ELETRÔ NICA DIGITAL

Alcantaro Corrêa
Presidente da FIESC

Sérgio Roberto Arruda
Diretor Regional do SENAI/SC

Antônio José Carradore
Diretor de Educação e Tecnologia do SENAI/SC

Marco Antônio Dociatti
Diretor de Desenvolvimento Organizacional do SENAI/SC

FIESC
CIESC
SESI
SENAI
IEL

Sistema Federação das Indústrias
do Estado de Santa Catarina

ELETRÔNICA DIGITAL

Florianópolis – 2004

Equipe Técnica:

Autor:
João Roberto Lorenzett
Projeto Gráfico:
Rafael Viana Silva
Capa:
Rafael Viana Silva
Samay Milet Freitas

S474e

SENAI. SC. Eletrônica digital.
Florianópolis: SENAI/SC, 2004. 63 p.

1. Eletrônica Digital. 2. Eletrônica Analógica. 3. Sistema Numérico.
I. Título

CDU: 621.22

Serviço Nacional de Aprendizagem Industrial
Departamento Regional de Santa Catarina
www.sc.senai.br

Rodovia Admar Gonzaga, 2765 – Itacorubi.
CEP 88034-001 - Florianópolis - SC
Fone: (048) 231-4221
Fax: (048) 231-4331

Este material faz parte do Programa SENAI SC de Recursos Didáticos
www.sc.senai.br/recursosdidaticos

LISTA DE FIGURAS

1 Noções Básicas.....11

1.1 Sinais analógicos.....11
1.2 Sinais digitais.....11
1.3 Conversão de números decimais em números binários.....13
1.4 Conversão de números decimais em números binários.....13

2 Variáveis e Funções Lógicas.....17

2.1 Função inversão (NOT)/Porta lógica NOT.....19
2.2 Função AND/Porta lógica AND.....19
2.3 Função OR/Porta lógica OR.....19
2.4 Outras portas lógicas.....19
2.5 Limitador Schmitt (Schmitt Trigger).....20
2.6 Saída em coletor aberto.....22
2.7 Saída tri-state.....22
2.8 Exemplo genérico de circuito que implementa a soma padrão de produto...24
2.9 Exemplo genérico de circuito que implementa o produto padrão de somas.25
2.10 Figura representariva de mapas de karnaugh.....26
2.11 Primeira etapa para solução do circuito, utilizando mapas de Karnaugh.....27
2.12 Segunda e terceira etapas para solução do circuito.....28

3 Circuitos Combinacionais.....29

3.1 Decodificador com 2 bits de entrada e 4 saída.....29
3.2 Circuito integrado 74138.....30
3.3 Codificador com 4 entradas e 4 bits de saída.....30
3.4 Diagrama para conversor de código.....31
3.5 Representação de multiplexador (MUX).....32
3.6 Implementação de circuito MUX com portas NOT, AND e OR.....32
3.7 CI multiplexador 74151.....33
3.8 Circuito multiplexador com coletor aberto.....34
3.9 Implementação de demultiplexador de 4 linhas.....34
3.10 Diagrama de blocos mostrando circuitos MUX e DEMUX no compartilhamento de linhas de transmissão.....35

4 Latch's e Flip-Flop's.....36

4.1 Latch com duas portas inversoras.....36
4.2 Latch SR (representação e tabela-verdade).....37
4.3 Representação latch SR.....37
4.4 Circuito anti-trepidação.....38
4.5 Latch controlado (com entrada enable/clock).....38
4.6 Diagrama de tempo do latch da figura 4.5.....39
4.7 Flip-flop mestre-escravo.....40
4.8 Circuito flip-flop tipo JK e sua tabela-verdade.....41
4.9 Diagrama de tempos mostrando ruído detectado por flip-flop tipo JK ou SR.....41
4.10 Diagrama de tempos comparativos entre flip-flop's metre-escravo com e sem data-lock-out.....42
4.11 Flip-flop mestre-escravo modificado (Flip-flop tipo D).....42

4.12 Modos de habilitação de flip-flop's.....43
4.13 Tempo de set-up.....43
4.14 Tempo de manutenção.....43

5 Registradores e Contadores.....44
5.1 Registrador simples para palavra de 4 bits.....44
5.2 Registrador de deslocamento 4 bits, com flip-flop´s tipo JK.....44
5.3 Contador síncrono, com configuração em anel e módulo 4.....46
5.4 Contador binário de 4 bits, síncrono.....46
5.5 Contador por pulsação.....47

6 Circuitos Aritméticos.....49
6.1 Circuito para soma e carry.....49
6.2 Circuito para subtração e carry (meio subtrator).....50
6.3 Circuito somador paralelo.....51
6.4 Somador série.....52
6.5 Circuito subtrador somador.....53

7 Memórias.....54
7.1 Diagrama de uma memória hipotética de 8 palavras de 6 bits.....54
7.2 Ligação de memórias em paralelo para aumentar número de bits por palavra..56
7.3 Ligação de memórias em paralelo para aumentar número de palavras.....56

8 Conversores D/A e A/D.....58
8.1 Conversor D/A.....58
8.2 Conversor D/A tipo somador.....58
8.3 Conversor D/A tipo R-2R.....59
8.4 Conversor A/D simultâneo ou conversor flash.....60
8.5 Conversor A/D de contagem crescente.....60
8.6 Sinal analógico e sua representação digital em um conversor A/D de contagem crescente.....61
8.7 Circuito conversor A/D de rastreamento e sinal analógico e sua representação digital em um conversor deste tipo.....61
8.8 Exemplo da utilização de conversor A/D e D/A em um sistema de controle de temperatura.....62
8.9 Circuito de amostra e retenção (sample-and-hold).....62

LISTA DE TABELAS

1 Noções Básicas.....11
1.1 Sistemas de Numeração.....14
1.2 Código Gray.....15

2 Variáveis e Funções Lógicas.....17
2.1 Tabelas-Verdade da Função Lógica.....17
2.2 Função de Duas Variáveis Lógicas.....18
2.3 Vantagens e Desvantagens das Famílias Lógicas.....21
2.4 Características das Famílias CMOS e TTL.....21
2.5 Operação OR.....23
2.6 Operação AND (E).....23
2.7 Operação Complemento (NOT ou NÃO).....23
2.8 Teoremas de Uma Variável.....23
2.9 Teoremas de Duas e Três Variáveis.....24
2.10 Obtenção de Soma Padrão de Produto a partir de Tabela-Verdade.....25
2.11 Obtenção de Produto Padrão de Somas a partir de Tabela-Verdade.....26
2.12 Tabela-Verdade para Circuito Hipotético.....27

3 Circuitos Combinacionais.....29
3.1 Saída EO e GS do Circuito Integrado 74148.....31

4 Latch's e Flip-Flop's.....36
4.1 Tabela-Verdade Latch SR com Portas NAND.....37

5 Registradores e Contadores.....44
5.1 Direção de Contagem Conforme Ligação e Tipo de Chaveamento em um Contador por Pulsação.....47
5.2 Entradas e Saídas do CI Contador 7490.....47

6 Circuitos Aritméticos.....48
6.1 Tabela-Verdade para Meio Somador.....48
6.2 Tabela-Verdade para Meio Subtrator.....50
6.3 Tabela-Verdade para Soma de 3 Bits.....50
6.4 Tabela-Verdade para Subtração de 3 Bits.....51
6.5 Cojunto de Funções da ULA 74181.....53

SUMÁRIO

1 Noções Básicas.....11
1.1 Eletrônica Analógica e Digital.....11
1.2 Sistemas de Numeração.....12
1.2.1 Sistema Decimal.....12
1.2.2 Sistema Binário.....12
1.2.3 Transformação de sistemas de numeração binário em decimal e vice-versa.....12
1.2.4 Sistema Octal.....14
1.2.5 Sistema Hexadecimal.....14
1.3 Codificação.....15
1.3.1 Código BCD.....15
1.3.2 Código Gray.....15
1.3.3Código Alfanumérico ASCII.....16

2 Variáveis e Funções Lógicas.....17
2.1 Variáveis Lógicas.....17
2.2 Funções Lógicas.....17
2.2.1 Funções de uma Variável Lógica.....17
2.2.2 Funções de duas Variáveis Lógicas.....18
2.3 Implementação de Sistemas Lógicos.....18
2.3.1 Função Inversão (NOT) / Porta Lógica NOT.....18
2.3.2 Função AND / Porta Lógica AND.....19
2.3.3 Função OR / Porta Lógica OR.....19
2.3.4 Outras Portas Lógicas.....19
2.3.5 Limitador Schmitt.....19
2.4 Famílias Lógicas.....20
2.5 Algebra de Boole.....22
2.5.1 Postulados Básicos.....23
2.5.2 Teoremas.....23
2.6 Formas Padrão de Funções Lógicas.....24
2.6.1 Soma Padrão de Produtos.....24
2.6.2 Produto Padrão de Somas.....25
2.7 Mapas de Karnaugh.....26

3 Cicuitos Combinacionais.....29
3.1 Decodificadores.....29
3.2 Codificadores.....30
3.2.1 Codificador com Prioridade.....31
3.3 Conversores de Códigos.....31
3.3.1 Decodificadores / Drives.....32
3.4 Multiplexadores (MUX).....32
3.4.1 Multiplexadores como Gerador de Função.....33
3.4.2 Multiplexação com Coletor Aberto.....33
3.4.3 Multiplexação com Saída Tri-State.....34
3.5 Demultiplexadores (DEMUX).....34
3.5.1 Mux e Demux em Comunicação.....35

3.6 Comparadores Digitais....35
3.7 Geração e Verificação de Bit de Paridade....35

4 Latch's e Flip-Flop's....36

4.1 Latch's....36
4.1.1 Latch SR....36
4.1.2 Chave sem Trepidação....37
4.1.3 Latch's Controlados....38
4.1.4 Sincronismo (Clocking)....39
4.2 Flip-Flop's....39
4.2.1 Flip-Flop Mestre-Escravo....39
4.2.2 Entradas Diretas....40
4.2.3 Flip-Flop JK....40
4.2.4 Flip-Flop Tipo T....41
4.2.5 Flip-Flop JK Gatilhado pela Borda....41
4.2.6 Flip-Flop's Mestre-Escravo com Data-Lock-Out....42
4.2.7 Flip-Flop Tipo D....42
4.2.8 Parâmetros dos Flip-Flop's....43

5 Registradores e Contadores....44

5.1 Registradores....44
5.1.1 Registradores de Deslocamento (Shift Register´s)....44
5.1.2 Formato Série e Paralelo....45
5.2 Contadores....45
5.2.1 Contadores Síncronos....45
5.2.1.1 Contador em Anel....45
5.2.1.2 Contador em Anel Torcido....46
5.2.1.3 Contador Síncrono em Código Binário....46
5.2.1.4 Contador Síncrono de Módulo Arbitrário....46
5.2.2 Contadores Assíncronos....47
5.2.2.1 Contadores por Pulsação (Ripple Counters)....47
5.2.3 Circuitos Integrados Contadores....47

6 Circuitos Aritméticos....49

6.1 Meio Somador....49
6.2 Meio Subtrador....49
6.3 Somador Inteiro....50
6.4 Subtrador Inteiro....51
6.5 Somador Paralelo....51
6.6 Somador Série....51
6.7 Representação em Complemento de Dois....52
6.8 Circuito Subtrador Somador....52
6.9 ULA (Unidade Lógica e Aritmética)....53

7 Memórias....54

7.1 Memória RAM (Memória de Acesso Aleatório)....54
7.2 Memórias ROM....55
7.2.1 Memórias ROM Programáveis (Prom's)....55

7.2.2 Memórias ROM Programáveis e Apagáveis (Eprom's).....55
7.3 Ligação de Memórias em Paralelo.....56
7.4 Memórias Série.....57

8 Conversores D/A e A/D.....58

8.1 Conversores Digitais/Analógicos (D/A).....58
8.1.1 Conversor D/A do Tipo Somador.....58
8.1.2 Conversor D/A Tipo R-2R.....59
8.2 Conversores Análogicos/Digitais (A/D).....59
8.2.1 Conversor A/D Simultâneo ou Conversor Flash.....59
8.2.2 Conversores de Contagem Crescente.....60
8.2.3 Conversores de Rastreamento.....61
8.2.4 Circuito de Amostra e Retenção (Sample-and-Hold).....62

Referências Bibliográficas.....63

CAPÍTULO 1
NOÇÕES BÁSICAS

1.1 Eletrônica Analógica e Digital

A diferença entre eletrônica analógica e digital é devida ao tipo de sinal processado.

O sinal analógico tem como principal característica a de que ele não tem descontinuidades no seu valor, ou seja, não varia bruscamente no tempo. Normalmente um circuito analógico responde a múltiplos níveis de tensão. A figura abaixo apresenta um sinal analógico variando continuamente no tempo (corrente alternada) e um sinal sem variação no tempo (corrente contínua).

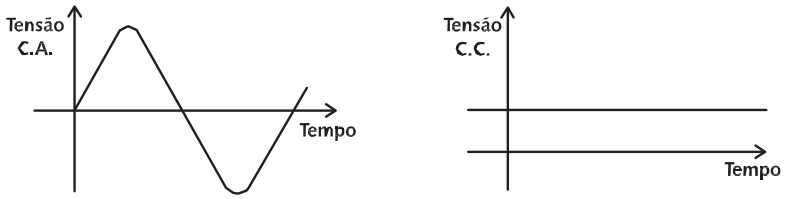


FIGURA 1.1: Sinais Analógicos

Já o sinal digital apresenta variações descontínuas no tempo, ou seja, normalmente o sinal varia bruscamente entre níveis definidos e conhecidos. Os circuitos digitais baseiam-se na representação de números (dígitos) binários; Portanto, normalmente respondem a apenas dois níveis de tensão, representativos destes números. Os gráficos abaixo demonstram dois sinais digitais: O primeiro varia entre 0 e 5 V e o segundo entre -5 e +5 V. Observe que o sinal não mantém-se entre os dois níveis por tempos que sejam consideráveis.

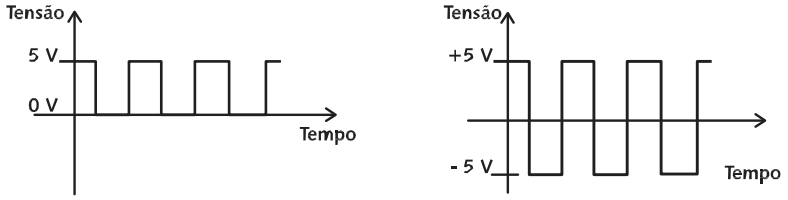


FIGURA 1.2: Sinais Digitais

Os circuitos analógicos e os digitais tem a mesma finalidade, qual seja: processar os sinais de entrada e fornecer sinais de saída. O que varia de um para outro é a filosofia de funcionamento. Cada tipo tem suas vantagens e desvantagens. Atualmente, os circuitos digitais tem avançado em áreas antes dominadas por dispositivos analógicos (como áudio e vídeo, por ex.), avan-

ço este proporcionado pelo aumento do poder de processamento do circuitos integrados.

1.2 Sistemas de Numeração

1.2.1 Sistema Decimal

Entre os sistemas numéricos existentes, o sistema decimal é o mais utilizado. Os símbolos ou dígitos empregados são os algarismos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Embora o sistema decimal possua somente dez símbolos, qualquer número acima disso pode ser expresso usando o sistema de peso por posicionamento, conforme o exemplo abaixo:

O decimal 3546, pode também ser escrito da seguinte forma:
$3 \times 10^3 + 5 \times 10^2 + 4 \times 10^1 + 6 \times 10^0$
$3000 + 500 + 40 + 6 = 3546$

O decimal 798, pode também ser escrito da seguinte forma :
$7 \times 10^2 + 9 \times 10^1 + 8 \times 10^0$
$700 + 90 + 8 = 798$

Dependendo do seu posicionamento, o dígito terá um peso. Quanto mais próximo da extrema esquerda do número estiver o dígito maior será a potência de dez que estará multiplicando o mesmo, ou seja, mais significativo será o dígito.

