

FERNANDA CECÍLIA CORREIA LIMA LOUREIRO

**TESTADOR SEMI-AUTOMÁTICO DE CIRCUITOS
INTEGRADOS DIGITAIS**

Dissertação apresentada ao Curso de **MES**
TRADO EM ENGENHARIA ELÉTRICA da Universi
dade Federal da Paraíba, em cumprimento
às exigências para obtenção do Grau de
Mestre

ÁREA DE CONCENTRAÇÃO: **Processamento da Informação**

WILLIAM LLOYD BRANDT
- Orientador -

CAMPINA GRANDE
SETEMBRO - 1985



L892t Loureiro, Fernanda Cecilia Correia Lima
Testador semi-automatico de circuitos integrados
digitais / Fernanda Cecilia Correia Lima Loureiro. -
Campina Grande, 1985.
86 f.

Dissertacao (Mestrado em Engenharia Eletrica) -
Universidade Federal da Paraiba, Centro de Ciencias e
Tecnologia.

1. Circuitos Integrados Digitais 2. Dissertacao I.
Brandt, William Lloyd, Dr. (orientador) II. Universidade
Federal da Paraiba - Campina Grande (PB) III. Titulo

CDU 621.3.049.77(043)

TESTADOR SEMI-AUTOMÁTICO DE CIRCUITOS
INTEGRADOS DIGITAIS

FERNANDA CECÍLIA CORREIA LIMA LOUREIRO

DISSERTAÇÃO APROVADA EM 20/09/85



WILLIAM LLOYD BRANDT

Orientador



WILSON GUERREIRO PINHEIRO

Componente da Banca



GURDIP SINGH DEEP

Componente da Banca



JOÃO MARQUES DE CARVALHO

Componente da Banca

CAMPINA GRANDE

SETEMBRO - 1985

DEDICATÓRIA

ao meu irmão **Eduardo** (in memoriam)

aos meus pais e tia **Titá**

aos meus filhos - **Clarissa Fernanda, Arthur César** e
Ricardo Augusto

AGRADECIMENTOS

Gostaria de agradecer a paciência e boa vontade dos que contribuíram para a realização deste trabalho, especialmente ao Prof. M.Sc. William Lloyd Brandt pela orientação, ao Prof. Ph.D. Gurdip Singh Deep pelo apoio, críticas e sugestões, ao Prof. Ph.D. Wilson Guerreiro Pinheiro pela revisão completa do texto e ao técnico Nildo Fernandes Torres pela montagem do sistema.

RESUMO

Este trabalho consiste no projeto e implementação prática de um testador semi-automático de circuitos integrados digitais que permite detetar falhas lógicas através da aplicação do teste funcional.

INDICE

	PÁGINA
CAPÍTULO 1 - INTRODUÇÃO	1
1.1 - Considerações Gerais.....	1
1.2 - Testadores de CI's.....	3
1.3 - Objetivo do Trabalho.....	4
CAPÍTULO 2 - O TESTE FUNCIONAL	6
2.1 - Descrição do Teste Funcional.....	6
2.2 - Implementação do Teste Funcional	8
CAPÍTULO 3 - IMPLEMENTAÇÃO DE UM TESTADOR SEMIAUTOMÁ TICO DE CI's DIGITAIS.....	10
3.1 - Descrição Geral do Sistema.....	10
3.2 - Descrição Detalhada dos Blocos do Sistema.....	12
3.2.1 - Matriz de Interconexão...	12
3.2.1.1 - Considerações Prelimina res.....	12

	PÁGINA
3.2.1.2 - Estrutura da Matriz de Interconexão.....	13
3.2.1.3 - Circuito Multiplexador.	14
a) Descrição.....	14
b) Decodificador.....	14
3.2.2 - Gerador de Códigos.....	31
3.2.3 - Teclado.....	42
3.2.3.1 - Introdução.....	42
3.2.3.2 - Circuito "Debounce"....	46
3.2.3.3 - Registradores.....	50
3.2.4 - Comparadores e Indicadores Luminosos.....	52
3.2.5 - Gerador de Sinais de Entrada e de Sincronismo...	56
3.3 - Funcionamento do Sistema.....	58
3.3.1 - Operação do Sistema.....	58
3.3.2 - Tempo de Resposta.....	62
3.3.3 - Confiabilidade.....	62
3.3.4 - Testes Efetuados.....	64
CAPÍTULO 4 - CONCLUSÕES.....	65

APÊNDICE A - CONFIGURAÇÃO DOS CI's	67
APÊNDICE B - CI's EXISTENTES NO MERCADO	75
APÊNDICE C - COMBINAÇÃO DE PINOS PARA ALIMENTAÇÃO.....	76
APÊNDICE D - PROGRAMAÇÃO DA TABELA DE CÓDIGOS E TABELA DE ENDEREÇOS.....	80
REFERÊNCIAS BIBLIOGRÁFICAS	87
BIBLIOGRAFIA CONSULTADA.....	88
ABSTRACT.....	89

LISTA DE ILUSTRAÇÕES

	PÁGINA
FIGURA 1.1 - Símbolo da porta AND de duas entradas...	7
FIGURA 3.1 - Diagrama de blocos Simplificado do Sistema.....	11
FIGURA 3.2 - Diagrama de blocos Geral da Matriz de Interconexão.....	14
FIGURA 3.3 - Diagrama de blocos do Circuito Multiplexador.....	16
FIGURA 3.4 - Diagrama de blocos do Circuito Multiplexador com ligações para o CIR e o CIT...	18
FIGURA 3.5 - a) Diagrama de blocos do circuito acoplado de V_{CC}	19
b) Circuito Acoplador de V_{CC} para um p_i no.....	19

FIGURA 3.6 - Diagrama Elétrico do Circuito Acoplador de V_{CC} com saída para um par de pinos..	21
FIGURA 3.7 - Diagrama Elétrico do Circuito Acoplador de V_{CC} com saída para todos os pinos...	22
FIGURA 3.8 - a) Diagrama de blocos do Circuito de Acoplamento de Ponto Comum.....	23
b) Circuito de Acoplamento de Ponto Comum.....	23
FIGURA 3.9 - Diagrama Elétrico Completo do Circuito Acoplador de Ponto Comum com saída para todos os pinos que podem ser Terminais Ponto Comum.....	24
FIGURA 3.10- a) Diagrama de blocos do Decodificador.	28
b) Implementação do Circuito Decodificador com Portas Discretas.....	28
FIGURA 3.11- Diagrama Elétrico da Matriz de Intercnexão.....	30
FIGURA 3.12- Diagrama Elétrico de parte do Circuito Decodificador que gera as linhas de controle para acoplar V_{CC}	32

FIGURA 3.13- Diagrama Elétrico de parte do Circuito Decodificador que gera as linhas de controle para o Circuito Acoplador de Ponto Comum.....	33
FIGURA 3.14- Soquetes ZIF.....	34
FIGURA 3.15- Diagrama de blocos do Circuito Gerador de Códigos.....	36
FIGURA 3.16- Diagrama do Gerador de Códigos.....	38
FIGURA 3.17- Diagrama Elétrico do Gerador de Códigos	40
FIGURA 3.18- Diagrama de blocos do Circuito do Teclado.....	44
FIGURA 3.19- Teclado.....	44
FIGURA 3.20- Diagrama de blocos do Circuito do Teclado Detalhado.....	45
FIGURA 3.21- Tensão na saída da chave devido ao efeito "bounce".....	47
FIGURA 3.22- Diagrama Lógico do CI 74123.....	49

FIGURA 3.23- Diagrama de Tempo do Circuito "Debounce"	49
FIGURA 3.24- Bloco Registradores da figura 3.20.....	51
FIGURA 3.25- Diagrama Elétrico do Circuito do Teclado	53
FIGURA 3.26- Diagrama Lógico do Circuito de Compara ção para um par de pinos.....	55
FIGURA 3.27- Diagrama de blocos do Circuito de Compa ração com Driver.....	55
FIGURA 3.28- Diagrama Elétrico do Circuito Comparador	57
FIGURA 3.29- Diagrama de blocos do Gerador de Entra das.....	59
FIGURA 3.30- Diagrama Elétrico do Gerador de Entradas	60

LISTA DE TABELAS

	PÁGINA
TABELA 1.1 - Tabela-verdade da porta AND.....	7
TABELA 3.1 - Tabela-verdade do Decodificador.....	26
TABELA 3.2 - Tabela-verdade do CI 74123.....	47
TABELA 3.3 - Taxa de Falhas dos Componentes do Siste <u>ma</u>	63

1 INTRODUÇÃO

1.1 - Considerações Gerais

É inegável a crescente importância dos circuitos digitais nos sistemas eletrônicos modernos. Sua faixa de aplicação abrange desde sistemas complexos como computadores e sistemas de telemetria por satélites até produtos eletrônicos mais simples como brinquedos, relógios, calculadoras etc.

Os sistemas lógicos convencionais baseados em relés estão sendo substituídos rapidamente por sistemas que utilizam circuitos digitais - os chamados sistemas lógicos digitais - , devido à maior confiabilidade e facilidade de manutenção que estes oferecem.

Com o desenvolvimento de circuitos integrados digitais mais aperfeiçoados, como os microprocessadores, os sistemas lógicos digitais adquiriram flexibilidade de programação, incluindo a capacidade de programação dinâmica.

Apesar de os circuitos integrados digitais possuírem alta confiabilidade, eles podem, no entanto, apresentar falhas durante sua operação no sistema em que estiverem inseridos. Estas falhas podem ter origem durante o processo de fabricação do CI ou durante operação enquanto parte de um sistema [1] .

É de suma importância prática a identificação de CI's defeituosos antes de serem empregados em qualquer sistema, já que implica na redução da probabilidade de falha do sistema como um todo. Além disso, a identificação prévia de um CI defeituoso é menos onerosa que a sua eliminação quando já faz parte de um sistema.

A identificação de CI's defeituosos pode ser feita através de testes que permitem verificar níveis lógicos de saída (teste funcional) potência dissipada e níveis mínimos e máximos de tensões e correntes de entrada e de saída (teste de parâmetros DC) ou respostas transitórias e tempo de propagação (teste de parâmetros AC).

O teste funcional é suficiente para a identificação de CI's que não executam a função lógica especificada pelo fabricante. O teste de parâmetros DC é normalmente feito como um complemento do teste funcional quando se deseja verificar os níveis DC adequados de tensões e correntes de entrada e de saída.

O teste de parâmetros AC é usualmente feito pelos fabricantes de CI's antes das etapas de metalização e de encapsulamento [2] ou por usuários que tenham aplicações extremamente críticas principalmente em termos de velocidade (tempo de resposta).

1.2 - Testadores de CI's

Os equipamentos de teste de parâmetros AC são frequentemente complexos e de elevado custo. Por outro lado, os testadores que realizam os testes funcional e de parâmetros DC são geralmente simples, de pequeno porte e de custo relativamente baixo. Os testadores de pequeno porte existentes na literatura são basicamente de dois tipos: 1) com lógica fixa e 2) com lógica programável [2].

Os testadores com lógica fixa foram os primeiros desenvolvidos [2]. São implementados em hardware e consequentemente, não podem ser reprogramados, a não ser através de modificações no circuito. São adequados a testes simples como o teste funcional. Os sistemas implementados com lógica fixa diferem basicamente entre si pela maneira como fornecem aos diferentes CI's sob teste a sequência de níveis lógicos de entrada e a tensão de alimentação adequadas. Essa sequência de níveis lógicos, bem como a tensão de alimentação, são conectados ao CI sob teste por um circuito comumente chamado de Matriz de Intercon

xão.

Nos primeiros sistemas a matriz de interconexão era implementada com chaves no console do testador |2|. Posteriormente, foram desenvolvidos equipamentos onde a Matriz de Interconexão foi implementada através de cartões programados conforme o tipo de CI.

Os testadores com lógica programável utilizam microprocessadores para controlar os testes e a geração de parâmetros DC de referência. A total flexibilidade de programação dos testes e a capacidade de programação dinâmica dos parâmetros DC de referência tornam este tipo de equipamento adequado a análises mais precisas e variadas, como é o caso, por exemplo, de testes nos estágios finais de controle de qualidade de componentes em fábricas de CI's.

1.3 - Objetivo do Trabalho

O objetivo deste trabalho é o projeto e implementação prática de um testador semi-automático de CI's digitais que permite detetar falhas lógicas através da aplicação do teste funcional.

Este testador, cuja finalidade principal é a utilização em laboratórios de ensino e de pesquisa, deve apresentar as seguintes características gerais:

- pequeno porte
- fácil manuseio
- rapidez na realização dos testes
- baixo custo

Em princípio, o testador deve ser capaz de testar a maioria dos CI's de 14 e 16 pinos das famílias TTL e CMOS.

2 O TESTE FUNCIONAL

2.1 - Descrição do Teste Funcional

O teste funcional consiste em verificar o funcionamento do CI através das especificações fornecidas pelo fabricante [3]. Esta verificação é realizada aplicando-se combinações de níveis lógicos às entradas do CI e comparando os níveis lógicos de saída do CI com os especificados pelo fabricante. Como ilustração considere-se a porta E de duas entradas mostrada na Figura 1.1. A tabela-verdade correspondente é apresentada na Tabela 1.1 mostrando os níveis lógicos da saída para cada combinação possível dos níveis lógicos (BAIXO e ALTO) das entradas. A aplicação do teste funcional a essa porta consiste em fornecer as quatro combinações possíveis dos níveis lógicos BAIXO (b) e ALTO (a) às entradas A e B e, em seguida, verificar se os níveis lógicos da saída C estão de acordo com os especificados na tabela-verdade.

O teste funcional pode ser aplicado a quaisquer circuitos combinacionais ou sequenciais [4]. De fato, a o

corrência de uma falha em um circuito combinacional irreduzível implica em alteração na função do circuito. Consequentemente, para pelo menos uma das combinações possíveis dos níveis lógicos de entrada o circuito defeituoso apresenta um ou mais níveis lógicos de saída diferente dos especificados.

Enquanto o teste para um circuito combinacional pode ser uma combinação de entradas, um teste para um circuito sequencial, em geral, requer uma sequência de entradas, visto que, os níveis lógicos de saída de um circuito sequencial em um determinado instante dependem não só dos níveis lógicos de entrada naquele instante, mas também do estado interno do circuito e, portanto, dos níveis lógicos anteriores das entradas [4].

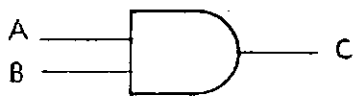


Fig. 1.1 - Símbolo da porta AND de duas entradas

ENTRADAS		SAÍDAS
A	B	C
b	b	b
b	a	b
a	b	b
a	a	a

Tabela 1.1 - Tabela verdade da porta AND

2.2 - Implementação do Teste Funcional

Este sistema foi desenvolvido com lógica fixa que se adequa perfeitamente bem ao objetivo de realizar o teste funcional. Na implementação deste foi utilizada uma técnica de teste característica dos sistemas de hardware. De fato, a verificação dos níveis lógicos de saída do CI sob teste é implementada como segue:

Toma-se um CI do mesmo tipo que o CI sob teste (CIT) que esteja funcionando bem para servir de referência. A ele são aplicados os mesmos sinais de entrada e a mesma tensão de alimentação que ao CIR. Como ambos os CI's foram projetados para executar a mesma função lógica, seja ela qual for, os níveis lógicos de saída do CI tomado como referência (CIR) e do CI sob teste devem ser iguais. Faz-se uma comparação entre o nível lógico de cada uma saída do CIT com o nível lógico da saída correspondente do CIR. Caso seja detetada alguma diferença, esta constituirá o indicativo de falha no CIT, já que os níveis lógicos de saída do CIR são considerados corretos.

Para que se consiga detetar o maior número possível de falhas, ou seja, para o teste ser mais completo, é necessário que :

- a) Todas as entradas do CIT e do CIR sejam ativadas simultaneamente.

b) Que sejam fornecidas ao CIT e ao CIR todas as com
binações possíveis dos níveis lógicos de entrada.

3 IMPLEMENTAÇÃO DE UM TESTADOR SEMI AUTOMÁTICO DE CI'S DIGITAIS

3.1 - Descrição Geral do Sistema

O testador semiautomático de CI's digitais implementado é mostrado no diagrama de blocos simplificado da Figura 3.1 constando de :

- 1) Um gerador de níveis lógicos de entrada.
- 2) Um gerador de códigos.
- 3) Um teclado acoplado ao gerador de códigos.
- 4) Uma Matriz de Interconexão que efetua as conexões entre o gerador de níveis lógicos de entrada e os CI's de teste e de referência e fornece a tensão de alimentação adequada a esses CI's. A efetivação dessas conexões pela Matriz de Interconexão é realizada sob o controle das informações codificadas sobre o CIT, geradas a partir do teclado e o gerador de códigos.
- 5) Comparadores com memória, cuja função é comparar os níveis lógicos de saída do CIT com os do CIR

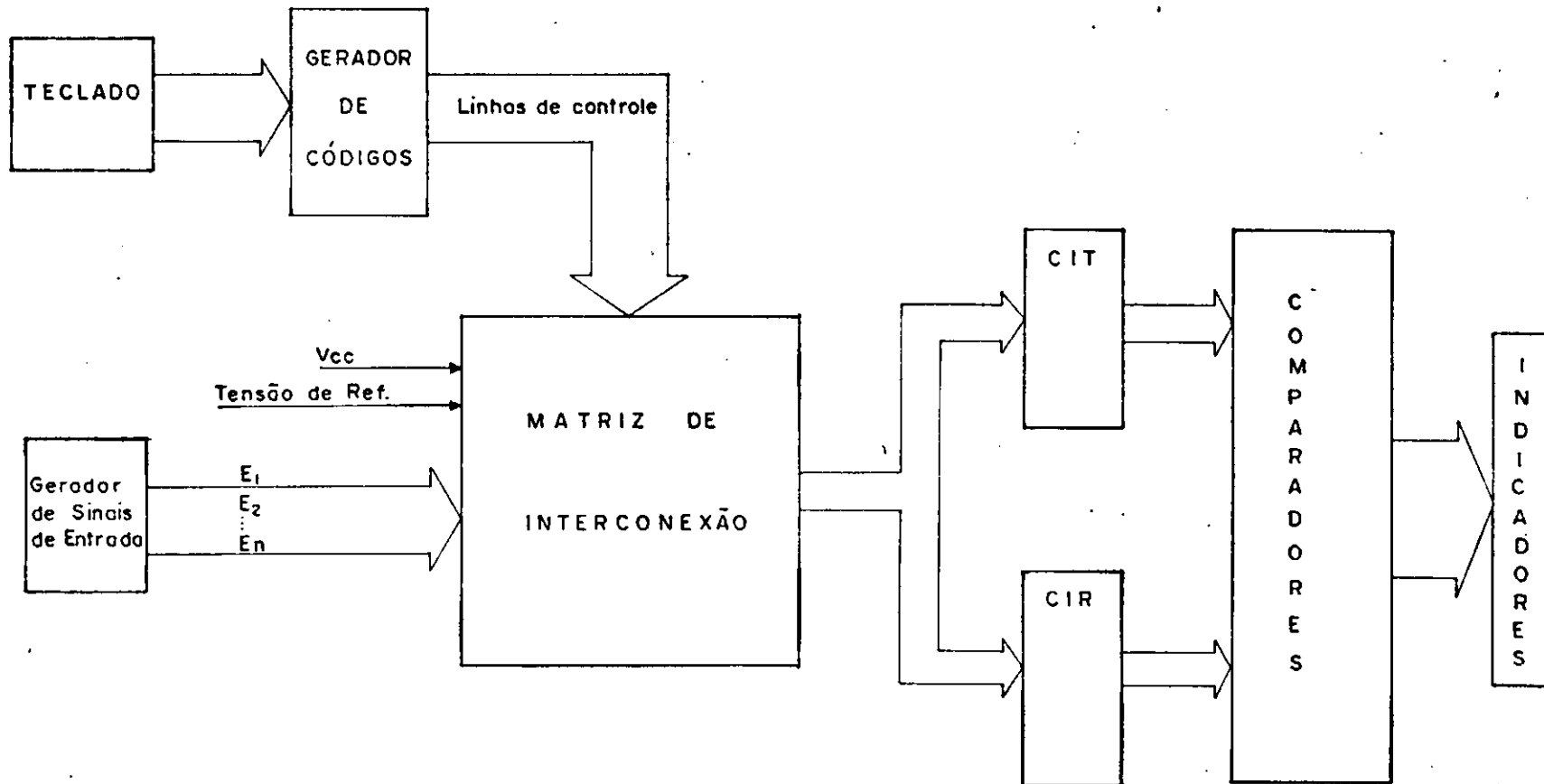


Fig. 3.1 Diagrama de blocos simplificado do sistema
 E_1, E_2, \dots, E_n — níveis lógicos de entrada
 Tensão de Referência — 0 V

e armazenar os resultados das comparações.

- 6) Indicadores luminosos (LED's) que mostram os resultados das comparações.

3.2 - Descrição Detalhada dos Blocos do Sistema

3.2.1 - Matriz de Interconexão

3.2.1.1 - Considerações Preliminares

A Matriz de Interconexão deve exercer sua função levando em conta as diferentes características dos CI's que poderão ser testados. As características que tornam os CI's diferentes entre si são a função executada, o número de entradas, o número de saídas, o tipo de alimentação, a configuração dos pinos (isto é, a definição da finalidade de cada pino) etc. Constata-se, na prática, (V. apêndice A), que existe uma grande variedade de configurações de pinos para CI's digitais de modo que um pino de mesma posição mas de CI's diferentes pode ter finalidades diferentes. Portanto, a Matriz de Interconexão deve ter a flexibilidade para ligar a cada pino do CIT e do CIR a tensão de alimentação ou qualquer uma das entradas.

