiconductor Corporation.

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
54LS	−55°C to +125°C
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	54LS244			DM74LS244			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.7			0.8	V
I_{OH}	High Level Output Current			−12			−15	mA
I_{OL}	Low Level Output Current			12			24	mA
T_A	Free Air Operating Temperature	−55		125	0		70	°C

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions			Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA					−1.5	V
HYS	Hysteresis ($V_{T+} - V_{T-}$) Data Inputs Only	V_{CC} = Min			0.2	0.4		V
V_{OH}	High Level Output Voltage	V_{CC} = Min, V_{IH} = Min V_{IL} = Max, I_{OH} = −1 mA		DM74	2.7			V
		V_{CC} = Min, V_{IH} = Min V_{IL} = Max, I_{OH} = −3 mA		54LS/DM74	2.4	3.4		
		V_{CC} = Min, V_{IH} = Min V_{IL} = 0.5V, I_{OH} = Max		54LS/DM74	2			
V_{OL}	Low Level Output Voltage	V_{CC} = Min V_{IL} = Max V_{IH} = Min	I_{OL} = 12 mA	54LS/DM74			0.4	V
			I_{OL} = Max	DM74			0.5	
I_{OZH}	Off-State Output Current, High Level Voltage Applied	V_{CC} = Max V_{IL} = Max V_{IH} = Min	V_O = 2.7V				20	μA
I_{OZL}	Off-State Output Current, Low Level Voltage Applied		V_O = 0.4V				−20	μA
I_I	Input Current at Maximum Input Voltage	V_{CC} = Max	V_I = 7V (DM74) V_I = 10V (54LS)				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max	V_I = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max	V_I = 0.4V		−0.5		−200	μA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)		54LS	−50		−225	mA
				DM74	−40			
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open	Outputs High			13	23	mA
			Outputs Low			27	46	
			Outputs Disabled			32	54	

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

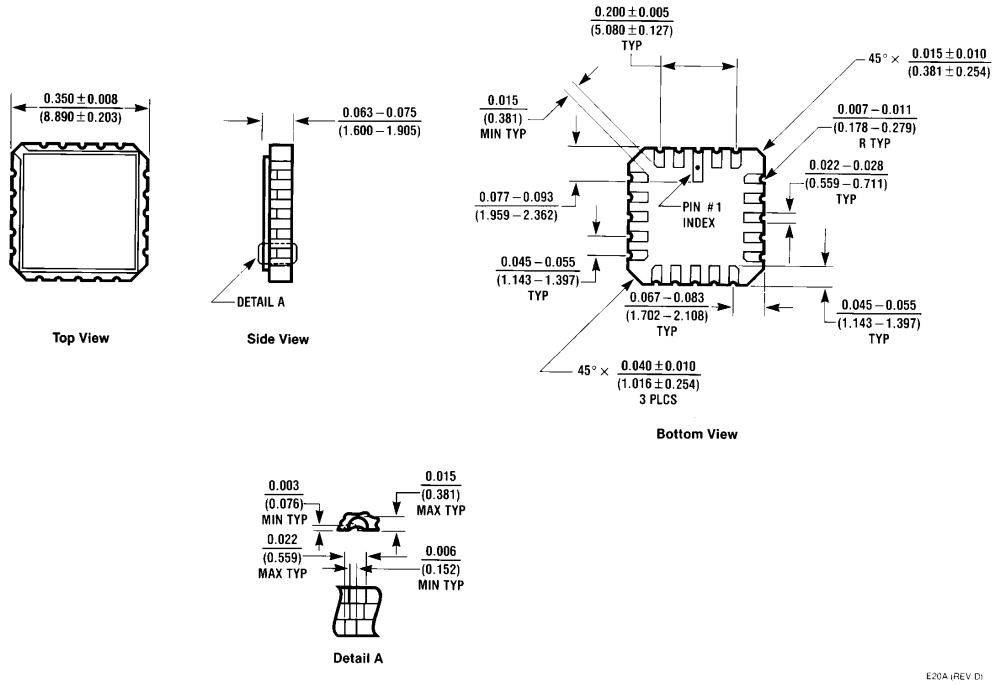
Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Switching Characteristics at $V_{CC} = 5V$, $T_A = 25°C$ (see Section 1 for Test Waveforms and Output Load)

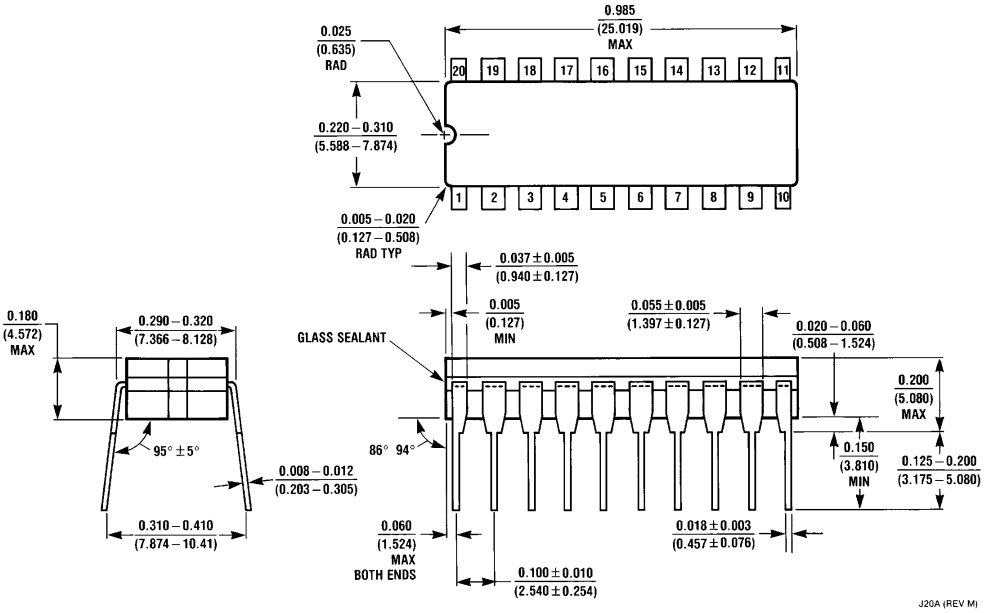
Symbol	Parameter	Conditions	54LS Max	DM74LS Max	Units
t_{PLH}	Propagation Delay Time Low to High Level Output	$C_L = 45$ pF $R_L = 667\Omega$	18	18	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	$C_L = 45$ pF $R_L = 667\Omega$	18	18	ns
t_{PZL}	Output Enable Time to Low Level	$C_L = 45$ pF $R_L = 667\Omega$	30	30	ns
t_{PZH}	Output Enable Time to High Level	$C_L = 45$ pF $R_L = 667\Omega$	23	23	ns
t_{PLZ}	Output Disable Time from Low Level	$C_L = 5$ pF $R_L = 667\Omega$	25	25	ns
t_{PHZ}	Output Disable Time from High Level	$C_L = 5$ pF $R_L = 667\Omega$	18	18	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	$C_L = 150$ pF $R_L = 667\Omega$		21	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	$C_L = 150$ pF $R_L = 667\Omega$		22	ns
t_{PZL}	Output Enable Time to Low Level	$C_L = 150$ pF $R_L = 667\Omega$		33	ns
t_{PZH}	Output Enable Time to High Level	$C_L = 150$ pF $R_L = 667\Omega$		26	ns

Note: 54LS Output Load is $C_L = 50$ pF for t_{PLH}, t_{PHL}, t_{PZL} and t_{PZH}.

Physical Dimensions inches (millimeters)

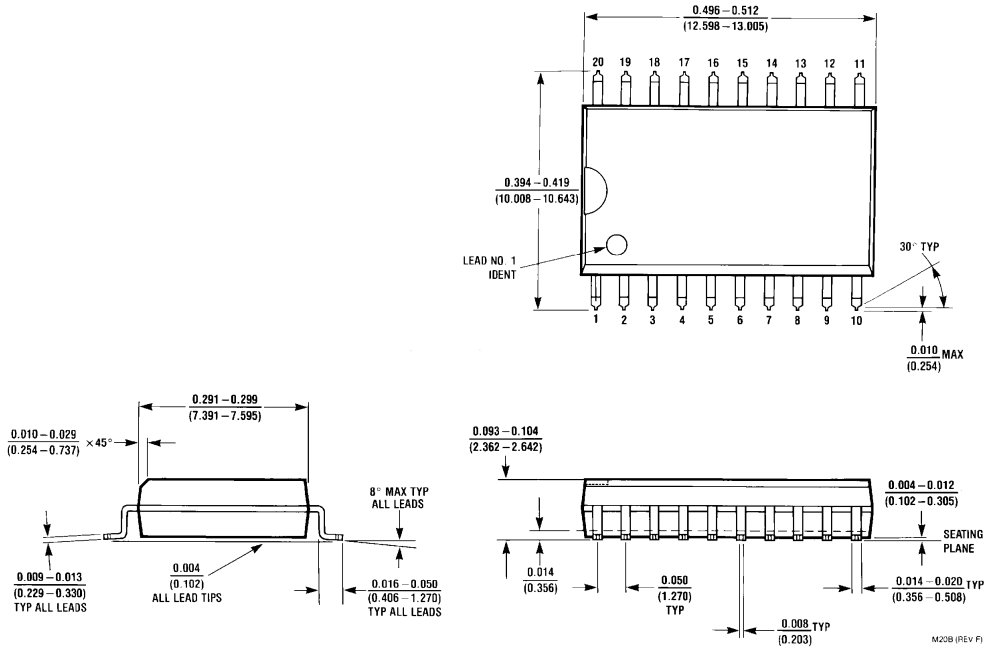


Ceramic Leadless Chip Carrier Package (E)
Order Number 54LS244LMQB
NS Package Number E20A

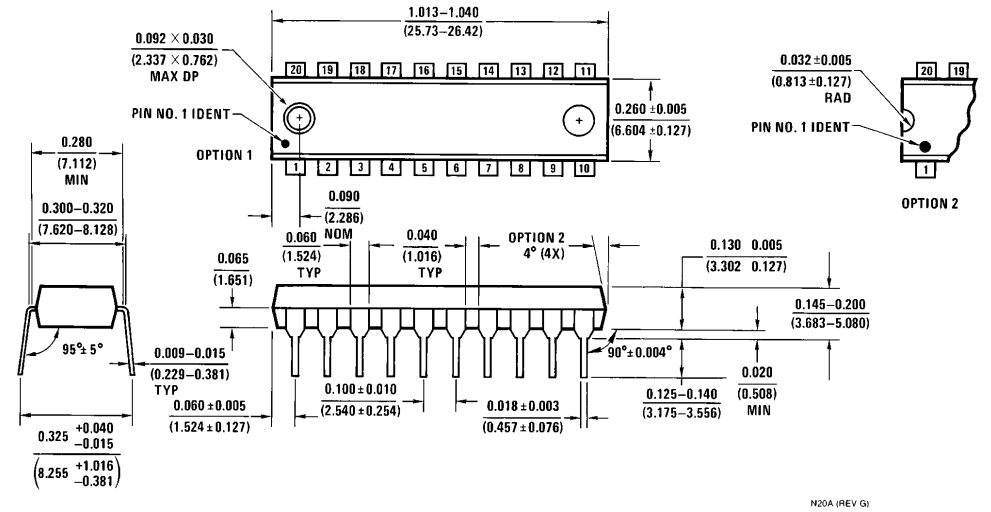


20-Lead Ceramic Dual-In-Line Package (J)
Order Number 54LS244DMQB
NS Package Number J20A

Physical Dimensions inches (millimeters) (Continued)

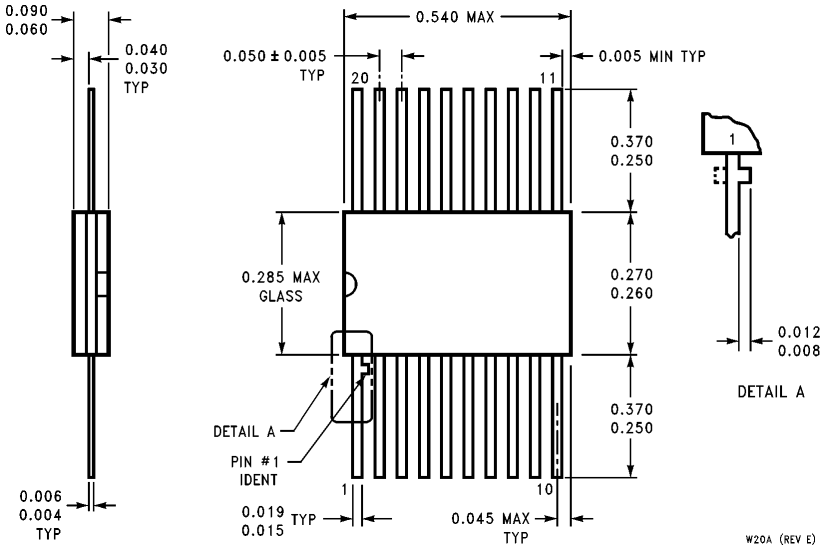


20-Lead Wide Small Outline Molded Package (M)
Order Number DM74LS244WM
NS Package Number M20B



20-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS244N
NS Package Number N20A

Physical Dimensions inches (millimeters) (Continued)



20-Lead Ceramic Flat Package (W)
Order Number 54LS244FMQB
NS Package Number W20A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

June 1989

54LS30/DM54LS30/DM74LS30
8-Input NAND Gate

General Description

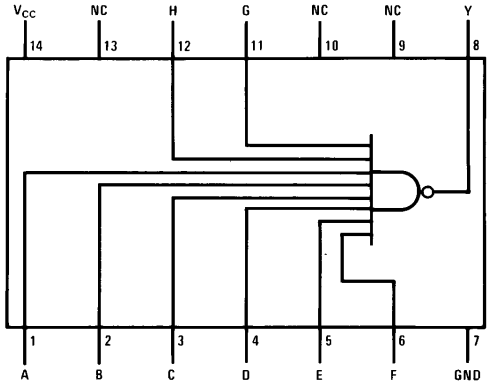
This device contains a single gate which performs the logic NAND function.