1.2.2 Sistema Binário

É o sistema de numeração mais utilizado em processamento de dados digitais, pois utiliza apenas dois algarismos (0 e 1), sendo portanto mais fácil de ser representado por circuitos eletrônicos (Os dígitos binários podem ser representados pela presença ou não de tensão).

O sistema de numeração binário utiliza a base 2. Cada posição de cada algarismo de um número binário corresponde uma potência de 2, analogamente ao sistema decimal, onde cada posição corresponde a uma potência de dez.

Exemplos de números em binário:
- 11001011001
- 1001100

Os dígitos binários chamam-se bits, proveniente da contração do inglês Binary Digit. Assim como no sistema decimal, dependendo do posicionamento, o algarismo ou bit terá um peso. O da extrema esquerda será o bit mais significativo (Most Significant Bit -MSB) e o da extrema direita o bit menos significativo (Least Significante Bit - LSB) .

Um conjunto de 8 bits é denominado byte. Um conjunto de 4 bits é denominado nybble.

1.2.3 Transformação de sistemas de numeração binário em decimal e vice-versa

É possível converter um número decimal em binário, dividindo-se o número decimal por 2 sucessivamente até obter-se o quociente 0. Toma-se a seqüência dos restos da divisão em ordem inversa e obtém-se o resultado em binário.

Exemplos:

Para converter o decimal 345 em binário, realiza-se a operação indicada abaixo:

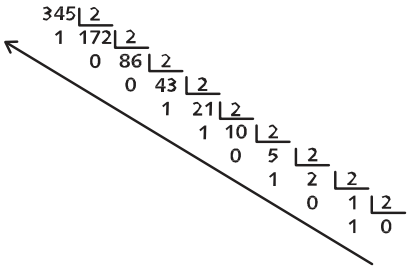


FIGURA 1.3: Conversão de Numeros Decimais em Numeros Binários

O correspondente binário do decimal 345 é, portanto, 101011001.

Para converter o decimal 47 em binário, realiza-se a operação indicada abaixo:

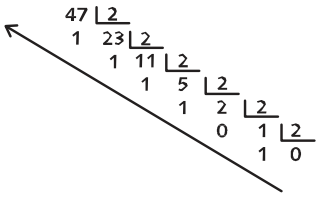


FIGURA 1.4: Conversão de Numeros Decimais em Numeros Binários

Correspondente binário do decimal 47 é, portanto, 101111.

O processo inverso, ou seja a conversão de um número binário em decimal, é também possível, multiplicando-se cada dígito binário por seu peso correspondente na base dois. A soma destes produtos resulta no equivalente decimal, conforme ilustram os exemplos abaixo:

Para transformar o número binário 1101 em decimal devemos proceder da seguinte forma:

$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$8 + 4 + 0 + 1 = 13$

O correspondente decimal do binário 1101 é, portanto, 13.

Para transformar o número binário 10111 em decimal devemos proceder da seguinte forma:

$1 \times 24 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

$16 + 0 + 4 + 2 + 1 = 23$

O correspondente decimal do binário 10111 é, portanto, 23

1.2.4 Sistema Octal

O sistema octal (base 8) é formado por oito símbolos ou dígitos : 0, 1, 2, 3, 4, 5, 6, 7 . Para representação de qualquer dígito em octal, necessitamos de três dígitos binários. Os números octais têm, portanto, um terço do comprimento de um número binário e fornecem a mesma informação. O sistema octal foi criado com o propósito de minimizar a representação de um número binário e facilitar a manipulação humana dos números.

1.2.5 Sistema Hexadecimal

O sistema hexadecimal (base 16) foi criado com o mesmo propósito do sistema octal (Minimizar a representação de um número binário). Se considerarmos quatro dígitos binários, ou seja, quatro bits, o maior número que se pode expressar com esses quatro dígitos é 1111 que é, em decimal, 15. Como 15 é o maior algarismo do sistema hexadecimal, com um único dígito hexadecimal podemos representar um conjunto de 4 bits.

O sistema hexadecimal, como o nome mesmo diz, possui 16 símbolos, de 0 a 15. Como não existem símbolos dentro do sistema arábico, que possam representar os números decimais entre 10 e 15 sem repetir os símbolos anteriores, foram usados símbolos literais: A, B, C, D, E e F, portanto, o sistema hexadecimal será formado por 16 símbolos alfanuméricos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

A tabela a seguir mostra um quadro resumo dos sistemas de numeração mais utilizados:

DECIMAL	BINÁRIO	HEXADECIMAL	OCTAL
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17
16	10000	10	20
...	...	...	...

TABELA 1.1: Sistemas de Numeração

1.3 Codificação

Codificar significa representar uma determinada informação por um cojunto de simbolos (códigos). Neste texto, "codificar" significa especificamente converter um dado ou uma informação numérica decimal ou alfabética em binário, pois os equipamentos digitais e os computadores processam a informação em binário, ao passo que as entradas e saídas desse sistema são acessadas pelo homem.

Existem diversas maneiras de realizar esta codificação (Existem diversos códigos em binário). Os principais serão vistos neste capítulo.

1.3.1 Código BCD

O código BCD (Binary Coded Decimal = Decimal Codificado em Binário) é muito utilizado em display´s, contadores, etc. Ele representa cada dígito decimal de 0 a 9 por quatro bits binários. Exemplo:

O número 360 em decimal é representado em BCD como:

3	6	0
0011	0110	0000

A vantagem do código BCD é que, na conversão de decimal para BCD precisamos examinar apenas um dígito de cada vez e na conversão de BCD para decimal examinamos apenas quatro dígitos binários (4 bits). No entanto, necessita mais dígitos do que o código binário puro.

1.3.2 Código Gray

O código Gray se caracteriza por variar apenas um bit na mudança de um número consecutivo para outro. O código Gray é às vezes referido como código 8421.

DECIMAL	BINÁRIO	GRAY
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
...	...	...

TABELA1.2: Código Gray

1.3.3 Código Alfanumérico ASCII

Neste código cada caráter alfanumérico corresponde a um número binário. É conhecido internacionalmente pelo nome de American Standard Code for Information Interchange. O código é utilizado principalmente na troca de informações ou dados entre computadores e seus sistemas periféricos, ou seja é o meio de comunicação entre os diversos sistemas. O código é apresentado em sete bits, portanto temos 128 combinações possíveis e podemos comunicar com isso até 128 caracteres, letras, números ou símbolos especiais ou de controle. Junto aos sete bits de código, é acrescentado um oitavo bit, como bit de paridade ou verificação e através dele se verifica se o que foi transmitido está correto ou não. O valor deste bit é definido pelo tipo de paridade.

CAPÍTULO 2
VARIÁVEIS E FUNÇÕES LÓGICAS

2.1 Variáveis Lógicas

Uma variável lógica é uma variável que atende os seguintes quesitos:

Só pode assumir um de dois estados possíveis (Por exemplo: aceso/apagado, alto/baixo, ligado/desligado, etc.)

Os dois valores possíveis devem ser tais que, baseados na lógica, sejam mutuamente exclusivos. (Se um circuito está ligado, não pode estar desligado).

O sistema binário é bastante adequado para a manipulação de variáveis lógicas, visto que só possui dois dígitos: 0 e 1.

2.2 Funções Lógicas

2.2.1 Funções de uma Variável Lógica

Supondo que A e B sejam variáveis lógicas e B dependa de A obedecendo a uma determinada regra chamada função lógica. Representa-se isto escrevendo B = f (A). Todas as maneiras possíveis de B variar com A estão representadas nas tabelas abaixo. Estas tabelas são chamadas tabelas-verdade da função lógica. Consideramos os dois valores possíveis para as variáveis lógicas como sendo 0 e 1.

A	B=f(A)
0	0
1	1

A	B=f(A)
0	1
1	0

A	B=f(A)
0	0
1	0

A	B=f(A)
0	1
1	1

TABELA 2.1: Tabelas-Verdade da Função Lógica

- Na primeira tabela temos a representação da função B=A
- Na segunda tabela temos a representação da função B=contrário de A
- Na terceira tabela temos a representação da função B=0
- Na quarta tabela temos a representação da função B=1

Podemos representar a função B=contrário de A como $B=\overline{A}$

Tabela verdade é uma tabela que representa todos os possíveis estados lógicos a que podem ser submetidas às entradas e saídas de um circuito digital.

2.2.2 Funções de Duas Variáveis Lógicas

Como existem 4 combinações de duas variáveis binárias, gerando quatro resultados; E estes quatro resultados podem combinar-se de 16 maneiras diferentes, temos então 16 funções possíveis para duas variáveis lógicas. A tabela abaixo apresenta as usualmente empregadas diretamente na prática:

A	0	0	1	1	
B	0	1	0	1	Função
	0	0	0	0	f=0
	0	0	0	1	f=A AND B
	0	0	1	1	f=A
	0	1	0	1	f=B
	0	1	1	0	f=A Exclusive-Or B
	0	1	1	1	f=A OR B
	1	0	0	0	f=A NOR B
	1	0	0	1	f=A Exclusive-Nor B
	1	0	1	0	f=NOT B
	1	1	0	0	f=NOT A
	1	1	1	0	f=A AND B
	1	1	1	1	f=1

TABELA 2.2: Funções de Duas Variáveis Lógicas

2.3 Implementação de Sistemas Lógicos

A implementação de sistemas lógicos em circuitos eletrônicos digitais pode ser feita considerando-se os dois estados possíveis para uma variável lógica como sendo a presença ou não de tensão em um determinado ponto do circuito, ou ainda como sendo dois níveis de tensão determinados. Por exemplo: em um determinado circuito digital os dois estados possíveis para cada ponto do circuito podem ser 0V ou 5V. Normalmente denominamos os níveis lógicos em um circuito digital de 0 e 1, ou Low e High, não importando quais os níveis de tensão que os representarão .

Os dispositivos digitais são constituídos de circuitos eletrônicos que, a partir dos níveis lógicos das entradas, fornecem as saídas de acordo com sua construção, obedecendo a uma determinada função. Estes dispositivos são denominados portas lógicas.

Porta lógica é, portanto, um circuito digital (dois estados) com uma ou mais entradas e uma saída.

2.3.1 Função Inversão (NOT) / Porta Lógica NOT

O Inversor é uma porta lógica que tem uma única entrada e cuja saída é o complemento da entrada. A notação da função e o símbolo lógico estão representados no desenho da página seguinte:

FIGURA 2.1: Função Inversão (NOT)/Porta Lógica NOT

2.3.2 Função AND / Porta Lógica AND

Na função AND, a saída tem nível lógico 1 somente quando as duas entradas tem nível lógico 1, conforme a tabela-verdade abaixo:

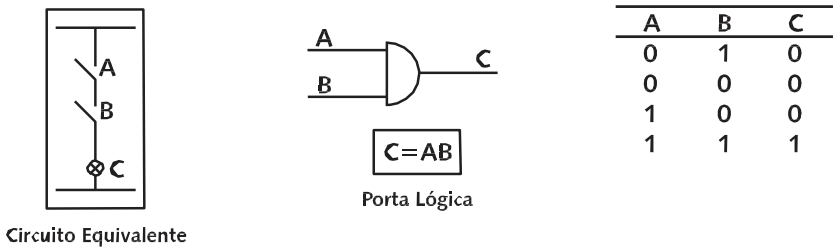


FIGURA 2.2: Função AND/Porta Lógica AND

2.3.3 Função OR / Porta Lógica OR

Na função OR, a saída C só terá nível lógico 1 se a entrada A for 1 ou a entrada B for 1 ou ambas as entradas forem 1, conforme a tabela-verdade abaixo:

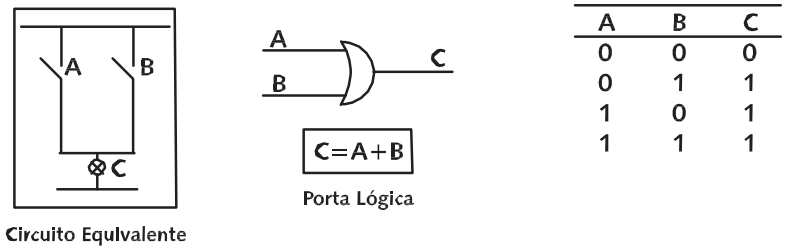


FIGURA 2.3: Função OR/Porta Lógica OR

2.3.4 Outras Portas Lógicas

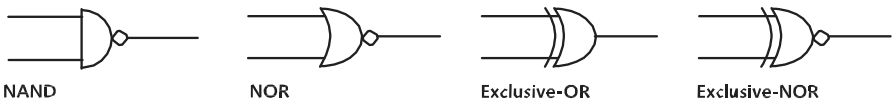


FIGURA 2.4: Outras Portas Lógicas

2.3.5 Limitador Schmitt

Um limitador Schmitt, também chamado de Schmitt Trigger é um circuito eletrônico utilizado para detectar se uma tensão ultrapassou um dado nível de referência. Ele tem dois estados estáveis e é utilizado como dispositivo de condicionamento de sinal: Se a tensão de entrada ultrapassar um determinado valor, a saída vai para nível alto. Se a tensão de entrada ficar abaixo de determinado valor, a saída vai para nível baixo.

O símbolo e um gráfico explicativo do funcionamento do schmitt trigger estão a seguir:

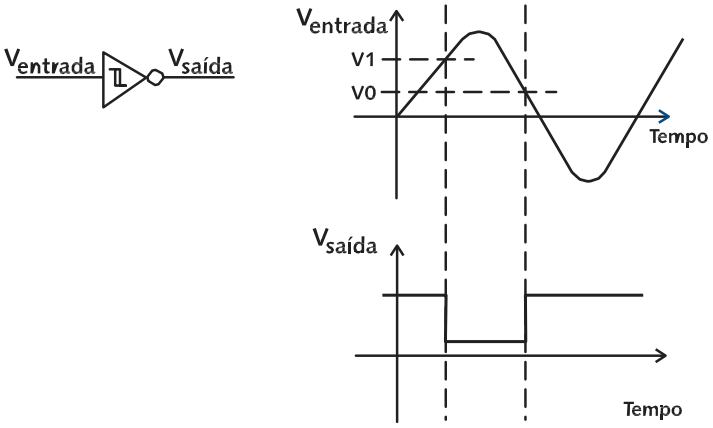


FIGURA 2.5: Limitador Schmitt (Schmitt Trigger)

2.4 Famílias Lógicas

As portas lógicas básicas dependendo da época em que foram desenvolvidas, do fabricante e da técnica, são construídas a partir de componentes discretos ou circuitos integrados.

Possuem vários tipos de construções ou montagens (implementação) tendo, portanto, vantagens ou desvantagens entre elas. Cada tipo de construção, embora possam representar a mesma função lógica, diferem quanto a fatores como: velocidade de operação, componentes empregados, potência consumida, etc. Assim, dependendo desses fatores, faz-se a opção por esta ou aquela família, segundo necessidade e interesse.

Um circuito integrado possui um tipo de integração que depende da época em que foi desenvolvido e do tamanho do circuito que estará contido nele. Assim, temos por convenção a integração em pequena escala - SSI (Small Scale Integration), que possui até 12 componentes em uma única pastilha de silício; a integração em média escala - MSI (Medium Scale Integration), com até 99 componentes; a integração em larga escala LSI (Large Scale Integration), com até 1.000 componentes em uma única pastilha; e a integração em larga escala VLSI (Very Large Scale Integration), até 100.000 componentes em uma única pastilha.

As famílias lógicas são constituídas de circuitos integrados construídos sob uma determinada técnica. As mais comuns são:

- família RTL (Resistor transistor logic);
- família DTL (Diode transistor logic);
- família TTL (Transistor transistor logic);
- família ECL (Emmiter Coupled logic);
- família IIL ou I^2L (Integrated injection logic);
- família MOS ou MOS FET (Metal oxid semiconductor, Field effect transistor);
- família CMOS (MOS complementar).