Por outro lado, as entradas do CIT devem ser ativadas simultaneamente. Para se efetuar o teste, portanto, as conexões aos pinos do CIT e do CIR devem ser implementadas

em paralelo.

3.2.1.2 - Estrutura da Matriz de Interconexão

O diagrama de blocos geral da Matriz de Interconexão é mostrado na Figura 3.2, consistindo de circuitos seletores (multiplexadores) em paralelo. Deve-se observar que existe um circuito multiplexador para cada pino do CIT. Os sinais de entrada e a tensão de alimentação são aplicados simultaneamente a todos os circuitos multiplexadores. Cada circuito multiplexador deve conectar ao pino, que lhe é correspondente, uma das fontes de sinais de entrada ou um dos terminais da fonte de alimentação (V_{CC} ou Ponto Comum), de acordo com a informação, sobre a finalidade do pino, recebida através de suas linhas de seleção.

3.2.1.3 - Circuito Multiplexador

a) Descrição

O diagrama de blocos do circuito multiplexador é apresentado na Figura 3.3. A multiplexação é efetuada, neste caso, em duas etapas: primeiramente, são multiplexados os sinais de entrada para o CIT ($\epsilon_1, \epsilon_2 \dots \epsilon_n$) por um circuito chamado aqui de multiplexador de sinais de entrada; em seguida, é realizada a multiplexação entre a saída do multiplexador de sinais de entrada com o sinal DC do terminal positivo da fonte de alimentação (V_{CC}) e a tensão

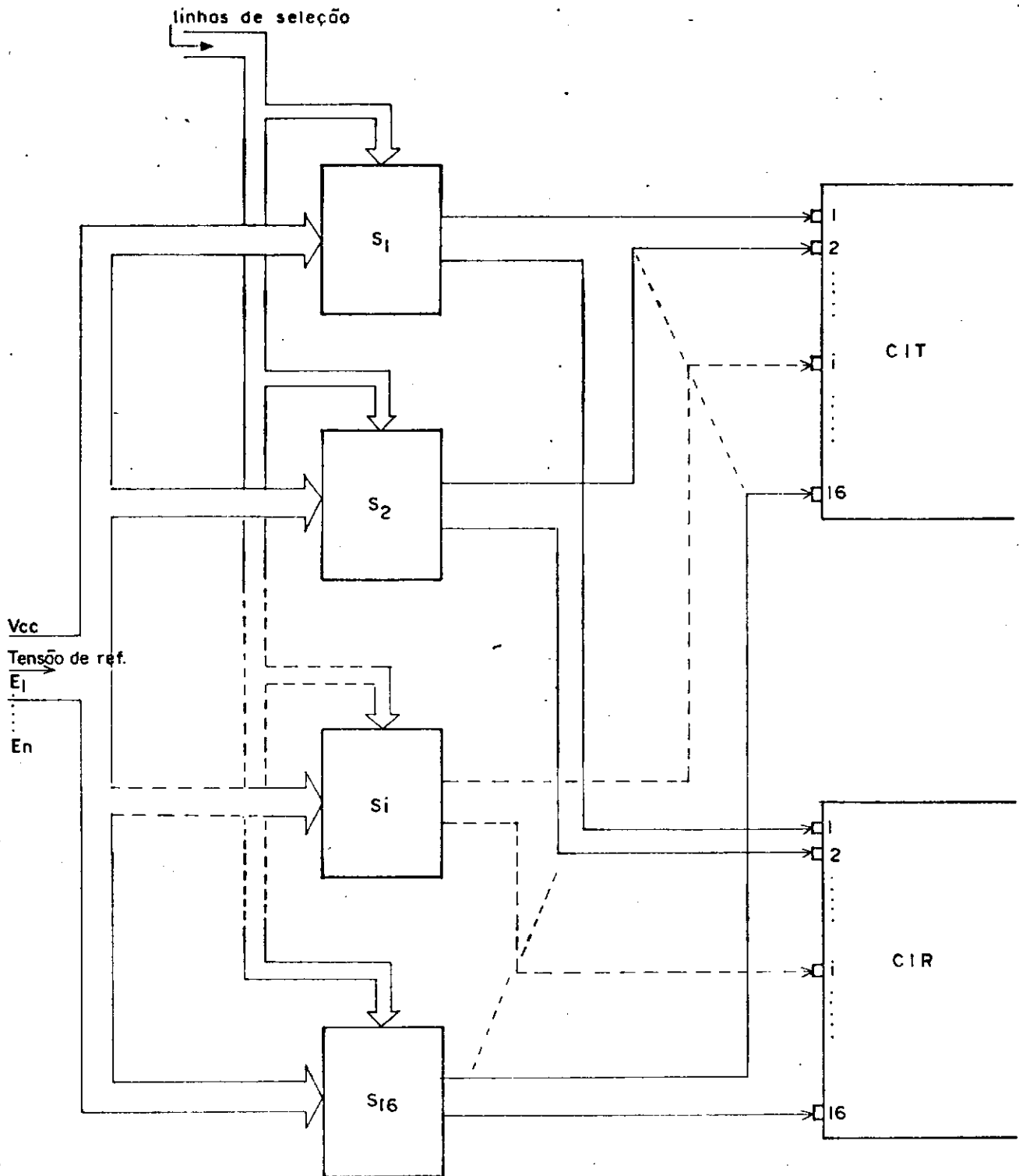


Fig. 3.2 Diagrama de blocos Geral da Matriz de Interconexão

V_{cc} — Tensão do terminal positivo da fonte

E_1, E_2, \dots, E_n — Sinais de entrada para o CIT e CIR

Tensão de ref. — zero volts

de referência (0 volt). Essa multiplexação é feita em duas etapas, porque é necessário utilizar circuitos multiplexadores de maior potência (analógicos) para acoplar a tensão de alimentação ao CIT e ao CIR, de modo a fornecer os níveis de tensão e de corrente adequados à alimentação dos mesmos. Os CI's digitais não podem fornecer corrente suficiente para isto. Por outro lado, é bastante um multiplexador digital para efetuar a multiplexação dos sinais de entrada.

Pelo estudo do diagrama de pinos dos vários tipos de CI's que serão testados (Apêndice A), constata-se que a grande maioria tem, no máximo, doze entradas. Para implementar o multiplexador de sinais de entrada, então, é suficiente utilizar um multiplexador digital 16 x 1 (CI 74150). A multiplexação da tensão V_{CC} , tensão de referência (0V), e a saída do multiplexador de sinais de entrada foi implementada através de chaves eletrônicas discretas controladas por linhas de controle (estas linhas de controle são geradas decodificando a informação sobre a finalidade do pino, recebidas através das linhas de seleção do multiplexador) (v. Figura 3.3). Como só uma das chaves é acionada de cada vez, não há problema em se ligar as saídas de todas as chaves ao mesmo pino do CIT.

As conexões descritas até aqui se referem apenas ao CIT. Essas mesmas conexões devem ser realizadas com o CIR. A finalidade de cada pino do CIT é a mesma do pino correspondente do CIR. Quando um pino do CIT, por exemplo, for

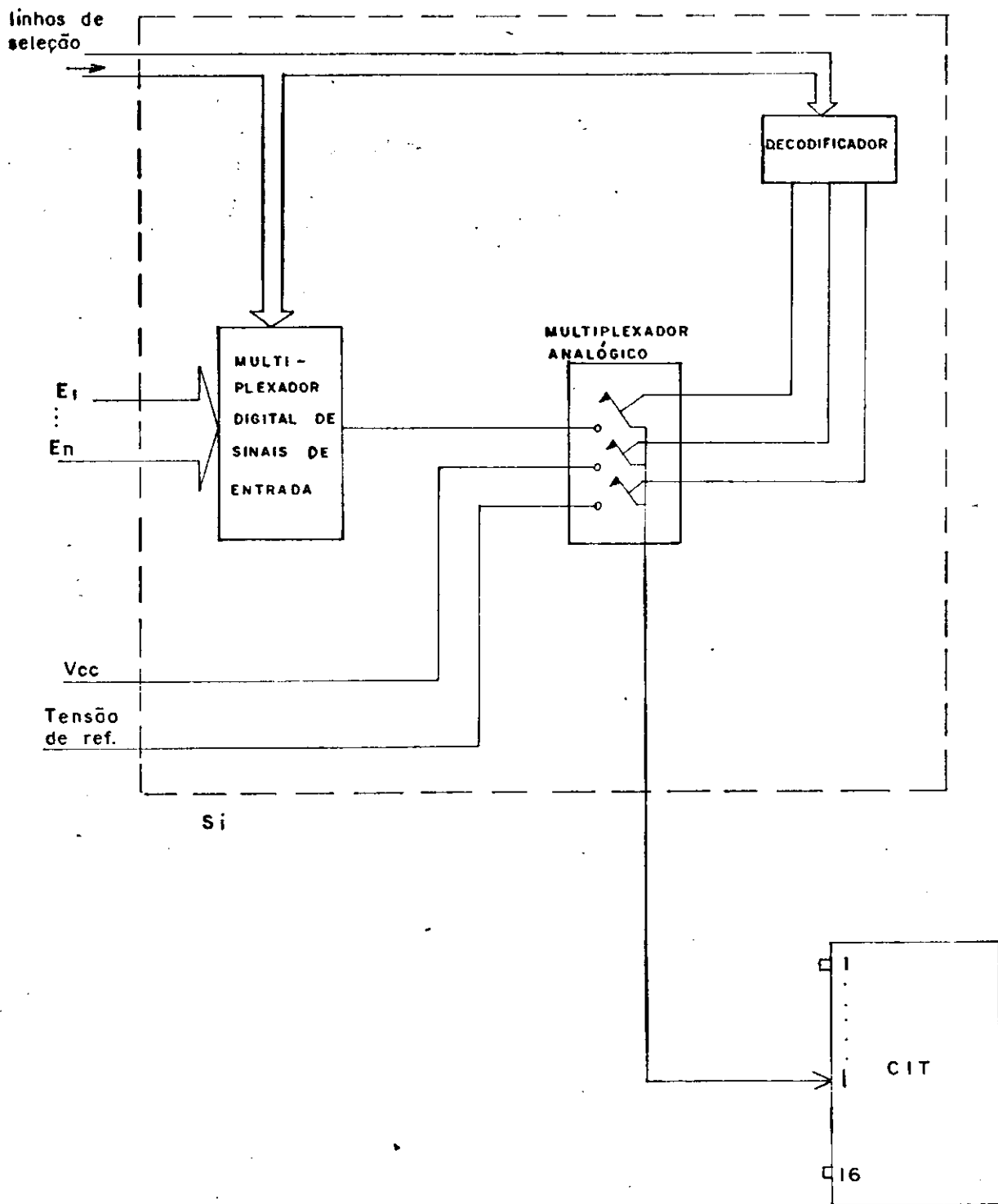


Fig. 3.3 Diagrama de blocos do Circuito Multiplexador

de entrada ou de conexão para tensão de alimentação, este deve estar interligado necessariamente com o pino correspondente do CIR. Quando o pino for de saída, no entanto, não deve de modo algum estar ligado ao correspondente do CIR, pois são os níveis de sinal destes pinos que precisam ser comparados. É necessária, portanto, uma chave entre cada pino do CIT e o seu correspondente do CIR, de modo a se ter a ligação ou o isolamento entre eles conforme seja necessário.

Viu-se, no início desta seção, que não é conveniente utilizar o mesmo tipo de chave para conectar a tensão de alimentação e sinal de entrada. Optou-se, então, por utilizar um conjunto de chaves eletrônicas discretas, idêntico ao utilizado para efetuar as ligações com o CIT, e em paralelo com este (V. Figura 3.4).

As chaves utilizadas para conectar a saída do multiplexador de sinais de entrada aos pinos do CIT e do CIR são chaves tri-estado (CI 74126). As chaves utilizadas para conectar a tensão V_{CC} da fonte de alimentação ao pino do CIT e do CIR, conjuntamente com o circuito da fonte de alimentação, constituem o **circuito acoplador de V_{CC}** (V. Figuras 3.5.a e 3.5.b).

A fonte de alimentação compreende uma fonte de tensão regulada comercial de +5V, um circuito limitador de corrente e um "driver". Este circuito foi implementado

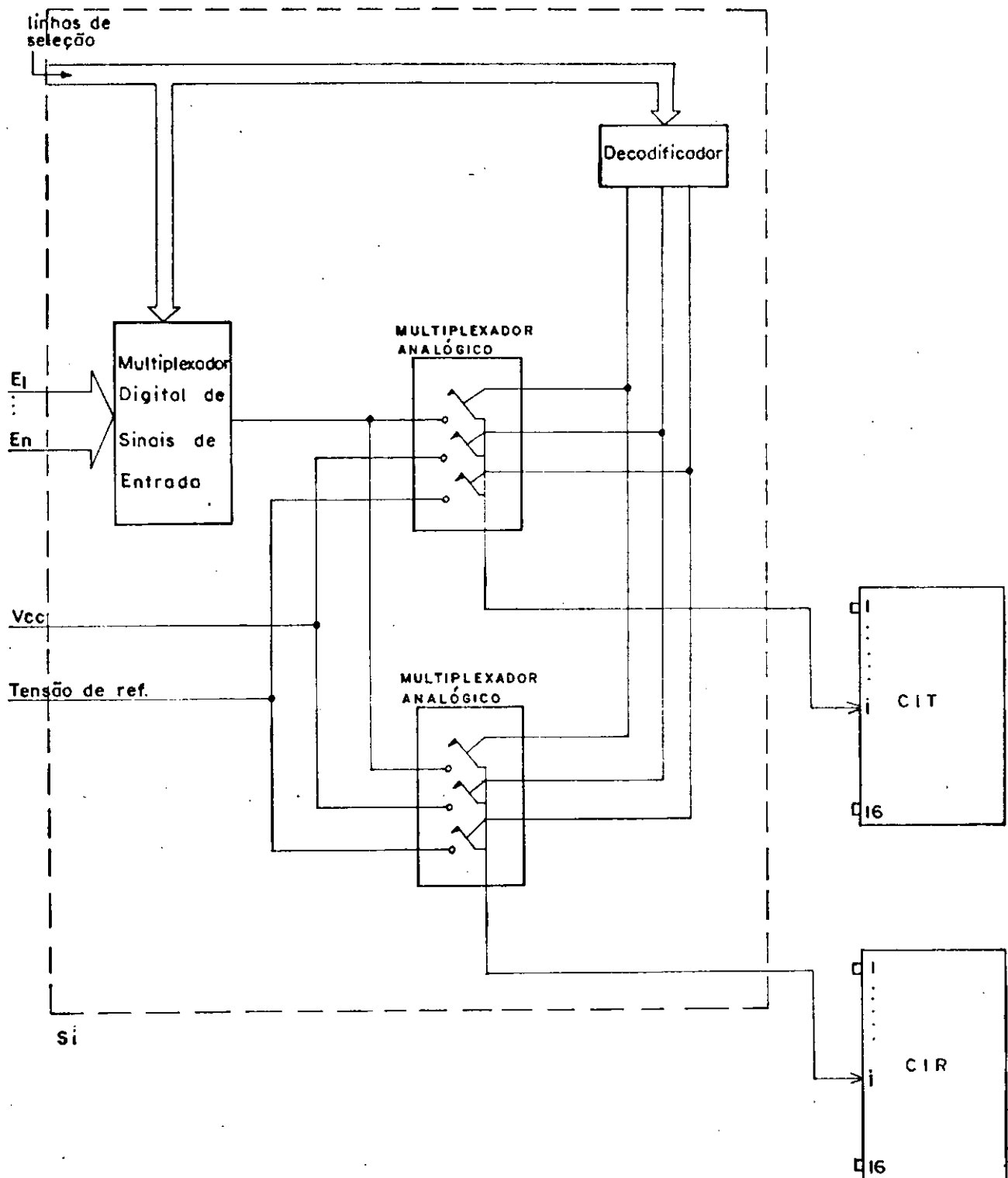


Fig. 3.4 Diagrama de blocos do circuito multiplexador com ligações para o CIT e o CIR

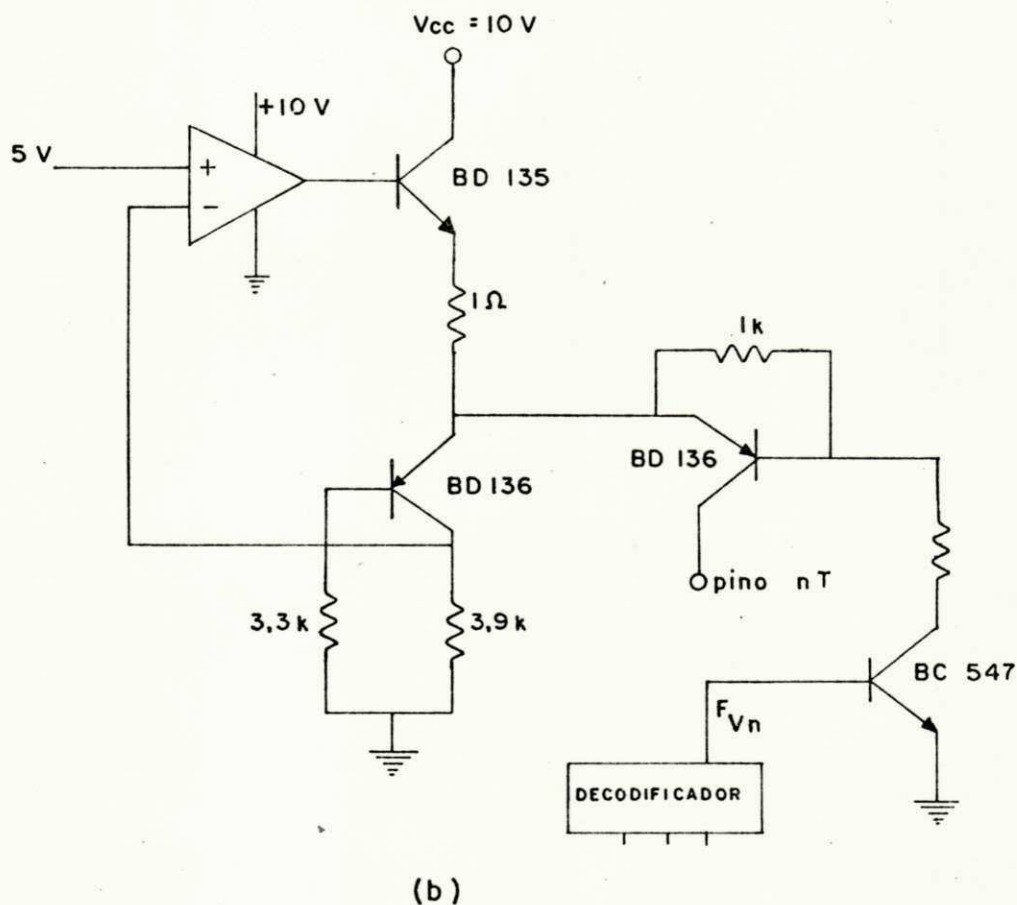
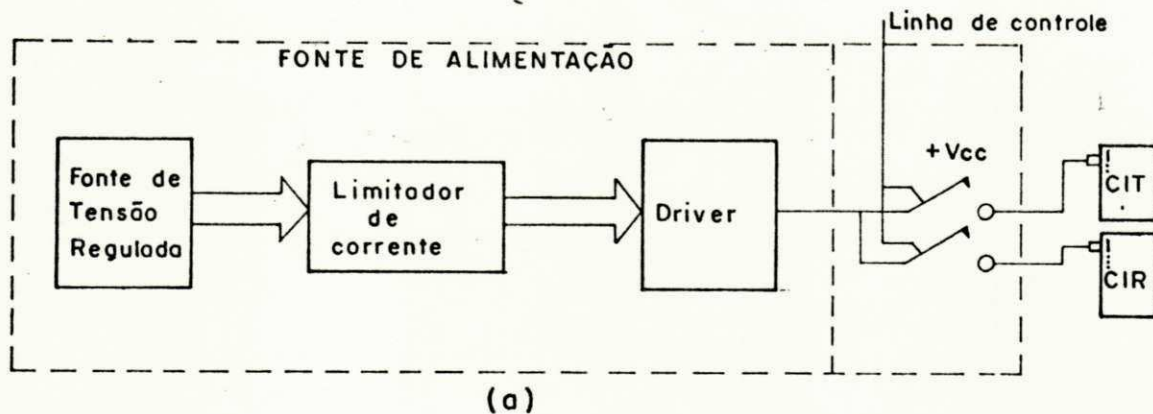


Fig. 3.5 a) Diagrama de blocos do circuito acoplador da tensão Vcc
b) Circuito Acoplador de Vcc para um pino
 F_{Vn} — Linha de controle da chave de acoplamento de Vcc para um pino ($n = 1, 2, \dots, 16$) do CIT

utilizando o CI 723 (V. Figura 3.6), fornecendo uma tensão de saída (igual a $(5 + V_{CE(sat)})$ Volts) no emissor do transistor PNP de saída que, em conjunto com a chave eletrônica implementada com transistor, garante um nível de tensão, +5V, aos terminais V_{CC} do CIT e do CIR. O diagrama elétrico do circuito acoplador de V_{CC} com saída para todos os pinos que podem ser terminais V_{CC} é mostrado na Figura 3.7.