Features

- Alternate Military/Aerospace device (54LS30) is available. Contact a National Semiconductor Sales Office/ Distributor for specifications.

Connection Diagram

Dual-In-Line Package



TL/F/6360–1

Order Number 54LS30DMQB, 54LS30FMQB, 54LS30LMQB, DM54LS30J, DM54LS530W, DM74LS30M or DM74LS30N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$$Y = \overline{ABCDEFGH}$$

Inputs	Output
A thru H	Y
All Inputs H	L
One or More Input L	H

H = High Logic Level

L = Low Logic Level

RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM54LS and 54LS	−55°C to +125°C
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM54LS30			DM74LS30			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.7			0.8	V
I_{OH}	High Level Output Current			−0.4			−0.4	mA
I_{OL}	Low Level Output Current			4			8	mA
T_A	Free Air Operating Temperature	−55		125	0		70	°C

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

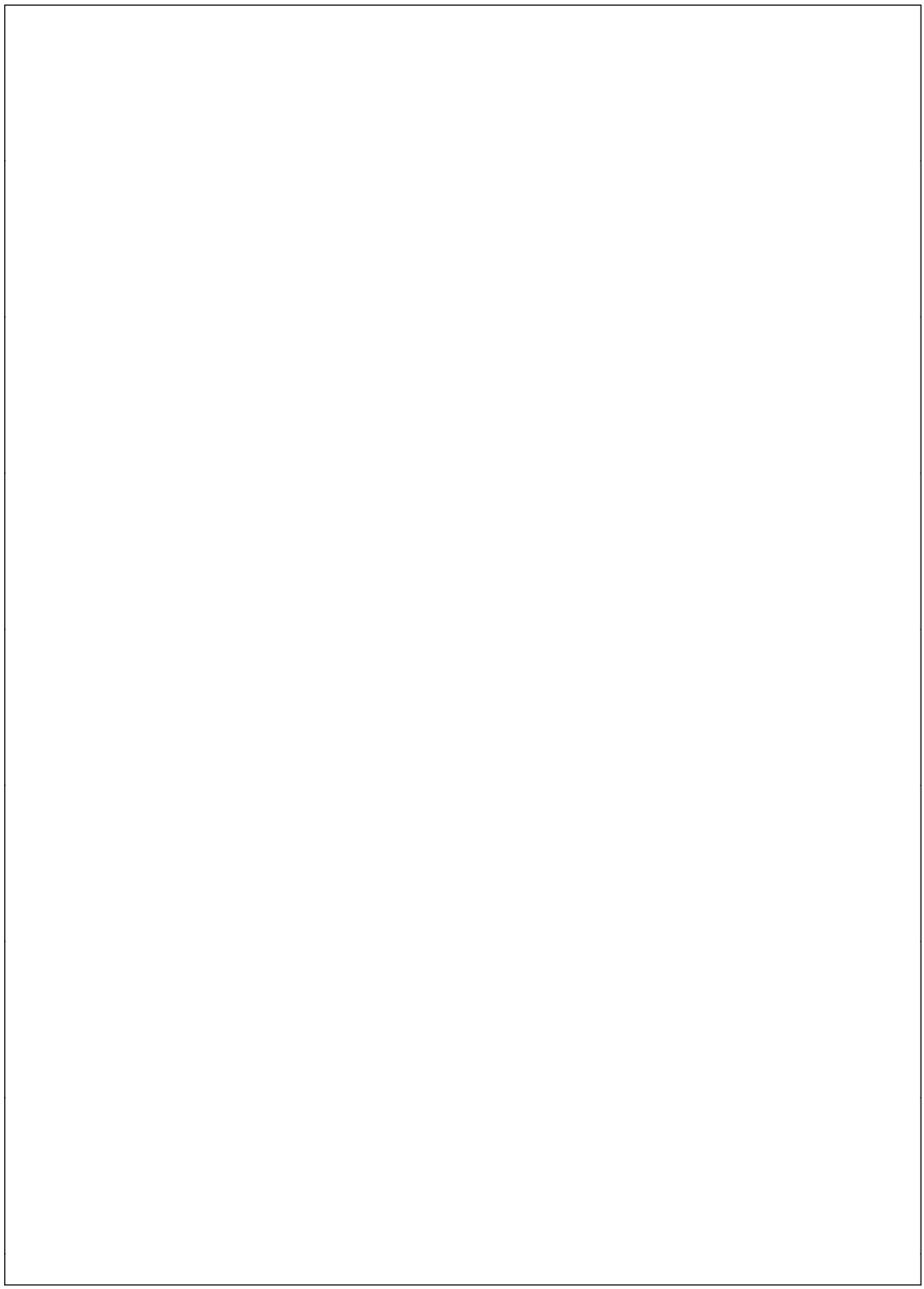
Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max	DM54	2.5	3.4		V
			DM74	2.7	3.4		
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IH} = Min	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 4 mA, V_{CC} = Min	DM74		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.4	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−20		−100	mA
			DM74	−20		−100	
I_{CCH}	Supply Current with Outputs High	V_{CC} = Max			0.35	0.5	mA
I_{CCL}	Supply Current with Outputs Low	V_{CC} = Max			0.6	1.1	mA

Switching Characteristics at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load)

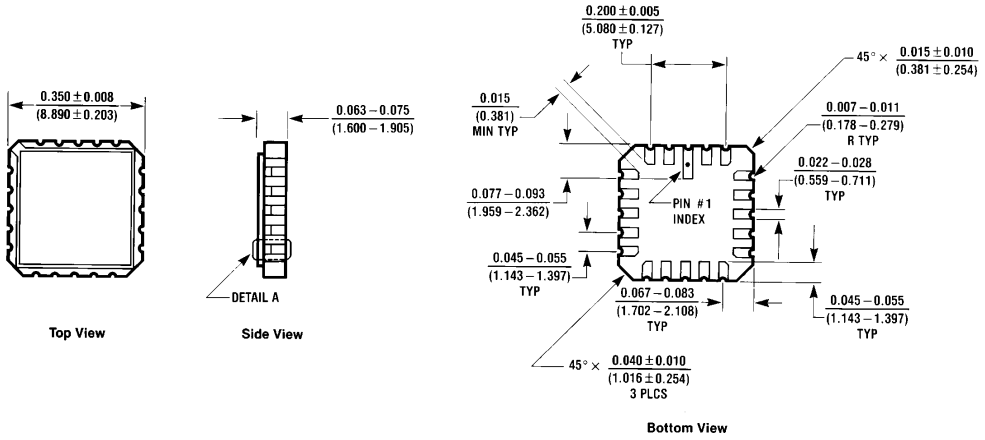
Symbol	Parameter	R_L = 2 kΩ				Units
		C_L = 15 pF		C_L = 50 pF		
		Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	4	12	5	18	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	4	15	5	20	ns

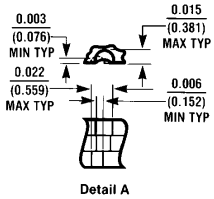
Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.



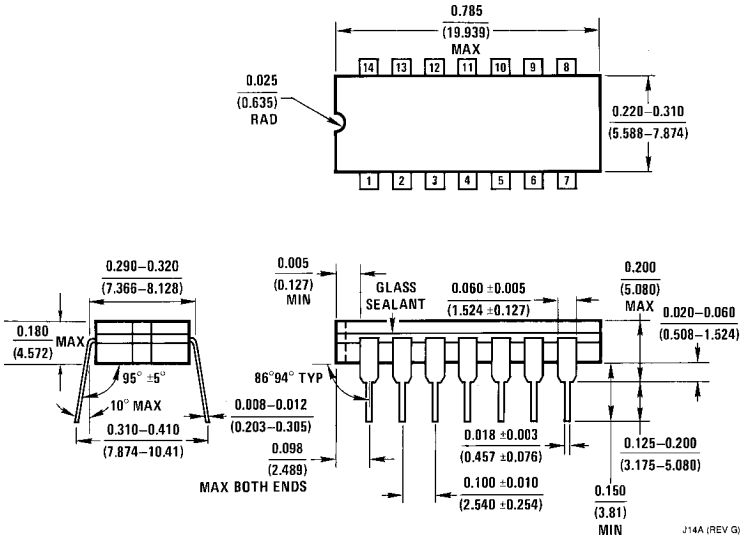
Physical Dimensions inches (millimeters)





E20A (REV D)

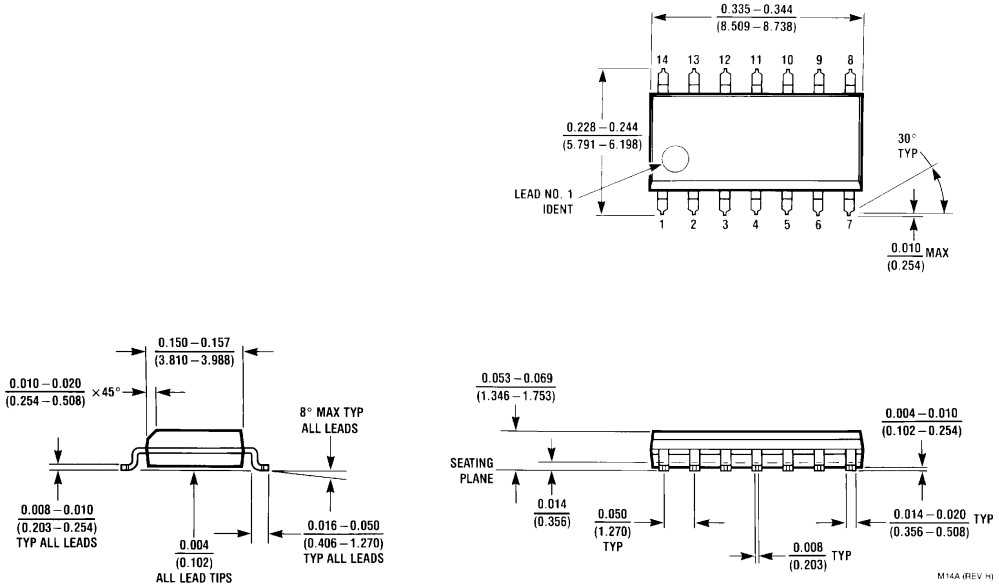
Ceramic Leadless Chip Carrier Package (E)
Order Number 54LS30LMQB
NS Package Number E20A



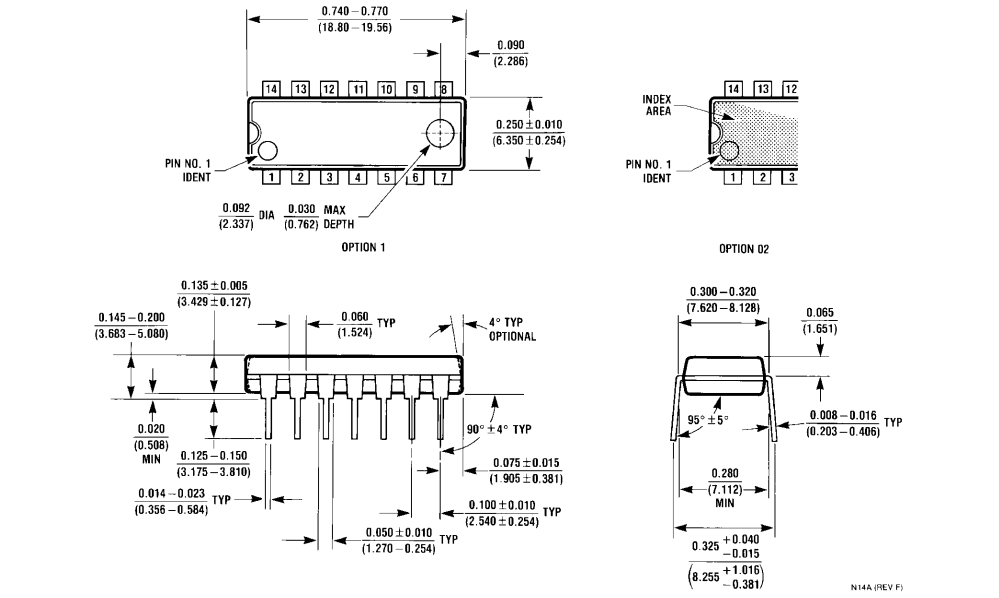
J14A (REV G)

14-Lead Ceramic Dual-In-Line Package (J)
Order Number 54LS30DMQB or DM54LS30J
NS Package Number J14A

Physical Dimensions inches (millimeters) (Continued)