Na tabela a seguir estão resumidas as vantagens e desvantagens de cada família:

FAMÍLIA	VANTAGENS	DESVANTAGENS
MOS (PMOS ou NMOS)	• Maior densidade de componentes por chip • Menor custo por componente • Baixo consumo de potência por função	• Baixa velocidade • Confiabilidade pouco elevada
CMOS	• Consumo e potência bem reduzida, se operado em baixas frequências • Boa imunidade a ruído, considerando o item anterior • Pode operar com outras famílias lógicas	• Suscetível a descargas estáticas de eletricidade • Velocidade relativamente baixa
RTL e DTL	• Baixa dissipação	• Baixa velocidade • Suscetível a ruído
TTL	• Alta velocidade de comutação • Baixo custo • Capacidade de fan-out elevada • Compatível com outros sistemas	• Suscetível a variações de fontes de alimentação • Baixa imunidade a ruído devido ao item anterior
ECL	• Alta velocidade • Capacidade de fan-out elevada	• Compatibilidade dificultosa • Custo elevado
IIL	• Alta densidade de porta por chip • Baixo consumo de potência em alta frequência	• Pouco difundido • Custo elevado

TABELA 2.3: Vantagens e Desvantagens das Famílias Lógicas

As famílias mais utilizadas na prática são a CMOS e a TTL. As principais características destas duas famílias estão detalhadas na tabela na página seguinte:

CARACTERÍSTICAS	CMOS	TTL
Capacidade de saída (fan-out) em portas	α	10 a 20
Dissipação de potência em mW	Depende da frequência de chaveamento, mas é consideravelmente menor que a família TTL	2 a 20
Atraso de propagação por portas em ns	Depende da carga, mas é consideravelmente maior que a família TTL	3 a 30
Teste de alimentação em V	5 a 15	5
Número de série	40	74 / 54

TABELA 2.4: Características das Famílias CMOS e TTL

Obs.: Atraso propagação (delay) é o tempo requerido para a saída da porta mudar de estado após as entradas terem mudado.

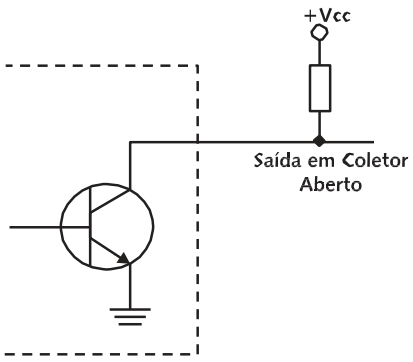


FIGURA 2.6: Saída em Coletor Aberto

Saídas em coletor aberto: A figura abaixo representa a saída de uma porta com coletor aberto. Observe que, para que o circuito funcione a contento, devemos utilizar um resistor externo ligado ao positivo da fonte de alimentação (chamado de resistor pull-up). Alguns circuitos integrados possuem saídas em coletor aberto e devem, portanto ser supridos de resistores nestas saídas.

Saída tri-state: Uma saída em tri-state está representada na figura abaixo. Este tipo de diminui o atraso provocado por capacitâncias parasitas nas portas. Observe que quando a chave S1 está fechada, a saída está em nível lógico zero; quando a chve S2 está fechada, a saída está em nível lógico um. Se nenhuma chave estiver fechada, a saída assume um terceiro estado denominado alta impedância (Hi-Z). É como se a saída ficasse desligada do resto do circuito. Quem diz se a porta está ou não em um estado de Hi-Z é um sinal de enable. Este tipo de configuração de saída é muito útil quando dois ou mais circuitos necessitam compartilhar o mesmo meio de transmissão.

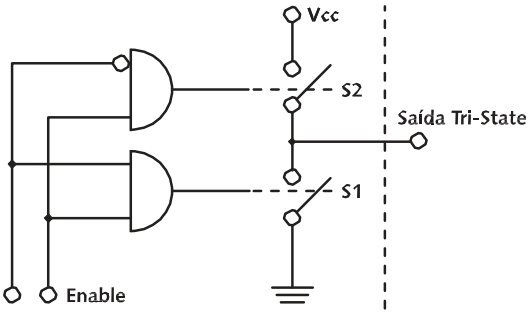


FIGURA 2.7: Saída Tri-State

2.5 Algebra de Boole

A álgebra criada por George Boole (1815 - 1864), facilita o entendimento operacional dos circuitos elétricos digitais. Partindo de premissas pré-estabelecidas, esta parte da matemática é muito importante para a determinação de circuitos lógicos, facilitando não só o entendimento operacional de tais circuitos, como também a sua simplificação e minimização.

2.5.1 Postulados Básicos

- Operação (+) também chamada de união, ou OR (OU).

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 1$

TABELA 2.5: Operação OR

- Operação (.) também chamada de interseção, ou AND (E).

$0 . 0 = 0$
$0 . 1 = 0$
$1 . 0 = 0$
$1 . 1 = 1$

TABELA 2.6: Operação AND (E)

- Operação Complemento, também chamada de inversão, ou NOT (NÃO)

$\overline{1} = 0$
$\overline{0} = 1$

TABELA 2.7: Operação Complemento (NOT ou NÃO)

2.5.2 Teoremas

Existem diversos teoremas envolvendo operações AND, OR e NOT que servem para simplificação de expressões contendo variáveis lógicas (provendo uma economia nos circuitos), e também para alteração da expressão, visando a utilização de dispositivos práticos diferentes. Usa-se os símbolos de adição para representar a função OR, o símbolo de multiplicação para representar a função AND e o sinal de barra sobre a variável para representar a função NOT. Os dois estados possíveis para as variáveis são 0 e 1.

Temos a seguir alguns teoremas de uma variável, úteis para simplificação de funções lógicas:

$\overline{\overline{A}} = A$	$A * 1 = A$
$A + 0 = 0$	$A * 0 = 0$
$A + 1 = 1$	$A * A = A$
$A + A = A$	$A * \overline{A} = 0$
$A + \overline{A} = 1$	

TABELA 2.8: Teorema de uma Variável

Temos a seguir alguns teoremas de duas e três variáveis, úteis para simplificação de funções lógicas:

$A+AB=A$
$A(A+B)=A$
$AB+A\overline{B}=A$
$(A+B)(A+\overline{B})=A$
$A+\overline{A}B=A+B$
$A(\overline{A}+B)=AB$
$A+BC=(A+B)(A+C)$
$A(B+C)=AB+AC$
$AB+\overline{A}C=(A+C)(\overline{A}+B)$
$(A+B)(\overline{A}+C)=AC+\overline{A}B$
$AB+\overline{A}C+BC=AB+\overline{A}C$
$(A+B)(\overline{A}+C)(B+C)=(A+B)(\overline{A}+C)$

TABELA 2.9: Teorema de Duas e Três Variáveis

Teorema de Morgan

$\overline{A*B*C...} = \overline{A}+\overline{B}+\overline{C}+...$
$\overline{A+B+C+...} = \overline{A}*\overline{B}*\overline{C}*$

Lei comutativa : A + B = B + A
Lei associativa : A . B = B . A
Lei distributiva : A + (B + C) = (A + B) + C

2.6 Formas Padrão de Funções Lógicas

2.6.1 Soma Padrão de Produtos

Soma padrão de produtos é uma estrutura de portas lógicas de dois níveis na qual as entradas do circuito são ligadas a portas AND, e as saídas destas portas AND são ligadas às entradas de uma porta OR. Qualquer função lógica pode ser expressa na forma de soma padrão de produtos. O desenho a seguir mostra um exemplo genérico.

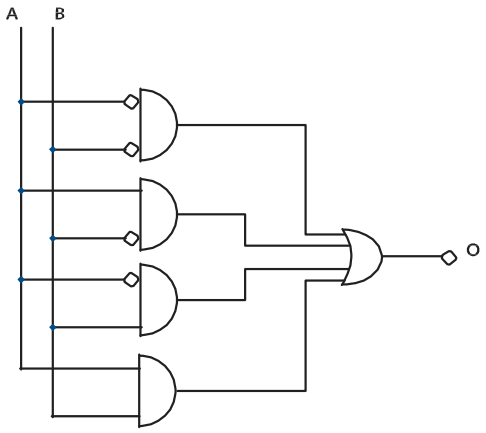


FIGURA 2.8: Exemplo Genérico de Circuito que Implementa a Soma Padrão de Produtos

Para expressar uma função como soma padrão de produtos, devemos modifica-la, se necessário, tendo em vista os teoremas e as propriedades de funções lógicas.

Para obter a soma padrão de produtos a partir de uma tabela-verdade, devemos selecionar as linhas da tabela que tiverem saída igual a um, conforme o exemplo abaixo:

ENTRADAS			SAÍDA	LINHA Nº
A	B	C	O	
0	0	0	1	0
0	0	1	0	1
0	1	0	1	2
0	1	1	1	3
1	0	0	0	4
1	0	1	0	5
1	1	0	1	6
1	1	1	1	7

$f = \Sigma\,(0,2,3,6,7)$

$f = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + \overline{A}\,BC + A\,B\,\overline{C} + ABC$

TABELA 2.10: Obtenção de Soma Padrão do Produto a partir de Tabela-Verdade

2.6.2 Produto Padrão de Somas

Produto padrão de somas é uma estrutura de portas lógicas de dois níveis na qual as entradas do circuito são ligadas a portas OR, e as saídas destas portas OR são ligadas às entradas de uma porta AND. Qualquer função lógica pode ser expressa na forma de produto padrão de somas.

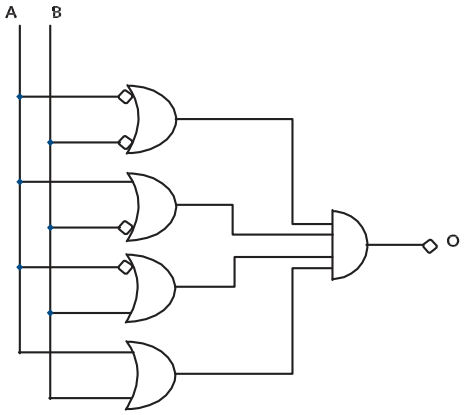


FIGURA 2.9: Exemplo Genérico de Circuito que Implementa o Produto Padrão de Somas

Para expressar uma função como produto padrão de somas, devemos modifica-la, se necessário, tendo em vista os teoremas e as propriedades de funções lógicas.

Para obter o produto padrão de somas a partir de uma tabela-verdade, devemos selecionar as linhas da tabela que tiverem saída igual a zero, invertendo as entradas, conforme o exemplo na página seguinte:

ENTRADAS			SAÍDA	LINHA Nº
A	B	C	O	
0	0	0	1	0
0	0	1	0	1
0	1	0	1	2
0	1	1	1	3
1	0	0	0	4
1	0	1	0	5
1	1	0	1	6
1	1	1	1	7

$f = \pi(1, 4, 5)$

$f = (A+B+\overline{C})(\overline{A}+B+C)(\overline{A}+B+\overline{C})$

TABELA 2.11: Obtenção de Produto Padrão do Somas a partir de Tabela-Verdade

2.7 Mapas de Karnaugh

Os mapas de Karnaugh são dispositivos úteis para simplificação e minimização de funções algébricas booleanas. Um mapa Karnaugh é uma figura geométrica como a mostrada abaixo:

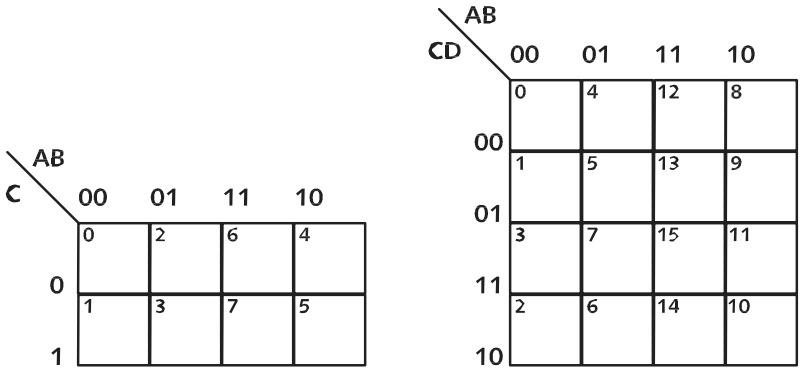


FIGURA 2.10: Figuras Representativas de Mapas de Karnaugh

A primeira figura mostra um mapa K para 3 variáveis (A,B e C) e a segunda mostra um mapa K para 4 variáveis (A,B,C e D). As vantagens do uso dos mapas K na simplificação de funções tornam-se mais evidentes nas funções com mais de 3 variáveis. É possível desenhar mapas K para qualquer número de variáveis.

No exemplo abaixo demonstramos um método simplificado de uso de um mapa K para 4 variáveis lógicas. Vamos projetar um circuito que tenha quatro entradas (A,B,C e D) e uma saída (O). Nosso circuito deve comportar-se conforme a tabela-verdade na página seguinte:

ENTRADAS				SAÍDA
A	B	C	D	O
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

TABELA 2.12: Tabela-Verdade para Circuitos Hipotéticos

Solução:

1) Desenhamos um mapa K de 4 variáveis de entrada e escrevemos o número 1 em cada quadrículo correspondente à linha da tabela verdade que tiver saída 1:

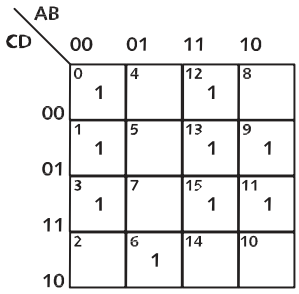


FIGURA 2.11: Primeira Etapa para Solução do Circuito, Utilizando Mapas de Karnaugh

2) Assinalamos os quadrículos com “um” que não estejam próximos a nenhum outro (são considerados próximos os quadrículos que estiverem se tocando diretamente pelas arestas ou aqueles que estiverem em posições análogas em linhas ou colunas nas extremidades opostas).

3) Assinalamos os grupos de quadrículos com “um” que estejam próximos. Os grupos de quadrículos devem ser os maiores possíveis. Cada quadrículo deve estar assinalado pelo menos uma vez. Observe a sequência dos desenhos na página seguinte:

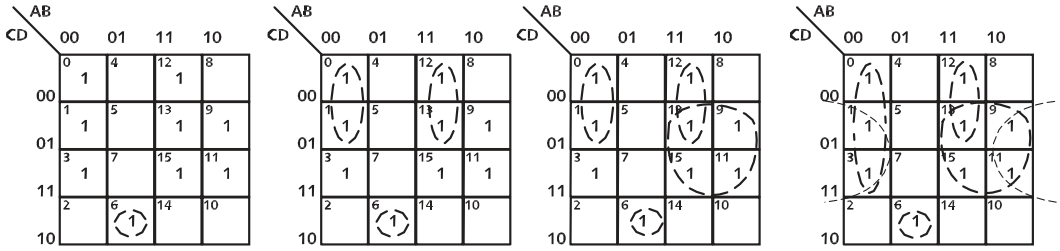


FIGURA 2.12: Segunda e Terceira Etapas para a Solução do Circuito

4) Obtemos a função simplificada que representa a tabela - verdade proposta, a partir da última versão do mapa K. Compare a expressão obtida abaixo com o último desenho da série mostrada acima

$$f(A,B,C,D) = \overline{A}BC\overline{D} + \overline{ABC} + A\overline{B}C + AD + \overline{B}D$$

CAPÍTULO 3
CIRCUITOS COMBINACIONAIS

Circuitos combinacionais são aqueles nos quais a saída é determinada pelas condições da entrada, ou seja, em um determinado momento a saída do circuito depende unicamente do estado da entrada do circuito naquele momento. Todos os circuitos digitais vistos até este ponto do curso são considerados combinacionais.