Por sua vez denomina-se **circuito acoplador de ponto comum** ao circuito que fornece a tensão de referência (0 Volts) ao terminal comum do CIT. Este circuito pode ser descrito como sendo um "driver" ligado a chaves eletrônicas discretas (V. Figuras 3.8.a e b). O "driver" foi implementado utilizando um amplificador operacional (tipo 741) dando uma tensão de saída negativa (igual a $-V_{CE(sat)}$ no transistor NPN de saída) que, em conjunto com as chaves eletrônicas implementadas com transistores, garante um nível de tensão, 0 Volts, ao ponto comum do CIT e do CIR. O diagrama elétrico completo incluindo saídas para todos os pinos que podem ser terminais ponto comum é mostrado na Figura 3.9.

b) Decodificador

Como foi visto anteriormente, a finalidade de um pino do CIT pode ser: conexão para V_{CC} (terminal V_{CC}), conexão para tensão de referência (ponto comum), conexão

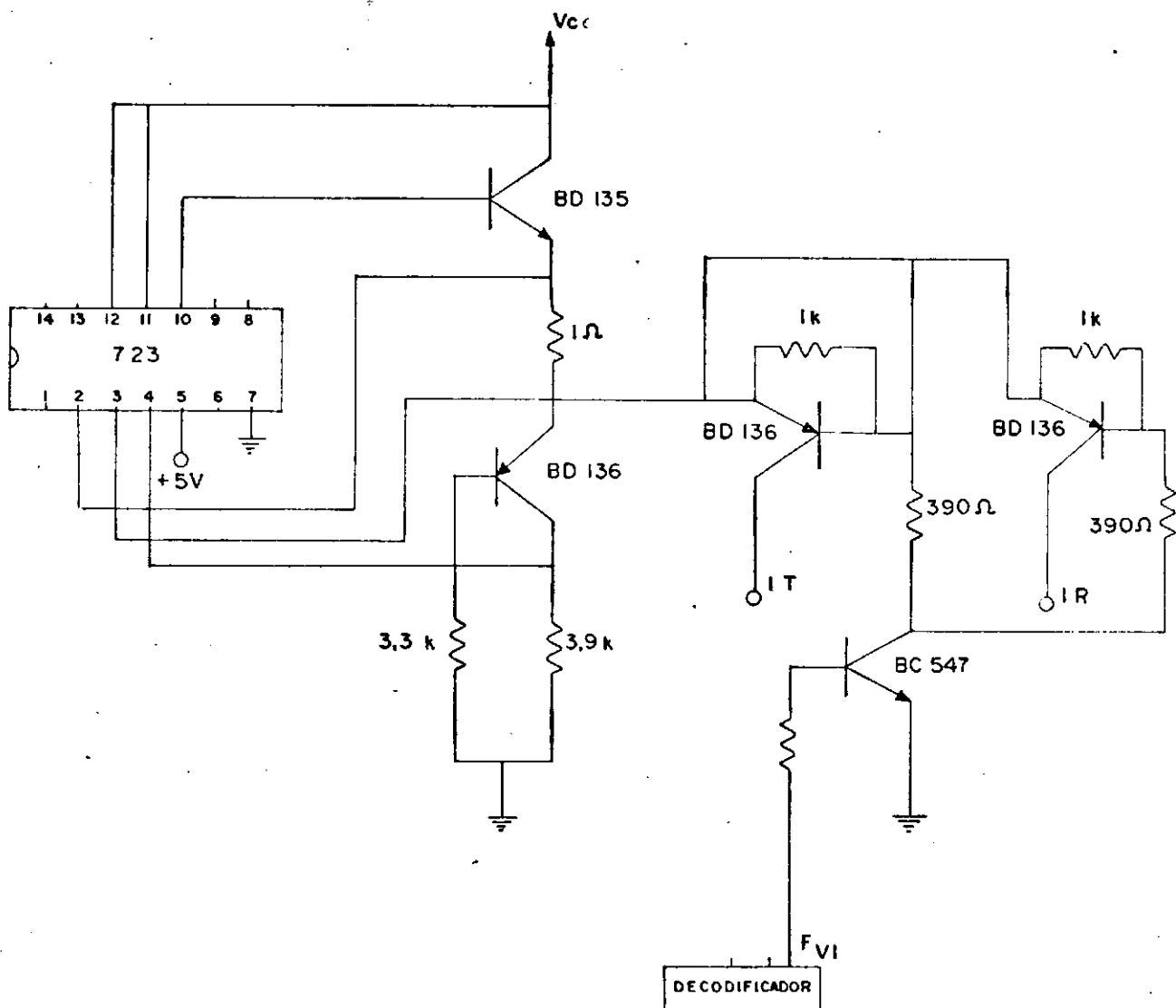


Fig. 3.6 - Diagrama elétrico do Circuito Acoplador da tensão Vcc com saída para o par de pinos [IT,IR]

F_{VI} - Linha de controle das chaves de acoplamento de Vcc para o par de pinos [IT,IR]

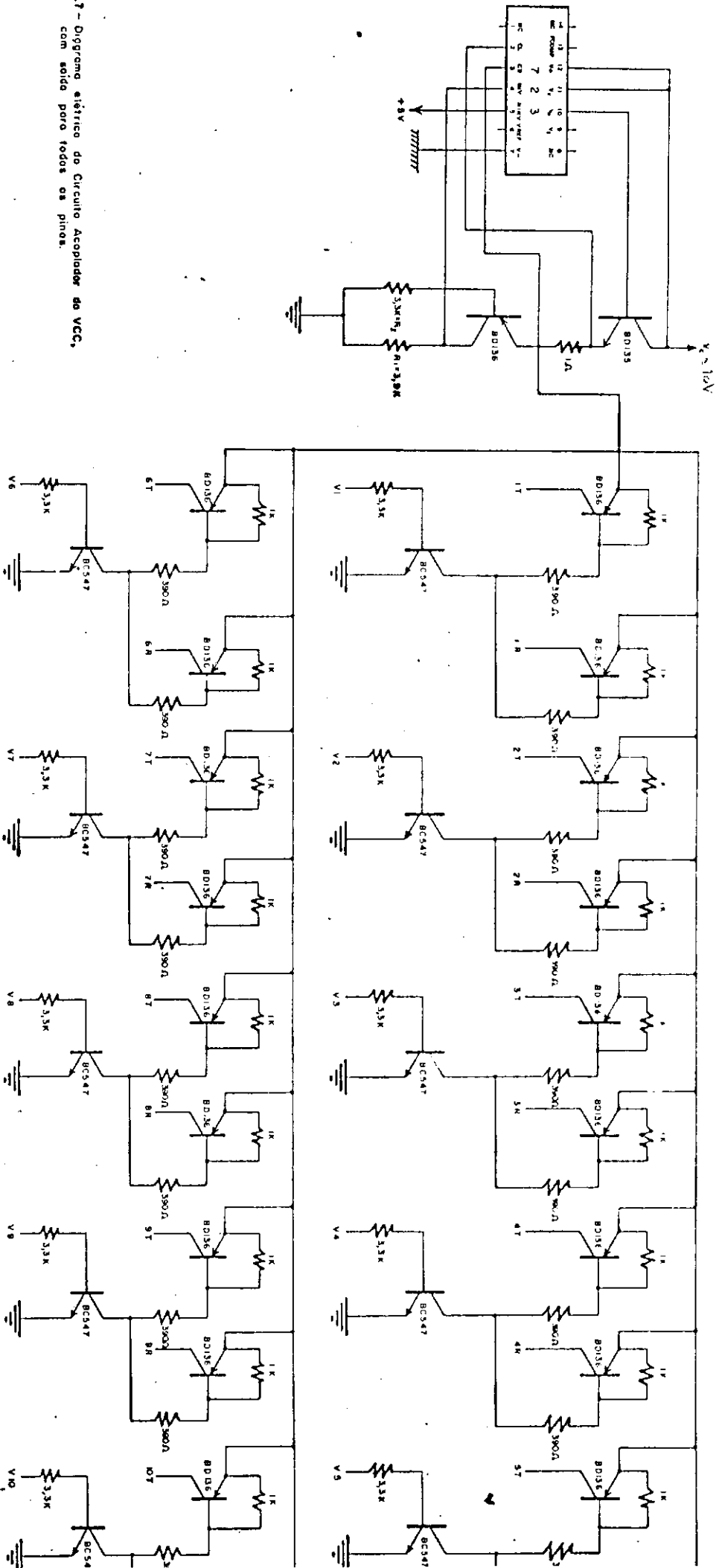
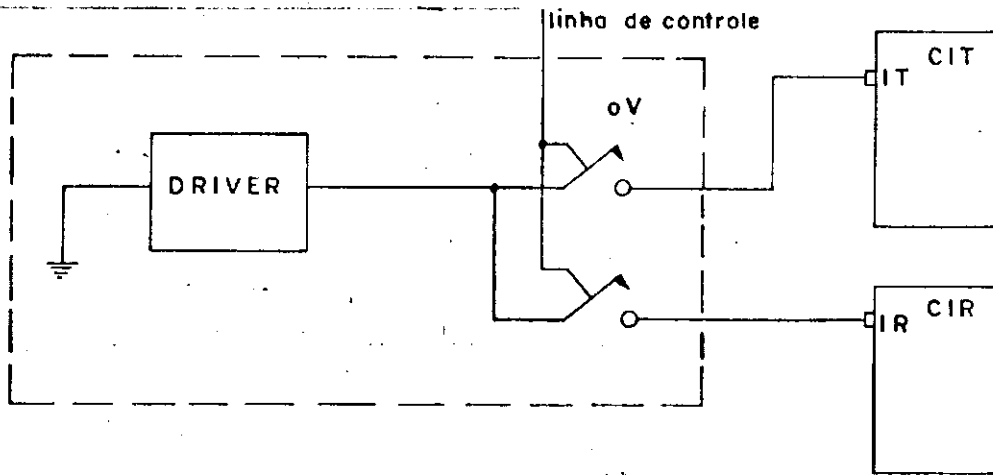
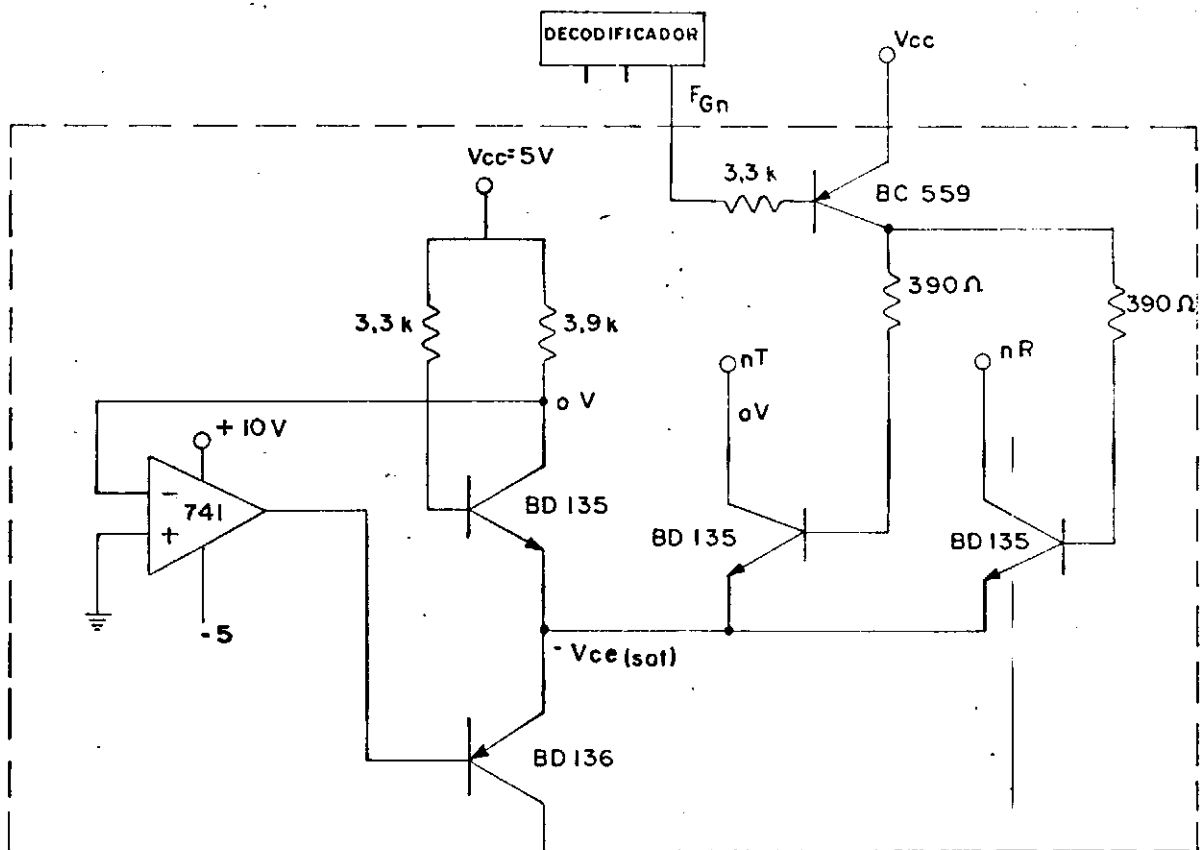


Fig. 3.1 - Diagrama elétrico do Circuito Acoplador da VCC, com saída para todos os pines.



(a)



(b)

Fig. 3.8 a) Diagrama de blocos do circuito acoplador de Ponto Comun
 b) Circuito acoplador de Ponto Comun para um par de pinos
 $|nT, nR|$ ($n=1,2,\dots,16$)

F_{Gn} — Linha de controle das chaves de acoplamento de Ponto Comun para um par de pinos (nT, nR)

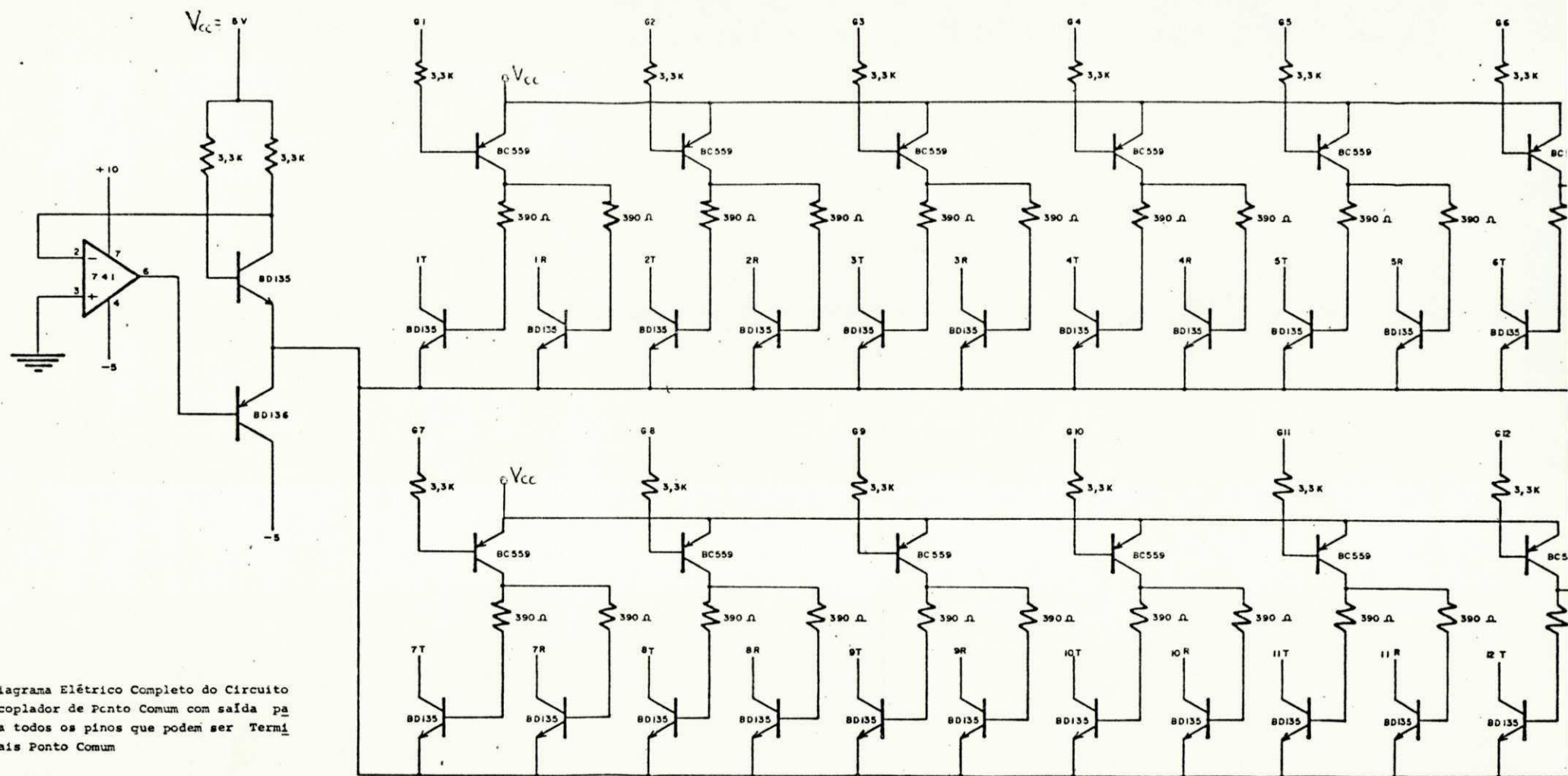


Fig. 3.9 - Diagrama Elétrico Completo do Circuito Acoplador de Ponto Comum com saída para todos os pinos que podem ser Terminais Ponto Comum

para entrada (dentre doze entradas possíveis) (Apêndice A) ou ainda pino de saída. Para codificar as diferentes finalidades (4) levando em conta as entradas possíveis (12) utiliza-se o código binário simples de 4 bits da maneira mostrada na tabela 3.1. As linhas de seleção da Matriz de Interconexão são constituídas, portanto, de quatro linhas de dados para cada pino, resultando em 64 linhas, considerando todos os 16 pinos do CIT. Cada circuito multiplexador recebe, então, através de quatro linhas de seleção, a palavra código de 4 bits que define a finalidade do pino a ele ligado. Esta informação é decodificada em cada circuito multiplexador da seguinte maneira:

- O multiplexador das fontes de sinais de entrada realiza a decodificação, ou seja, identificação dos sinais de entrada, internamente no CI (74150). O multiplexador implementado com chaves discretas necessita de um decodificador para gerar os sinais de controle para as chaves.

O projeto do decodificador necessário a cada circuito multiplexador é mostrado a seguir. Foi desenvolvido a partir da tabela 3.1 onde está contida a associação das palavras código às diferentes finalidades do pino e aos diferentes tipos de entrada. Como entrada do decodificador tem-se as quatro linhas de dados contendo o código de quatro bits que define a finalidade do pino. Como saída tem-se três linhas para controle das chaves eletrônicas discretas: 1) Chaves de acoplamento de V_{cc}

ASSOCIAÇÃO DAS PALAVRAS CÓDIGO	D	C	B	A	P.COMUM	V _{CC}	ENTRADA
P. Comum	0	0	0	0	0	0	0
P. Comum	0	0	0	1	0	0	0
V _{CC}	0	0	1	0	1	1	0
Saída	0	0	1	1	1	0	0
Ent. 1	0	1	0	0	1	0	1
Ent. 2	0	1	0	1	1	0	1
Ent. 3	0	1	1	0	1	0	1
Ent. 4	0	1	1	1	1	0	1
Ent. 5	1	0	0	0	1	0	1
Ent. 6	1	0	0	1	1	0	1
Ent. 7	1	0	1	0	1	0	1
Ent. 8	1	0	1	1	1	0	1
Ent. 9	1	1	0	0	1	0	1
Ent.10	1	1	0	1	1	0	1
Ent.11	1	1	1	0	1	0	1
Ent.12	1	1	1	1	1	0	1

Tabela 3.1 - Tabela Verdade do Decodificador

para o CIT e o CIR. 2) Chaves de acoplamento de ponto comum para o CIT e o CIR e 3) Chaves tri-estado para conectar a saída do multiplexador de sinais de entrada para o CIT e o CIR.

O diagrama lógico do decodificador é mostrado na Figura 3.10.a. As expressões lógicas para as saídas do decodificador foram obtidas a partir da tabela 3.1 utilizando Mapa de Karnaugh, quando necessário, como se segue:

Mapa de Karnaugh para a saída F_{entrada} .