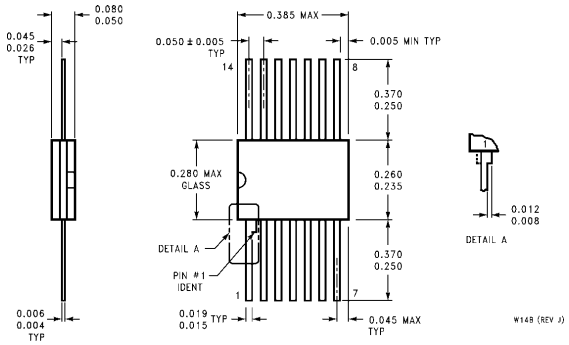


14-Lead Small Outline Molded Package (M)
Order Number DM74LS30M
NS Package Number M14A



14-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS30N
NS Package Number N14A

Physical Dimensions inches (millimeters) (Continued)



14-Lead Ceramic Flat Package (W)
Order Number 54LS30FMQB or DM54LS30W
NS Package Number W14B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

May 1992

DM54LS373/DM74LS373, DM54LS374/DM74LS374 TRI-STATE® Octal D-Type Transparent Latches and Edge-Triggered Flip-Flops

DM54LS373/DM74LS373, DM54LS374/DM74LS374
TRI-STATE Octal D-Type Transparent Latches and Edge-Triggered Flip-Flops

General Description

These 8-bit registers feature totem-pole TRI-STATE outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance state and increased high-logic level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers. (Continued)

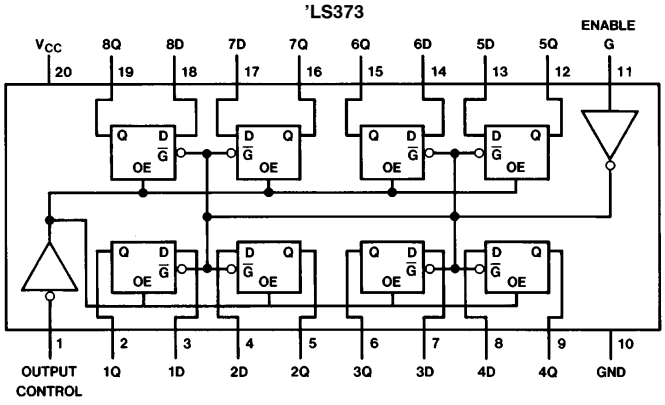
Features

- Choice of 8 latches or 8 D-type flip-flops in a single package
- TRI-STATE bus-driving outputs
- Full parallel-access for loading
- Buffered control inputs
- P-N-P inputs reduce D-C loading on data lines

Connection Diagrams

Dual-In-Line Packages

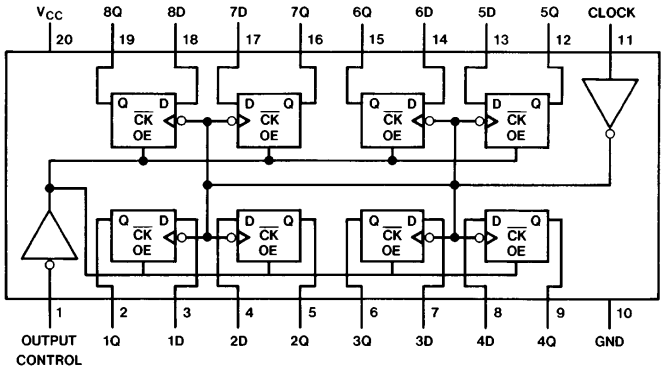
'LS373



Order Number DM54LS373J, DM54LS373W, DM74LS373N or DM74LS373WM
See NS Package Number J20A, M20B, N20A or W20A

TL/F/6431–1

'LS374



Order Number DM54LS374J, DM54LS374W, DM74LS374WM or DM74LS374N
See NS Package Number J20A, M20B, N20A or W20A

TL/F/6431–2

TRI-STATE® is a registered trademark of National Semiconductor Corp.

RRD-B30M105/Printed in U. S. A.

General Description (Continued)

The eight latches of the DM54/74LS373 are transparent D-type latches meaning that while the enable (G) is high the Q outputs will follow the data (D) inputs. When the enable is taken low the output will be latched at the level of the data that was set up.

The eight flip-flops of the DM54/74LS374 are edge-triggered D-type flip flops. On the positive transition of the clock, the Q outputs will be set to the logic states that were set up at the D inputs.

A buffered output control input can be used to place the eight outputs in either a normal logic state (high or low logic levels) or a high-impedance state. In the high-impedance state the outputs neither load nor drive the bus lines significantly.

The output control does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered even while the outputs are off.

Function Tables

DM54/74LS373

Output Control	Enable G	D	Output
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

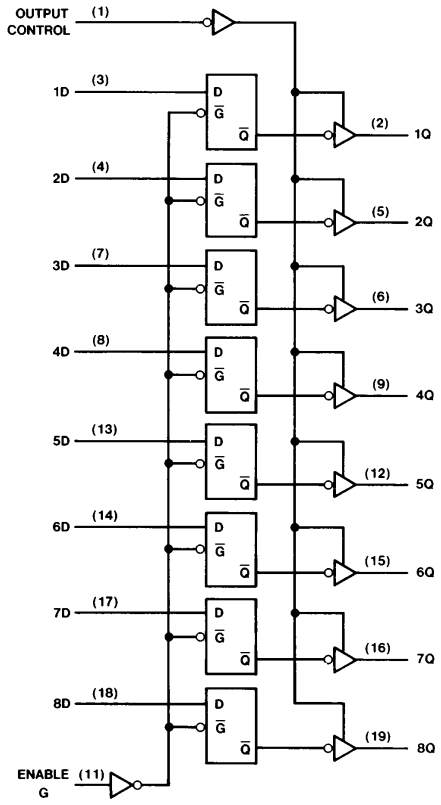
DM54/74LS374

Output Control	Clock	D	Output
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

H = High Level (Steady State), L = Low Level (Steady State), X = Don't Care
↑ = Transition from low-to-high level, Z = High Impedance State
Q_0 = The level of the output before steady-state input conditions were established.

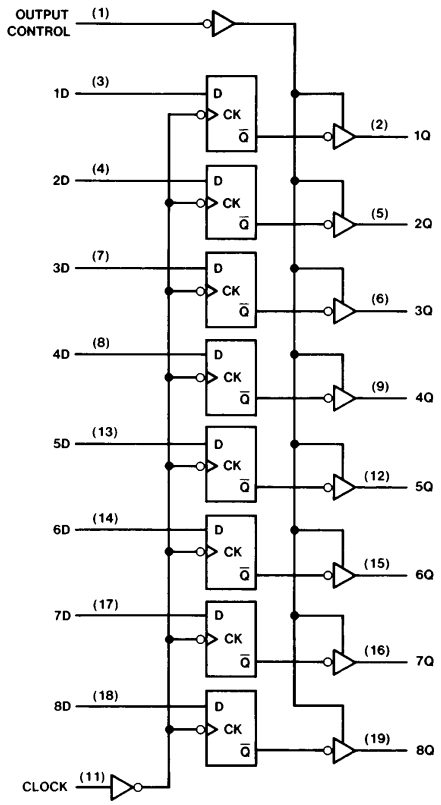
Logic Diagrams

DM54/74LS373
Transparent Latches



TL/F/6431–3

DM54/74LS374
Positive-Edge-Triggered Flip-Flops



TL/F/6431–4

Absolute Maximum Ratings (See Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Storage Temperature Range	−65°C to +150°C
Operating Free Air Temperature Range	
DM54LS	−55°C to +125°C
DM74LS	0°C to +70°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter		DM54LS373			DM74LS373			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Votage		2			2			V
V_{IL}	Low Level Input Voltage				0.7			0.8	V
I_{OH}	High Level Output Current				−1			−2.6	mA
I_{OL}	Low Level Output Current				12			24	mA
t_W	Pulse Width (Note 2)	Enable High	15			15			ns
		Enable Low	15			15			
t_{SU}	Data Setup Time (Notes 1 & 2)		5 ↓			5 ↓			ns
t_H	Data Hold Time (Notes 1 & 2)		20 ↓			20 ↓			ns
T_A	Free Air Operating Temperature		−55		125	0		70	°C

Note 1: The symbol (↓) indicates the falling edge of the clock pulse is used for reference.

Note 2: $T_A = 25°C$ and $V_{CC} = 5V$.

'LS373 Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	DM54	2.4	3.4		V
			DM74	2.4	3.1		
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 12 mA, V_{CC} = Min	DM74			0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.4	mA
I_{OZH}	Off-State Output Current with High Level Output Voltage Applied	V_{CC} = Max, V_O = 2.7V, V_{IH} = Min, V_{IL} = Max				20	μA
I_{OZL}	Off-State Output Current with Low Level Output Voltage Applied	V_{CC} = Max, V_O = 0.4V, V_{IH} = Min, V_{IL} = Max				−20	μA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−20		−100	mA
			DM74	−50		−225	
I_{CC}	Supply Current	V_{CC} = Max, OC = 4.5V, D_n, Enable = GND			24	40	mA

'LS373 Switching Characteristics at $V_{CC} = 5V$ and $T_A = 25°C$

(See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	From (Input) To (Output)	$R_L = 667\Omega$				Units
			$C_L = 45$ pF		$C_L = 150$ pF		
			Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	Data to Q		18		26	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Data to Q		18		27	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Enable to Q		30		38	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Enable to Q		30		36	ns
t_{PZH}	Output Enable Time to High Level Output	Output Control to Any Q		28		36	ns
t_{PZL}	Output Enable Time to Low Level Output	Output Control to Any Q		36		50	ns
t_{PHZ}	Output Disable Time from High Level Output (Note 3)	Output Control to Any Q		20			ns
t_{PLZ}	Output Disable Time from Low Level Output (Note 3)	Output Control to Any Q		25			ns

Note 1: All typicals are at $V_{CC} = 5V$, $T_A = 25°C$.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Note 3: $C_L = 5$ pF.

Recommended Operating Conditions

Symbol	Parameter		DM54LS374			DM74LS374			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage		2			2			V
V_{IL}	Low Level Input Voltage				0.7			0.8	V
I_{OH}	High Level Output Current				−1			−2.6	mA
I_{OL}	Low Level Output Current				12			24	mA
t_W	Pulse Width (Note 4)	Clock High	15			15			ns
		Clock Low	15			15			
t_{SU}	Data Setup Time (Notes 1 & 4)		20 ↑			20 ↑			ns
t_H	Data Hold Time (Notes 1 & 4)		1 ↑			1 ↑			ns
T_A	Free Air Operating Temperature		−55		125	0		70	°C

Note 1: The symbol (↑) indicates the rising edge of the clock pulse is used for reference.

Note 4: $T_A = 25°C$ and $V_{CC} = 5V$.

'LS374 Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	DM54	2.4	3.4		V
			DM74	2.4	3.1		
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 12 mA, V_{CC} = Min	DM74		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.4	mA
I_{OZH}	Off-State Output Current with High Level Output Voltage Applied	V_{CC} = Max, V_O = 2.7V, V_{IH} = Min, V_{IL} = Max				20	μA
I_{OZL}	Off-State Output Current with Low Level Output Voltage Applied	V_{CC} = Max, V_O = 0.4V, V_{IH} = Min, V_{IL} = Max				−20	μA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−50		−225	mA
			DM74	−50		−225	
I_{CC}	Supply Current	V_{CC} = Max, D_n = GND, OC = 4.5V			27	45	mA

'LS374 Switching Characteristics at V_{CC} = 5V and T_A = 25°C

(See Section 1 for Test Waveforms and Output Load)

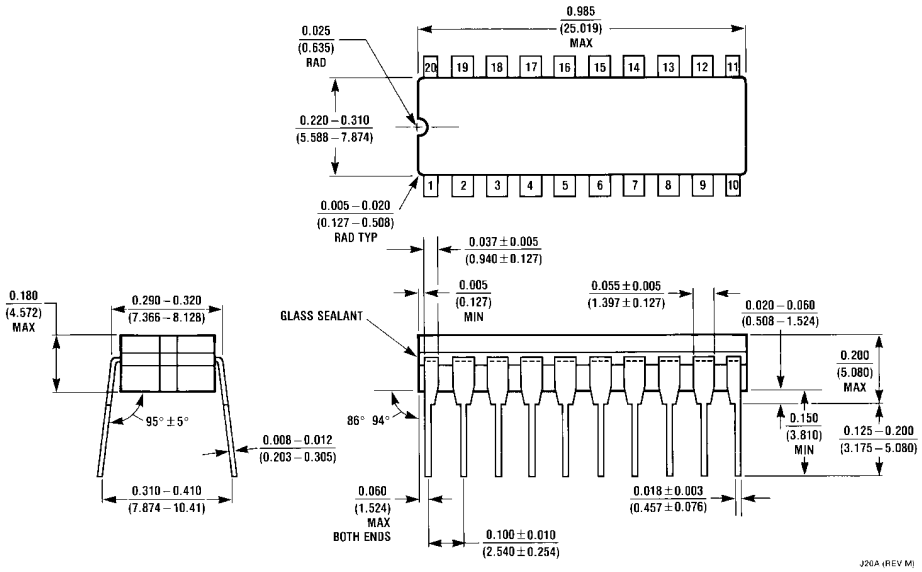
Symbol	Parameter	R_L = 667Ω				Units
		C_L = 45 pF		C_L = 150 pF		
		Min	Max	Min	Max	
f_{MAX}	Maximum Clock Frequency	35		20		MHz
t_{PLH}	Propagation Delay Time Low to High Level Output		28		32	ns
t_{PHL}	Propagation Delay Time High to Low Level Output		28		38	ns
t_{PZH}	Output Enable Time to High Level Output		28		44	ns
t_{PZL}	Output Enable Time to Low Level Output		28		44	ns
t_{PHZ}	Output Disable Time from High Level Output (Note 3)		20			ns
t_{PLZ}	Output Disable Time from Low Level Output (Note 3)		25			ns