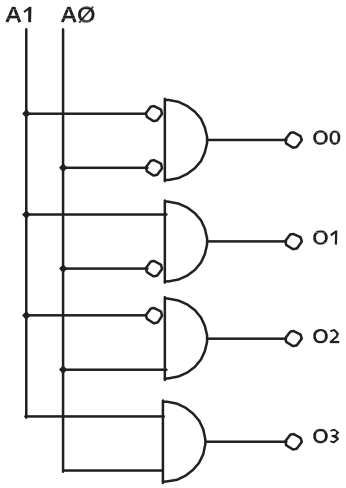
Existem alguns circuitos combinacionais clássicos, muito utilizados na prática, e que merecem ser estudados separadamente. São eles:

- decodificadores;
- codificadores;
- multiplexadores (MUX);
- demultiplexadores (DEMUX);
- comparadores.

3.1 Decodificadores

Um circuito decodificador, usualmente, é considerado aquele que, a partir de uma entrada fornecida em um código qualquer, produz uma saída em código decimal.

A figura a seguir mostra um decodificador com uma entrada em binário com dois bits e uma saída em decimal de 4 dígitos, bem como a tabela verdade do circuito:



A1	A0	O0	O1	O2	O3
0	0	1			
0	1		1		
1	0			1	
1	1				1

FIGURA 3.1: Decodificador com 2 bits de entrada e 4 saídas

Uma característica importante de um decodificador é que , para cada entrada A1,A0 existe uma e apenas uma saída habilitada (ativa), permanecendo as demais desabilitadas.

Um exemplo de CI decodificador é o 74138, que possui quatro entradas para código binário (endereço), e 8 vias de saída para código decimal. O 74138 possui ainda uma entrada de enable ativada por três entradas (duas em nível baixo e uma em nível alto).

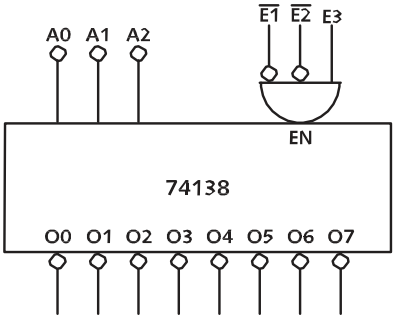
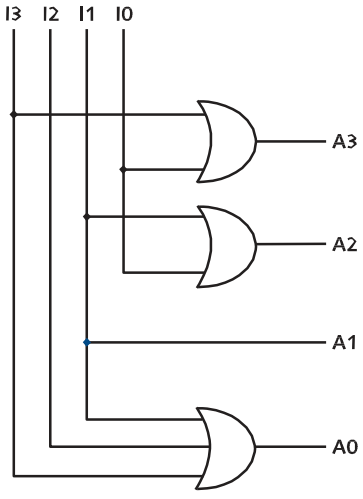


FIGURA 3.2: Circuito Integrado 74138 (Decodificador)

3.2 Codificadores

Um codificador executa a opração inversa do decodificador: Para cada linha escolhida (ativada) , uma palavra de código aparecerá nas linhas de saída. A palavra de código da saída geralmente é única para cada linha selecionada na entrada, mas não precisa ser necessariamente assim. Abaixo temos um exemplo de um possível decodificador com quatro linhas de entrada e quatro linhas de saída em um código arbitrário



I3	I2	I1	I0	A3	A2	A1	A0
1				1	0	0	1
	1			0	1	0	1
		1		0	0	1	1
			1	1	1	0	0

FIGURA 3.3: Codificador com 4 Entradas e 4 bits de Saída

3.2.1 Codificador com Prioridade

Sistemas digitais freqüentemente possuem sensores que indicam quando alguma ação é necessária. Por exemplo: Um sensor de passagem pode indicar que uma pessoa passou por uma porta, gerando um sinal digital (a passagem do nível lógico 0 para nível lógico 1 em um condutor, por ex.). Este sinal é a indicação de que algo deve ser feito (incrementar um contador, ligar um circuito, etc.), ou seja é uma solicitação de atendimento . Normalmente um codificador é utilizado para aceitar as linhas de solicitação de atendimento e apresentar o código que corresponde ao endereço do componente do sistema que faria o atendimento à solicitação. Se duas solicitações de atendimento forem feitas ao mesmo tempo, temos que ter um esquema de prioridades, ou seja, será atendida a solicitação que tiver prioridade mais alta.

Um circuito integrado comercial que implementa um codificador com prioridade é o 74148. Este possui todas as entradas e linhas de controle ativas quando baixas. O CI aceita 8 entradas (I7 a I0) e provê três saídas (A2 a A0), que podem gerar $2^3 = 8$ endereços.

Possui ainda uma entrada de habilitação (EI) e duas saídas : EO e GS.

As saídas EO e GS servem para indicar quando há uma solicitação de atendimento ou não, conforme a tabela abaixo:

EI	Solicitação de Atendimento	EO	GS
1	Irrelevante	1	1
0	Não	0	1
0	Sim	1	0

TABELA 3.1: Saídas EO e GS do Circuito Integrado 74148

3.3 Conversores de Código

Conversor de código é o circuito lógico que faz a tradução de uma informação codificada de uma determinada maneira para um código diferente. Um conversor de código pode ser construído ligando um decodificador e um codificador em cascata, conforme a figura abaixo:

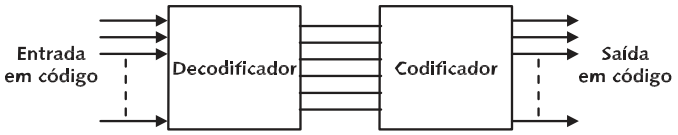


FIGURA 3.4: Diagrama para a Criação de Código

Existem conversores de código implementados em circuitos integrados disponíveis comercialmente. Um exemplo típico é o conversor de código BCD para display de 7 segmentos, no qual podemos entrar com um código BCD de 4 bits e teremos na saída um código de 7 bits adequado para acionar um display de 7 segmentos que mostre o número da entrada em forma decimal ou hexa.

Às vezes os conversores de código são denominados pelos fabricantes de integrados de decodificadores.

3.3.1 Decodificadores / Drivers

Existem alguns integrados que possuem saídas de coletor aberto com capacidades relativamente elevadas de dissipação de potência, como por exemplo os integrados 7406 (Buffers/Drivers inversores) e 7407 (Buffers/Drivers não-inversores), sendo adequados para alimentação direta de pequenas cargas como lâmpadas (da ordem de algumas dezenas de mA, em até 30 V).

Existem também CI's decodificadores da família 74 com drivers incorporados, que são chamados decodificadores / drivers. Um exemplo é o CI 74141 que é um decodificador de BCD para decimal.

3.4 Multiplexadores (MUX)

Um multiplexador é um circuito lógico digital que possui um determinado número de entradas e apenas uma saída. O circuito, através de uma determinação externa, conecta uma única entrada à saída. Um multiplexador, portanto, faz o papel de uma chave digital múltipla, como a que está representada abaixo:

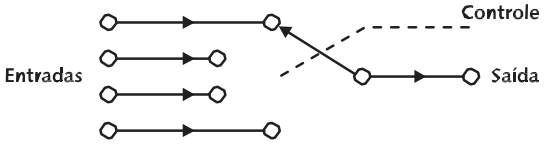


FIGURA 3.5: Representação de um Multiplexador (MUX)

Uma implementação prática de um circuito MUX é demonstrada na figura abaixo. No caso, as entradas S0 e S1 determinam, através de um código binário, qual das entradas (I0 a I3) estará conectada com a saída (O). Um circuito MUX tem sua aplicação mais óbvia no compartilhamento de uma única via de transmissão de dados por mais que um circuito, dividindo o tempo.

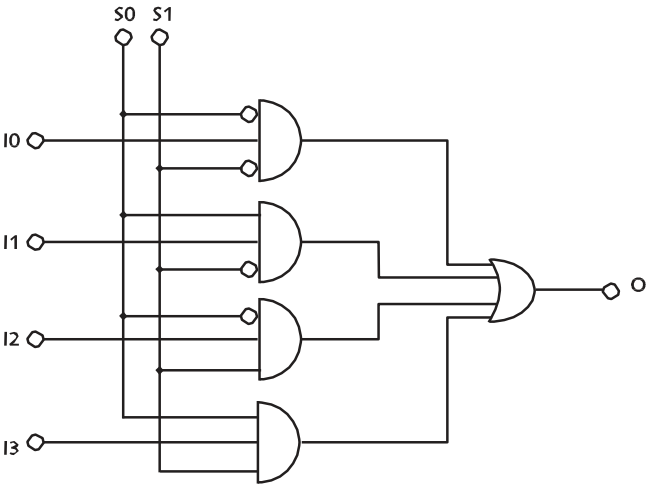


FIGURA 3.6: Implementação de Circuito MUX com Portas NOT, AND e OR

3.4.1 Multiplexador como Gerador de Funções

Um circuito MUX pode ser utilizado para gerar uma função lógica arbitrária das variáveis selecionadoras. Para isto, devemos escrever a função arbitrária na forma de soma de produtos das variáveis de seleção e ativamos as entradas correspondentes a cada parcela da soma. Por exemplo: se desejarmos criar, através de um multiplexador, um circuito que apresente na saída a função:

$$O = \overline{S1} + S1\,\overline{S0}$$

Escrevemos a função como soma de produtos das duas variáveis de seleção (S1 e S2):

$$O = \overline{S1}(S0 + \overline{S0}) + S1\,\overline{S0}$$
$$O = \overline{S1}\,S0 + \overline{S1}\,\overline{S0} + S1\,\overline{S0}$$

A primeira parcela da soma, no circuito de portas da página anterior, será gerado fazendo I1 = 1. A segunda parcela será gerada fazendo I0 = 1. A terceira parcela será gerada com I2 = 1.

Um exemplo de CI multiplexador é o 74151, que possui três entradas de seleção e 8 vias de entrada. O 74151 possui ainda uma entrada de enable ativada por nível baixo e duas saídas complementares.

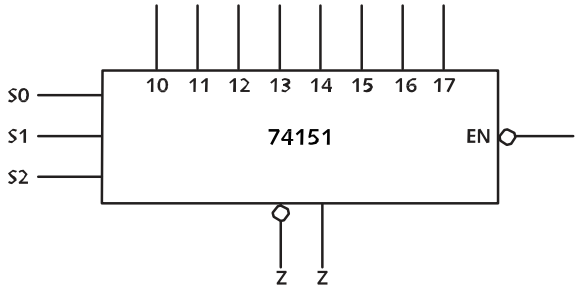


FIGURA 3.7: CI Multiplexador 74151

3.4.2 Multiplexação com Coletor Aberto

Se tivermos diversos chips com saídas de coletor aberto, e necessitamos multiplexar estas saídas, não é necessário usar um CI multiplexador adicional, pois se ligarmos todas as saídas a um ponto comum , com um resistor de pull-up (resistor ligado ao positivo da fonte de alimentação), o circuito irá funcionar perfeitamente bem, desde que apenas um integrado esteja habilitado de cada vez.

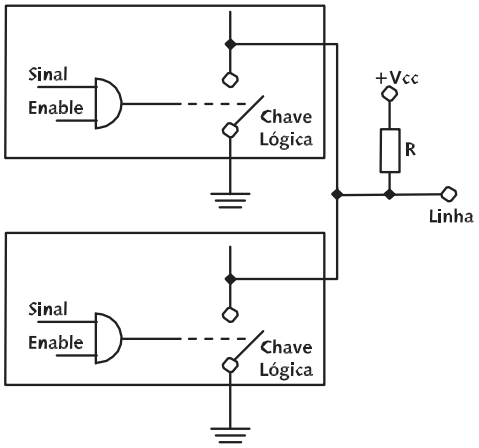


FIGURA 3.8: Circuito Multiplexador com Coletor Aberto

3.4.3 Multiplexação com Saída Tri-State

Circuitos integrados com saídas em tri-state também podem ser multiplexados sem o auxílio de um circuito especial, visto que as saídas estão desligadas (em alta impedância - HiZ) se o circuito correspondente estiver desabilitado. Basta, portanto, ligar todas as saídas a um ponto comum e habilitar um circuito de cada vez.

A linha de dados compartilhada por mais de um circuito é chamada BUS ou Barramento.

3.5 Demultiplexadores (DEMUX)

Na prática, normalmente utilizamos um BUS para muitas fontes de sinal diferentes, recorrendo à multiplexação. O Demultiplexador é um circuito que recebe um sinal do BUS e o dirige para o receptor adequado. Um demultiplexador pode ser considerado um decodificador no qual, em cada porta AND foi acrescentada uma entrada adicional que é ligada ao BUS.

Um demultiplexador de uma linha para quatro linhas é mostrado na figura abaixo:

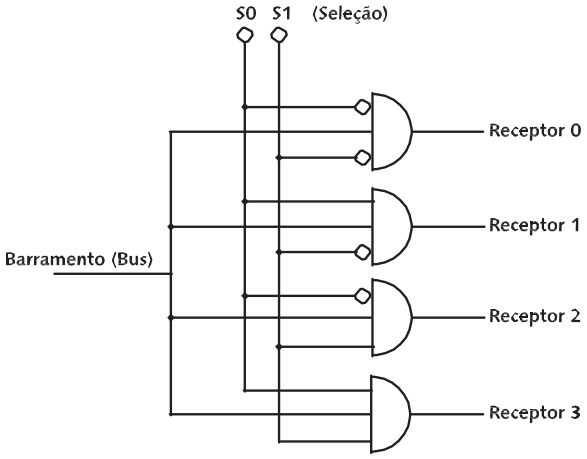


FIGURA 3.9: Implementação de Demultiplexadores de 4 Linhas

3.5.1 Mux e Demux em Comunicação

Circuitos multiplexadores e Demultiplexadores podem ser utilizados para tornal possível o compartilhamento de uma única linha de transmissão para vários emissores e receptores.

Neste caso, os usuários compartilham a linha no tempo, conforme representado no desenho abaixo. O bloco denominado "contador" e o clock do sistema serão estudados em detalhes mais à frente.

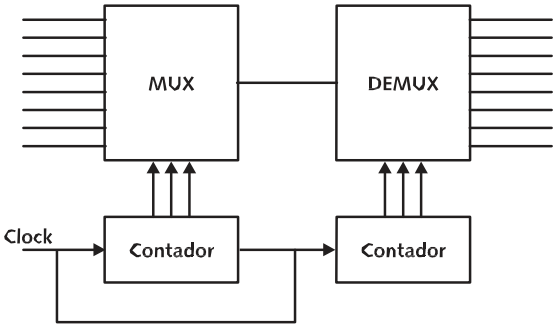


FIGURA 3.10: Diagrama de Blocos Mostrando Circuitos MUX e DEMUX no compartilhamento de linha de transmissão

3.6 Comparadores Digitais

Uma grande parte dos circuitos lógicos digitais, normalmente, utilizam circuitos especializados em comaparar dois números binários e determinar se estes são iguais ou diferentes, e qual dos dois é maior. Para o caso em que os números possuem apenas um bit, o circuito é simples e requer apenas algumas portas simples. Para números maiores o circuito torna-se mais complexo, e existem CI's dedicados para este fim.

O CI 7485 é um circuito de comparação de magnitude que possui duas entradas de quatro bits (A e B) e três saídas (A>B, A<B , A=B). Possui ainda três entradas de cascateamento para possibilitar a comparação entre números com mais que 4 bits.

3.7 Geração e Verificação de Bit de Paridade

O bit de paridade é um bit que se adiciona a uma palavra binária, para diminuir o risco de perda de integridade de dados em uma transmissão, por exemplo. Existem dois tipos de paridade: Par e ímpar. Na paridade par, o bit adicionado à palavra deve ser tal que o total de bits "1" seja par. Na paridade ímpar, o bit adicionado à palavra deve ser tal que o total de bits "1" seja ímpar.

Na recepção da palavra, existe um circuito para verificação da paridade. Se alguma falha na transmissão ocorreu, provavelmente será detectada. O CI 74180 é um gerador e verificador de paridade, e pode verificar uma palavra de 9 bits, gerando o 10º bit de paridade. O 74180 possui 8 entradas para a palavra a ser verificada (A a H), 2 entradas (ODD e EVEN) para o 9º bit ou para ligação em cascata, e 2 saídas (□odd e □even) e que indicam a paridade das entradas.