DC BA	00	01	11	10
00	0	1	1	1
01	0	1	1	1
11	0	1	1	1
10	0	1	1	1

$$F_{\text{entrada}} = D + C$$

A saída $F_{\text{ponto comum}}$ foi obtida diretamente da tabela tomando os termos máximos ("Maxterms") já que as chaves para acoplar o ponto comum são nível baixo ativo.

$$F_{\text{ponto comum}} = D + C + B$$

$$\text{A saída } F_{V_{CC}} = \overline{D} \cdot \overline{C} \cdot B \cdot \overline{A} = \overline{D + C + B + A}$$

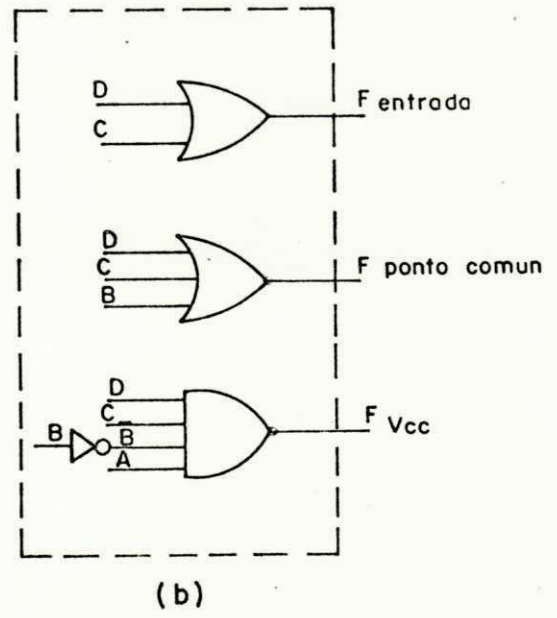
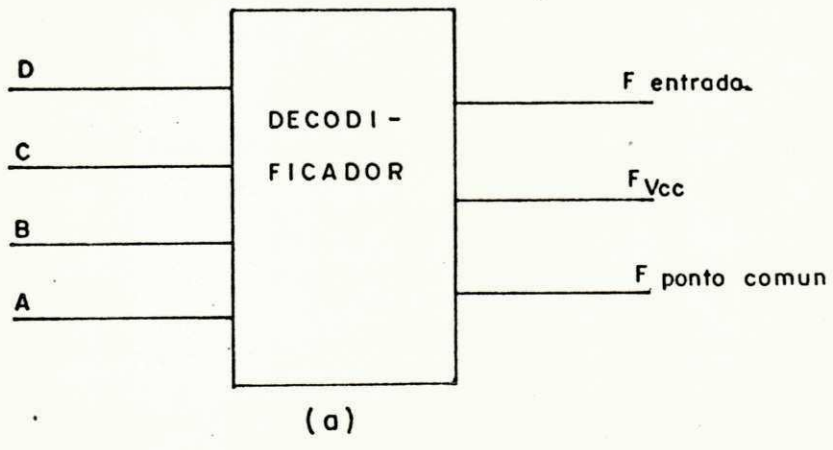


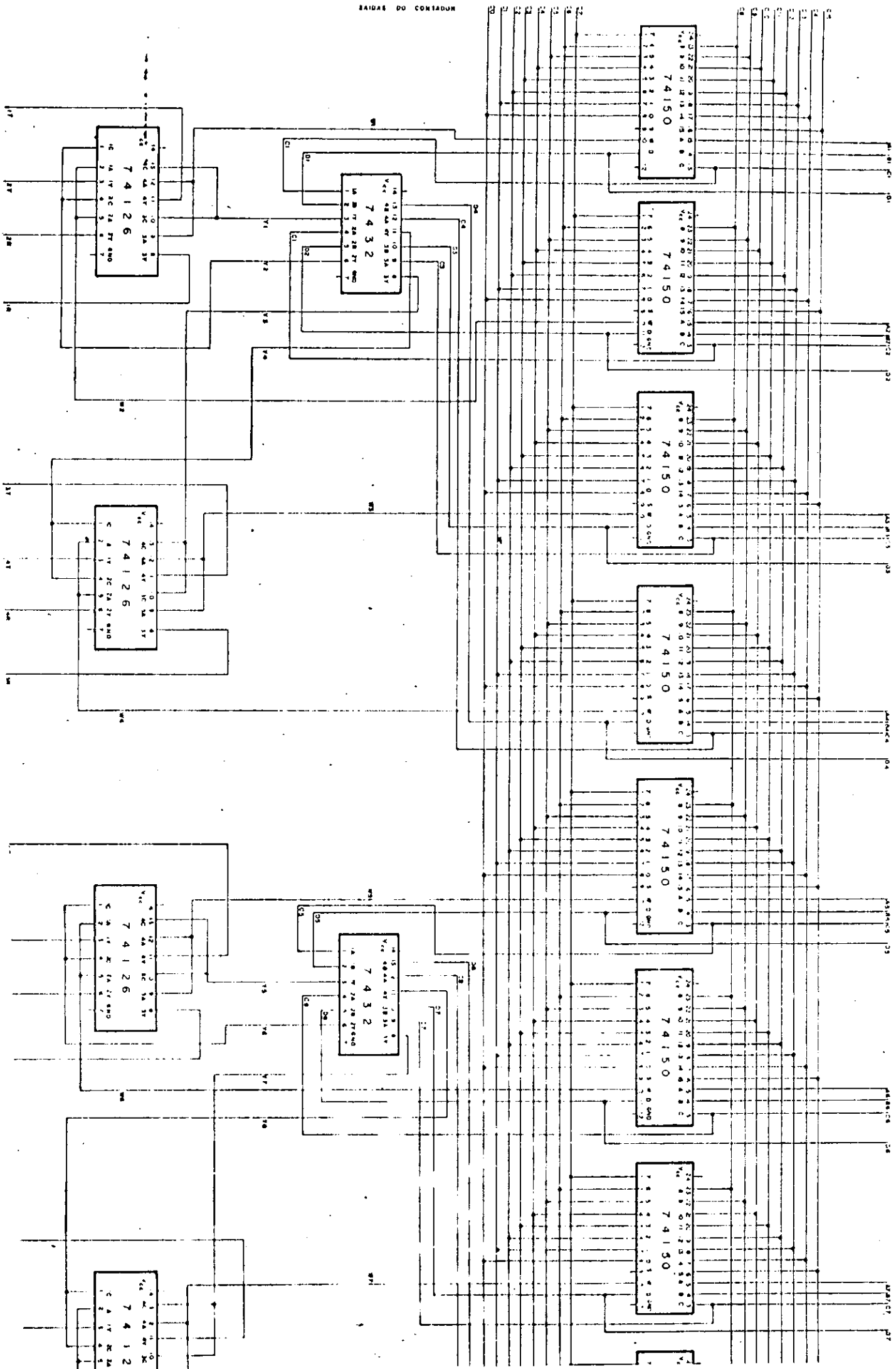
Fig. 3.10 a) Diagrama de blocos do decodificador
b) Implementação do circuito decodificador

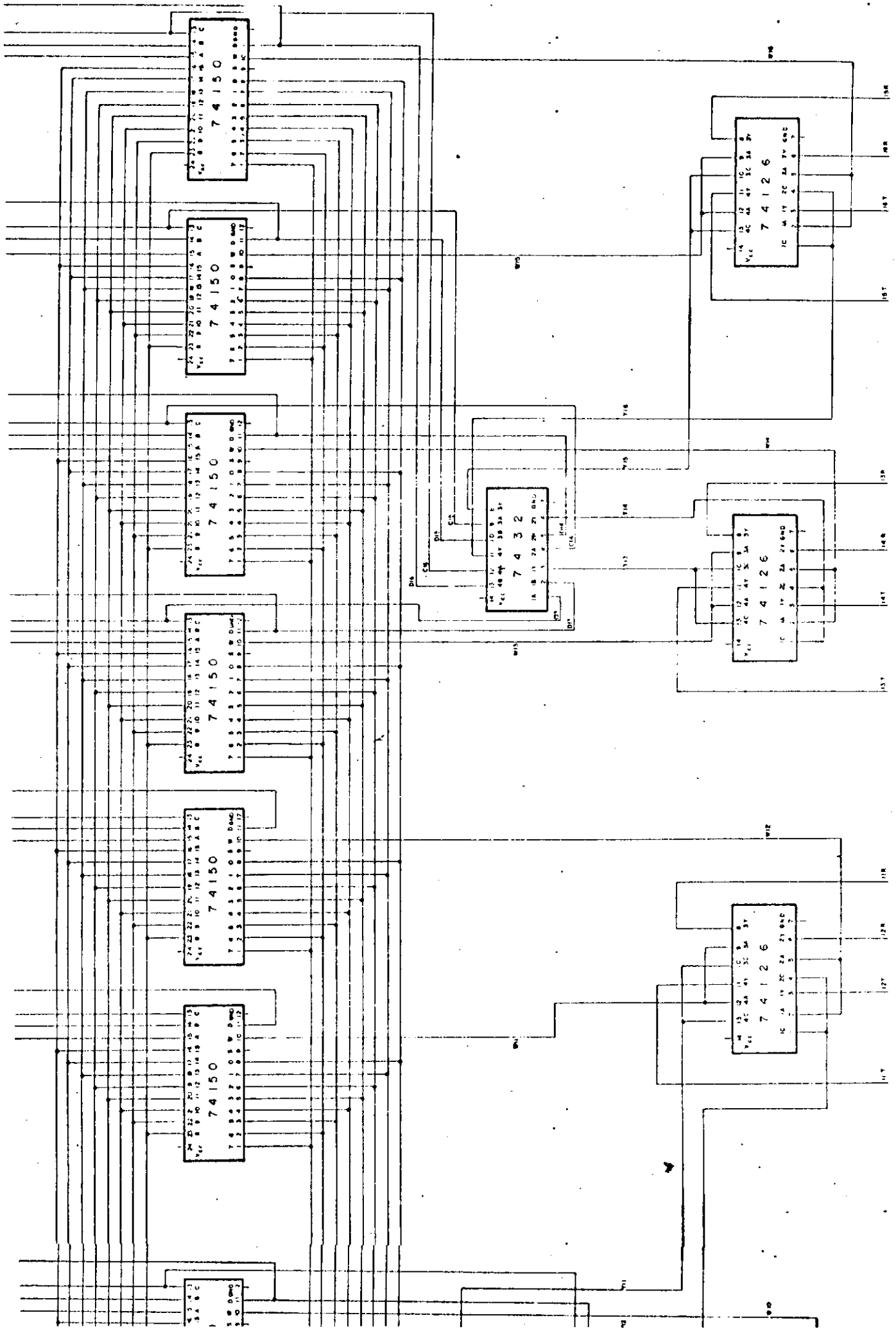
O decodificador de cada circuito multiplexador, a princípio, é implementado como mostra a Figura 3.10.b; com uma porta NOU 4 entradas, uma porta NOU 3 entradas, uma porta inversora e um OU - duas entradas.

Pelo estudo das configurações de CIT (Apêndice A) tem-se que apenas dez dos dezesseis pinos do CIT podem servir de conexão para V_{CC} (terminal V_{CC}) enquanto doze podem ser terminal ponto comum. Com isto, reduz-se mais o número total de portas necessárias para o decodificador de alguns circuitos multiplexadores (assim como se reduz o número de chaves eletrônicas discretas). O número total de CI's necessários para implementar os dezesseis decodificadores é quinze, ou seja, cinco duplos NOU de 4 entradas, quatro triplos NOU de 3 entradas, dois hex-inversores e quatro quádruplos OU de 2 entradas.

Caso se deseje utilizar decodificadores já integrados em pastilhas, são necessárias dezesseis pastilhas decodificadoras mais caras que portas discretas, o que não é vantajoso.

O diagrama elétrico da Matriz de Interconexão é mostrado na Figura 3.11, a menos dos circuitos de acoplamento de V_{CC} e de Ponto Comum e de parte dos decodificadores que geram as linhas de controle para o acoplamento de V_{CC} e de Ponto Comum. Neste diagrama são mostrados os dezesseis multiplexadores de fontes de sinais de entrada (CI's 74150),





as chaves tri-estado (CI's 74126) para conexão das saídas dos multiplexadores de sinais de entrada e a parte dos decodificadores que controla estas chaves (CI's 7432). A parte dos circuitos decodificadores responsável pela geração das linhas de controle das chaves de acoplamento de V_{cc} é mostrada na Figura 3.12 e a parte responsável pela geração das linhas de controle das chaves de acoplamento do Ponto Comum é mostrada na Figura 3.13.

As conexões com o CIT e CIR são efetuadas através de soquetes tipo ZIF (V. Figura 3.14).

3.2.2 - Gerador de Códigos

O circuito gerador de códigos deve receber como entrada o número do CIT (em código BCD) enviado pelo operador através do teclado, e fornecer à Matriz de Intercaptação um padrão de bits, que correspondem à informação codificada da função dos pinos do CIT.

As diferentes finalidades de um pino foram codificadas utilizando o código binário simples de quatro bits, conforme a Tabela 3.1. Como foi visto anteriormente (seção 3.2.1.3.b) a finalidade de cada pino é codificada com quatro bits. Desse modo o padrão de bits fornecido pelo gerador de códigos é uma palavra-código de sessenta e quatro bits capaz de conter a informação sobre a finalidade de to

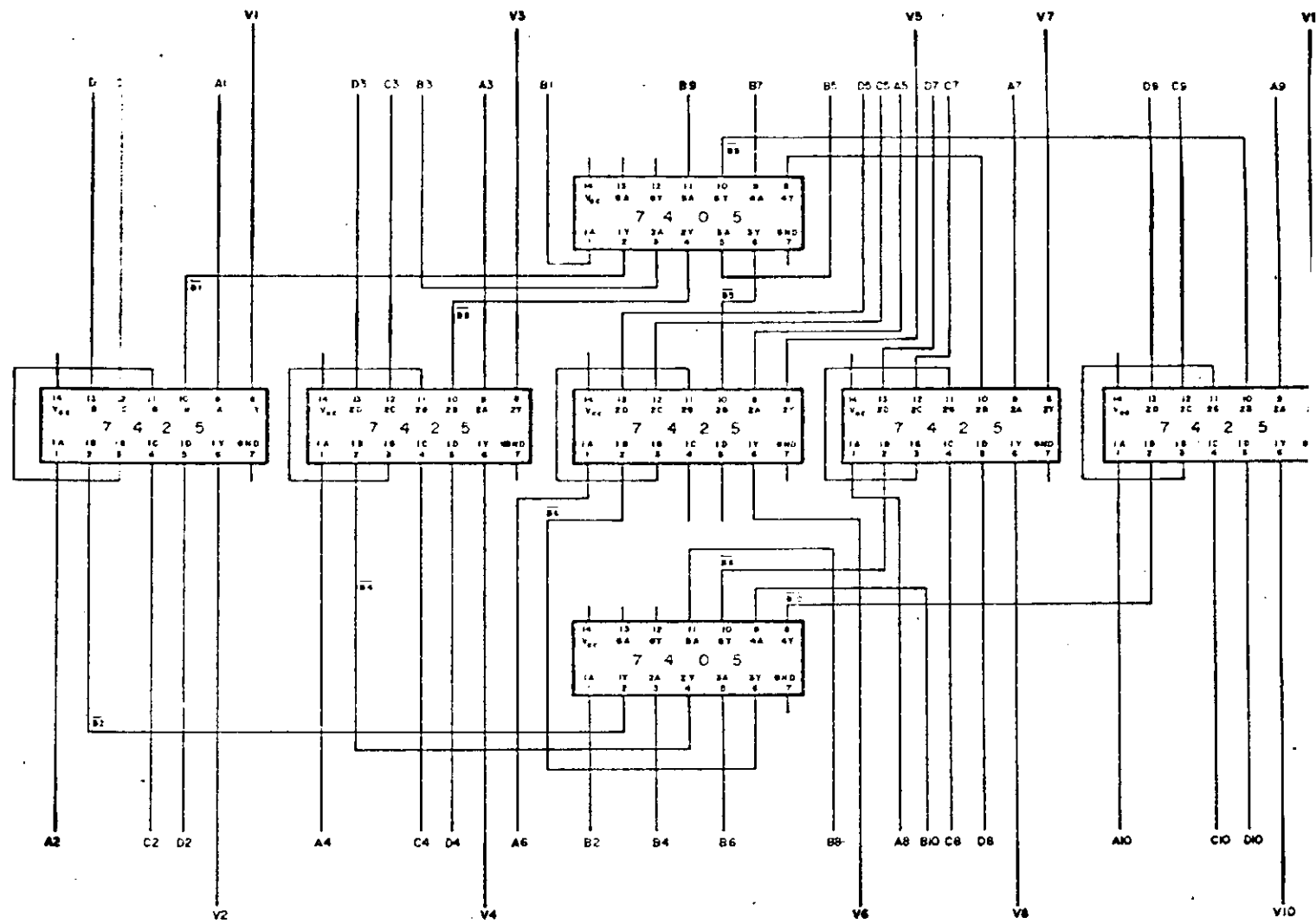


Fig. 3.12 - Diagrama elétrico de parte do Circuito Decodificador que gera as linhas de controle para copiar a tensão de VCC.

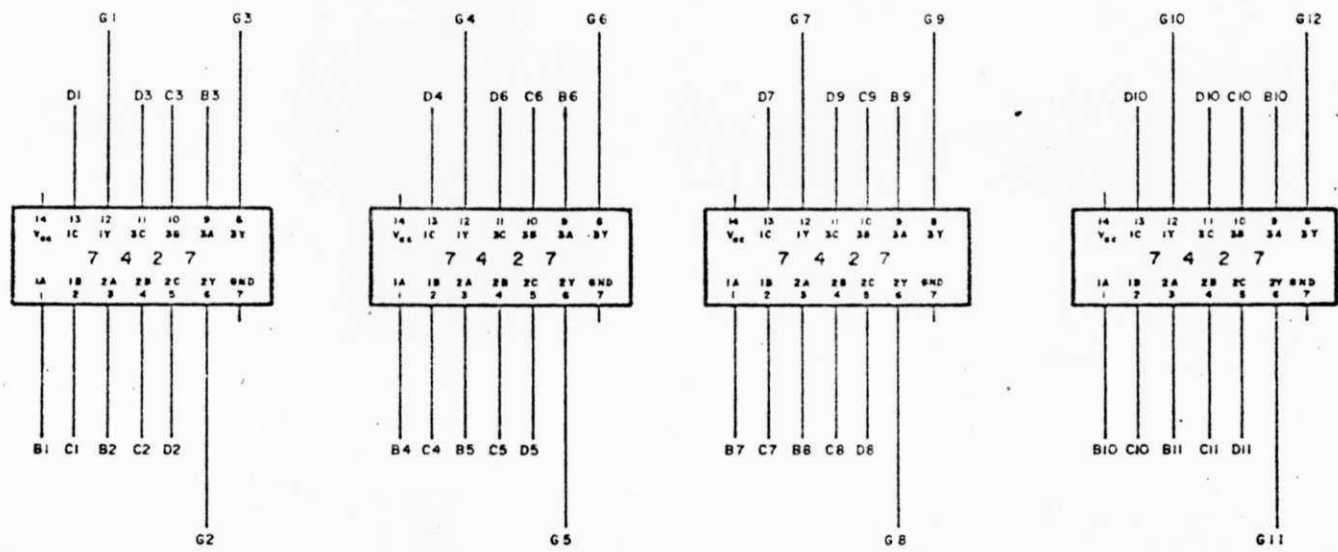


Fig. 3.13 - Diagrama elétrico de parte do Circuito Decodificador que gera as linhas de controle para o Circuito Acondicionador de ponto comum.

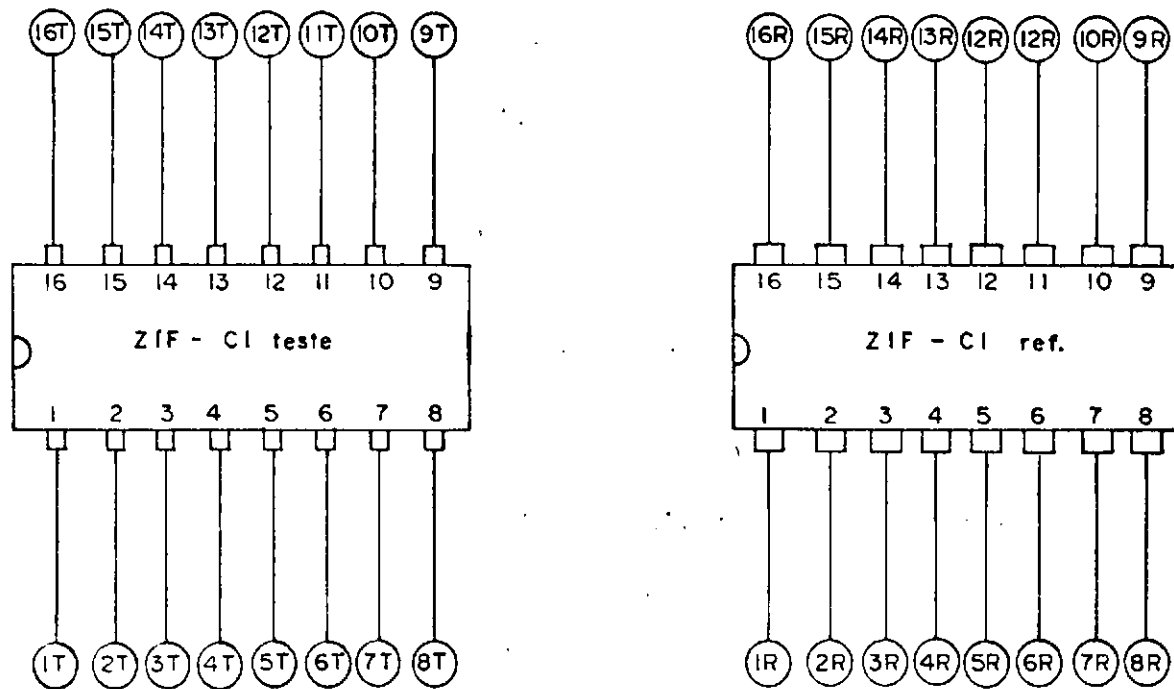


Fig. 3.14 Soquetes ZIF para o CIT e o CIR com numeração dos pinos

dos os pinos do CIT. O gerador de códigos consiste, basicamente, em uma memória, aqui denominada tabela de códigos, organizada como tabela de dados. Em cada endereço da memória está contida a palavra-código de sessenta e quatro bits correspondente a uma configuração distinta de CIT.

Ao se testar um CI, coloca-se inicialmente o número deste CIT nas linhas de endereço da memória e esta, por sua vez, fornece na saída o conteúdo deste endereço, que é a palavra-código correspondente à configuração de pinos desse CI.