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

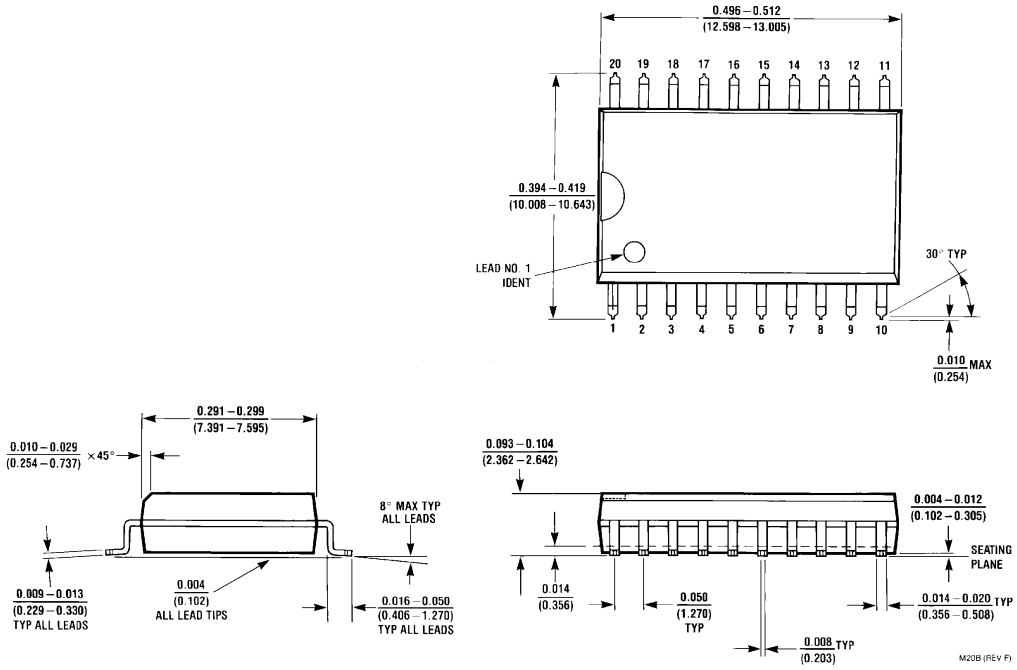
Note 3: C_L = 5 pF.

Physical Dimensions inches (millimeters)

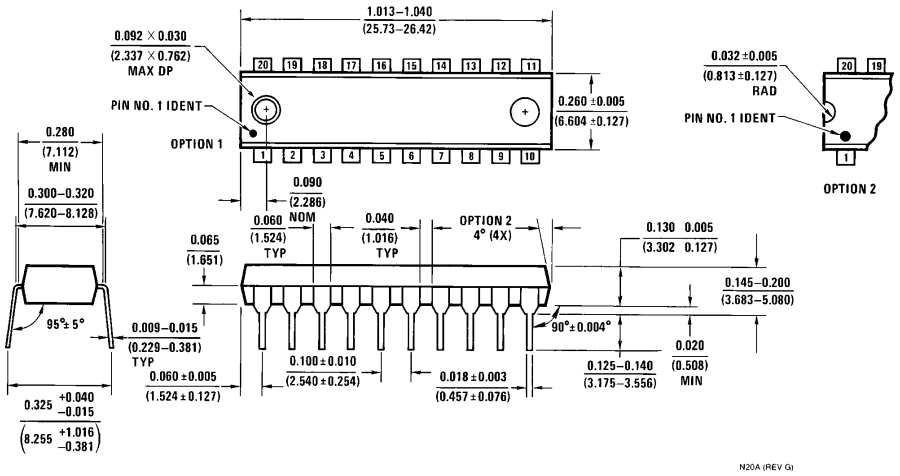


20-Lead Ceramic Dual-In-Line Package (J)
Order Number DM54LS373J or DM54LS374J
NS Package Number J20A

Physical Dimensions inches (millimeters) (Continued)

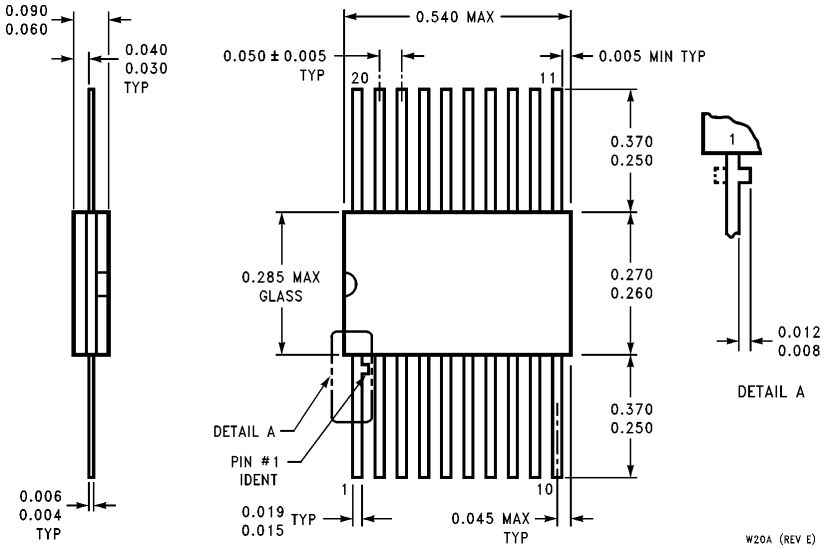


20-Lead Wide Small Outline Molded Package (M)
Order Number DM74LS373WM or DM74LS374WM
NS Package Number M20B



20-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS373N and DM74LS374N
NS Package Number N20A

Physical Dimensions inches (millimeters) (Continued)



20-Lead Ceramic Flat Package (W)
Order Number DM54LS373W or DM54LS374W
NS Package Number W20A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

June 1989

54LS74/DM54LS74A/DM74LS74A Dual Positive-Edge-Triggered D Flip-Flops with Preset, Clear and Complementary Outputs

General Description

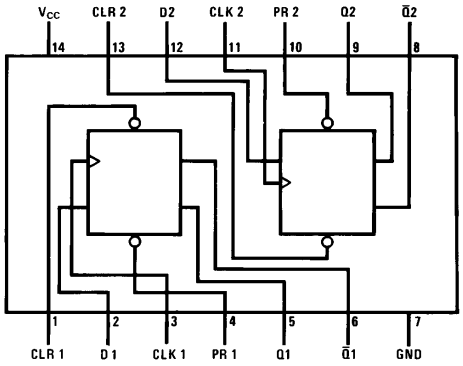
This device contains two independent positive-edge-triggered D flip-flops with complementary outputs. The information on the D input is accepted by the flip-flops on the positive going edge of the clock pulse. The triggering occurs at a voltage level and is not directly related to the transition time of the rising edge of the clock. The data on the D input may be changed while the clock is low or high without affecting the outputs as long as the data setup and hold times are not violated. A low logic level on the preset or clear inputs will set or reset the outputs regardless of the logic levels of the other inputs.

Features

- Alternate military/aerospace device (54LS74) is available. Contact a National Semiconductor Sales Office/ Distributor for specifications.

Connection Diagram

Dual-In-Line Package



TL/F/6373–1

Order Number 54LS74DMQB, 54LS74FMQB, 54LS74LMQB, DM54LS74AJ, DM54LS74AW, DM74LS74AM or DM74LS74AN
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

Inputs				Outputs	
PR	CLR	CLK	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\overline{Q}_0$

H = High Logic Level

X = Either Low or High Logic Level

L = Low Logic Level

↑ = Positive-going Transition

* = This configuration is nonstable; that is, it will not persist when either the preset and/or clear inputs return to their inactive (high) level.

Q_0 = The output logic level of Q before the indicated input conditions were established.

54LS74/DM54LS74A/DM74LS74A Dual Positive-Edge-Triggered D Flip-Flops with Preset, Clear and Complementary Outputs

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM54LS and 54LS	−55°C to +125°C
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter		DM54LS74A			DM74LS74A			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage		2			2			V
V_{IL}	Low Level Input Voltage				0.7			0.8	V
I_{OH}	High Level Output Current				−0.4			−0.4	mA
I_{OL}	Low Level Output Current				4			8	mA
f_{CLK}	Clock Frequency (Note 2)		0		25	0		25	MHz
f_{CLK}	Clock Frequency (Note 3)		0		20	0		20	MHz
t_W	Pulse Width (Note 2)	Clock High	18			18			ns
		Preset Low	15			15			
		Clear Low	15			15			
t_W	Pulse Width (Note 3)	Clock High	25			25			ns
		Preset Low	20			20			
		Clear Low	20			20			
t_{SU}	Setup Time (Notes 1 and 2)		20 ↑			20 ↑			ns
t_{SU}	Setup Time (Notes 1 and 3)		25 ↑			25 ↑			ns
t_H	Hold Time (Note 1 and 4)		0 ↑			0 ↑			ns
T_A	Free Air Operating Temperature		−55		125	0		70	°C

Note 1: The symbol (↑) indicates the rising edge of the clock pulse is used for reference.

Note 2: $C_L = 15$ pF, $R_L = 2$ kΩ, $T_A = 25°C$, and $V_{CC} = 5V$.

Note 3: $C_L = 50$ pF, $R_L = 2$ kΩ, $T_A = 25°C$, and $V_{CC} = 5V$.

Note 4: $T_A = 25°C$ and $V_{CC} = 5V$.

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	DM54	2.5	3.4		V
			DM74	2.7	3.4		
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 4 mA, V_{CC} = Min	DM74		0.25	0.4	
I_I	Input Current @Max Input Voltage	V_{CC} = Max, V_I = 7V	Data			0.1	mA
			Clock			0.1	
			Preset			0.2	
			Clear			0.2	
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V	Data			20	μA
			Clock			20	
			Clear			40	
			Preset			40	
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V	Data			−0.4	mA
			Clock			−0.4	
			Preset			−0.8	
			Clear			−0.8	
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−20		−100	mA
			DM74	−20		−100	
I_{CC}	Supply Current	V_{CC} = Max (Note 3)			4	8	mA

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

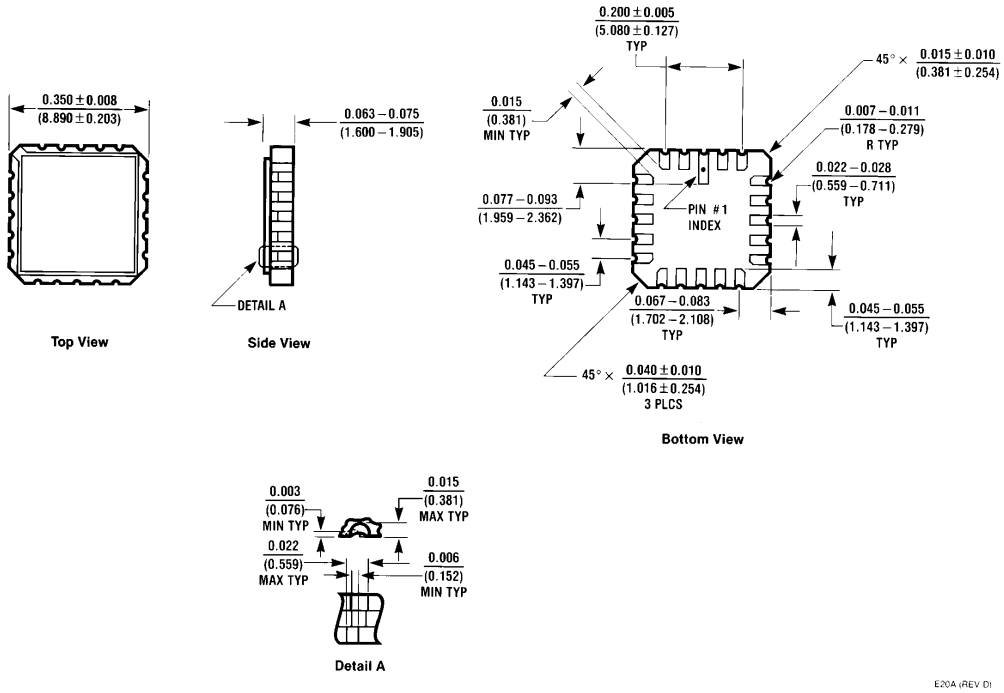
Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second. For devices, with feedback from the outputs, where shorting the outputs to ground may cause the outputs to change logic state an equivalent test may be performed where V_O = 2.25V and 2.125V for DM54 and DM74 series, respectively, with the minimum and maximum limits reduced by one half from their stated values. This is very useful when using automatic test equipment.

Note 3: With all outputs open, I_{CC} is measured with CLOCK grounded after setting the Q and $\overline{Q}$ outputs high in turn.