CAPÍTULO 4
LATCH'S AND FLIP-FLOP'S

4.1 Latch's

Um latch é um circuito lógico que, sem intervenção externa, permanece indefinidamente em um determinado estado (High ou low, 1 ou 0). Ao contrário dos circuitos combinacionais, o latch não depende unicamente das suas entradas para determinar suas saídas, visto que na ausência de entradas ele ainda mantém um nível lógico definido.

No desenho abaixo está representado um latch construído com duas portas inversoras:

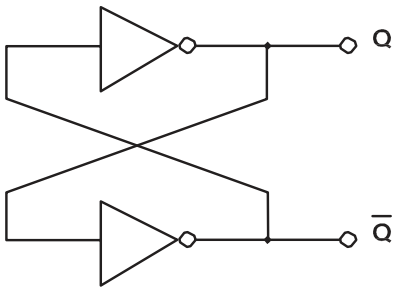


FIGURA 4.1: Latch com Duas Portas Inversoras

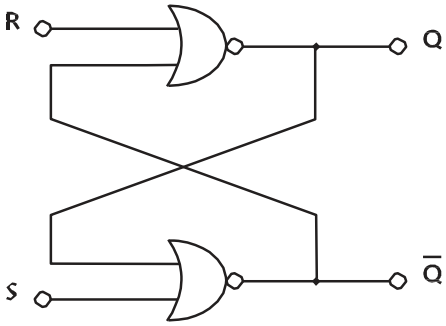
Um latch pode ser usado para armazenar , ou seja registrar ou ainda memorizar um
valor 0 ou 1 (chamado bit).

Um conjunto de 8 latch's pode ser considerado um registrador de palavras de 8 bits onde "escrevemos" um valor ou dado.

Os dois estados possíveis de um latch são chamados de set (quando Q=1) e reset (quando Q=0). O estado reset é chamado às vezes de "clear", ou seja, limpo. Para levar o latch da figura acima para o estado setado ou resetado, basta forçar a entrada Q para o nível lógico 1 ou 0, respectivamente.

41.1 Latch SR

Podemos construir um latch utilizando portas NOR, conforme desenho a seguir, obtendo assim duas entradas chamadas entradas de controle.O comportamento do circuito está resumido na tabela verdade a seguir:



S	R	Q	Q/
0	0	x	x
0	1	0	1
1	0	1	0
1	1	não utilizada	

Obs: x = mantém estado anterior

FIGURA 4.2: Latch SR (Representação e Tabela-Verdade)

O mesmo circuito que está representado acima, se for implementado com portas NAND, obedecerá a tabela abaixo:

S	R	Q	Q/
0	0	não utilizada	
0	1	1	0
1	0	0	1
1	1	x	x

Obs: x = mantém estado anterior

TABELA 4.1: Tabela-Verdade Latch SR com Portas NAND

O símbolo do latch SR está representado no desenho abaixo:

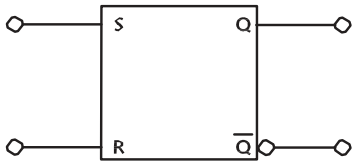


FIGURA 4.3: Representação Latch SR

O circuito integrado 74279 possui quatro latchs SR implementados com portas NAND, sendo que dois deles tem as entradas S e R normais e dois deles tem três entradas :

S1, S2 e R (para serem setados necessitam de sinais em duas entradas).

4.1.2 Chave sem Trepidação

Uma característica das chaves mecânicas de liga/desliga é que o contato trepida (bouces) diversas vezes antes de repousar em um estado determinado. Isto pode causar problemas em um circuito digital, já que a abertura

ou fechamento de uma chave não irá causar a simples alteração de nível lógico 1 para 0 ou vice-versa, e sim irá gerar um trem de pulsos geralmente indesejável. Um circuito como o representado no desenho abaixo elimina o efeito da trepidação da chave, já que, se a chave trepidar, as entradas do latch permanecerão em nível lógico 0, mantendo as saídas com seu valor inalterado.

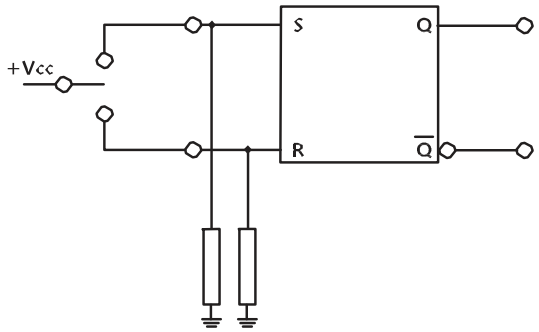


FIGURA 4.4: Circuito Anti-Trepidação

4.1.3 Latch's Controlados

Um latch controlado ou dinâmico nada mais é do que um latch que possui uma entrada adicional de habilita (Enable), que recebe um sinal de controle (strobe) habilitando o latch a registrar o dado presente em suas entradas (R e S). Um exemplo de latch deste tipo está representado no desenho abaixo.

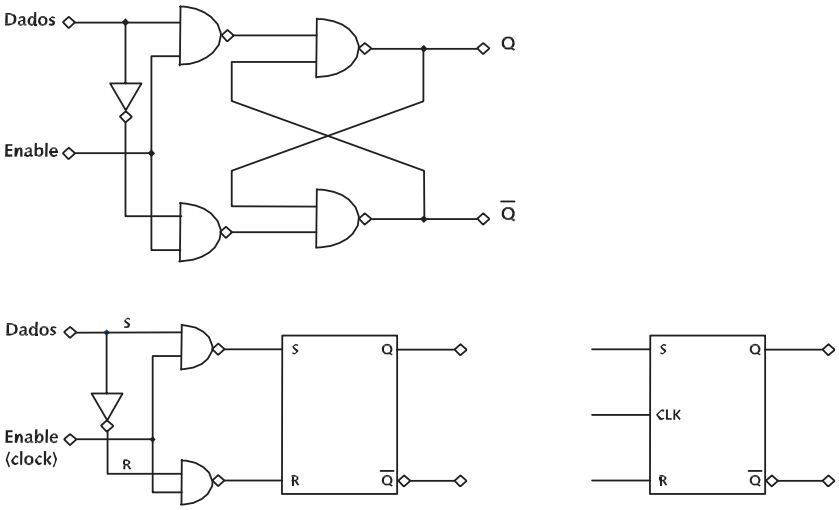


FIGURA 4.5: Latch Controlado (com Entrada Enable/Clock)

Diz-se que este tipo de latch tem a característica de ser transparente, já que a saída segue a entrada sempre que o latch estiver habilitado. Um latch que possui apenas uma entrada de dados é denominado latch tipo D. A seguir temos um diagrama de tempos representando o funcionamento do latch da figura acima.

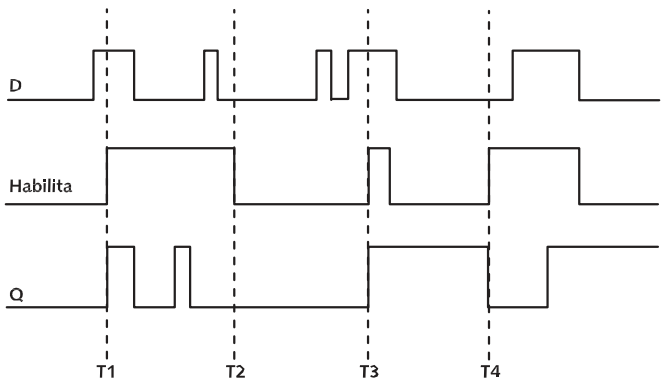


FIGURA 4.6: Diagrama de Tempos do Latch da Figura 4.5

O circuito integrado 74373 possui oito latch's transparentes tipo D, com saídas tri-state controladas por um pino de Enable e entradas controladas por um pino de Enable também.

4.1.4 Sincronismo (Clocking)

Um sistema síncrono é aquele em que aplicamos à sua entrada Enable um sinal (Clock) que executa transições entre os níveis alto e baixo, habilitando e desabilitando o circuito nos momentos apropriados. A cada ciclo alto-baixo, o processamento do circuito avança um passo. A velocidade do processamento depende da frequência destes ciclos. O tempo destinado ao ciclo do clock depende dos atrasos de propagação dos dispositivos físicos empregados.

4.2 Flip-Flop's

Os latch's controlados apresentam deficiências em sistemas síncronos, pois tem suas saídas inteiramente, ou em parte, ligadas à entradas de outros latch's. Neste caso encontramos problemas quando desejamos ler e escrever um dado em um latch no mesmo período do relógio (clock). Circuitos que superam estas deficiências são chamados de Flip-Flop's.

4.2.1 Flip-Flop Mestre-Escravo

O flip-flop mestre-escravo é um circuito lógico armazenador não-transparente, no qual acoplamos dois latch's em sequência, com um mesmo sinal de habilitação (clock) para os dois, sendo que no segundo latch (escravo) o sinal é invertido, conforme desenho a seguir:

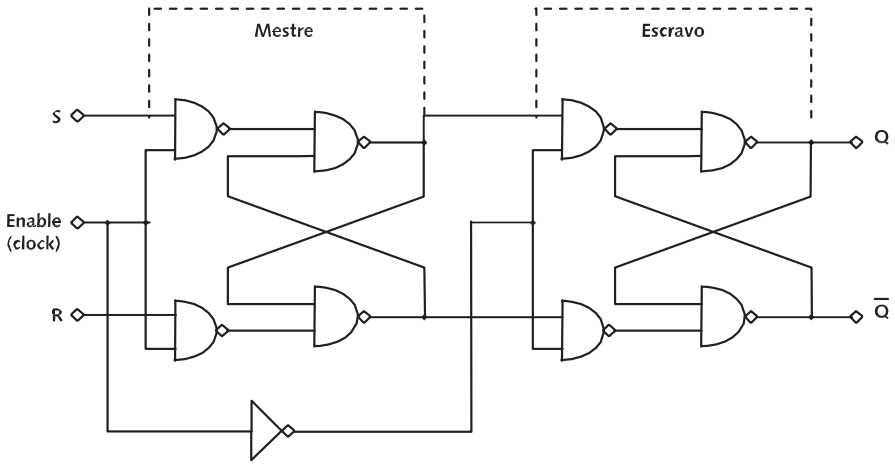


FIGURA 4.7: Flip-Flop Mestre-Escravo

No flip-flop mestre-escravo, ocorre o seguinte:

Quando o pino de clock passa de 0 para 1, o valor lido na entrada é armazenado no latch mestre.

Quando o pino de clock passa de 1 para 0, o valor armazenado no latch mestre passa para a saída do latch escravo.

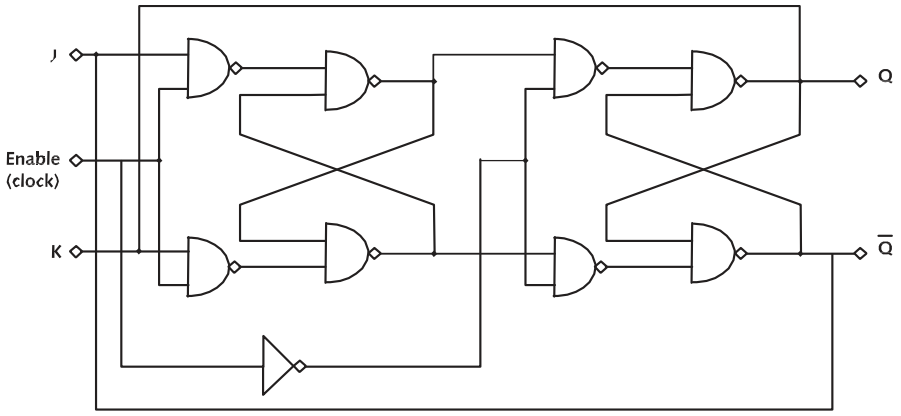
Este modo de funcionamento permite a construção de dispositivos chamados registradores de deslocamento ou shift-register's , bastante utilizados na prática.

4.2.2 Entradas Diretas

Entradas diretas são entradas que podem setar(preset) ou resetar(clear) o flip-flop sem depender do sinal do clock. Estas entradas são às vezes chamadas de entradas assíncronas. As entradas diretas prevalecem totalmente sobre as entradas síncronas.

4.2.3 Flip-Flop JK

O flip-flop JK é um flip-flop modificado para que, na condição R=1 e S=1, a saída inverta-se (altere o seu estado) . A representação de um circuito de flip-flop tipo JK, bem como a sua tabela-verdade está no desenho a seguir:



J	K	Q	Q/
0	0	x	x
0	1	0	1
1	0	1	0
1	1	y	y

Obs: x = mantém estado anterior
y = inverte o estado anterior

FIGURA 4.8: Circuito Flip-Flop tipo JK e sua Tabela-Verdade

4.2.4 Flip-Flop Tipo T

Um flip-flop tipo T é um flip-flop que chaveia a saída quando a entrada de controle for ativa e não altera-se quando a entrada for inativa. O flip-flop tipo T pode ser construído ligando-se juntas as entradas J e K de um flip-flop JK.

4.2.5 Flip-Flop JK Gatilhado pela Borda

O flip-flop mestre-escravo (Tanto no modo SR como JK), tem a seguinte desvantagem: Se alguma alteração rápida (ruído) ocorrer durante um pulso ativo do clock, esta alteração pode ser detectada e registrada, conforme mostra o diagrama de tempos representado a seguir:

O flip-flop gatilhado pela borda resolve o problema apresentado ante-

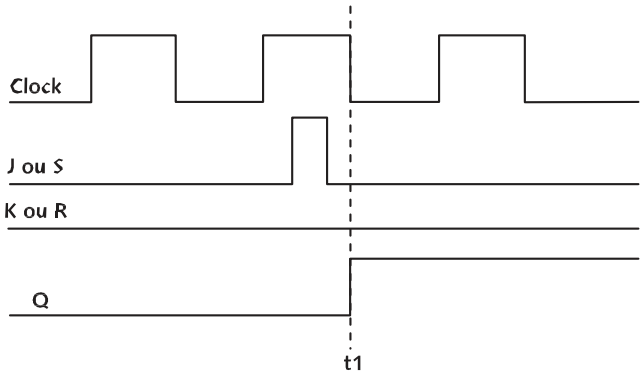


FIGURA 4.9: Diagrama de Tempos Mostrando Ruído Detectado por Flip-Flop tipo JK ou SR

riormente, já que a saída é a resposta à entrada, mas somente aos dados imediatamente anteriores à transição de gatilho do sinal de relógio.

4.2.6 Flip-Flop´s Mestre-Escravo com Data-Lock-Out

Existem flip-flop's do tipo mestre-escravo nos quais os dados são guardados no mestre durante a borda positiva do clock, e são trasnferidos para a saída durante a borda negativa do pulso do clock. Estes são denominados flip-flop's mestre-escravo com Data-Lock-Out.

Neste flip-flop, pode haver mudança na entrada durante o nível positivo do clock, pois a informação da entrada já foi armazenada no mestre na borda positiva do clock. Abaixo temos o diagrama de tempos que compara as formas de onda para um flip-flop mestre-escravo normal e um com data-lock-out

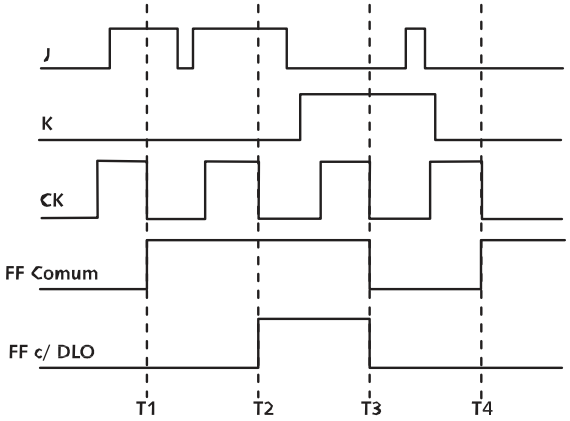


FIGURA 4.10: Diagrama de Tempos Comparativos entre Flip-Flop's Mestre-Escravo com e sem Data-Lock-Out

4.2.7 Flip-Flop Tipo D

Um flip-flop tipo D é um flip-flop tipo mestre-escravo, modificado da forma indicada no desenho abaixo. Em um flip-flop tipo D, o dado presente na entrada aparece na saída Q, em resposta à uma transição negativa do relógio.