Existem alguns tipos de CI's que possuem a mesma configuração de pinos, alguns tipos que não são mais utilizados, e ainda outros não poderão ser testados por este sistema, como é o caso, por exemplo, de memórias. Devido a isso os números de identificação de CI's que podem ser testados por este sistema não são seguidos (Apêndice A). Consequentemente não é vantajoso usar o número de identificação do CIT como endereço direto da memória que contém as palavras-código (Tabela de códigos) correspondentes às configurações dos CI's, pois isto redundaria em espaços vazios na memória. Por esse motivo é utilizada uma outra memória, com tamanho de palavra bem menor, como memória auxiliar, para se efetuar o endereçamento de modo indireto da tabela de códigos e, assim, obter uma compactação da tabela de códigos, o que é mais interessante (V. Figura 3.15). Agora, a memória auxiliar, chamada tabela de endereços, é que

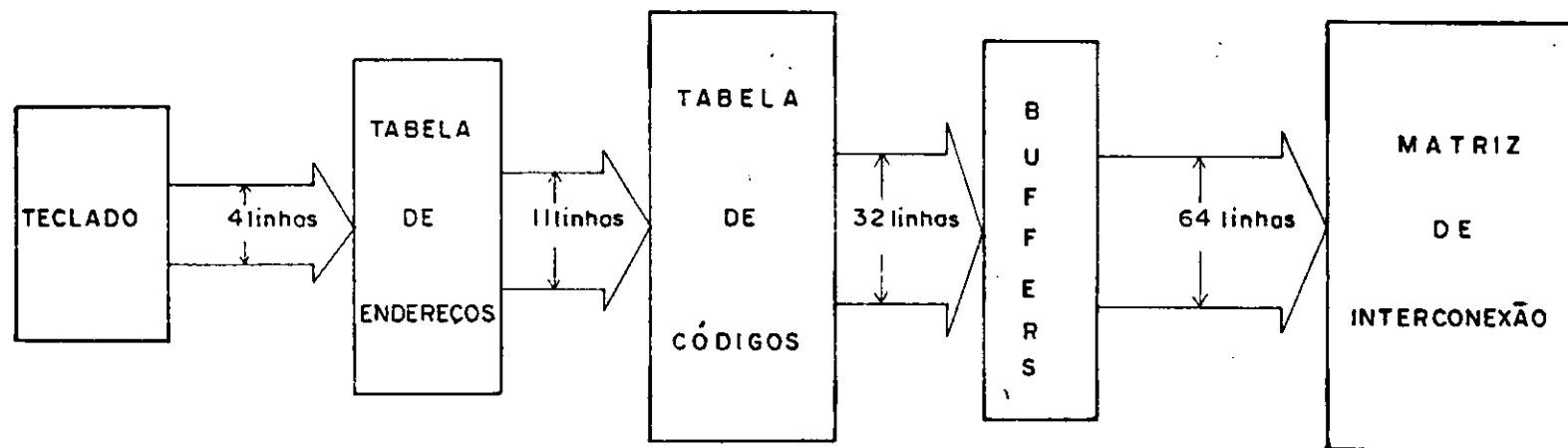


Fig. 3.15 Diagrama de blocos do Gerador de Códigos

vai conter os endereços (da tabela de códigos) das palavras código correspondentes aos CI's de teste. A tabela de endereços recebe o número do CIT e fornece na sua saída o endereço da locação da tabela de códigos que contém a palavra-código correspondente àquele CIT. A Tabela de códigos, então, coloca na saída a palavra-código, a qual é enviada à Matriz de Interconexão.

As memórias mais comumente utilizadas possuem palavras de 8 bits. Para implementar a tabela de códigos, então, seriam necessários oito memórias em paralelo para se obter a palavra de 64 bits. Optou-se, então, por utilizar "buffers" para armazenar a palavra-código fornecida pela tabela de códigos, de modo a permitir a leitura por partes, da memória.

A tabela de códigos foi organizada com palavras duplas de 2 x 32 bits. A primeira linha de endereço (A_0) é utilizada como linha de controle para endereçar a primeira parte da palavra ($A_0 = 0$) ou a segunda ($A_0 = 1$). Deste modo, foram utilizadas quatro memórias em paralelo, com palavras de oito bits, para implementar a tabela de códigos. Os "buffers" são oito registradores de oito bits (CI's 74198) (V. Figura 3.16).

A última linha de endereço (A_{10}) foi utilizada para selecionar a região da memória onde estão armazenadas as informações sobre CI's da família TTL ($A_{10} = 0$) ou da famí

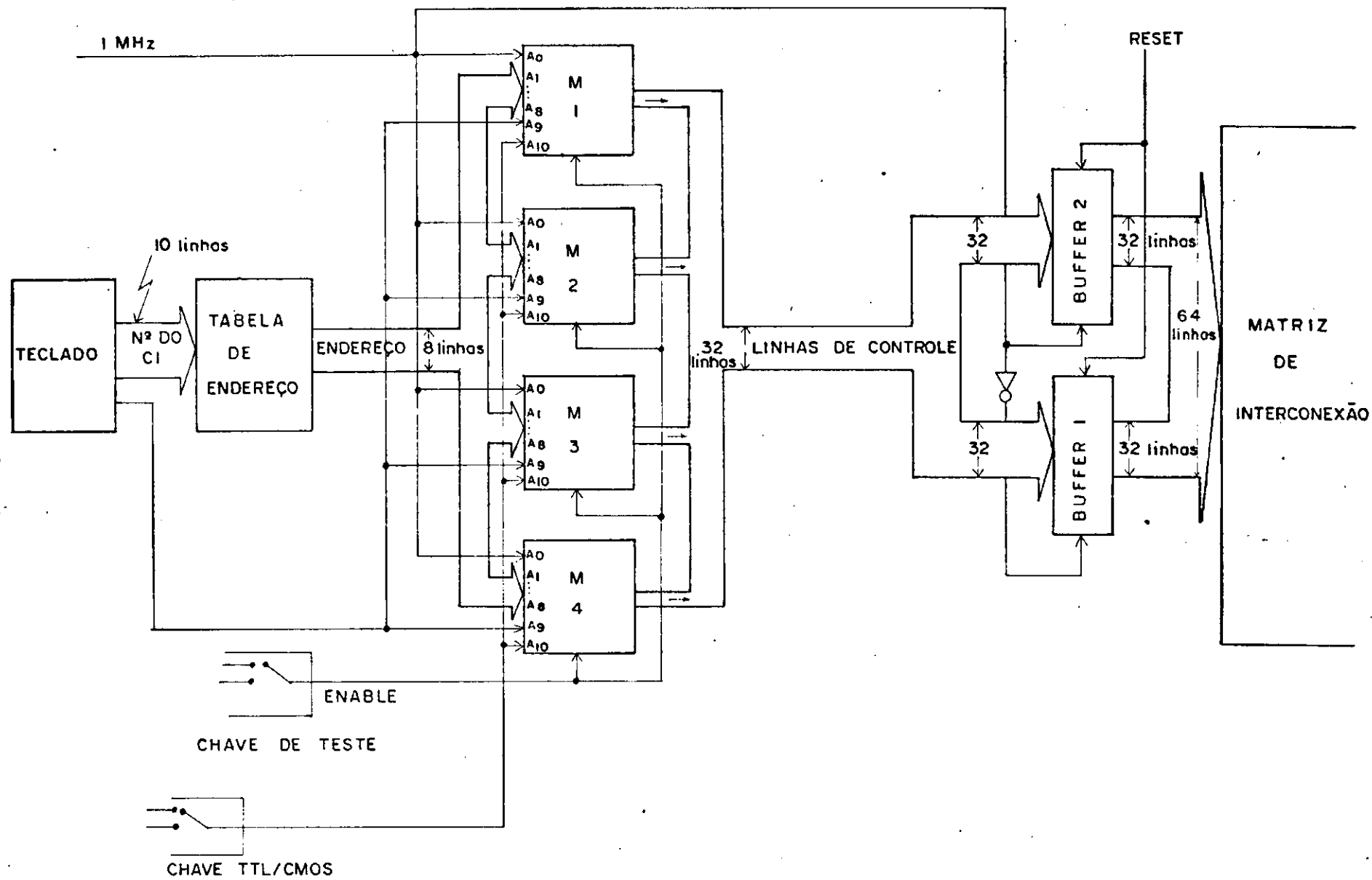


Fig. 3.16 - Diagrama do Gerador de Códigos (detalhado)

lia CMOS ($A_{10} = 1$). Esta seleção é feita através de uma chave (TTL-CMOS) acionada pelo operador.

O diagrama elétrico do Gerador de Códigos é mostrado na Figura 3.17. A tabela de códigos foi implementada com quatro memórias com capacidade 2Kx8, cada, do tipo 2716. A tabela de endereços foi implementada por uma memória 1Kx8 tipo 2708. As onze linhas necessárias para endereçar a tabela de códigos foram fornecidas como se segue:

A_0 - linha que seleciona a parte da palavra dupla a ser lida. É ligado a uma saída do gerador de entradas com frequência $f_1 = 1$ MHz

$A_0 = 0$, para a primeira parte da palavra,

$A_0 = 1$ para a segunda parte.

$A_1, A_2 \dots A_8$ - linhas fornecidas pela tabela de endereços.

A_9 - linha de endereço fornecida diretamente do teclado.

A_{10} - linha controlada por uma chave que identifica a família do CIT. $A_{10} = 0$ para a família TTL e $A_{10} = 1$ para a família CMOS.

Ao se testar um CI, o seu número, fornecido pelo te

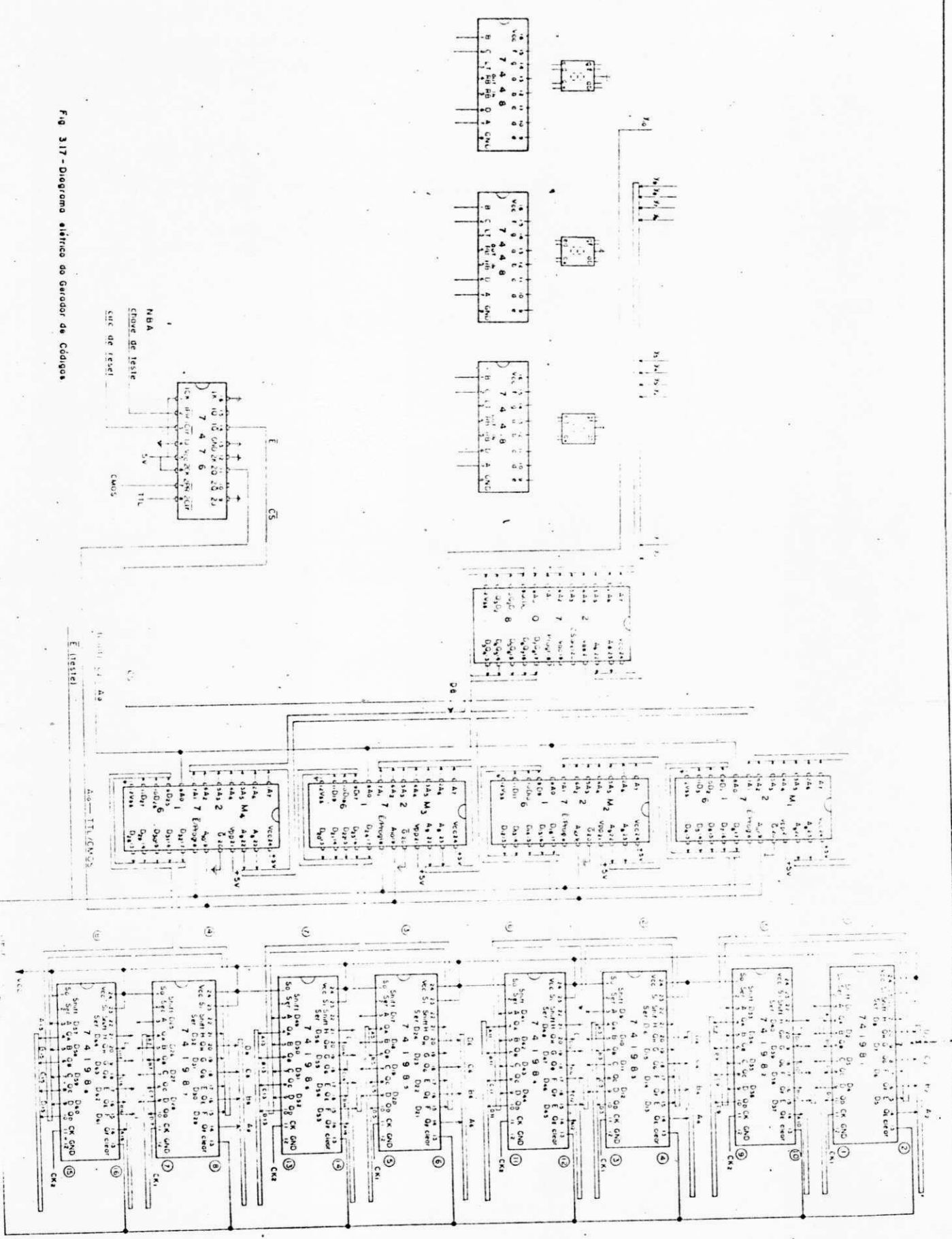


Fig 317 - Diagrama elétrico do Gerador de Códigos

clado, fica disponível na entrada da tabela de endereços do CI que, por sua vez, fornece o endereço da palavra-código, endereço esse que fica disponível na entrada da tabela de códigos. A palavra-código correspondente, porém, só é armazenada nos "buffers" quando o "enable" da memória (tabela de códigos) for ativado. Isto é feito utilizando uma chave (chave de teste) acionada pelo operador. Esta mesma chave é utilizada para desativar o "clear" dos "buffers". Os "buffers" estão inicialmente no nível lógico zero ("clear" ativado), pois isto equivale a colocar todos os pinos do CIT em ponto comum, evitando assim, conexão errada nos pinos. Quando o operador está seguro de que o número do CIT teclado está correto, então, aciona a chave de teste, desativando, assim, o "clear", dos buffers e iniciando o armazenamento da palavra-código. Quando $A_0 = 0$ a primeira parte da palavra-código é armazenada numa determinada região dos "buffers". Quando $A_0 = 1$ a segunda parte da palavra-código é armazenada em outra região dos "buffers". Estas duas regiões compõem a palavra completa. As saídas dos "buffers" estão ligadas diretamente à Matriz de Interconexão. As conexões dos pinos do CIT, consequentemente, também são efetuadas em duas etapas. Primeiramente, os pinos do CIT correspondentes à primeira parte da palavra-código são conectados. Em seguida, são conectados os pinos restantes. O fato de não se ter conexão simultânea de todos os pinos, não acarreta danos ao CI porque os pinos não conectados no intervalo entre essas duas etapas permanecem no nível lógico zero (ponto comum).

As palavras código para algumas configurações de CI's são apresentadas no Apêndice B.

3.2.3 - Teclado

3.2.3.1 - Introdução

O número de identificação de um CI consta de quatro ou cinco dígitos, dos quais os dois primeiros identificam a família lógica e a série, e os restantes, o tipo (função) do CI.

Em relação aos CI's existentes no mercado, pode-se dizer que o número que identifica o tipo do CI não ultrapassa 799.

No presente sistema, a identificação da família do CI é feita através de uma chave TTL/CMOS, acionada pelo operador, enquanto que a identificação do tipo do CI é feita com três dígitos fornecidos pelo operador através de um teclado.

O número do CIT que é enviado para o Gerador de Códigos servindo de endereço direto para a tabela de endereços e de endereço indireto para a tabela de códigos, é inicialmente fornecido por um Teclado e armazenado em re

gistradores. As saídas desses registradores são ligadas diretamente às linhas de endereço da tabela de endereços, no gerador de códigos (V. Figura 3.18).

O circuito do Teclado compreende o teclado propriamente dito e o circuito de interface.

O teclado utilizado é simples e em forma de matriz com doze teclas dispostas em quatro colunas e três linhas (V. Figura 3.19).

O circuito de interface compreende o circuito de codificação do teclado, registradores para armazenar o número do CI e o circuito para "debouncing" (V. Figura 3.20).

O circuito de codificação do teclado foi implementado utilizando a técnica de varredura de linhas. Consiste na varredura de um conjunto de linhas com um "1 (ou 0) circulante" e no exame das outras linhas para a detecção do pressionamento de alguma tecla. Foi utilizado um contador acoplado a um decodificador para realizar a varredura das colunas. Para a detecção do pulso gerado pela tecla foi utilizado um codificador de prioridade ligado às linhas.

Quando uma tecla é pressionada, um pulso de sinalização é enviado pelo codificador e a varredura é interrompida. O código correspondente à tecla pressionada é

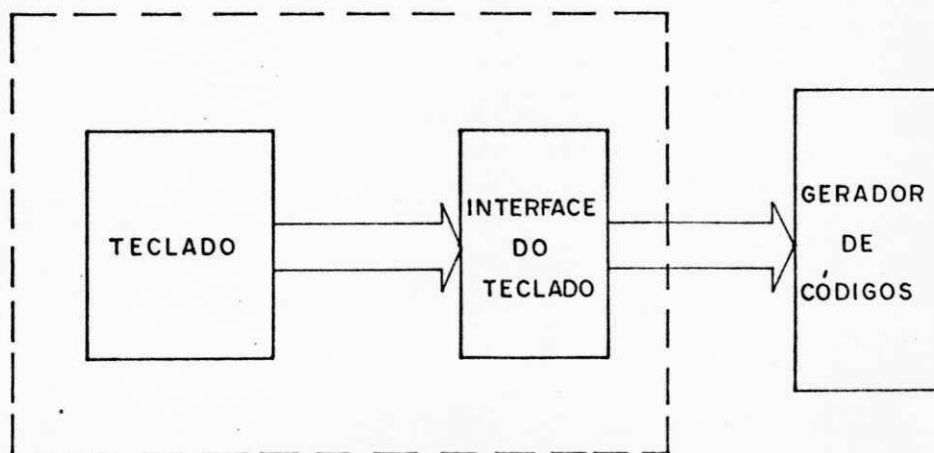


Fig. 3.18 Diagrama de blocos do Circuito do Teclado

0	1	2	3
4	5	6	7
8	9	*	#

Fig. 3.19 Teclado

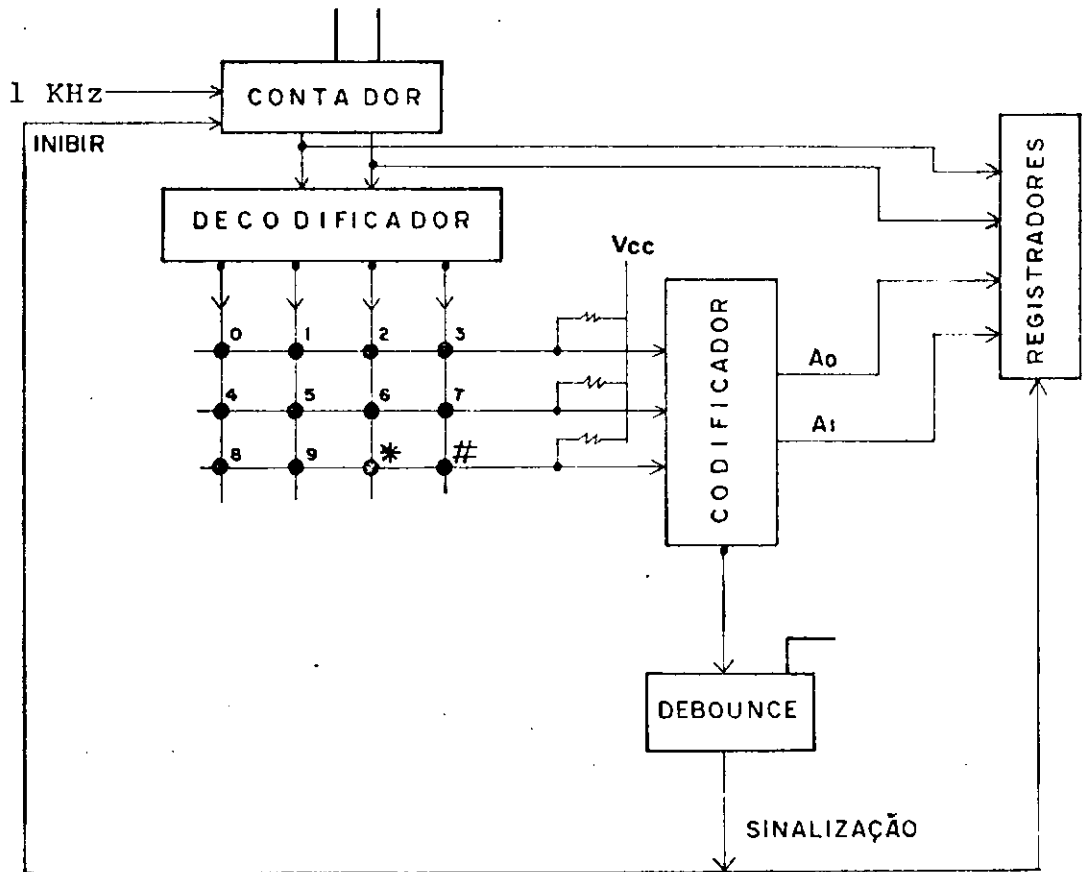


Fig. 3.20 Diagrama de blocos do Circuito do Teclado Detalhado

lido do contador e da saída do codificador, já que estes indicam a coluna e a linha, respectivamente. Como o codificador é de prioridade, se duas teclas forem acionadas simultaneamente, somente uma delas será codificada, justamente aquela que tem maior prioridade em relação às linhas e que, em relação às colunas, é ativada primeiro pelo "1 circulante".

3.2.3.2 - Circuito "Debounce"

Quando uma chave é acionada, o contato oscila por um curto intervalo de tempo antes de efetuar a ligação ("Keybounce"). O mesmo fato ocorre quando a chave abre (V. Figura 3.21).