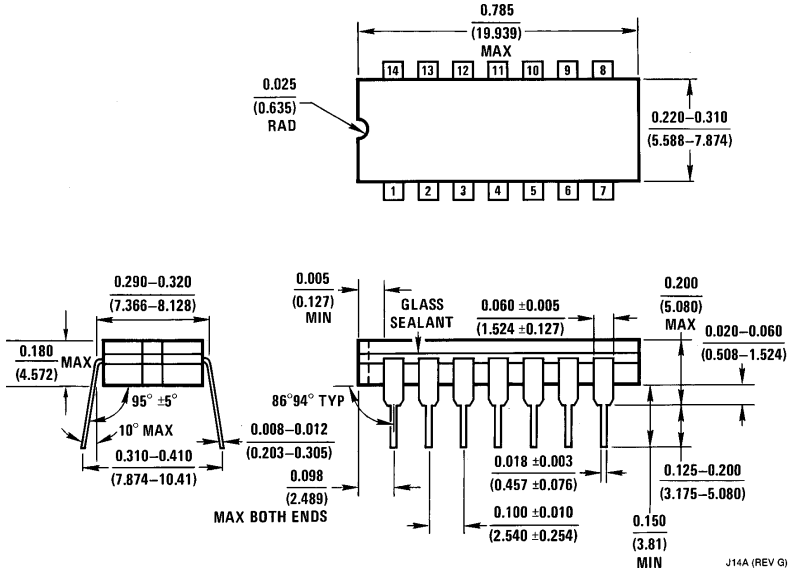
Switching Characteristics at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	From (Input) To (Output)	R_L = 2 kΩ				Units
			C_L = 15 pF		C_L = 50 pF		
			Min	Max	Min	Max	
f_{MAX}	Maximum Clock Frequency		25		20		MHz
t_{PLH}	Propagation Delay Time Low to High Level Output	Clock to Q or $\overline{Q}$		25		35	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Clock to Q or $\overline{Q}$		30		35	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Preset to Q		25		35	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Preset to $\overline{Q}$		30		35	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Clear to $\overline{Q}$		25		35	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Clear to Q		30		35	ns

Physical Dimensions inches (millimeters)

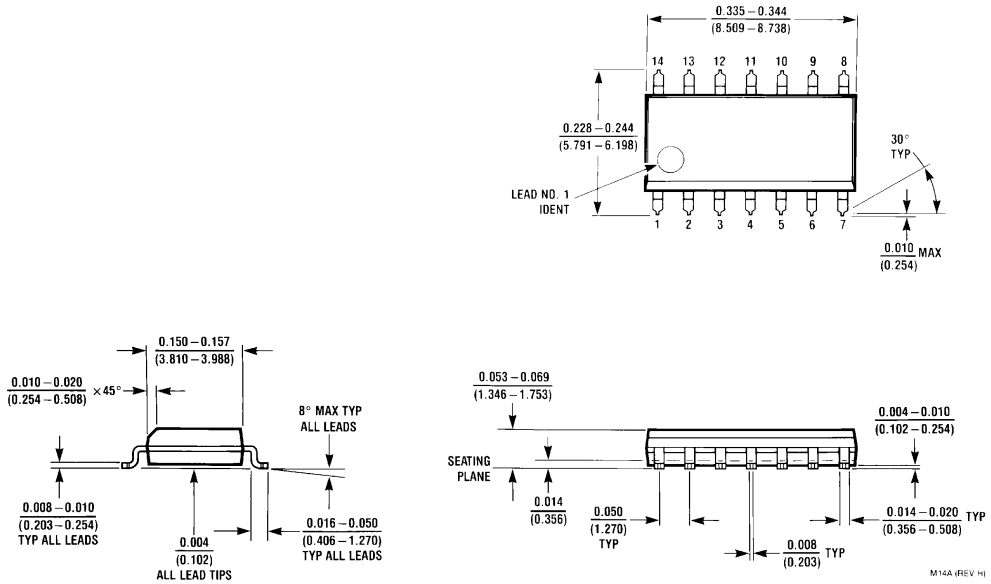


Ceramic Leadless Chip Carrier Package (E)
Order Number 54LS74LMQB
NS Package Number E20A

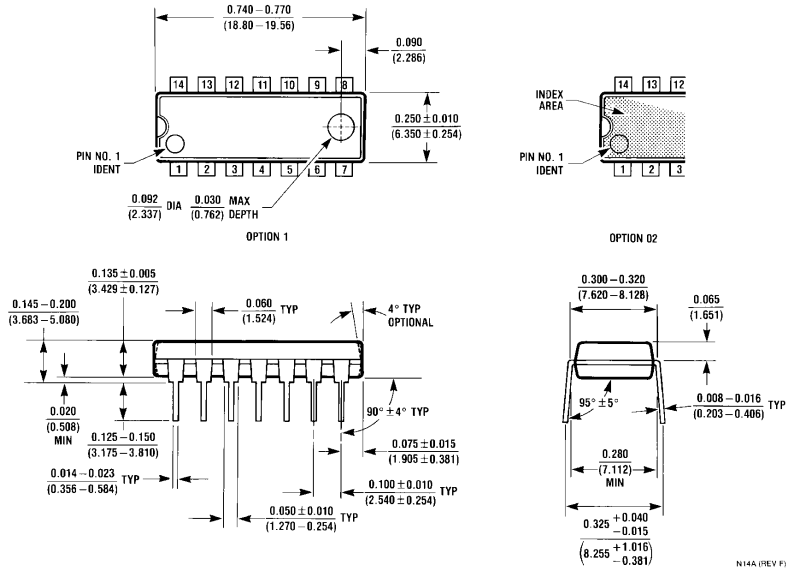


14-Lead Ceramic Dual-In-Line Package (J)
Order Number 54LS74DMQB or DM54LS74AJ
NS Package Number J14A

Physical Dimensions inches (millimeters) (Continued)

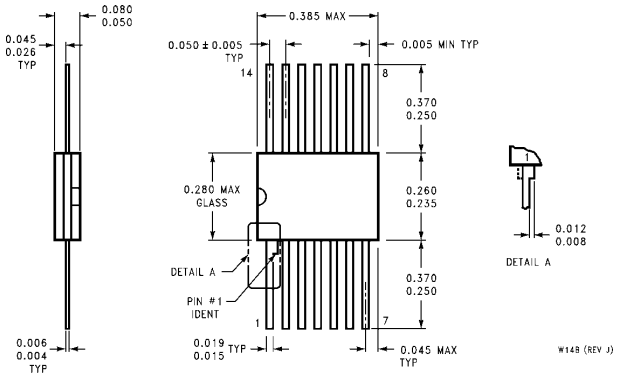


14-Lead Small Outline Molded Package (M)
Order Number DM74LS74AM
NS Package Number M14A



14-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS74AN
NS Package Number N14A

Physical Dimensions inches (millimeters) (Continued)



14-Lead Ceramic Flat Package (W)
Order Number 54LS74FMQB or DM54LS74AW
NS Package Number W14B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

February 1995

LM78XX Series Voltage Regulators

General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expanded to make the LM78XX series of regulators easy to use and mininize the number of external components. It is not necessary to bypass the output, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

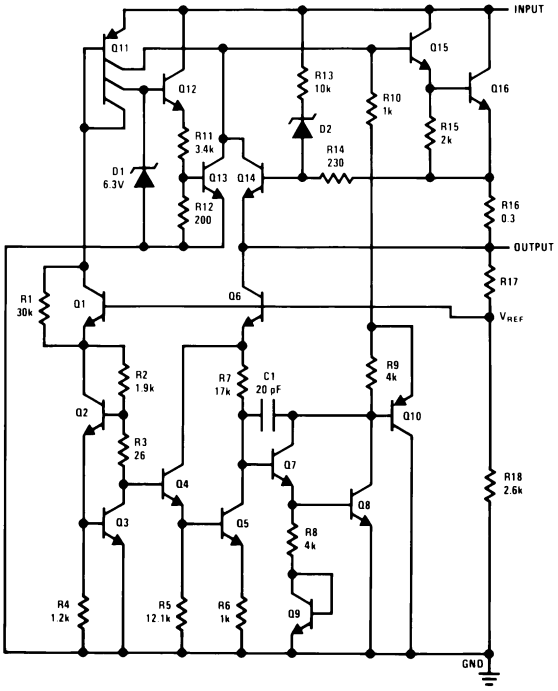
Features

- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

Voltage Range

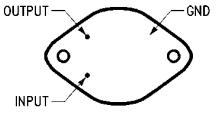
LM7805C	5V
LM7812C	12V
LM7815C	15V

Schematic and Connection Diagrams



TL/H/7746–1

Metal Can Package
TO-3 (K)
Aluminum

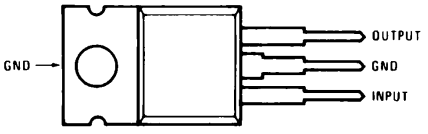


TL/H/7746–2

Bottom View

Order Number LM7805CK,
LM7812CK or LM7815CK
See NS Package Number KC02A

Plastic Package
TO-220 (T)



TL/H/7746–3

Top View

Order Number LM7805CT,
LM7812CT or LM7815CT
See NS Package Number T03B

RRD-B30M115/Printed in U. S. A.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Input Voltage (V_O = 5V, 12V and 15V)	35V
Internal Power Dissipation (Note 1)	Internally Limited
Operating Temperature Range (T_A)	0°C to +70°C
Maximum Junction Temperature	
(K Package)	150°C
(T Package)	150°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	
TO-3 Package K	300°C
TO-220 Package T	230°C

Electrical Characteristics LM78XXC (Note 2) 0°C ≤ Tj ≤ 125°C unless otherwise noted.

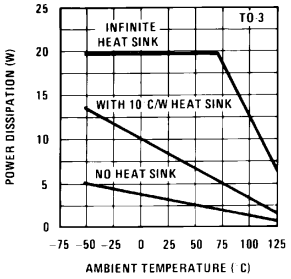
Output Voltage				5V			12V			15V			Units
Input Voltage (unless otherwise noted)				10V			19V			23V			
Symbol	Parameter	Conditions		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
V_O	Output Voltage	Tj = 25°C, 5 mA ≤ I_O ≤ 1A		4.8	5	5.2	11.5	12	12.5	14.4	15	15.6	V
		P_D ≤ 15W, 5 mA ≤ I_O ≤ 1A V_{MIN} ≤ V_{IN} ≤ V_{MAX}		4.75 (7.5 ≤ V_{IN} ≤ 20)		5.25	11.4 (14.5 ≤ V_{IN} ≤ 27)		12.6	14.25 (17.5 ≤ V_{IN} ≤ 30)		15.75	V V
ΔV_O	Line Regulation	I_O = 500 mA	Tj = 25°C		3	50		4	120		4	150	mV
			ΔV_{IN}		(7 ≤ V_{IN} ≤ 25)			14.5 ≤ V_{IN} ≤ 30)			(17.5 ≤ V_{IN} ≤ 30)		V
			0°C ≤ Tj ≤ +125°C			50			120			150	mV
			ΔV_{IN}		(8 ≤ V_{IN} ≤ 20)			(15 ≤ V_{IN} ≤ 27)			(18.5 ≤ V_{IN} ≤ 30)		V
		I_O ≤ 1A	Tj = 25°C			50			120			150	mV
			ΔV_{IN}		(7.5 ≤ V_{IN} ≤ 20)			(14.6 ≤ V_{IN} ≤ 27)			(17.7 ≤ V_{IN} ≤ 30)		V
			0°C ≤ Tj ≤ +125°C			25			60			75	mV
			ΔV_{IN}		(8 ≤ V_{IN} ≤ 12)			(16 ≤ V_{IN} ≤ 22)			(20 ≤ V_{IN} ≤ 26)		V
ΔV_O	Load Regulation	Tj = 25°C	5 mA ≤ I_O ≤ 1.5A		10	50		12	120		12	150	mV
			250 mA ≤ I_O ≤ 750 mA			25			60			75	mV
		5 mA ≤ I_O ≤ 1A, 0°C ≤ Tj ≤ +125°C				50			120			150	mV
I_Q	Quiescent Current	I_O ≤ 1A	Tj = 25°C			8			8			8	mA
			0°C ≤ Tj ≤ +125°C			8.5			8.5			8.5	mA
ΔI_Q	Quiescent Current Change	5 mA ≤ I_O ≤ 1A				0.5			0.5			0.5	mA
		Tj = 25°C, I_O ≤ 1A				1.0			1.0			1.0	mA
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}			(7.5 ≤ V_{IN} ≤ 20)			(14.8 ≤ V_{IN}≤ 27)			(17.9 ≤ V_{IN} ≤ 30)		V
		I_O ≤ 500 mA, 0°C ≤ Tj ≤ +125°C				1.0			1.0			1.0	mA
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}			(7 ≤ V_{IN} ≤ 25)			(14.5 ≤ V_{IN}≤ 30)			(17.5 ≤ V_{IN} ≤ 30)		V
V_N	Output Noise Voltage	T_A = 25°C, 10 Hz ≤ f ≤ 100 kHz			40			75			90		μV
$\frac{\Delta V_{IN}}{\Delta V_{OUT}}$	Ripple Rejection	f = 120 Hz	I_O ≤ 1A, Tj = 25°C or	62	80		55	72		54	70		dB
			I_O ≤ 500 mA	62			55			54			dB
			0°C ≤ Tj ≤ +125°C										
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}		(8 ≤ V_{IN} ≤ 18)			(15 ≤ V_{IN} ≤ 25)			(18.5 ≤ V_{IN} ≤ 28.5)			V
R_O	Dropout Voltage	Tj = 25°C, I_{OUT} = 1A			2.0			2.0			2.0		V
	Output Resistance	f = 1 kHz			8			18			19		mΩ
	Short-Circuit Current	Tj = 25°C			2.1			1.5			1.2		A
	Peak Output Current	Tj = 25°C			2.4			2.4			2.4		A
	Average TC of V_{OUT}	0°C ≤ Tj ≤ +125°C, I_O = 5 mA			0.6			1.5			1.8		mV/°C
V_{IN}	Input Voltage Required to Maintain Line Regulation	Tj = 25°C, I_O ≤ 1A			7.5			14.6			17.7		V

Note 1: Thermal resistance of the TO-3 package (K, KC) is typically 4°C/W junction to case and 35°C/W case to ambient. Thermal resistance of the TO-220 package (T) is typically 4°C/W junction to case and 50°C/W case to ambient.