Existe um circuito alternativo disponível mais econômico que o flip-flop mestre-escravo modificado, que tem gatilhamento na borda positiva da transição do relógio.

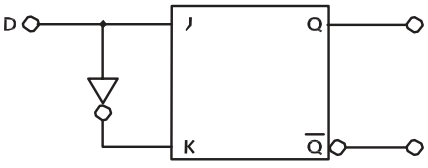


FIGURA 4.11: Flip-Flop Mestre-Escravo Modificado (Flip-Flop tipo D)

Utilizando outros circuitos não mostrados aqui, os flip-flop's podem ser habilitados, na prática, de quatro maneiras diferentes:

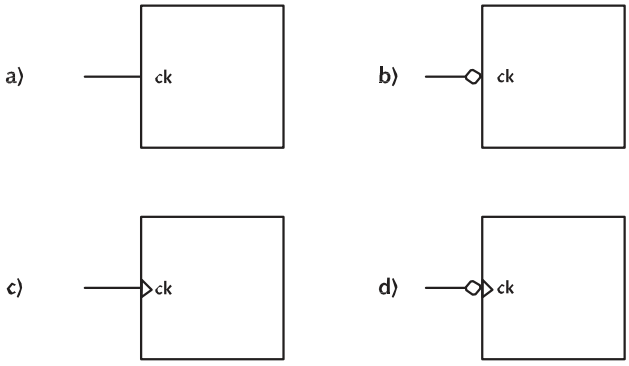


FIGURA 4.12: Modos de Habilitação de Flip-Flop's

a) É habilitado quando o clock está em nível 1; b) É habilitado quando o clock está em nível 0; c) É habilitado na transição de 0 para 1 do clock; d) É habilitado na transição de 1 para 0 do clock.

4.2.8 Parâmetros dos Flip-Flop´s

a) Tempo de Set-Up - $t_{Set\text{-}Up}$

É o tempo mínimo que o sinal deve permanecer nas entradas antes da ocorrência do pulso de gatilhamento

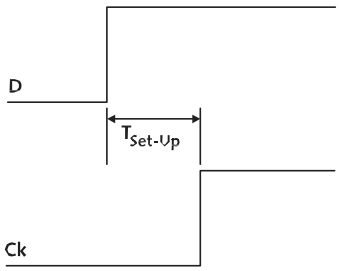


FIGURA 4.13: Tempo de Set-Up

b) Tempo de Manutenção - t_{hold}

É o tempo que o sinal deve permanecer nas entradas após a ocorrência do pulso de gatilhamento.

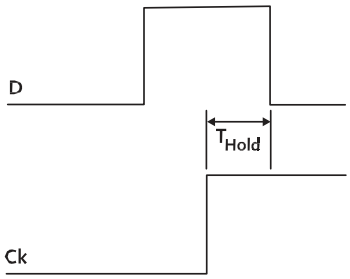


FIGURA 4.14: Tempo de Manutenção

c) Freqüência máxima - f_{max}

É a maior freqüência de pulsos de clock que pode ser aplicado ao dispositivo, mantendo chaveamento confiável.

CAPÍTULO 5
REGISTRADORES E CONTADORES

5.1 Registradores

Registradores são um grupo de elementos (flip-flop's, por ex.) capazes de guardar uma informação, e que funcionam juntos em uma unidade. Os registradores mais simples armazenam uma palavra binária que pode ter n bits. Existem também registradores mais complexos como os de deslocamento (shift register's) e contadores que, além de armazenar a palavra, podem executar uma determinada operação sobre esta palavra.

O desenho abaixo representa um registrador simples para palavra de quatro bits. Os registradores normalmente são mais complexos que o mostrado abaixo, tendo sinais de controle que habilitam sua leitura ou escrita.

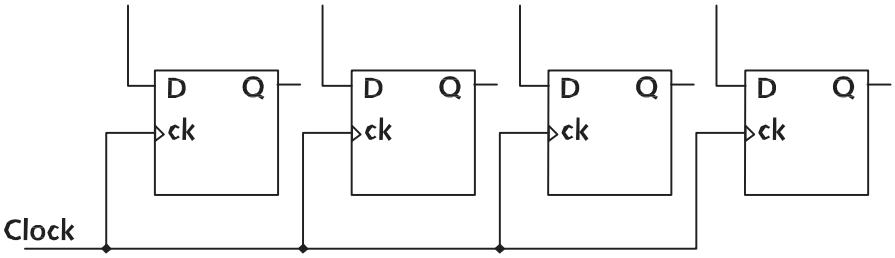


FIGURA 5.1: Registrador Simples para Palavra de 4 Bits

5.1.1 Registradores de Deslocamento (Shift Register's)

Uma característica bastante útil em um registrador é a possibilidade de efetuar um deslocamento , isto é, a possibilidade de um dado registrado em um bit deslocar-se através das unidades de armazenamento (flip-flop's). Um registrador de deslocamento é um dispositivo síncrono que pode ser utilizado para armazenar n bits com apenas um pino de entrada.

O desenho a seguir representa um registrador de deslocamento de 4 bits, utilizando flip-flop's do tipo JK.

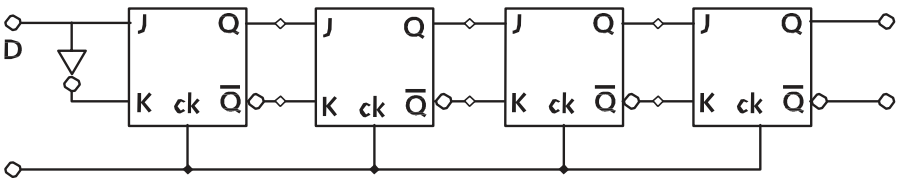


FIGURA 5.2: Registrador de Deslocamento de 4 Bits, com Flip-Flop's tipo JK

Existem registradores de deslocamento com circuitos lógicos próprios para a inversão de ligação entre os flip-flop's, de forma que possamos inverter o sentido do deslocamento.

Se ligarmos a saída do último flip-flop à entrada do primeiro, obtemos um registrador de deslocamento que não perde o último bit, mas sim transfere o seu valor para o primeiro. Este tipo de circuito é denominado registrador com rotação.

O CI 74194 é um registrador de deslocamento universal bidirecional de quatro bits. O CI 74 164 é um registrador de deslocamento de 8 bits.

5.1.2 Formato Série e Paralelo

Os dados digitais são apresentados e transmitidos de duas formas : Série e paralelo. No formato série, os dados são conduzidos (transmitidos) por um único fio. No formato paralelo, necessitamos de tantos fios quantos forem os bits a serem transmitidos. O formato série é mais econômico, mas é menos rápido. Frequentemente, necessitamos fazer a conversão da apresentação de dados em série para paralelo e vice-versa. O circuito registrador de deslocamento é adequado para este fim.

5.2 Contadores

Um contador é um arranjo de flip-flop's que avança de um estado para outro em resposta a um evento. Este evento pode ser a passagem de uma peça por uma esteira na linha de produção, a passagem de uma pessoa por uma roleta, etc. , e geralmente é traduzido na forma de um pulso de clock. Observe que um relógio digital nada mais é do que um contador em que o evento a ser contado é um período de tempo.

Existem dois tipos de contadores: Os **contadores síncronos,** nos quais o sinal de clock é aplicado a todos os flip-flop's componentes do conjunto ao mesmo tempo e os **contadores assíncronos** , nos quais o sinal aplicado ao clock de cada flip-flop pode diferir.

5.2.1 Contadores Síncronos

5.2.1.1 Contador em Anel

O contador em anel é um registrador de deslocamento ligado em configuração de rotação, no qual carregamos previamente o valor 1 em apenas um dos flip-flop's. Este tipo de contador tem, portanto, uma saída em decimal. No desenho na página seguinte está representado um contador em anel com módulo 4:

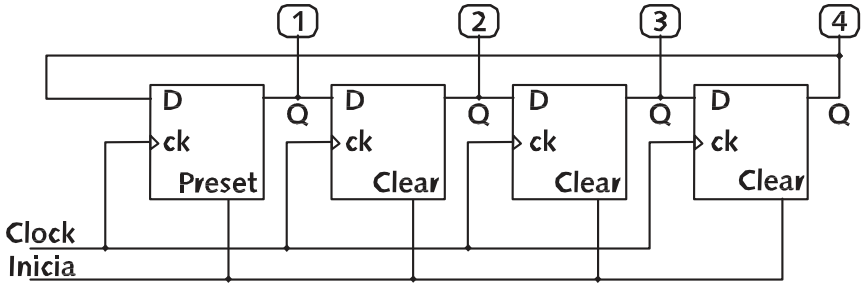


FIGURA 5.3: Contador Síncrono, com Configuração em Anel e Módulo 4

5.2.1.2 Contador em Anel Torcido

O contador em anel torcido, ou contador Johnson, é idêntico ao da figura anterior, com excessão que para fechar o anel, ligamos a saída Q\ do último flip-flop à entrada D do primeiro. Com este arranjo, obtemos 8 combinações possíveis para quatro flip-flop's, ao contrário do circuito da Figura 5.3, que só tinha módulo 4. Este tipo de circuito, no entanto, necessitará de uma lógica adicional (um decodificador) para obter uma saída aceitável.

5.2.1.3 Contador Síncrono em Código Binário

Um circuito utilizando flip-flop's tipo T , montados como mostra a figura abaixo é um contador em binário de 4 bits, ou seja, pode contar até 16. Lembre que os flip-flop's tipo JK com as entradas J e K ligadas juntas é um flip-flop tipo T. Este tipo de contador é mais lento que o contador em anel, visto que , dependendo da situação, o sinal deve se propagar através de uma cascata de portas AND.

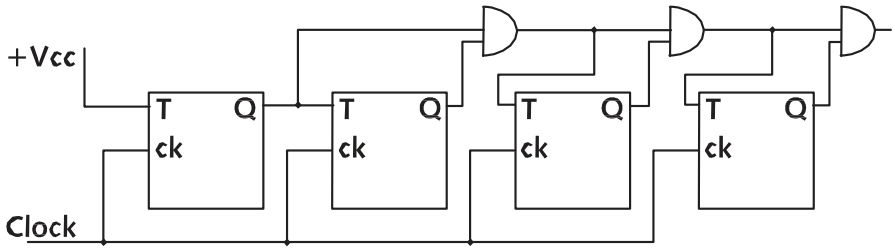


FIGURA 5.4: Contador Binário de 4 Bits, Síncrono

5.2.1.4 Contadores Síncronos de Módulo Arbitrário

Para construir um contador de módulo arbitrário, começamos com um número de flip-flop's suficiente para prover um número de estados igual ou maior que o módulo. Em seguida devemos decidir quais estados entre os possíveis devem ser eliminados. Devemos então decidir a ordem pela qual o contador deve passar pelos estados não eliminados.

Finalmente devemos projetar uma lógica combinacional de forma que o contador passe de estado para estado conforme desejamos. Os métodos específicos para o projeto de um gerador de módulo arbitrário fogem do escopo deste documento.

5.2.2 Contadores Assíncronos

5.2.2.1 Contadores por Pulsação (Ripple Counters)

Também chamado contador Série. Este tipo de contador é construído ligando-se flip-flop's tipo T ligados em cascata conforme o desenho abaixo. A entrada T do flip-flop deve ser mantida em nível 1 para que o flip-flop opere em modo de chaveamento. O relógio externo (clock) só é ligado no primeiro flip-flop. Para que o efeito do clock possa ser sentido em um determinado flip-flop, este deve se propagar através de todos os flip-flop's anteriores.

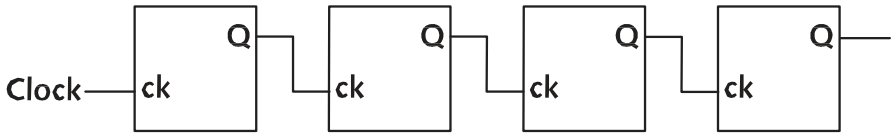


FIGURA 5.5: Contador por Pulsação

Um contador por pulsação pode usar flip-flop's que chaveiam na borda positiva ou na borda negativa do sinal aplicado na entrada de clock. Pode-se ainda ligar a entrada de clock de cada flip-flop à saída Q ou Q\ do flip-flop anterior. Dependendo do tipo de arranjo, o contador contará para cima ou para baixo.

LIGAÇÃO DE ENTRADA DE CLOCK	CHAVEAMENTO NA BORDA DE CLOCK	DIREÇÃO DE CONTAGEM
Q	Negativa	Crescente
Q	Positiva	Decrescente
Q\	Negativa	Crescente
Q\	Positiva	Decrescente

TABELA 5.1: Direção de Contagem Conforme Ligação e Tipo de Chaveamento em um Contador por Pulsação

5.2.3 Circuitos Integrados Contadores

Existe um grande número de circuitos integrados contadores , sendo que alguns possuem circuitos bastante complexos. Todos baseiam o funcionamento em flip-flop's. Na família TTL 74, os mais simples são o 7490, 7492 e 7493.

O 7490 é um contador de décadas (módulo 10). Internamente temos um conjunto de 3 flip-flop's que compõe um contador módulo 5 e mais um flip-flop que é um contador módulo 2. Se ligarmos externamente a saída QA (pino 12) à entrada B (pino 1), obtemos um contador de décadas. Neste caso o sinal de entrada é conectado à entrada A (pino 14). O 7490 possui 4 entradas de reset que determinam as saídas, conforme a tabela a seguir:

ENTRADAS RESET				SAÍDAS			
R0(1)	R0(2)	Rg(1)	Rg(2)	QD	QC	QB	QA
1	1	0	x	0	0	0	0
1	1	x	0	0	0	0	0
x	x	1	1	1	0	0	1
x	0	x	0	Contagem			
0	x	0	x	Contagem			
0	x	x	0	Contagem			
x	0	0	x	Contagem			

TABELA 5.2: Entradas e Saídas do CI Contador 7490

O 7492 possui 4 flip-flop's (A,B,C e D) interligados de maneira que possam ser usados das seguintes maneiras:

- flip-flop A pode ser usado separadamente como contador módulo 2;
- os flip-flop's B, C e D podem ser usados como contador módulo 6;
- os quatro flip-flop's podem ser usados como contador módulo 12.

O 7493 é um contador de pulsação com quatro flip-flop's, sendo que o primeiro é isolado dos demais, formando assim três possibilidades de uso :

- 1 contador módulo 2;
- 1 contador de pulsação módulo 8;
- 1 contador de pulsação módulo 16, interligando externamente o primeiro flip-flop aos demais.

Pode-se interligar dois circuitos 7493, como qualquer outro circuito integrado contador, para formar um contador de pulsação maior. Existem integrados que possuem uma saída de carry que facilitam esta interligação.

CAPÍTULO 6
CIRCUITOS ARITMÉTICOS

Na maioria dos circuitos digitais existem circuitos denominados aritméticos que executam as operações de adição, subtração, multiplicação e divisão de números binários. Existem também dispositivos flexíveis tais como as Unidades Aritméticas e Lógicas (ULA's), que são fornecidas em forma de CI's e podem executar diversas operações, dependendo de como forem programadas (conforme o estado de determinadas entradas que determinam a função a executar).