Quando uma tecla é pressionada, portanto, há uma geração de pulsos estreitos antes da estabilização do contato. Ao ser liberada a tecla, ocorre o mesmo tipo de oscilação, com efeito análogo. Esta oscilação indesejável precisa ser eliminada. Para isto, é necessário um circuito que faça o "debounce", ou seja, que elimine a oscilação da chave. Este circuito é implementado utilizando um circuito monoestável com redisparo (V. Figura 3.22). O funcionamento deste circuito é ilustrado na tabela 3.2. Quando se liga este circuito, da maneira mostrada no diagrama da Figura 3.22, ocorre o seguinte: A saída G do codificador

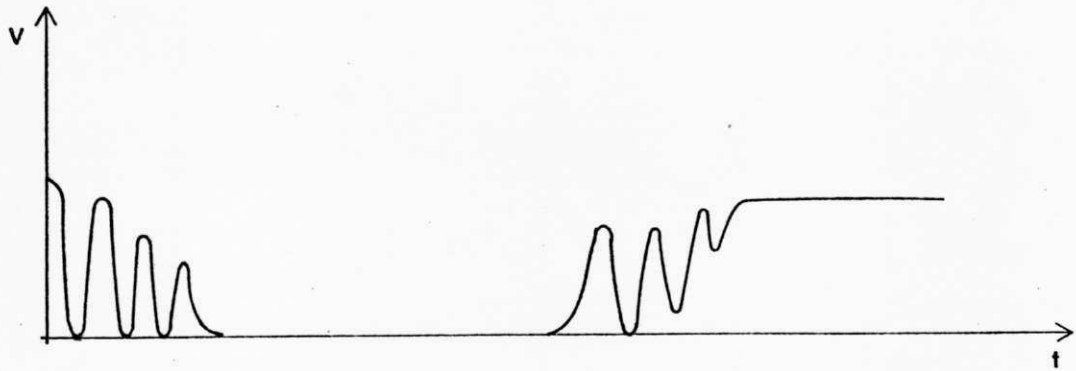


Fig. 3.21 - Tensão na saída da chave devido ao efeito "bounce"

ENTRADAS		SAÍDAS			
CLEAR		A	B	Q	\bar{Q}
	L	X	X	L	H
→	X	H	X	L	H
	X	X	L	L	H
→	H	L	↑	⌊	⌋
→	H	↓	H	⌊	⌋

Tabela 3.2 Tabela-verdade do CI 74123

acompanha a oscilação da chave. Como esta está ligada à entrada A do monoestável, a forma de onda em A é do tipo mostrado no diagrama de tempo da Figura 3.23. A entrada B do monoestável está ligada à saída do oscilador, com período (ζ) bem menor que a largura do pulso gerado pelo monoestável (T_w) ($\zeta \ll T_w$), para causar o "redisparo", ou seja, manter a resposta do monoestável. Quando uma tecla é pressionada, (instante t_0), a entrada A cai de 1 para 0, acompanha o "bounce". Isto faz com que o monoestável gere um pulso na saída com largura T_w ($\bar{Q} = 0$). Este tempo é suficiente para que ocorra a estabilização da tecla, eliminando, assim, o bounce. Como o tempo entre as transições de descida na entrada B é bem menor que T_w ($\zeta \ll T_w$), estas causam o redisparo no monoestável, mantendo a saída \bar{Q} em zero. Isto permanece até que a tecla seja liberada. A entrada A novamente acompanha o "bounce", porém o redisparo em B pouco antes do "bounce" garante que a saída \bar{Q} ainda se mantenha em zero o tempo suficiente (T_w) para o contato estabilizar fazendo novamente o "debounce". Com a tecla liberada, G vai para o nível lógico 1, ou seja, $A = 1$ e $\bar{Q} = 1$.

A saída \bar{Q} é utilizada para inibir o contador responsável pela varredura do teclado e ainda como entrada para o circuito que vai armazenar os códigos, correspondentes às teclas pressionadas, ao se teclar o número do CIT.

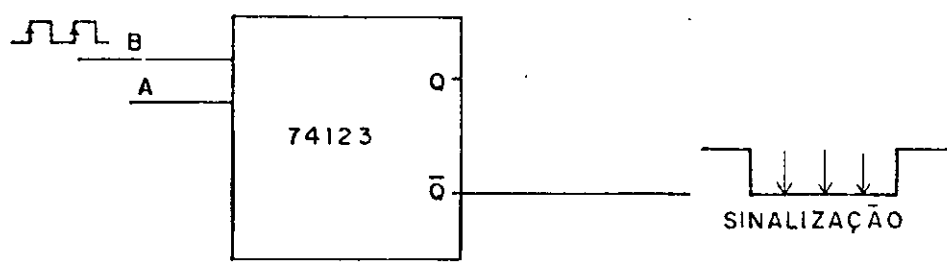


Fig. 3.22 Diagrama lógico do 74123

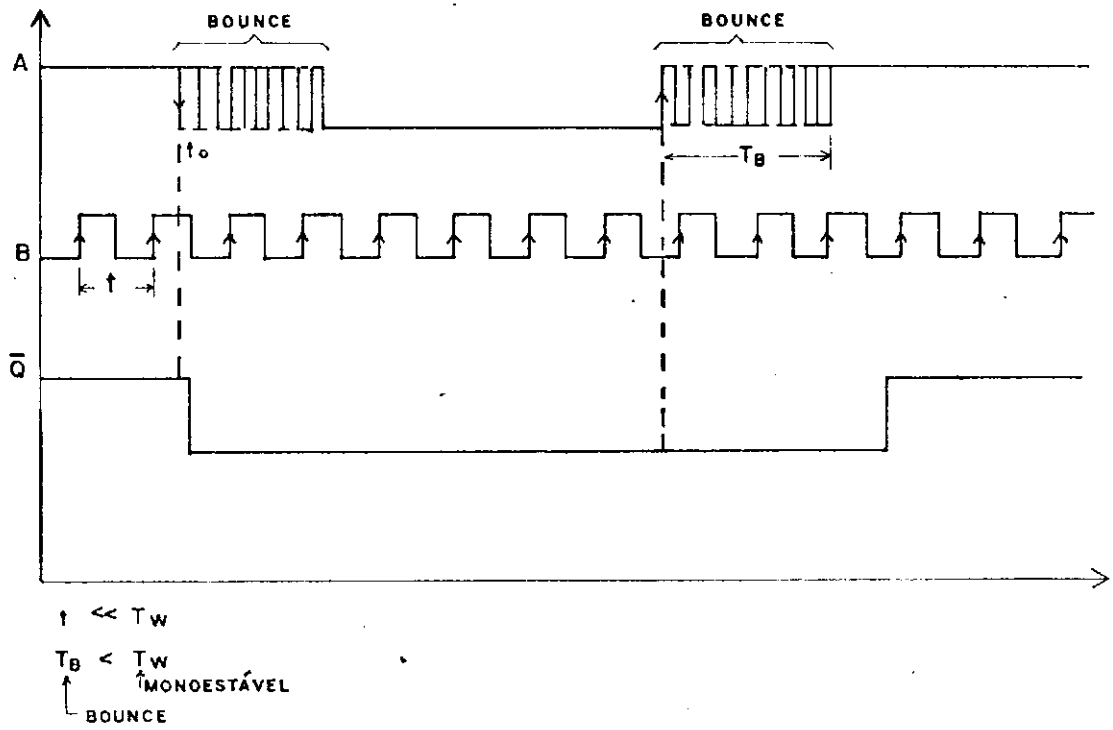


Fig. 3.23 Diagrama de tempo do circuito "Debounce"

3.2.3.3 - Registradores

Ao se pressionar uma tecla, o dígito corres
pondente é codificado em BCD. Para fornecer o número de um
CIT, deve-se teclar os três dígitos, um de cada vez, e ar
mazená-los à medida que forem teclados. Isto é feito utili
zando três registradores de quatro bits, um contador de
dois bits e um decodificador (um em quatro) conforme ilus
tra o diagrama da Figura 3.24. A saída \bar{Q} do monoestável é
utilizada para incrementar o contador a cada dígito tecla
do. O estado do contador ($\neq 0$) identifica a ordem do dígi
to teclado (primeiro, segundo ou terceiro dígito, a partir
do mais significativo). O estado do contador é decodifica
do e as saídas do decodificador ativam os três registrado
res de modo a armazenar os dígitos na ordem em que forem
digitados.

O número é digitado na mesma ordem em que é
lido, mas, para isto, deve ser digitado sempre com três dí
gitos, mesmo quando for menor que 100. Por exemplo, o CI
7432 é identificado da seguinte maneira:

- 1) Aciona-se a chave TTL/CMOS para a posiçã
o que identifica a família TTL.
- 2) Tecla-se o dígito 0.
- 3) Tecla-se o dígito 3.

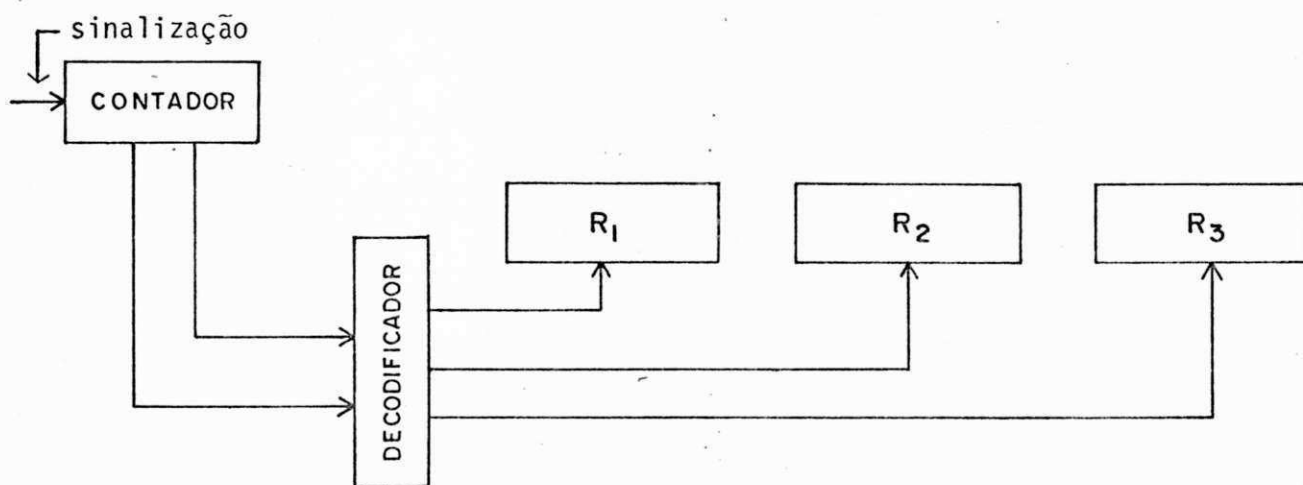


Fig. 3.24 Bloco registradores da Fig. 3.20

4) Tecla-se o dígito 2.

O diagrama elétrico completo do circuito do Teclado é mostrado na Figura 3.25. Os contadores utilizados para varredura do teclado e direcionamento dos dígitos que identificam o tipo do CIT são implementados com circuitos integrados tipo CI 7473. Para os decodificadores foi utilizado o CI 74139. O codificador de prioridade |5|, o monoestável de redisparo e os registradores de quatro bits são implementados, respectivamente, pelos CI's 74148, 74123 e 74173.

Para visualizar o número teclado, que identifica o tipo do CIT, foram utilizados três displays de sete segmentos, conforme se vê na Figura 3.17, ligado às saídas do bloco Registradores através dos conversores de código BCD-sete-segmentos (CI's tipo 7448).

3.2.4 - Comparadores e Indicadores Luminosos

A função dos comparadores é efetuar a verificação dos níveis de saída do CIT, comparando-os com os níveis de saída do CIR. Qualquer pino do CIT pode ser terminal de saída. Deve existir, então, um circuito de comparação para todos os pares de pinos (pino do CIT e seu correspondente do CIR). Num teste de CI, são comparados não somente os níveis de saída do CIT com os do CIR, mas também os níveis

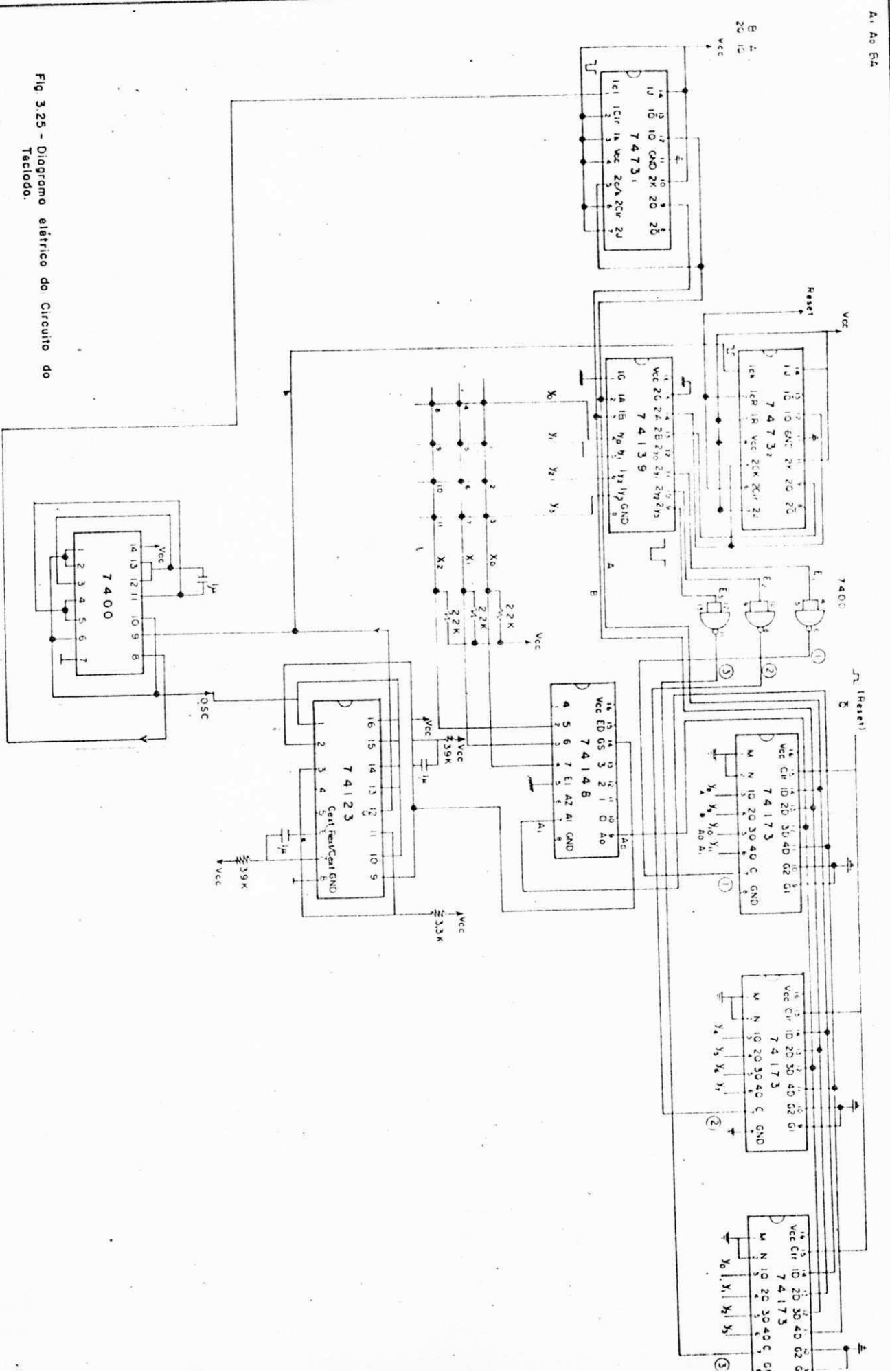


Fig 3.25 - Diagrama elétrico do Circuito do Teclado.

de sinais de entrada e de tensão de alimentação. A verifi
cação das saídas deve ser efetuada para todas as combina
ções de níveis de entrada possíveis, ou seja, se um CIT
tiver por exemplo, oito entradas, o número de comparações
deve ser $2^8 = 256$. Os resultados das comparações são arma
zenados em registradores (um flip-flop para cada pino do
CIT) e devem ser atualizados a cada mudança na entrada. O
diagrama lógico (Figura 3.26) mostra o circuito de compara
ção para um par de pinos. A comparação é efetuada utilizan
do portas OU-exclusivo de duas entradas. A atualização do
resultado da comparação é realizada através do OU-lógico en
tre o resultado já armazenado (em flip-flop) e o novo re
sultado da comparação. Isto porque, se existir alguma di
ferença entre os níveis de saída do CIR e do CIT, para al
guma combinação de entrada, este resultado deve ser cons
tantemente lembrado mesmo que os níveis de saída estejam
corretos para as combinações de entrada restantes.

Os resultados das comparações entre os níveis de saí
da do CIT e do CIR devem ser mostrados continuamente. Para
isto são utilizados diodos emissores de luz (LEDs), confor
me mostra o diagrama da Figura 3.26, alimentados pelos
flip-flops (observe-se que, a cada par de pinos do CIT e
do CIR, corresponde um LED). Quando existe diferença entre
o nível lógico de um pino de saída do CIT e o nível lógico
do correspondente no CIR, o LED correspondente a esse par
de pinos acende e permanece aceso até ser completado o tes
te.

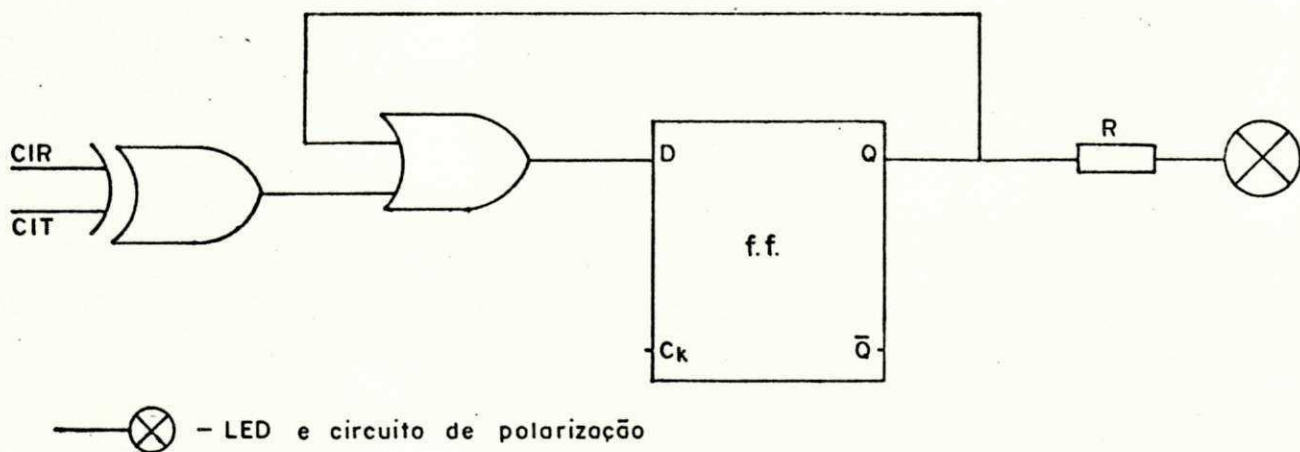


Fig. 3.26 Diagrama lógico do Circuito de Comparação para um par de pinos

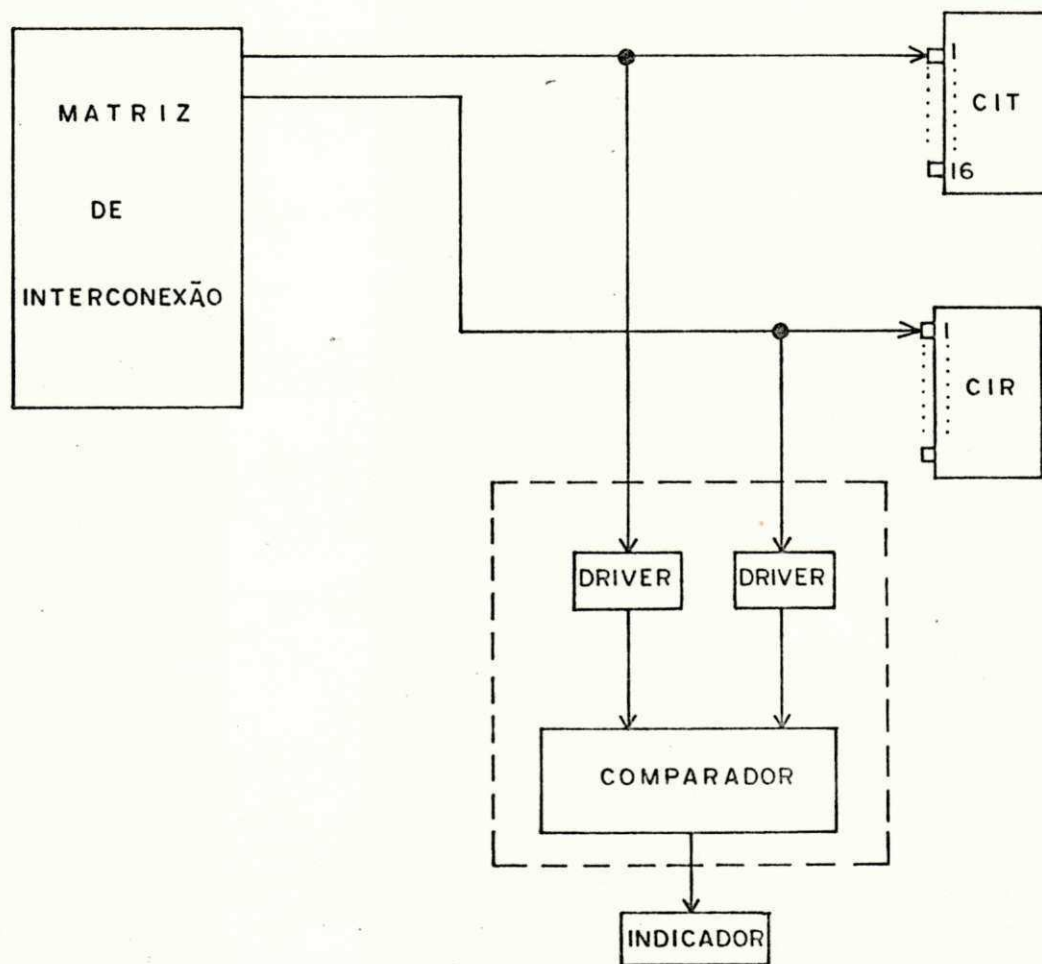


Fig. 3.27 Diagrama de blocos do Circuito de Comparação com Driver

Este sistema foi implementado com CI's TTL. Se o CI for da família CMOS, não poderá alimentar diretamente o comparador, já que este é da família TTL. Foi utilizado, então, um circuito de interface ("driver") entre os CI's de teste e de referência e o circuito comparador (V. Figura 3.27).