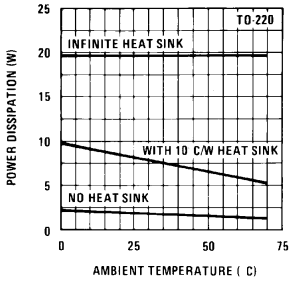
Note 2: All characteristics are measured with capacitor across the input of 0.22 μF, and a capacitor across the output of 0.1μF. All characteristics except noise voltage and ripple rejection ratio are measured using pulse techniques (t_w ≤ 10 ms, duty cycle ≤ 5%). Output voltage changes due to changes in internal temperature must be taken into account separately.

Typical Performance Characteristics

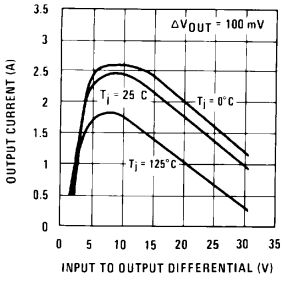
Maximum Average Power Dissipation



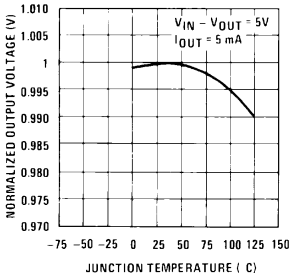
Maximum Average Power Dissipation



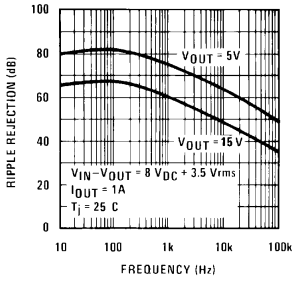
Peak Output Current



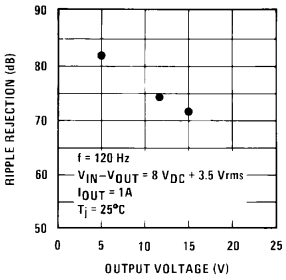
Output Voltage (Normalized to 1V at Tj = 25°C)



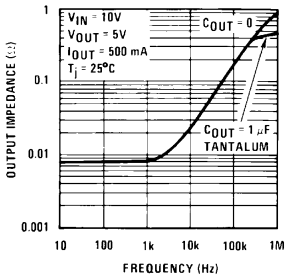
Ripple Rejection



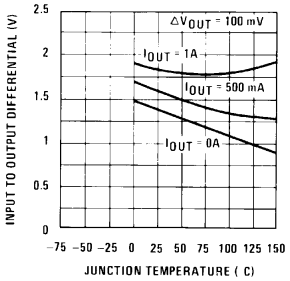
Ripple Rejection



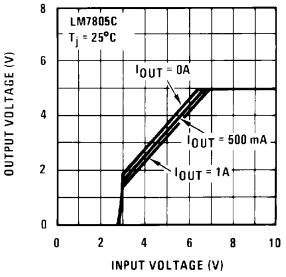
Output Impedance



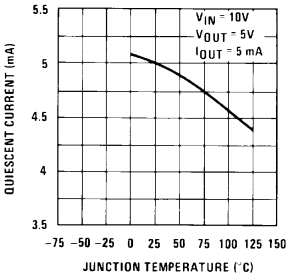
Dropout Voltage



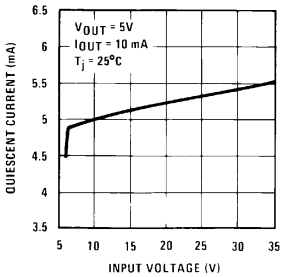
Dropout Characteristics



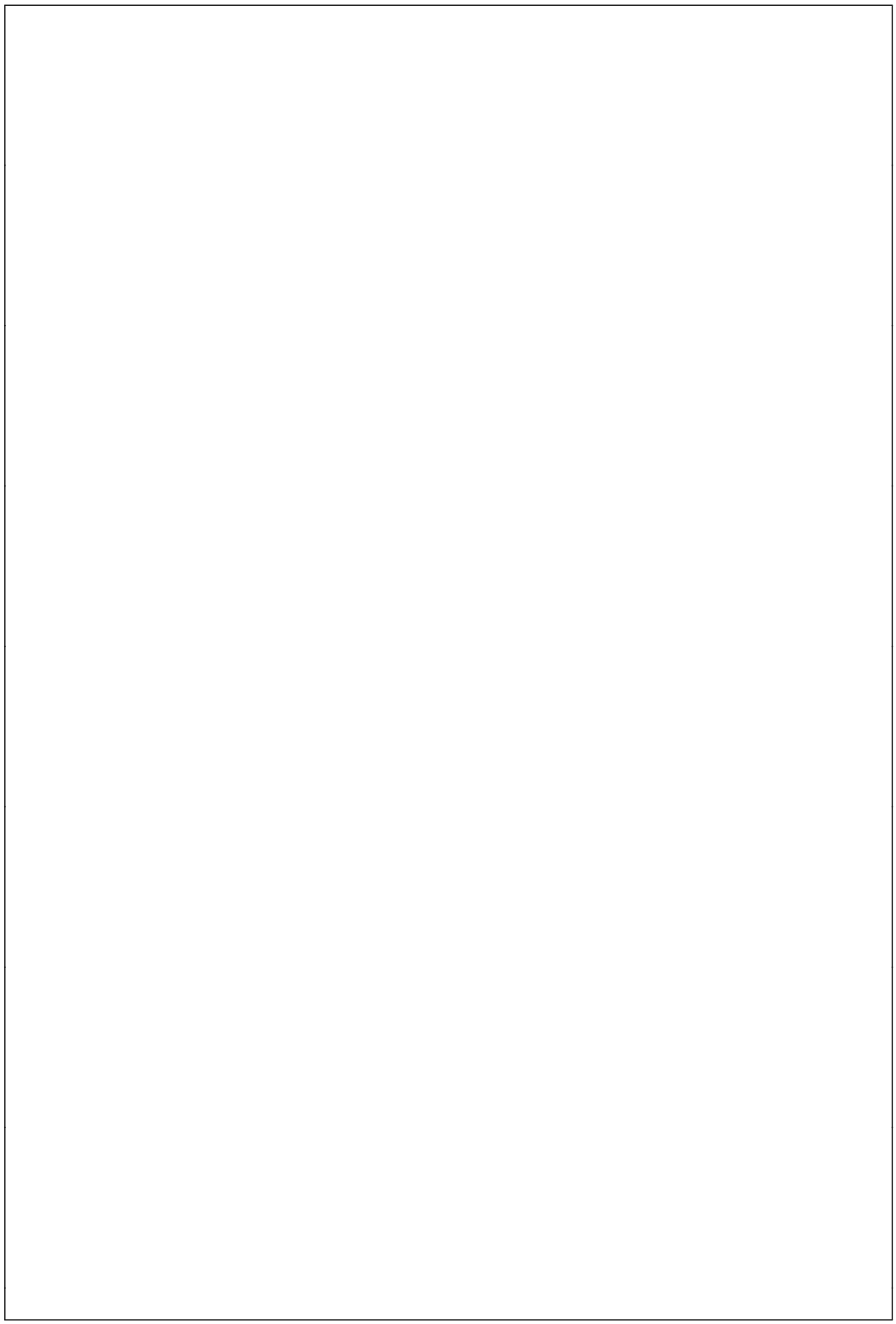
Quiescent Current



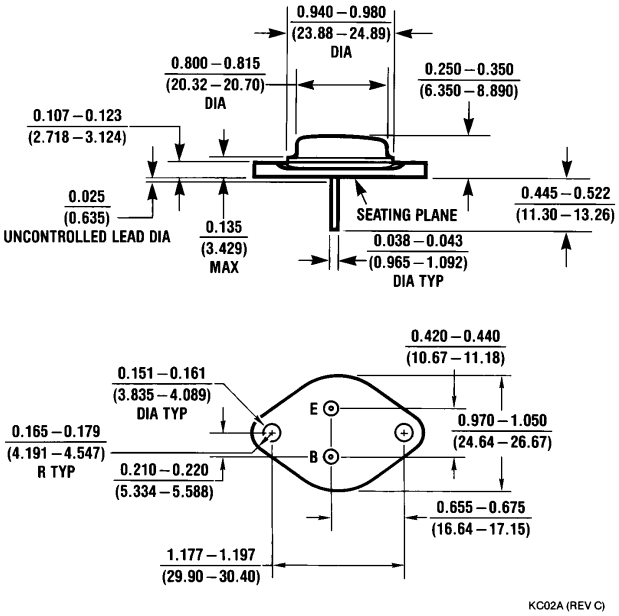
Quiescent Current



TL/H/7746–4

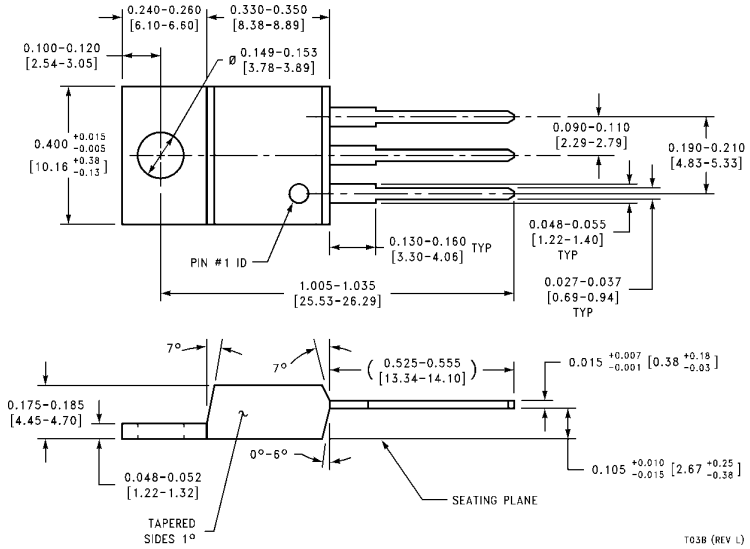


Physical Dimensions inches (millimeters)



Aluminum Metal Can Package (KC)
Order Number LM7805CK, LM7812CK or LM7815CK
NS Package Number KC02A

Physical Dimensions inches (millimeters) (Continued)



TO-220 Package (T)
Order Number LM7805CT, LM7812CT or LM7815CT
NS Package Number T03B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National Semiconductor

November 1994

LM79XX Series 3-Terminal Negative Regulators

General Description

The LM79XX series of 3-terminal regulators is available with fixed output voltages of −5V, −8V, −12V, and −15V. These devices need only one external component—a compensation capacitor at the output. The LM79XX series is packaged in the TO-220 power package and is capable of supplying 1.5A of output current.

These regulators employ internal current limiting safe area protection and thermal shutdown for protection against virtually all overload conditions.

Low ground pin current of the LM79XX series allows output voltage to be easily boosted above the preset value with a resistor divider. The low quiescent current drain of these devices with a specified maximum change with line and load ensures good regulation in the voltage boosted mode.

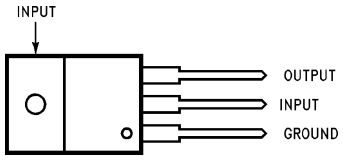
For applications requiring other voltages, see LM137 data sheet.

Features

- Thermal, short circuit and safe area protection
- High ripple rejection
- 1.5A output current
- 4% tolerance on preset output voltage

Connection Diagrams

TO-220 Package



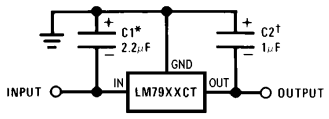
TL/H/7340–14

Front View

Order Number LM7905CT, LM7912CT or LM7915CT
See NS Package Number TO3B

Typical Applications

Fixed Regulator



TL/H/7340–3

*Required if regulator is separated from filter capacitor by more than 3″. For value given, capacitor must be solid tantalum. 25 μF aluminum electrolytic may be substituted.

†Required for stability. For value given, capacitor must be solid tantalum. 25 μF aluminum electrolytic may be substituted. Values given may be increased without limit.

For output capacitance in excess of 100 μF, a high current diode from input to output (1N4001, etc.) will protect the regulator from momentary input shorts.

RRD-B30M115/Printed in U. S. A.

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Input Voltage	
(V_o = −5V)	−25V
(V_o = −12V and −15V)	−35V
Input-Output Differential	
(V_o = −5V)	25V
(V_o = −12V and −15V)	30V
Power Dissipation (Note 2)	Internally Limited
Operating Junction Temperature Range	0°C to +125°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	230°C

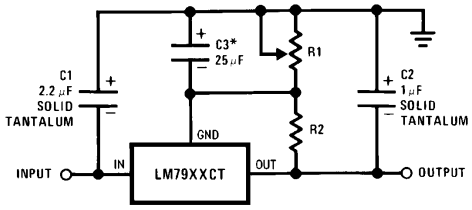
Electrical Characteristics

Conditions unless otherwise noted: I_{OUT} = 500 mA, C_{IN} = 2.2 μF, C_{OUT} = 1 μF, 0°C ≤ T_J ≤ +125°C, Power Dissipation ≤ 1.5W.