6.1 Meio Somador

O meio somador é um circuito destinado a somar dois bits. A soma de dois bits obedece as regras da tabela verdade a seguir:

A	B	S	Carry ou "Vai um"
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

TABELA 6.1: Tabela-Verdade para Meio Somador

A partir da última tabela , pode-se chegar ao circuito que calcula a soma e o carry:

$S = A \oplus B$
$C = AB$

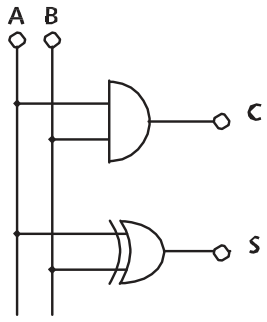


FIGURA 6.1: Circuito para Soma e Carry (Meio Somador)

6.2 Meio Subtrator

É um circuito básico destinado a subtrair dois bits. A subtração de dois bits obedece as regras da tabela a seguir:

A	B	S	Carry ou "Pede Emprestado"
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

TABELA 6.2: Tabela-Verdade para Meio Subtrator

A partir da tabela acima, pode-se chegar ao circuito que calcula a subtração e o carry:

$S = A \oplus B$
$C = \overline{A}B$

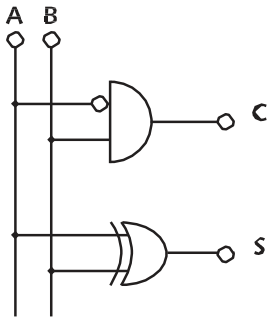


FIGURA 6.2: Circuito para Subtração e Carry (Meio Subtrador)

6.3 Meio Inteiro

O somador descrito neste item é utilizado para soma de dois números binários, cada qual contendo mais que um bit. Nestes casos, quase sempre há necessidade de somar-se três bits (os dois números mais um eventual "vai um"). A tabela verdade que rege a soma de três bits é a que está representada à seguir:

A	B	Carry in	S	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

TABELA 6.3: Tabela-Verdade para Soma de 3 Bits

A partir da tabela anterior, chega-se à expressão que calcula a soma de três bits:

$S = A \oplus B \oplus C$
$C = C(A \oplus B) + AB$

6.4 Subtrator Inteiro

Da mesma forma que o somador, quando se deseja subtrair dois números binários com mais de um bit cada, quase sempre há necessidade de efetuar uma operação envolvendo 3 bits. Neste caso, a tabela-verdade que rege a subtração é a mostrada abaixo. O carry é o "pede emprestado", no caso.

A	B	Carry in	S	Carry out
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

TABELA 6.4: Tabela-Verdade para Subtração de 3 Bits

A partir da tabela acima, chega-se à expressão que calcula a subtração de três bits:

$$S = A \oplus B \oplus C$$
$$C = C(\overline{A \oplus B}) + \overline{AB}$$

6.5 Somador Paralelo

O somador paralelo soma todos os bits das parcelas a serem somados simultaneamente, conforme mostra a figura abaixo. Isto aumenta a velocidade da soma, mas requer mais hardware.

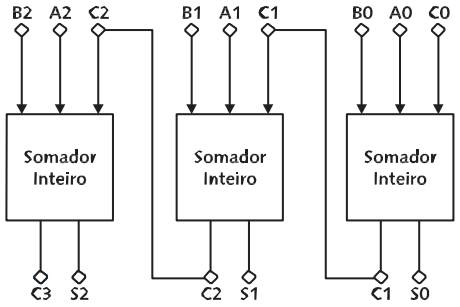


FIGURA 6.3: Circuito Somador Paralelo

6.6 Somador Série

O circuito somador série é mais lento que o somador paralelo, mas gasta menos hardware.

O circuito está representado no desenho abaixo e funciona da seguinte forma: As parcelas da soma são carregadas em dois registradores de deslocamentos de n bits. Temos à disposição um registrador de deslocamento de N+1 bits para registrar o resultado. Temos um clock comum a todo o conjunto. A cada pulso de clock, o resultado da soma de dois bits é guardado e dois bits mais significativos são apresentados ao circuito somador inteiro.

Um flip-flop tipo D "atrasa" o bit de carry resultante da soma de forma que este seja apresentado no próximo ciclo.

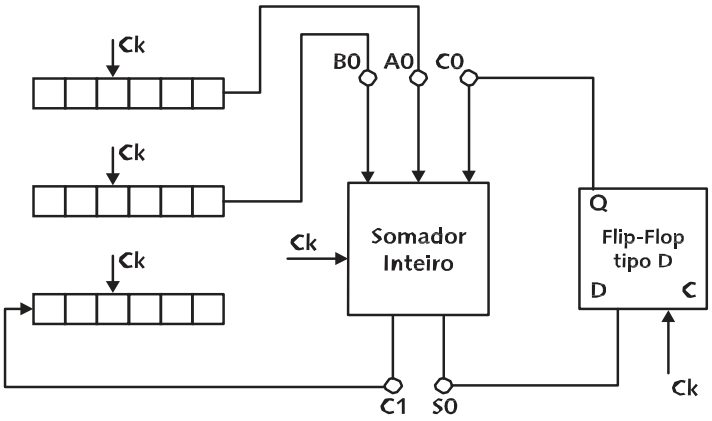


FIGURA 6.4: Somador Série

O 7483 é um circuito integrado somador de 4 bits que aceita dois números de quatro bits como entradas e fornece na saída a soma de quatro bits e um bit de carry.

6.7 Representação em Complemento de Dois

Boa parte dos circuitos digitais utiliza apenas um circuito para soma e subtração. Neste processo, no caso de efetuarmos uma subtração, representa-se o subtraendo como um número negativo e efetua-se uma soma comum. A maneira mais usual de representação de números negativos em binário é denominada complemento de dois. Para achar o complemento de dois de um determinado número binário basta inverter este número e somar 1 ao resultado. Da mesma forma, se um resultado tiver 1 no bit mais significativo, isto significa que este é um número negativo, e devemos calcular o seu complemento de dois.

Exemplo:

30 - 25 = 5
30 = 11110
25 = 11001
25 em comp. de dois = 00111
  11110
+ 00111
= 00101 = 5

6.8 Circuito Subtrator Somador

Na figura 6.5 temos representado um circuito somador subtrator paralelo de 4 bits que utiliza o processo de complemento de dois. Quando a entrada de seleção estiver em 0, a operação realizada pelos blocos somadores inteiros é **A+B**. Quando a entrada de seleção estiver em 1, a operação rea-

lizada é **A+$\overline{B}$+1** como $\overline{B}$**+1** é o complemento de dois de B, na verdade estamos realizando a subtração de A e B.

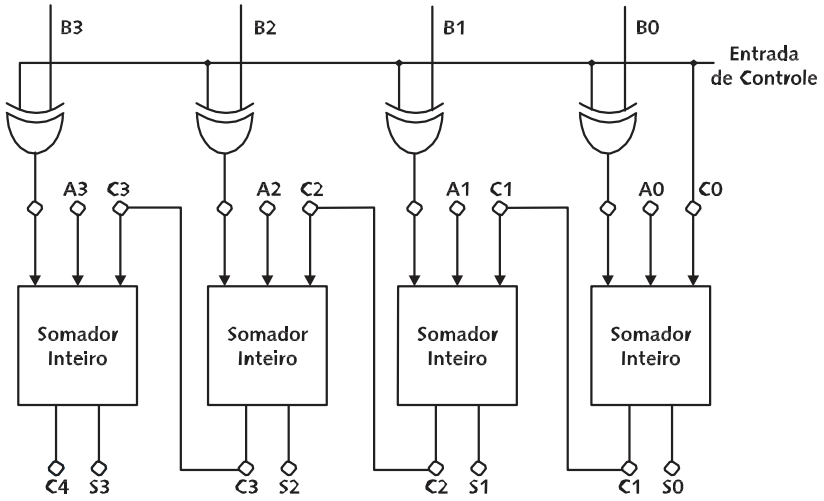


FIGURA 6.5: Circuito Subtrador Somador

6.9 ULA (Unidade Lógica e Aritmética)

Uma ULA é um dispositivo lógico combinacional que aceita duas entradas A e B e gera uma função na saída F que está relacionada às entradas por uma operação lógica ou aritmética. A maneira pela qual as saídas estão relacionadas com as entradas depende do bits seletores de função S0, S1, etc.

O circuito Integrado 74181 é uma ULA básica de linha 74. Possui duas entradas para palavras de quatro bits (A e B), uma saída de quatro bits (F) para o resultado da operação, uma entrada de quatro bits para seleção de operação (S), um bit para seleção de modo (m), uma entrada para o carry-in (o carry-in não afeta as operações lógicas), uma saída para o carry-out, e uma saída indicativa de igualdade (A=B).

A tabela abaixo resume as funções da ULA 74181

SELEÇÃO				MODO = 1	MODO = 0 (Operações Aritméticas)	
S3	S2	S1	S0	(Funções Lógicas)	Cn = 0 (com carry)	Cn = 1 (sem carry)
L	L	L	L	f = $\overline{A}$	f = A mais 1	f = A
L	L	L	H	f = $\overline{A + B}$	f = (A + B) mais 1	f = A + B
L	L	H	L	f = $\overline{A}$ B	f = (A + $\overline{B}$) mais 1	f = A + $\overline{B}$
L	L	H	H	f = 0	f = 0	f = menos 1 (comp. dois)
L	H	L	L	f = $\overline{A B}$	f = A mais A $\overline{B}$ mais 1	f = A mais A $\overline{B}$
L	H	L	H	f = $\overline{B}$	f = (A + B) mais A $\overline{B}$ mais 1	f = (A + B) mais A $\overline{B}$
L	L	H	L	f = A $\oplus$ B	f = A menos B	f = A menos B menos 1
L	H	H	H	f = A $\overline{B}$	f = A $\overline{B}$	f = A B menos 1
H	L	L	L	f = $\overline{A}$ + B	f = A masi A B mais 1	f = A masi A B
H	L	L	H	f = $\overline{A \oplus B}$	f = A mais B mais 1	f = A mais B
H	L	H	L	f = B	f = (A + $\overline{B}$) mais A B mais 1	f = (A + $\overline{B}$) mais A B
H	L	H	H	f = A B	f = A B	f = A B menos 1
H	H	L	L	f = 1	f = A mais A mais 1	f = A mais A *
H	H	L	H	f = A + $\overline{B}$	f = (A + B) mais A mais 1	f = (A + B) mais A
H	H	H	L	f = A + B	f = (A + $\overline{B}$) mais A mais 1	f = (A + $\overline{B}$) mais A
H	H	H	H	f = A	f = A	f = A menos 1

* cada bit é deslocado para a próxima posição mais significativa

TABELA 6.5: Conjunto de Funções da ULA 74181

CAPÍTULO 7
MEMÓRIAS

Já vimos que através de dispositivos eletrônicos como registradores, podemos armazenar uma palavra de n bits. Memórias são dispositivos utilizados para armazenar palavra binárias na ordem de centenas de milhares. Pode-se utilizar flip-flop's para o armazenamento em memórias ou outro dispositivo qualquer que sirva para este fim.

Os circuitos de memória normalmente tem as seguintes entradas e saídas:

- algumas vias de entrada de dados para gravação e algumas saídas para leitura (que fisicamente podem ser as mesmas);
- algumas vias para endereço , que selecionará qual registrador será lido / escrito, de acordo com um código (endereço de memória);
- um pino que habilita o circuito (Chip Select - CS). Se o circuito não estiver habilitado, as saídas permanecem em alta impedância;
- um pino de leitura / escrita, que habilita uma destas duas operações ou apenas leitura, dependendo do tipo de memória.

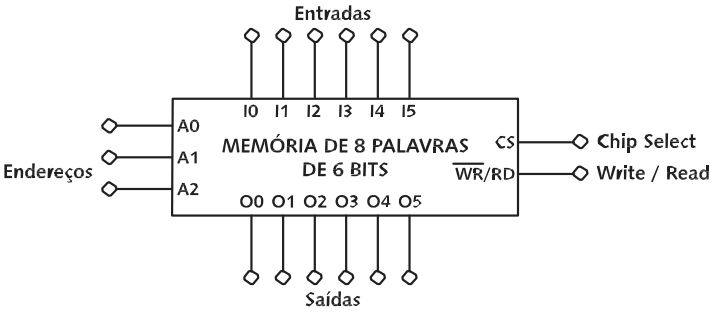


FIGURA 7.1: Diagrama de uma Memória Hipotética de 8 Palavras de 6 Bits

7.1 Memória RAM (Memória de Acesso Aleatório)

Memória RAM é uma memória de leitura e escrita, isto é, que pode ser gravada com um determinado valor e este valor pode ser posteriormente lido. Além disso, podemos acessar qualquer registrador desejado aleatoriamente para ler ou escrever uma palavra. A memória RAM comum necessita de alimentação elétrica para manter a integridade de seus dados. É por este motivo, pertencente ao grupo de memórias voláteis.

Quanto à sua construção, as memórias RAM podem ser de dois tipos básicos: RAM estática e RAM dinâmica.

Na memória RAM estática, os bits são armazenados em flip-flop's individuais e permanecem armazenados indefinidamente enquanto o circuito for alimentado.

A memória RAM dinâmica armazena os bits através de carga em diminutos capacitores.

Como um capacitor deste tipo ocupa muito menos espaço que um flip-flop em um CI, a memória dinâmica resulta bem mais compacta que a estática. Em compensação, o bit em um capacitor permenece íntegro por apenas uma fração de tempo (aprox. 2 ms), devido às fugas. Para contornar este problema este tipo de memória deve ter um circuito auxiliar que verifique temporariamente os capacitores e os recaregue, se for necessário. Esta operação é denominada refresh.

A maioria das memórias tem saídas em coletor aberto ou tri-state para permitir a ligação em paralelo e consequentemente melhorar a capacidade de manuseio de dados. Assim, quando o Chip Select não estiver ativo, o componente ficará em estado de alta impedância, e não se pode nem escrever na memória nem ler os seus conteúdos. Isto significa que a memória estará desconectada de seus componentes externos.

A operação de gravação ou escrita é feita colocando-se os dados nas linhas de entrada, habilitando-se o chip , colocando-se os sinais de endereço na posição desejada e habilitando a escrita da memória. Deste modo os dados das linhas de entrada serão escritos na posição selecionada.

Do mesmo modo, a operação de leitura é feita habilitando-se o chip , colocando-se os sinais de endereço na posição desejada e habilitando a leitura da memória. Deste modo os dados da posição de memória selecionada ficarão disponíveis na saída, para leitura.

7.2 Memórias ROM

Uma memória ROM (Read Only Memory) é um tipo de memória no qual podemos ler, mas não escrever. Os conteúdos são fixos e inalterados, sendo estabelecidos na hora da fabricação.

Em uma ROM, os conteúdos não precisam ser alterados. Portanto não necessitamos de flip-flop's ou dispositivos semelhantes. Uma ROM na verdade nada mais é do que um conversor de código e pode ser construído a partir de dispositivos mais simples e baratos que as portas normalmente utilizadas.

7.2.1 Memórias ROM Programáveis (Prom's)

Existem circuitos de ROM que permitem que o usuário estabeleça as informações que serão armazenadas, ao invés do fabricante. Estas memórias são chamadas de memórias PROM (Memórias de leitura programáveis). A gravação só pode ser feita uma única vez e não mais alterada. Normalmente a gravação é feita através da queima de elos fusíveis que determinam se a posição de memória conterá "um" ou "zero".

7.2.2 Memórias ROM Programáveis e Apagáveis (Eprom's)

Na EPROM, os dados são armazenados em dispositivos baseados em MOSFET's. Estes dispositivos fazem ou não a conexão (guardam bit "um" ou

"zero") conforme haja ou não carga elétrica na porta do transistor. A programação é feita através de um programador de EPROM's. Uma característica importante é a de que a exposição à luz ultravioleta forte (por aproximadamente 30 min.) permite a fuga das cargas, apagando a memória. O apagamento possibilita uma nova programação (gravação).

7.3 Ligação de Memórias em Paralelo

A ligação de memórias em paralelo é utilizada quando o número de palavras e/ou o número de bits por palavra disponível em um determinado CI não é adequado.