O diagrama elétrico do circuito comparador para todos os pinos é mostrado na Figura 3.28. Os "drivers" utilizados foram CI's tipo 4050. São mostradas as portas OU-EXCLUSIVO (CI tipo 7486), as portas OU (CI tipo 7432) e os registradores (CI tipo 74198).

Os registradores são atualizados com uma frequência de 0,5 MHz que é maior que a maior frequência de sinal de entrada, de modo a armazenar os resultados das comparações para cada mudança de sinal de entrada. Os registradores são ativados depois que as conexões adequadas tenham sido efetuadas pela Matriz de Interconexão. Isto pode ser feito utilizando um circuito temporizador (tipo monoestável) ligado à chave de teste.

3.2.5 - Gerador de Sinais de Entrada e de Sincronismo

O gerador de sinais de entrada e de sincronismo ou simplesmente **Gerador de Entradas** deve gerar todas as combi

nações possíveis de sinais de entrada. O número máximo de entradas (visto no ítem 3.2.3.2) é doze. Para implementar este circuito foi utilizado um contador binário simples de 16 estágios. Doze saídas do contador são os sinais de entrada para o CIT e CIR e as saídas restantes são utilizadas para efetuar o sincronismo na transmissão de informações entre o Gerador de Códigos e a Matriz de Interconexão e entre o Circuito Comparador e os Indicadores. O diagrama lógico do gerador de sinais de entrada é mostrado na Figura 3.29 e o diagrama elétrico na Figura 3.30.

3.3 - Funcionamento do Sistema

3.3.1 - Operação do Sistema

A operação do sistema é realizada seguindo os seguintes procedimentos:

- I) O CIT e o CIR são inseridos nos respectivos soquetes (tipo ZIF)
- II) Inicialmente o sistema é zerado, através do circuito de "reset".
- III) Em seguida, o operador fornece o número do CIT

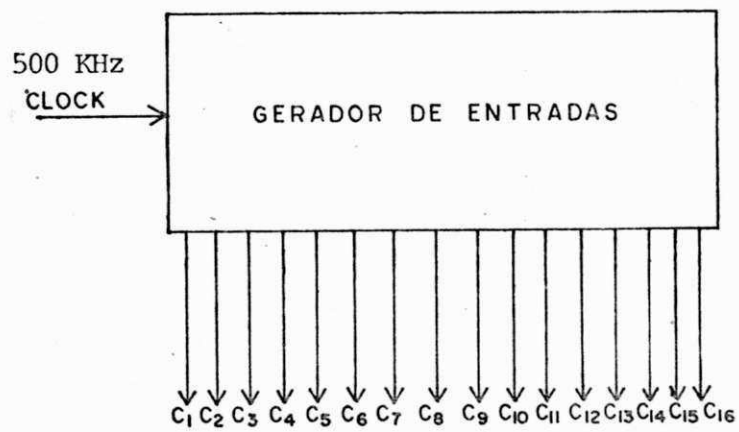


Fig. 3.29 Diagrama de blocos do Gerador de Entradas

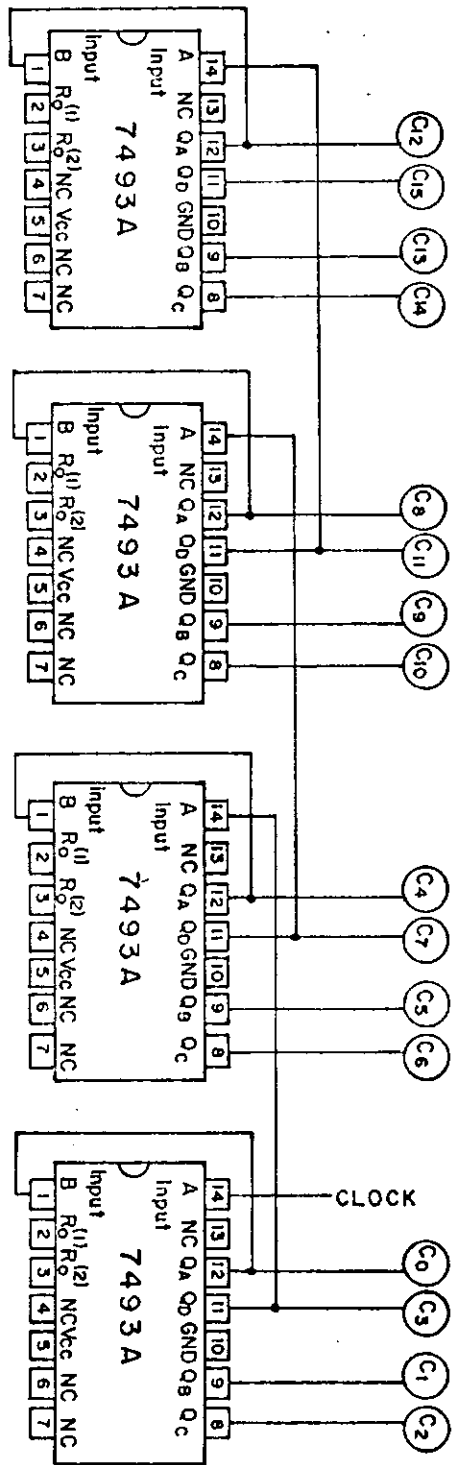


Fig. 3.30 Diagrama elétrico do Gerador de Entradas

posicionando adequadamente a chave TTL/CMOS (implementada por uma chave comum de dois pólos ligada a um flip-flop (CI 7476) da maneira mostrada na Figura 3.17) e teclando logo após os três dígitos que identificam o CIT. Este número fica armazenado no teclado e é disponível na entrada do gerador de códigos, mais precisamente, na entrada da tabela de endereços. Esta tabela coloca o endereço da palavra código, correspondente ao CIT, na entrada da tabela de códigos.

- IV) O operador, então, aciona a chave de teste (V. ítem 3.2.2) (implementada também com uma chave de 2 posições ligada a um flip-flop como mostrado na Figura 3.17) para armazenar a palavra código nos "buffers". Estes, por sua vez, fornecem à Matriz de Interconexão a palavra-código armazenada e esta efetua as conexões adequadas.
- V) Os resultados das comparações são armazenados e mostrados somente quando todas as conexões adequadas tenham sido efetuadas, ou seja, quando o sistema estiver estabilizado.

3.3.2 - Tempo de Resposta

O tempo de resposta do sistema, ou seja, o tempo para ser efetuado todo o teste em um CI, depende do gerador de níveis lógicos de entrada. O maior tempo de teste é o necessário para gerar as 2^{12} combinações possíveis de níveis lógicos de entrada.

A menor frequência de entrada (f_{\min}) é:

$$f_{\min} = \frac{2\text{MHz}}{2^{15}} \approx 61 \text{ Hz}$$

O maior período de sinal de entrada (T_{\max}) é então:

$$T_{\max} = \frac{1}{f_{\min}} = \frac{1}{61} \approx 16 \text{ ms}$$

3.3.3 - Confiabilidade do Sistema

Uma das medidas estabelecidas para dar uma medida da confiabilidade do sistema implementado [5] é o tempo médio entre falhas - MTBF (Mean Time Between Failures).

A taxa de falha é definida como $\frac{1}{\text{MTBF}}$.

Se a taxa de falha de cada componente for conhecida é possível calcular a taxa de falha de todo o sistema como a somatória das taxas de falha de cada componente.

COMPONENTES DO SISTEMA	QUANTIDADE	TAXA DE FALHA %/1000 hrs	TAXA DE FALHA P/ COMPONENTE
Capacitor	05	0,02	0,10
Resistor	100	0,002	0,20
CI	80	0,15	1,2
Transistor	70	0,04	2,8
Junção de "Wire-wrap"	8.000	0,00002	0,16

TABELA 3.3 - Taxas de falha dos componentes do sistema.

A tabela 3.3 apresenta as taxas de falha dos componentes utilizados no sistema, incluindo as junções "wire-wrap".

A taxa de falha total do sistema, isto é, a soma das taxas de falha de todos os componentes é 4,46%/1.000 hrs. Portanto, o tempo médio entre falhas (MTBF) é :

$$\text{MTBF} = 22.421 \text{ horas}$$

Estes resultados significam que, se 100 sistemas idênticos estiverem funcionando continuamente, é provável

que após 1.000 horas 5 (4,46%) dos sistemas tenham falha do.

3.3.4 - Testes Efetuados

A capacidade do sistema de identificar defeitos foi comprovada simulando falhas através da utilização, como CIT e CIR, de circuitos integrados, em bom estado, com configurações idênticas, porém com funções lógicas diferentes.

Foram efetuados testes numa série de CI's de diferentes tipos e defeitos diversos. Em todos os casos de falha, o sistema indicou os pinos de saída aos quais correspondiam os defeitos. O sistema foi capaz ainda de indicar casos de curto circuito em terminais de entrada.

4 CONCLUSÕES

O desempenho do testador semi-automático de CI's implementado é bom, já que permite identificar defeitos na maioria dos CI's da família TTL/CMOS, através da indicação dos pinos correspondentes aos níveis lógicos de saída incorretos e ainda dos pinos de entrada em curto-circuito.

O tempo de resposta do sistema é de 16ms, sendo, por tanto, bastante satisfatório neste caso em que o equipamento é acionado diretamente pelo usuário.

A confiabilidade do sistema é apenas razoável, tendo em vista o elevado número de transistores e de CI's utilizados.

O custo do sistema é considerado baixo, pois, em sua implementação, foram utilizados os CI's mais populares da família TTL (com exceção dos "drivers" tipo 4050), além de outros componentes de baixo custo (transistores, resistores, etc).

A principal limitação do sistema desenvolvido é o fato de utilizar lógica fixa, não tendo, portanto, flexibilidade de mudança de programação dos testes. Os testes em circuitos sequenciais, por exemplo, são incompletos, já que o sistema pode testar uma única sequência de sinais de entrada.

Utilizando as modernas técnicas de integração, é possível melhorar a confiabilidade do sistema através de sua compactação em um número bem menor de circuitos integrados.

Por outro lado, o sistema pode ser implementado com microprocessadores, os quais, além de melhorar a confiabilidade do sistema por se tratar de um tipo de CI com um grande número de funções já compactadas, permitem, principalmente, flexibilidade de mudanças de programação. Consequentemente, um sistema que utilize microprocessadores tem outras vantagens, como possibilidade de efetuar testes de parâmetros DC com programação dinâmica, testes mais completos de circuitos sequenciais, além de manutenção mais fácil.

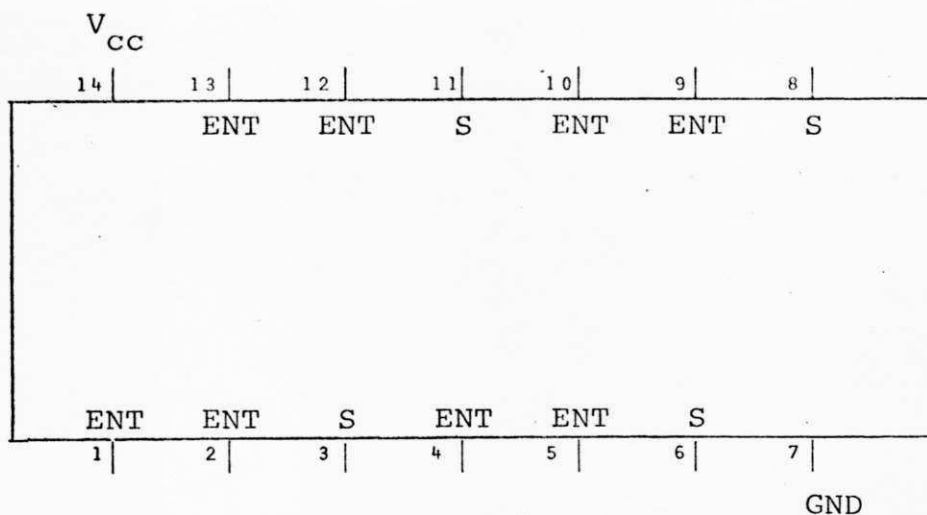
APÊNDICE A

CONFIGURAÇÃO DOS CI's

A partir do estudo das configurações de CI's variados constata-se que diferentes tipos de CI's podem ter a mesma configuração.

São apresentados aqui, algumas configurações e os CI's correspondentes.

Configuração 1

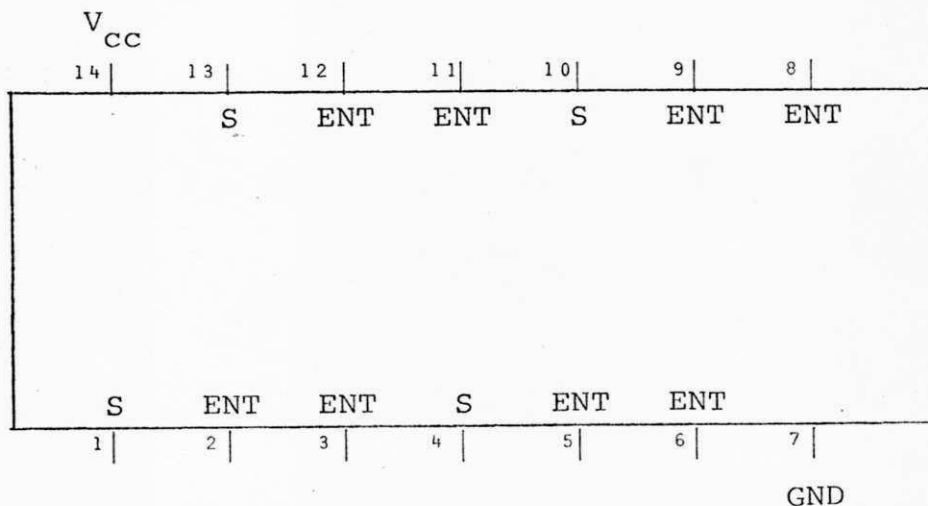


CI's

TTL -74XXX 14 pinos

00, H00*, S00, LS00, H01*, 03, S03, LS03, LS26, 37, LS37
 37, LS38, 132, S132, LS132
 08, LS08, 09, LS09

Configuração 2



CI's

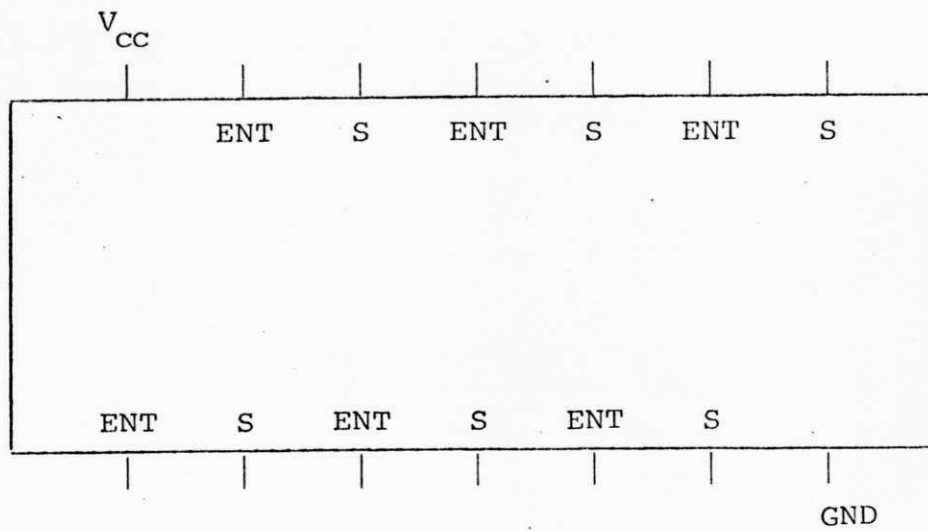
TTL - 74XXX 14 pinos

01, 39

02, L02, LS03, S02, 28, LS28, 33, LS33

14 pinos

Configuração 3



CI's

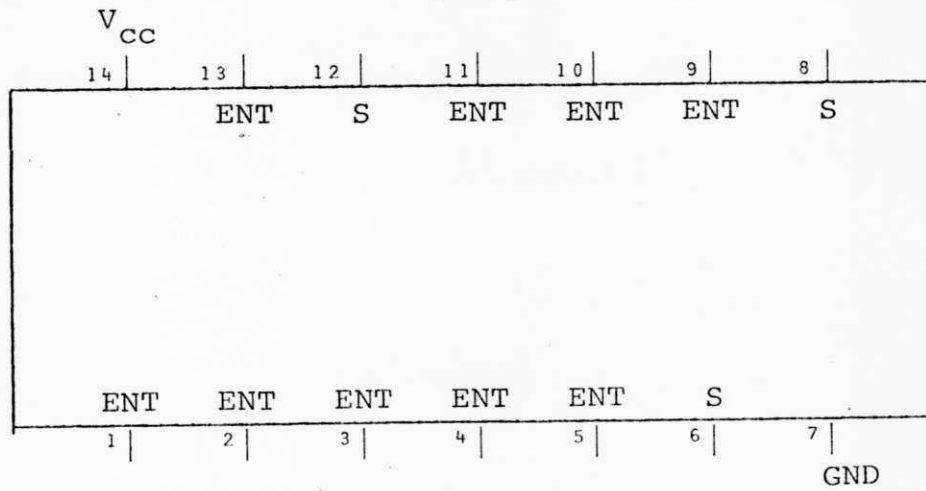
TTL 74XXX

04, H04*, L04, LS04, S04, 07, 17, 05, H05*, LS05, S05

CA*

06, 14, LS14, 16

Configuração 4



CI's

TTL 74XXX

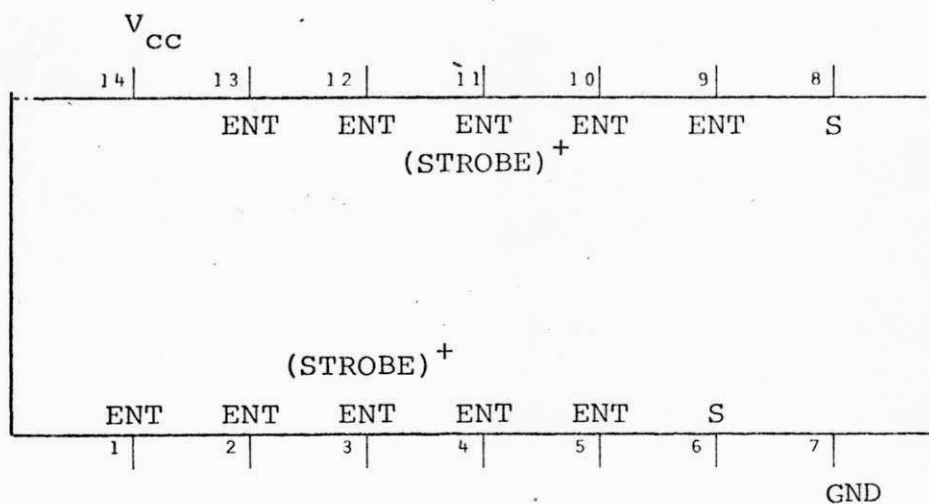
10, H10*, L10, LS10, S10, H11*, LS11, S11, 12, H15*, LS15