Part Number			LM7905C			Units
Output Voltage			−5V			
Input Voltage (unless otherwise specified)			−10V			
Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	
V_O	Output Voltage	T_J = 25°C	−4.8	−5.0	−5.2	V
		5 mA ≤ I_{OUT} ≤ 1A, P ≤ 15W	−4.75		−5.25	V
				(−20 ≤ V_{IN} ≤ −7)		V
ΔV_O	Line Regulation	T_J = 25°C, (Note 3)		8	50	mV
				(−25 ≤ V_{IN} ≤ −7)		V
				2	15	mV
				(−12 ≤ V_{IN} ≤ −8)		V
ΔV_O	Load Regulation	T_J = 25°C, (Note 3)				
		5 mA ≤ I_{OUT} ≤ 1.5A		15	100	mV
		250 mA ≤ I_{OUT} ≤ 750 mA		5	50	mV
I_Q	Quiescent Current	T_J = 25°C		1	2	mA
ΔI_Q	Quiescent Current Change	With Line			0.5	mA
				(−25 ≤ V_{IN} ≤ −7)		V
		With Load, 5 mA ≤ I_{OUT} ≤ 1A			0.5	mA
V_n	Output Noise Voltage	T_A = 25°C, 10 Hz ≤ f ≤ 100 Hz		125		μV
	Ripple Rejection	f = 120 Hz	54	66		dB
				(−18 ≤ V_{IN} ≤ −8)		V
	Dropout Voltage	T_J = 25°C, I_{OUT} = 1A		1.1		V
I_{OMAX}	Peak Output Current	T_J = 25°C		2.2		A
	Average Temperature Coefficient of Output Voltage	I_{OUT} = 5 mA, 0 C ≤ T_J ≤ 100°C		0.4		mV/°C

Typical Applications (Continued)

Variable Output



TL/H/7340–2

*Improves transient response and ripple rejection. Do not increase beyond 50 μF.

$$V_{OUT} = V_{SET}\left(\frac{R1 + R2}{R2}\right)$$

Select R2 as follows:

LM7905CT	300Ω
LM7912CT	750Ω
LM7915CT	1k

Electrical Characteristics (Continued) Conditions unless otherwise noted: I_{OUT} = 500 mA, C_{IN} = 2.2 µF, C_{OUT} = 1 µF, 0°C ≤ T_J ≤ +125°C, Power Dissipation = 1.5W.

Part Number			LM7912C			LM7915C			Units
Output Voltage			−12V			−15V			
Input Voltage (unless otherwise specified)			−19V			−23V			
Symbol	Parameter	Conditions	Min	Typ	Max	Min	Typ	Max	
V_O	Output Voltage	T_J = 25°C	−11.5	−12.0	−12.5	−14.4	−15.0	−15.6	V
		5 mA ≤ I_{OUT} ≤ 1A, P ≤ 15W	−11.4		−12.6	−14.25		−15.75	V
			(−27 ≤ V_{IN} ≤ −14.5)			(−30 ≤ V_{IN} ≤ −17.5)			V
ΔV_O	Line Regulation	T_J = 25°C, (Note 3)		5	80		5	100	mV
			(−30 ≤ V_{IN} ≤ −14.5)			(−30 ≤ V_{IN} ≤ −17.5)			V
				3	30		3	50	mV
			(−22 ≤ V_{IN} ≤ −16)			(−26 ≤ V_{IN} ≤ −20)			V
ΔV_O	Load Regulation	T_J = 25°C, (Note 3)							
		5 mA ≤ I_{OUT} ≤ 1.5A		15	200		15	200	mV
		250 mA ≤ I_{OUT} ≤ 750 mA		5	75		5	75	mV
I_Q	Quiescent Current	T_J = 25°C		1.5	3		1.5	3	mA
ΔI_Q	Quiescent Current Change	With Line			0.5			0.5	mA
			(−30 ≤ V_{IN} ≤ −14.5)			(−30 ≤ V_{IN} ≤ −17.5)			V
		With Load, 5 mA ≤ I_{OUT} ≤ 1A			0.5			0.5	mA
V_n	Output Noise Voltage	T_A = 25°C, 10 Hz ≤ f ≤ 100 Hz		300			375		µV
	Ripple Rejection	f = 120 Hz	54	70		54	70		dB
			(−25 ≤ V_{IN} ≤ −15)			(−30 ≤ V_{IN} ≤ −17.5)			V
	Dropout Voltage	T_J = 25°C, I_{OUT} = 1A		1.1			1.1		V
I_{OMAX}	Peak Output Current	T_J = 25°C		2.2			2.2		A
	Average Temperature Coefficient of Output Voltage	I_{OUT} = 5 mA, 0 C ≤ T_J ≤ 100°C		−0.8			−1.0		mV/°C

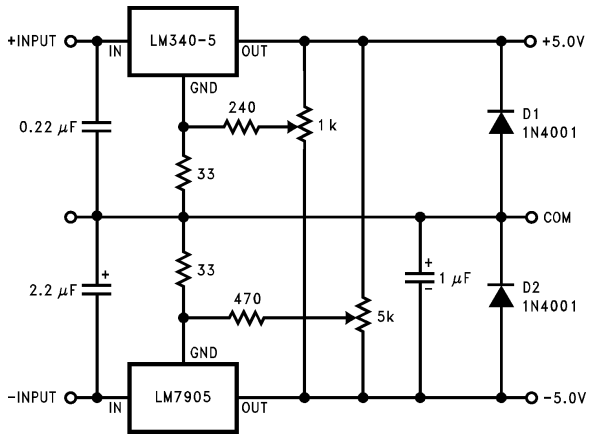
Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but do not guarantee Specific Performance limits. For guaranteed specifications and test conditions, see the Electrical Characteristics.

Note 2: Refer to Typical Performance Characteristics and Design Considerations for details.

Note 3: Regulation is measured at a constant junction temperature by pulse testing with a low duty cycle. Changes in output voltage due to heating effects must be taken into account.

Typical Applications (Continued)

Dual Trimmed Supply



TL/H/7340–4

Design Considerations

The LM79XX fixed voltage regulator series has thermal overload protection from excessive power dissipation, internal short circuit protection which limits the circuit's maximum current, and output transistor safe-area compensation for reducing the output current as the voltage across the pass transistor is increased.

Although the internal power dissipation is limited, the junction temperature must be kept below the maximum specified temperature (125°C) in order to meet data sheet specifications. To calculate the maximum junction temperature or heat sink required, the following thermal resistance values should be used:

Package	Typ θ_{JC} °C/W	Max θ_{JC} °C/W	Typ θ_{JA} °C/W	Max θ_{JA} °C/W
TO-220	3.0	5.0	60	40

$$P_{D\,MAX} = \frac{T_{J\,Max} - T_A}{\theta_{JC} + \theta_{CA}} \text{ or } \frac{T_{J\,Max}\,T_A}{\theta_{JA}}$$

$\theta_{CA} = \theta_{CS} + \theta_{SA}$ (without heat sink)

Solving for T_J:

$T_J = T_A + P_D(\theta_{JC} + \theta_{CA})$ or

$= T_A + P_D\theta_{JA}$ (without heat sink)

Where:

T_J = Junction Temperature

T_A = Ambient Temperature

P_D = Power Dissipation

θ_{JA} = Junction-to-Ambient Thermal Resistance

θ_{JC} = Junction-to-Case Thermal Resistance

θ_{CA} = Case-to-Ambient Thermal Resistance

θ_{CS} = Case-to-Heat Sink Thermal Resistance

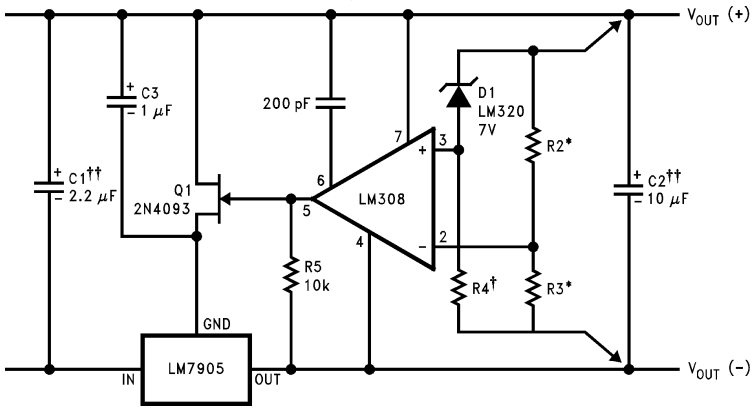
θ_{SA} = Heat Sink-to-Ambient Thermal Resistance

Typical Applications (Continued)

Bypass capacitors are necessary for stable operation of the LM79XX series of regulators over the input voltage and output current ranges. Output bypass capacitors will improve the transient response by the regulator.

The bypass capacitors, (2.2 μF on the input, 1.0 μF on the output) should be ceramic or solid tantalum which have good high frequency characteristics. If aluminum electrolytics are used, their values should be 10 μF or larger. The bypass capacitors should be mounted with the shortest leads, and if possible, directly across the regulator terminals.

High Stability 1 Amp Regulator



TL/H/7340–5

Load and line regulation < 0.01% temperature stability ≤ 0.2%

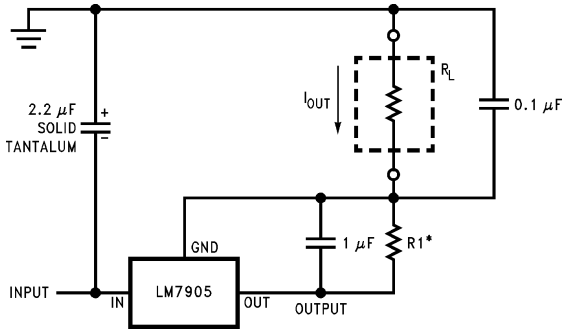
†Determine Zener current

††Solid tantalum

*Select resistors to set output voltage. 2 ppm/°C tracking suggested

Typical Applications (Continued)

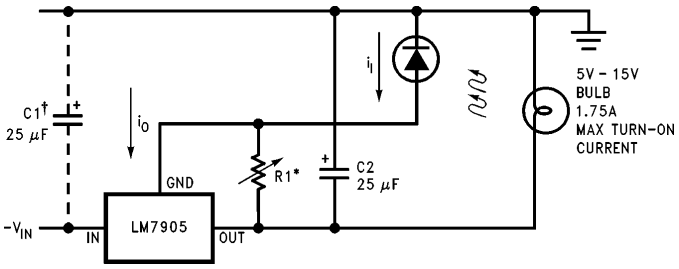
Current Source



*$I_{OUT} = 1\ mA + \frac{5V}{R1}$

TL/H/7340–7

Light Controller Using Silicon Photo Cell



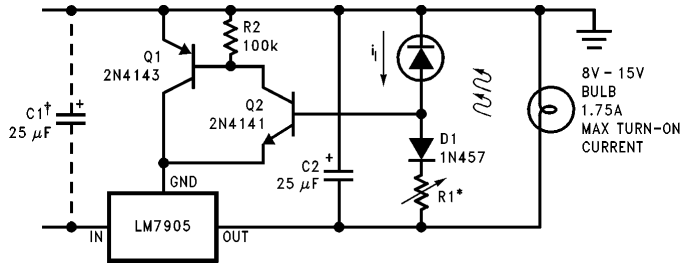
TL/H/7340–8

*Lamp brightness increase until $i_I = i_O$ ($\approx$ 1 mA) + 5V/R1.

†Necessary only if raw supply filter capacitor is more than 2″ from LM7905CT

Typical Applications (Continued)

High-Sensitivity Light Controller

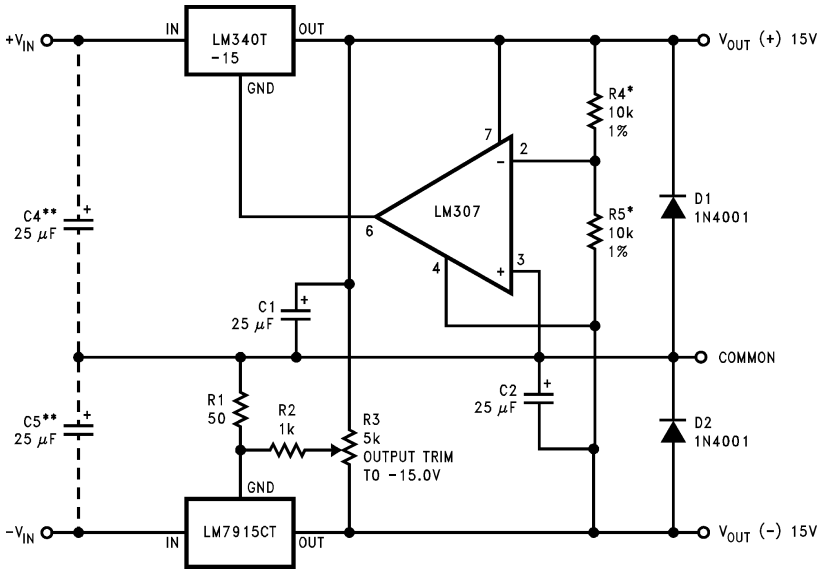


TL/H/7340–9

*Lamp brightness increases until i_I = 5V/R1 (I_I can be set as low as 1 μA)

†Necessary only if raw supply filter capacitor is more than 2″ from LM7905

±15V, 1 Amp Tracking Regulators



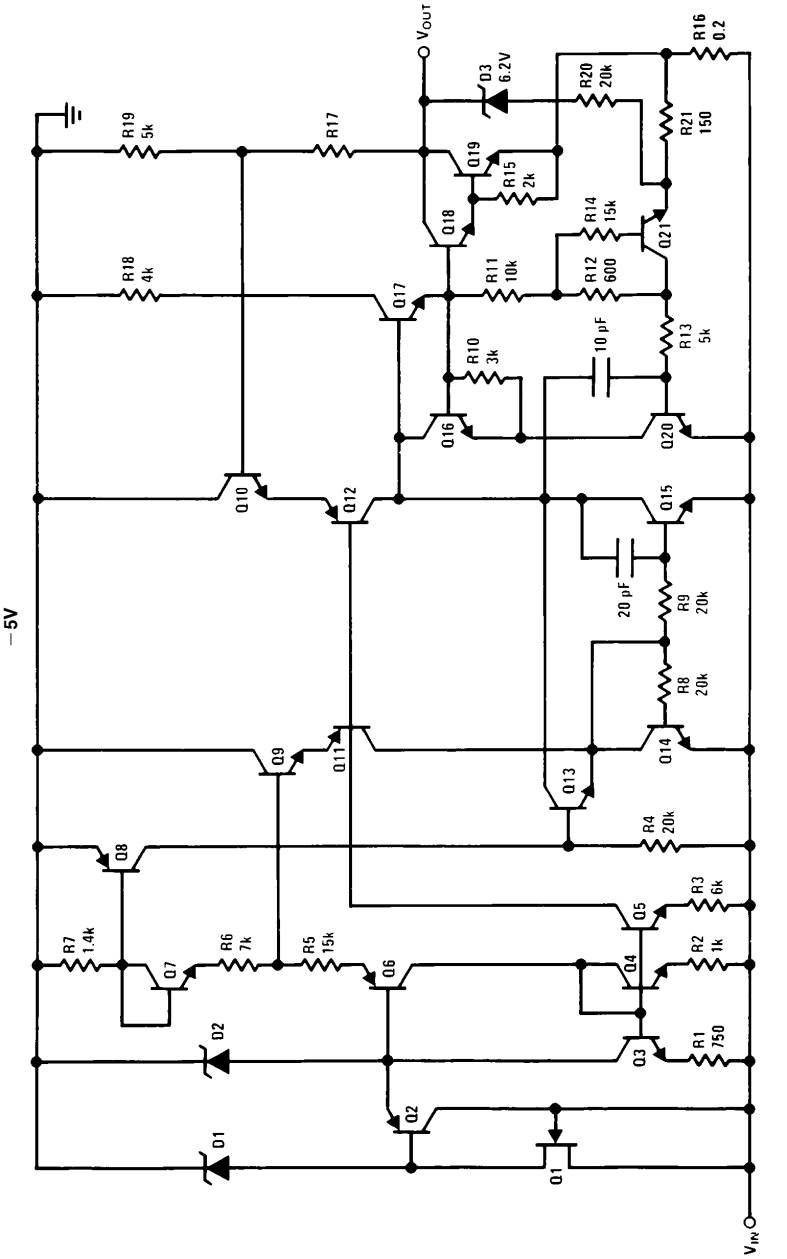
TL/H/7340–1

	(−15)	(+15)
Load Regulation at ΔI_L = 1A	40 mV	2 mV
Output Ripple, C_{IN} = 3000 μF, I_L = 1A	100 μVrms	100 μVrms
Temperature Stability	50 mV	50 mV
Output Noise 10 Hz ≤ f ≤ 10 kHz	150 μVrms	150 μVrms

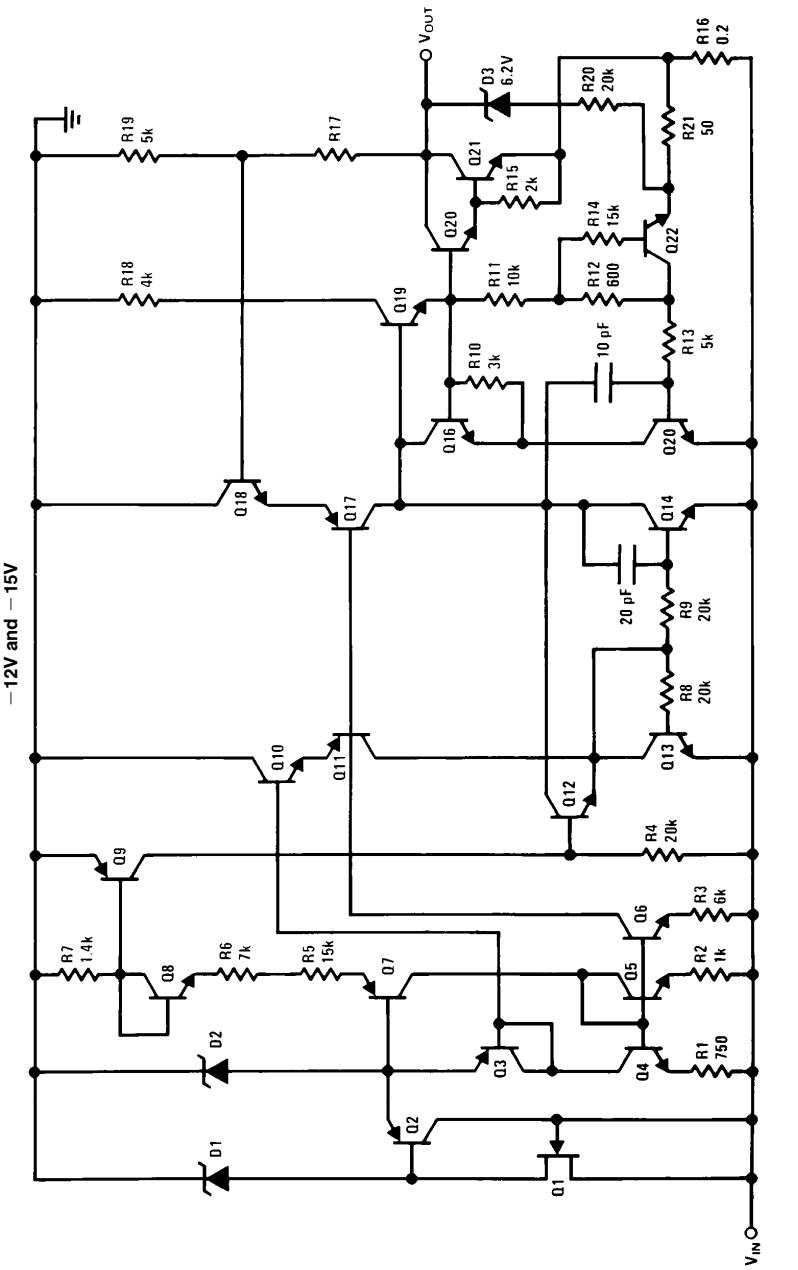
*Resistor tolerance of R4 and R5 determine matching of (+) and (−) outputs.

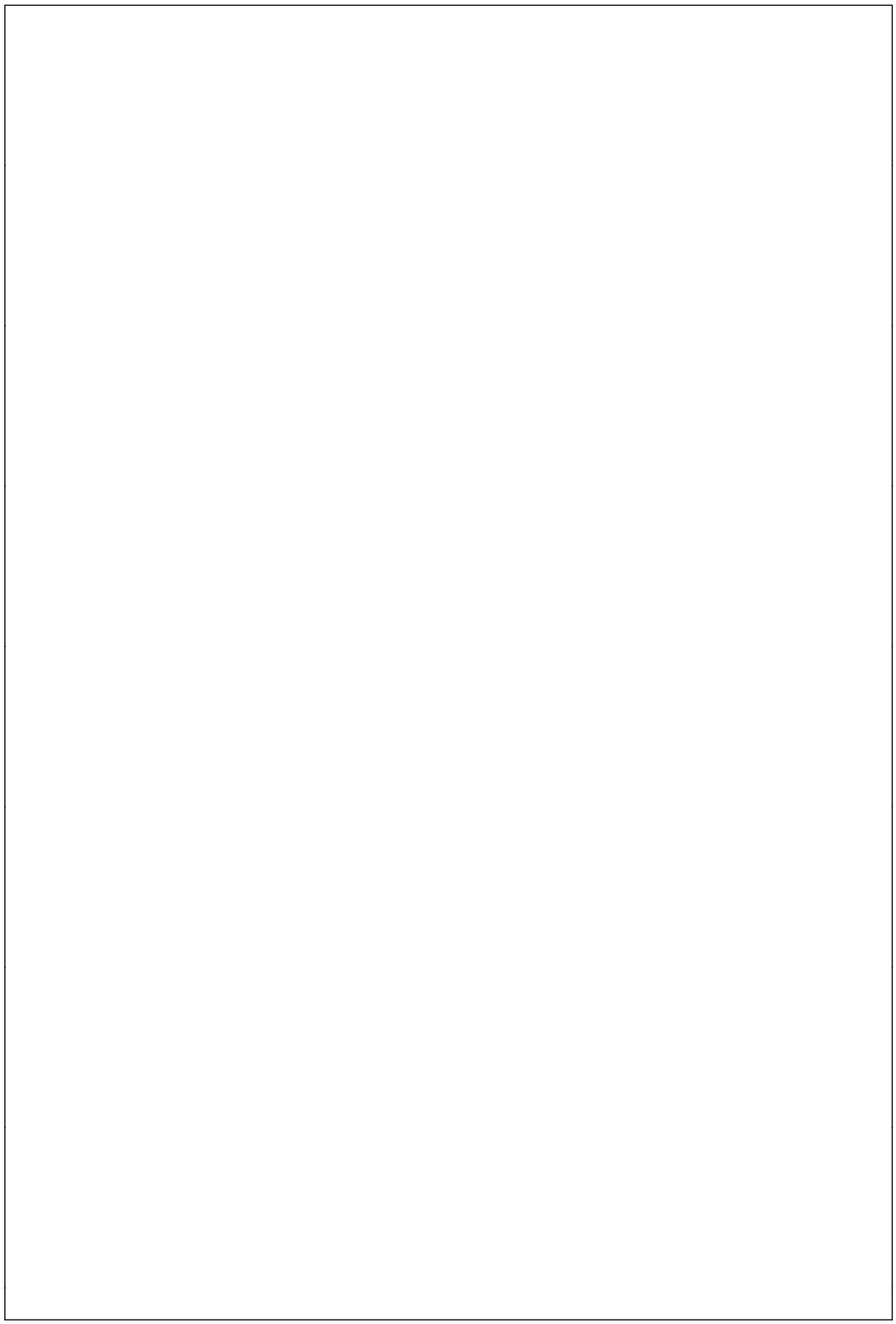
**Necessary only if raw supply filter capacitors are more than 3″ from regulators.

Schematic Diagrams

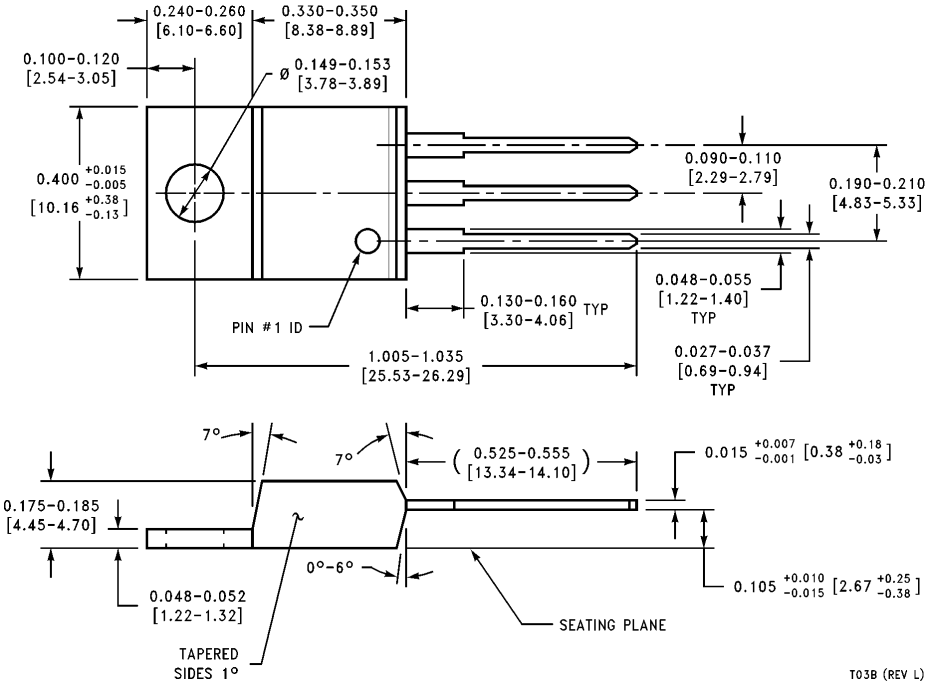


Schematic Diagrams (Continued)





Physical Dimensions inches (millimeters)



TO-220 Outline Package (T)
Order Number LM7905CT, LM7912CT or LM7915CT
NS Package Number T03B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408