S15

27, LS27

* CA - coletor aberto

14 pinos **Configuração 7**



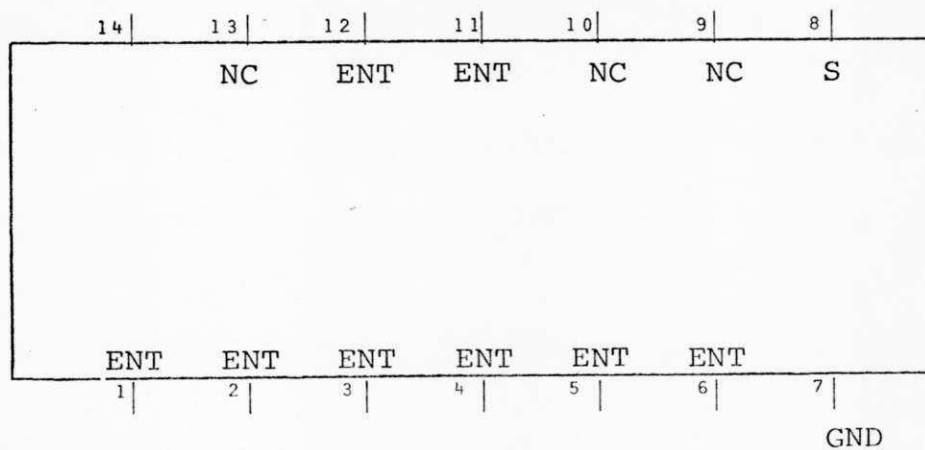
CI'S

TTL 74XXX

25,

Configuração 8

14 pinos

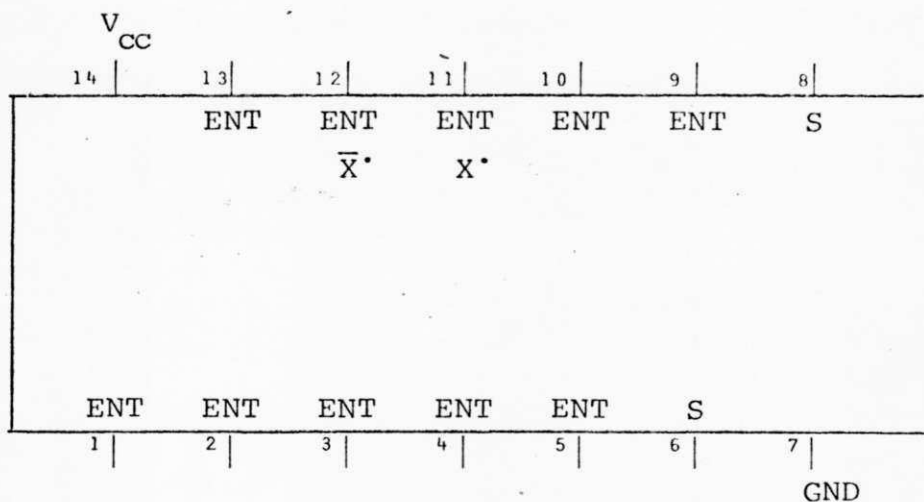


CI's

TTL 74XXX

30, H^{*}30, L30, LS30, S30,

Configuração 9



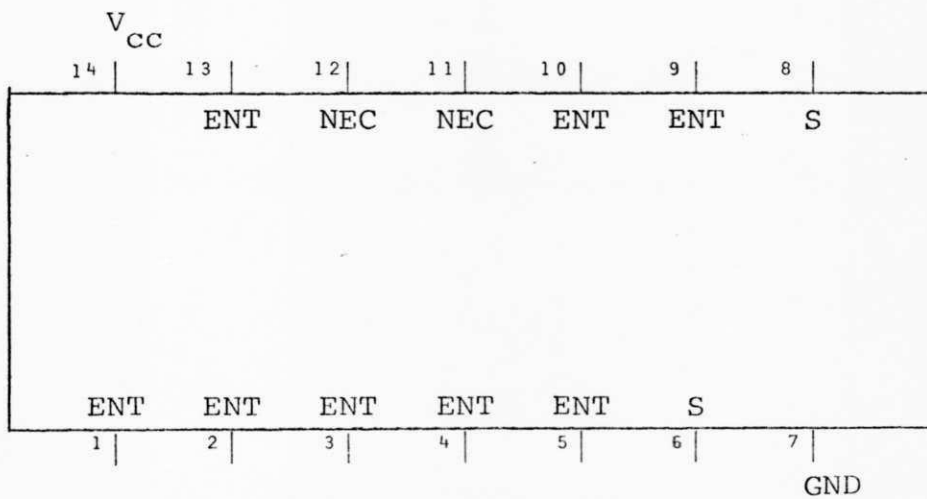
CI's

TTL 74XXX

S0, H50,

Configuração 10

14 pinos

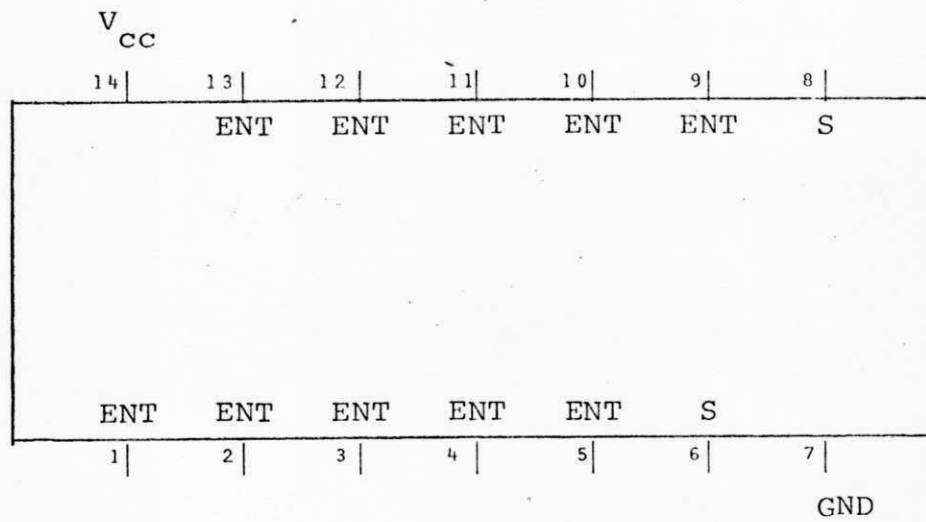


CI's

TTL 74XXX

51, H51, S51

Configuração 11

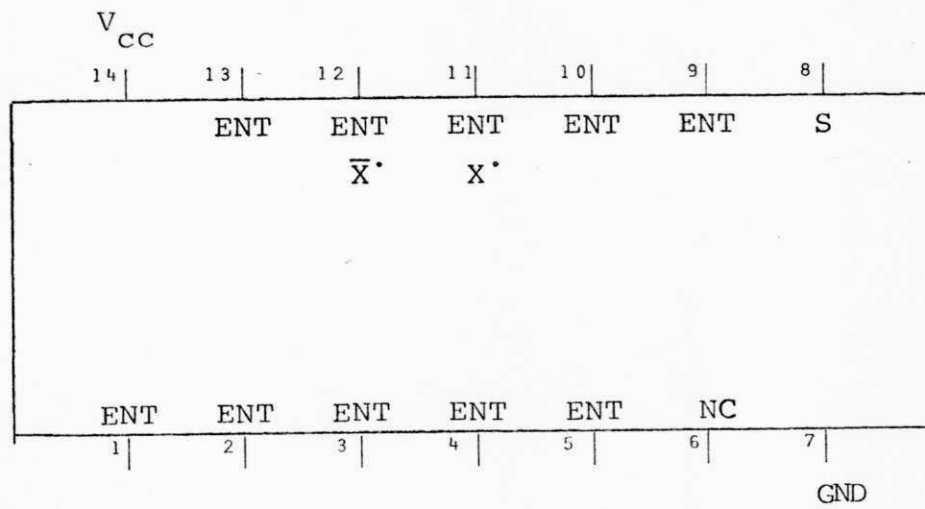


CI's

TTL 74XXX

L51, LS51

Configuração 12

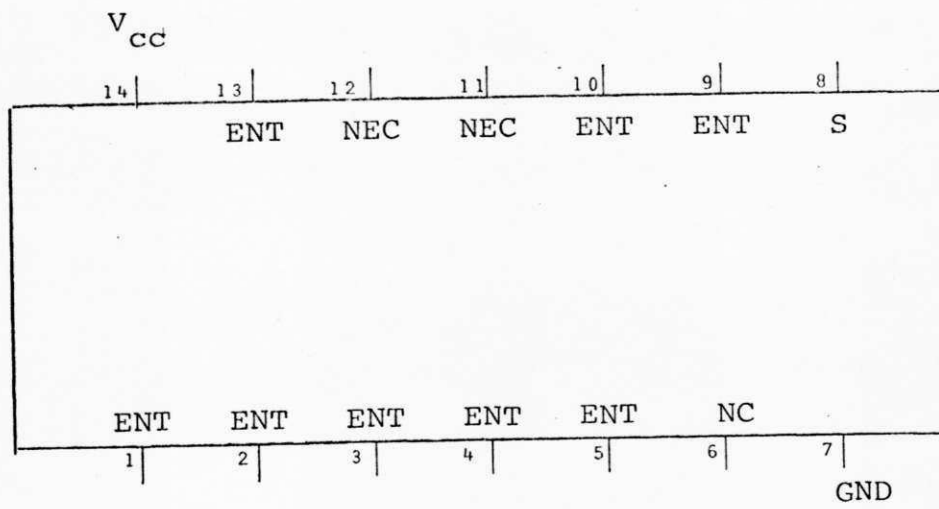


CI's

TTL 74XXX

53,

Configuração 13

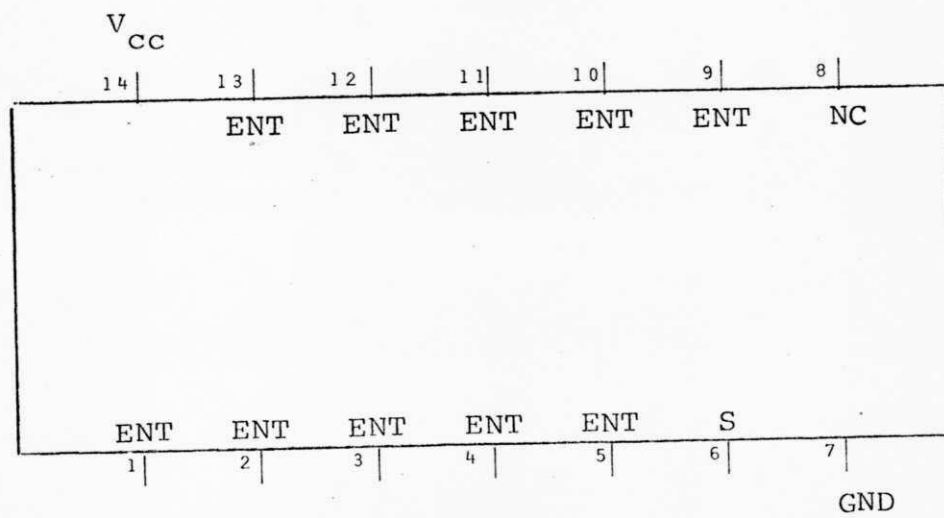


CI's

TTL 74XXX

54

Configuração 14



CI's

TTL 74XXX

LS54, L54

APÊNDICE B

CI's EXISTENTES NO MERCADO

Os números que identificam os CI's TTL, da série 7400, atualmente existentes no mercado, estão na faixa 7400 a 74799.

Uma pesquisa feita nos manuais de CI's TTL e CMOS de alguns fabricantes mais populares, mostrou que nesta faixa (7400 a 74799) muitos CI's não são mais fabricados ou disponíveis comercialmente.

Na relação seguinte são fornecidos os números de identificação que não aparecem nos manuais TTL, na faixa entre 7400 e 74180: 7418, 7419, 7424, 7429, 7431, 7435, 7436, 7441, 7456, 7457, 7458, 7459, 7463, 7466, 7467, 7468, 7469, 74104, 74105, 74109, 74115, 74117, 74118, 74119, 74124, 74127, 74129, 74130, 74131, 74137, 74138.

APÊNDICE C

COMBINAÇÕES DE PINOS PARA ALIMENTAÇÃO

São apresentados aqui, em forma de tabelas, várias combinações de pinos para alimentação (V_{CC} , V_{DD} , V_{CS} , GND) e os CI's correspondentes.

TABELA C.1 - CIRCUITOS INTEGRADOS DIGITAIS - TTL

Nº DE PINOS	V _{CC}	GND	CI's
14	14	7	00, 01*, 02, 03*, 04, 05*, 06*, 07* , 08, 09*, 10, 11, 12*, 13, 14, 15* , 16*, 17*, 20, 21, 22*, 25, 26, 27 , 28, 30, 32, 33*, 37, 38*, 40, 50 , 51, L51, LS51, 52, 53, H53, 54 , H54, L54, LS54, 55, L55, LS55, 60, 61, 62, 64, 65, 70, H71, L71, 72, 74, H78, H101, H102, 107, H108 , 110, LS113, S113, LS114, S114 , 121, L121, 122, L122, 125, 126 , 128, 132, L132, S140, S260, 80 , 86, LS86, S86, L86, H87, 95A , LS95A, 136, LS136, 164, L164...
14	4	10	^m 81A
14	5	10	90A, L90, 92A, 93A, 91A, L91
14	4	11	73, L78, LS78, H103, 82, L93, L95

* - coletor aberto

m - memória

TABELA C.1 - (continuação)

Nº DE PINOS	V _{CC} ' GND	CI'S
16	16 8	23, 109, LS109, 111, LS112, S112, 123, L123, S133, S134, 279, 42A, L42, 43A, L43, 44A, L44, 45, 46A, L46, 47A, L47, 48, 85, S85, L85, 88A ^m , 82 ^m , 97, L98, 120, S135, LS138, S138, LS139, S139, 142, 145, 147, 148, 151A, LS151, S151, 153, LS153, S153, LS155, 155, 156, 157, S157, S158, 160, 161, 162, 163, 165, 166, 167, 170, 173, 174 ...
16	5 12	75, L75, 83A, 84A ^m , LS83, 96, L86, 94, 141
16	5 13	76, H106, L99

TABELA C.2 - CIRCUITOS INTEGRADOS DIGITAIS - CMOS

Série - CD40

Nº DE PINOS	V _{DD}	V _{SS}	CI's
14	14	7	00A, 00B, 00UB, 01A, 01B, 01UB, 02A, 02B, 02UB, 06A, 06B,...
14	3	14	4045A, 45B
14	14	7	V _{CC} 1 4037A

TABELA C.3 - CIRCUITOS INTEGRADOS DIGITAIS - CMOS

Séries - 40, 45, 47
22

Nº DE PINOS	V _{DD}	V _{SS}	V _{CC}	CI's
16	16	8	-
16	16	8	1	109B
16	-	8	1	4049A, 49UB
16	16	8	7	51B, 52B, 53B, 54B, 55B, 56B
8	8	4	-	107BE

APÊNDICE D

PROGRAMAÇÃO DA TABELA DE CÓDIGOS E TABELA DE ENDEREÇOS

D.1 - Tabela de Códigos

A Tabela de Códigos deve conter as palavras-código que representam as configurações dos CI's.

Apresentamos aqui as palavras-código correspondentes às configurações mostradas no Apêndice A, baseadas na associação de códigos da Tabela 3.1.

Os endereços das locações, onde estão armazenadas as palavras-código na Tabela de Códigos, são armazenados na Tabela de Endereços mostrada mais adiante, neste apêndice.

Exemplo de como é feita a codificação de uma configuração de CI.

CONFIGURAÇÃO 1

PINOS	FINALIDADE	CÓDIGO HEXADECIMAL	CÓDIGO BINÁRIO
P1	Entrada	04	0100
P2	Entrada	05	0101
P3	Saída	03	0011
P4	Entrada	06	0110
P5	Entrada	07	0111
P6	Saída	03	0011
P7	Terra	00	0000
P8	Saída	03	0011
P9	Entrada	04	0100
P10	Entrada	09	1001
P11	Saída	03	0011
P12	Entrada	A	1010
P13	Entrada	B	1011
P14	VCC	02	0010
P15	-	-	XXXX
P16	-	-	XXXX

A palavra código de 64 bits, correspondente à configuração 1, é formada como se segue (em hexadecimal)

Código Hexadecimal	Pinos P ₁₆ P ₁₅		P ₁₄ P ₁₃		P ₁₂ P ₁₁		P ₁₀ P ₉		P ₈ P ₇		P ₆ P ₅		P ₄ P ₃		P ₂ P ₁	
		X	X	2	B	A	3	9	4	3	0	3	7	6	3	5

As restantes configurações são codificadas de maneira análoga.

A Tabela D1 apresenta as palavras-código correspondentes às várias configurações mostradas do Apêndice B.

A Tabela de Códigos é implementada com 4 memórias (2716) com palavras de 8 bits, em paralelo.

Para armazenar a palavra-código são usadas palavras duplas (2 x 8 bits) de cada memória. A palavra código é armazenada da seguinte maneira, nas memórias.

MEMÓRIAS	M4	M3	M2	M1
1 ^a palavra A ₀ =0	P ₈ P ₇	P ₆ P ₅	P ₄ P ₃	P ₂ P ₁
2 ^a palavra A ₀ =1	P ₁₆ P ₁₅	P ₁₄ P ₁₃	P ₁₂ P ₁₁	P ₁₀ P ₉

Exemplo:

Palavra-código correspondente à configuração 1 armazenada na Tabela de Códigos.

Memória Endereço (hexa)	M ₄	M ₃	M ₂	M ₁
0000	3 0	3 7	6 3	5 4
0001	X X	2 B	A 3	9 8

D.2 - Tabela de Endereços

A Tabela de Endereços deve conter os endereços onde estão armazenadas as palavras-código, correspondentes às configurações dos CI's na Tabela de Códigos.

A cada tipo de CI corresponde uma locação na Tabela de Endereços.

O conteúdo desta locação é o endereço (na Tabela de Códigos) da palavra-código correspondente aquele tipo de CI. Para se ter acesso a este endereço, na saída da Tabela de Endereços, é bastante fornecer na entrada da mesma Tabela o número de identificação do tipo de CI.

TABELA D.1 - PALAVRAS-CÓDIGO CORRESPONDENTES ÀS CONFIGURAÇÕES DOS CI's
MOSTRADOS NO APÊNDICE A

Configurações	PALAVRAS-CÓDIGO															
	Pinos	P ₁₆	P ₁₅	P ₁₄	P ₁₃	P ₁₂	P ₁₁	P ₁₀	P ₉	P ₈	P ₇	P ₆	P ₅	P ₄	P ₃	P ₂
01	X	X	2	B	A	3	9	8	3	0	3	7	6	3	5	4
02	X	X	2	3	B	A	3	9	8	0	7	6	3	5	4	3
03	X	X	2	9	3	8	3	7	3	0	3	6	3	5	3	4
04	X	X	2	C	3	B	A	9	3	0	3	8	7	6	5	4
05	X	X	2	B	A	3	9	8	3	0	3	7	6	3	5	4
06	2	D	C	B	2	A	9	3	0	3	8	7	2	6	5	4
07	X	X	2	B	A	2	9	8	3	0	3	7	6	2	5	4
08	X	X	2	3	B	A	3	3	3	0	9	8	7	6	5	4
09	X	X	2	D	C	B	A	9	3	0	3	8	7	6	5	4
10	X	X	2	D	C	B	A	9	3	0	3	8	7	6	5	4
11	X	X	2	D	C	B	A	9	3	0	3	8	7	6	5	4
12	X	X	2	E	D	C	B	A	3	0	9	8	7	6	5	4
13	X	X	2	C	3	3	B	A	3	0	9	8	7	6	5	4
14	X	X	2	E	D	C	B	A	9	0	3	8	7	6	5	4

TABELA D.2 - CONTEÚDO DA TABELA DE ENDEREÇOS E DA TABELA DE
CÓDIGOS

TABELA DE ENDEREÇOS 2708		TABELA DE CÓDIGOS 2716	
Endereço (Nº do CI) (74)	Conteúdo (Endereço da pa lavra código)	Endereço (Incluindo A ₀)	Conteúdo (Configuração)
00	00	00 01	Conf. 1
01	01	02 03	Conf. 2
02	01	02 03	Conf. 2
03	00	00 01	Conf. 1
04	02	04 05	Conf. 3
05	02	04 05	Conf. 3
06	02	04 05	-
07	02	04 05	Conf. 3
08	00	00 01	Conf. 1
09	00	00 01	Conf. 1
10	03	06 07	Conf. 4
11	03	06 07	Conf. 4

Exemplo: O CI 7400 tem a palavra-código correspondente à sua configuração armazenada nas locações 000 e 001 na Tabela de Códigos. Estes endereços estão armazenados na Tabela de Endereços como sendo 00 (a linha A₀ referencia a 1^a e a 2^a parte da palavra).

REFERÊNCIAS BIBLIOGRÁFICAS

01. BREUER, Melvin A. & FRIEDMAN, Arthur D. Diagnosis & Reliable Design of Digital Systems California , USA, Computer Science Press, Inc., 1976, 308 p.
02. LALIOTS, Theodore A. & BRUMETT, Tom. A Microprocessor Controller DIC Test System. Computer Magazine . Microprocessors Design and Applications. USA , 8, (10),: 60-7, Oct., 1975.
03. ELECTRONICS ENGINEERING SERIES. "Eletronics, Testing & Measurement". Hampshire, UK, William F. Waller . The Macmillan Press Limited, 1972.
04. FRIEDMAN, Arthur D. & MENON, Premachandran R. "Fault De_utection in Digital Circuits" New Jersey, USA , Bell Telephone Laboratories, Incorporated, 1971. 220 p.
05. LESEA, Austin & ZAKS, Rodnay, "Microprocessor Interfa_ucing Techniques" 2. ed. USA, Sybex Inc., 1977 . 413 p.

BIBLIOGRAFIA CONSULTADA

01. John R. ed. Projetos com circuitos Integrados TTL. Texas Instruments Incorporated, 1971. Rio de Janeiro. Editora Guanabara Dois, 1978. 325p.
02. 1975 RCA SOLID STATE DATA BOOKS. Linear Integrated Circuits. USA, RCA Corporation, Jan, 1975.
03. FAIRCHILD CAMERA AND INSTRUMENT CORPORATION. Full Line Condensed Catalog. California, 1978. 15-08 p.
04. RCA SOLID STATE. RCA COS/MOS Integrated Circuits. USA, RCA Corporation, Sept., 1980.
05. NATIONAL SEMICONDUCTOR. CMOS Data book USA. National Semiconductor Corporation, 1978. 7-7 p.

ABSTRACT

The present dissertation is the design and implementation of a semiautomatic digital integrated circuit tester for detection of logical faults by applying functional test.