Abaixo está representado a uma ligação em paralelo para aumentar o número de bits por palavra:

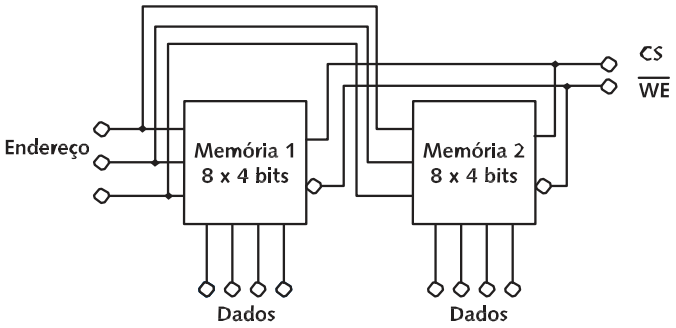


FIGURA 7.2: Ligação de Memórias em Paralelo para Aumentar Número de Bits por Palavra

O próximo desenho mostra uma ligação em paralelo para aumentar o número de palavras do conjunto:

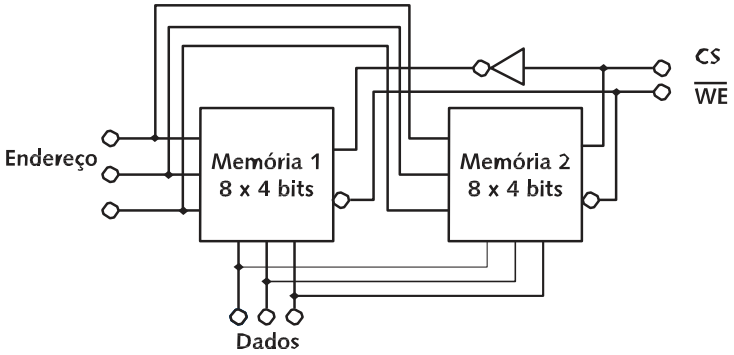


FIGURA 7.3: Ligação de Memórias em Paralelo para Aumentar Número de Palavras

Para possibilitar flexibilidade no número de bits por palavra, os fabricantes oferecem circuitos de memória com um só bit por palavra. Após selecionar um circuito com o número adequado de palavras, construímos uma memória com n bits/palavra através da ligação em paralelo de n circuitos.

7.4 Memórias Série

As memórias vistas até agora são todas de acesso aleatório, isto é, qualquer posição de memória pode ser acessada gastando aproximadamente o mesmo tempo, colocando o código adequado nos pinos de endereço. Existe outro tipo de memória denominado memórias-série, em que os dados são guardados seqüencialmente em uma ordem pré-determinada.

Isto significa que o tempo de acesso à memória-série depende do número de endereços interpostos entre o endereço desejado e o endereço presente. As memórias série são mais econômicas que as de acesso aleatório, mas em compensação são mais lentas.

Existem dois tipos principais de memórias-série:

Aquelas nas quais as palavras aparecem na saída na mesma ordem em que foram escritas. Elas são implementadas utilizando shift-register's ligados em anel. Por isso, o dispositivo é às vezes chamado de first-in-first-out (FIFO);

Aquelas nas quais as palavras aparecem na saída na ordem inversa em que foram escritas. Elas são implementadas utilizando shift-register's bidirecionais. Por isso, o dispositivo é às vezes chamado de last-in-first-out (LIFO).

CAPÍTULO 8
CONVERSORES D/A e A/D

É comum acontecerem situações em que se precisa trabalhar com sinal digital ao invés do analógico e vice-versa. Para resolver estas situações foram criados dispositivos denominados conversores Analógico/Digital (A/D), que convertem um sinal analógico em digital e conversores (D/A), que fazem a operação inversa.

Nos conversores A/D e D/A, a importância maior é dada a amplitude do sinal, de entrada e de saída respectivamente, de maneira que exista uma exata correspondência entre o sinal analógico e o quantificado em digital e vice-versa.

8.1 Conversores Digitais/Analógicos (D/A)

O conversor D/A converte n bits de sinal, presente em sua entrada, em um correspondente sinal analógico de saída e no tempo, de modo que, a cada valor binário colocado rm sua entrada, fornecerá somente um nível analógico de tensão na saída:

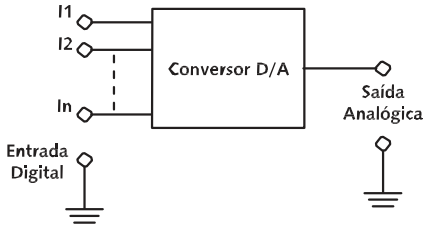


FIGURA 8.1: Conversor D/A

8.1.1 Conversor D/A do Tipo Somador

Este conversor soma os níveis lógicos correspondentes a cada bit, sendo que o bit mais significativo corresponde com um nível de tensão maior ou proporcional na soma, se comparado aos bits menos significativos.

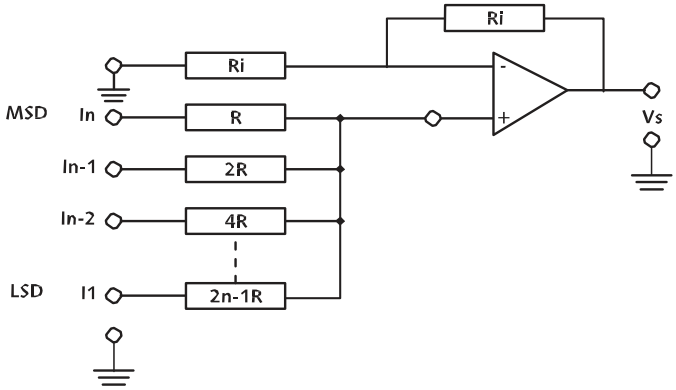


FIGURA 8.2: Conversor D/A Tipo Somador

Quanto menos significativo é o bit, maior é a resistência, que dobra a cada bit. Consequentemente, maior é a queda de tensão sobre a mesma e menor é a tensão com que contribui na saída, uma vez que os níveis de tensão de cada bit digital de A1 a An, são iguais.

O amplificador operacional na saída permite um ajuste de ganho do circuito. Como exemplo, temos abaixo a equação que expressa o valor da tensão de saída Vs a partir da tensão de entrada Vi para 3 bits de entrada:

$$V_s = \left(1 + \frac{Rf}{Ri}\right)\frac{Vi}{7}(4i_3 + 2i_2 + 1i_1)$$

8.1.2 Conversor D/A Tipo R-2R

O conversor somador apresenta o inconveniente utilizar resistores de valores múltiplos uns dos outros . Já os do tipo R-2R só utiliza resistores de dois valores, formando uma malha, conforme mostrado na figura abaixo.

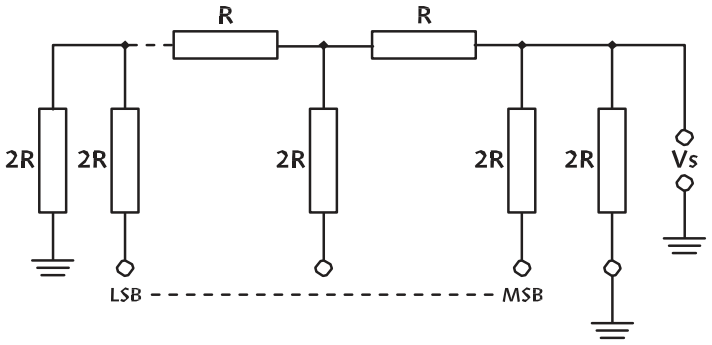


FIGURA 8.3: Conversor D/A Tipo R-2R

A expressão geral para qualquer número de bits i de entrada é mostrada abaixo:

$$V_s = \frac{Rf}{2R} + \frac{Vi}{Ri}(2^{n-1}i_n + \ldots\ldots + 2^1 i_2 + i_1)$$

8.2 Conversores Analógicos/Digitais (A/D)

O processo de conversão do sinal analógico em digital é bem mais complexo que o inverso. Existem diversos métodos desenvolvidos para tal, sendo que os principais descreveremos a seguir. É importante ressaltar que, para aplicações que exijam confiabilidade e/ou precisão maiores, existem circuitos integrados dedicados para a conversão A/D no mercado.

8.2.1 Conversor A/D Simultâneo ou Conversor Flash

Este método de conversão baseia-se no uso de uma série de comparadores ligados ao sinal analógico de entrada. Cada comparador tem sua segunda entrada ligada a uma tensão de referência diferente, fazendo que a saída dos comparadores seja ativada conforme o valor da tensão de entrada. Ligando-se as saídas dos comparadores a um conversor de código, teremos um número binário que corresponde ao sinal de entrada.

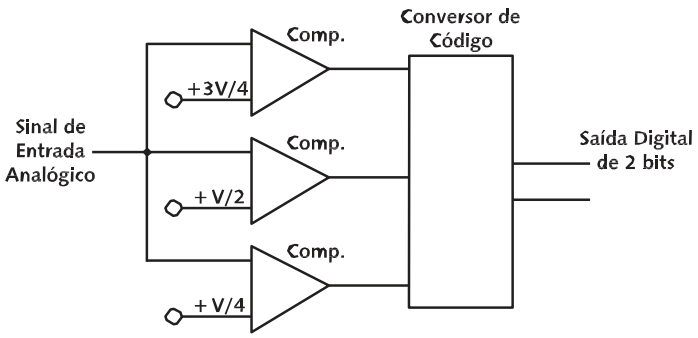


FIGURA 8.4: Conversor A/D Simultâneo ou Conversor Flash

O conversor A/D simultâneo é utilizado somente até 3 ou 4 bits de resolução de saída e tem uma velocidade de conversão bastante elevada.

8.2.2 Conversores de Contagem Crescente

Observe o desenho abaixo:

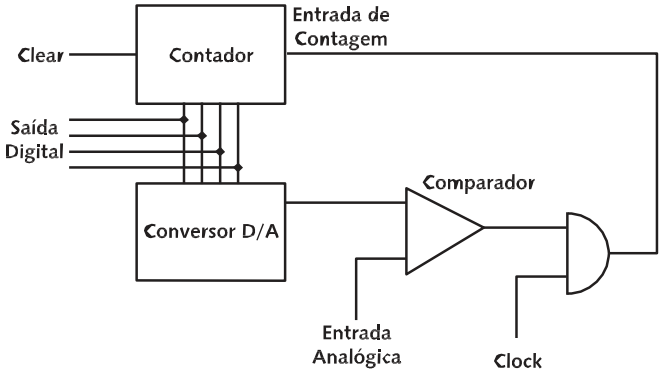


FIGURA 8.5: Conversor A/D de Contagem Crescente

O funcionamento do circuito é o seguinte: Um pulso de clear reseta o contador fazendo com que a saída do conversor D/A vá para zero. Se a entrada analógica apresentar uma determinada tensão, aparecerá um sinal na saída do comparador, que habilita o clock e faz com que o contador aumente até estabilizar no número binário que representa a entrada.

Periodicamente, deve-se fazer a leitura do contador e resetá-lo para reiniciar o processo. Logicamente o conversor apenas tem o número binário correto nas suas saídas em determinados períodos de tempo, o que limita sua utilização.

Na página seguinte temos a representação do funcionamento de um conversor A/D de quatro bits do tipo contagem crescente:

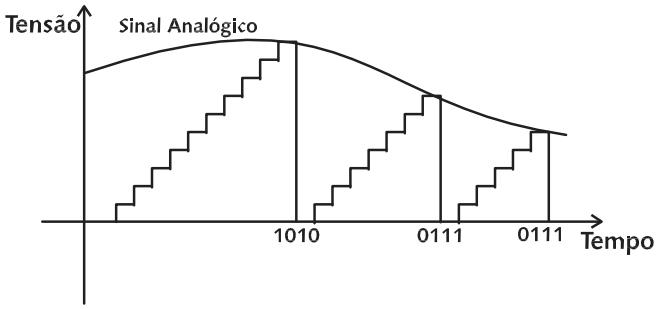


FIGURA 8.6: Sinal Analógico e sua Representação Digital em um Circuito Conversor de A/D de Contagem Crescente

8.2.3 Conversores de Rastreamento

Neste tipo de conversor, acrescentamos um contador bidirecional (up-down) ao circuito. A saída do comparador é ligada ao controle up-down do contador. Se o sinal de entrada for maior que o dígito binário na saída, o contador conta para cima e vice-versa. Isto faz com que a saída siga (rastreie) a entrada. O circuito e o gráfico representativo estão na figura abaixo:

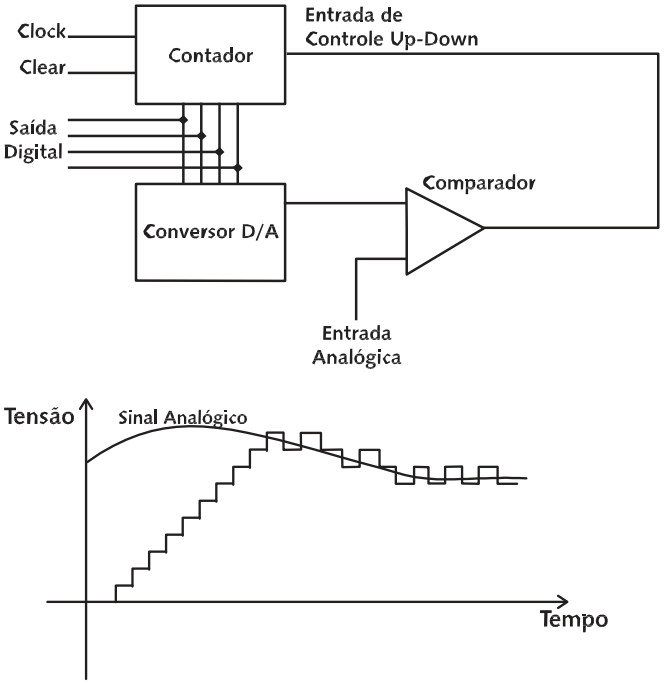


FIGURA 8.7: Circuito Conversor A/D de Rastramento / Sinal Analógico e sua Representação digital em um conversor deste tipo

Os conversores A/D e D/A são muito utilizados em aplicações como armazenamento (gravação) e transmissão de áudio e vídeo, visto que os sinais digitais são menos passíveis de introdução de ruídos e podem ter verificação de integridade e processamento mais fáceis. Além disso em qualquer circuito de interface entre variáveis do mundo real (como movimento, velocidade, temperatura, massa, pressão, umidade, posição, etc), e um circuito

digital, os circuitos conversores mostram-se úteis, conforme mostra o circuito de controle de temperatura sugerido abaixo:

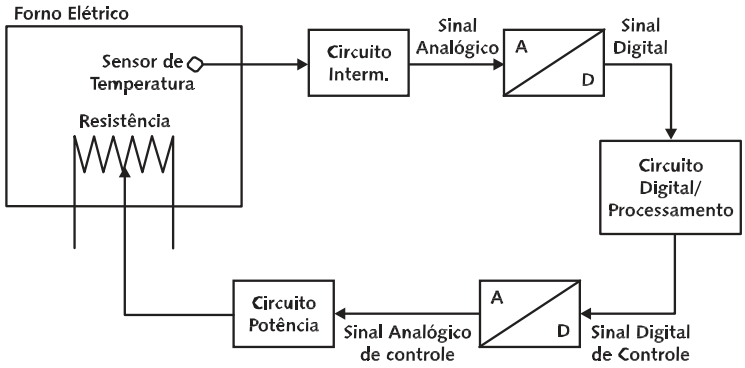


FIGURA 8.8: Exemplo da Utilização de Conversores A/D e D/A em um Sistema de Controle de Temperatura

8.2.4 Circuito de Amostra e Retenção (Sample-and-Hold)

Um conversor A/D requer uma certa quantidade de tempo, chamada tempo de conversão, para transformar um sinal analógico em um sinal digital correspondente. Se o sinal analógico muda durante o tempo de conversão, a saída do conversor pode apresentar erro. Para impedir isso, um circuito de amostra e retenção é utilizado para ler o sinal no início da conversão e armazená-lo em um capacitor durante o tempo de conversão restante.

A figura 8.9 mostra um circuito deste tipo. A1 e A2 são buffer's. S é uma chave normalmente representada por um circuito a transistor.

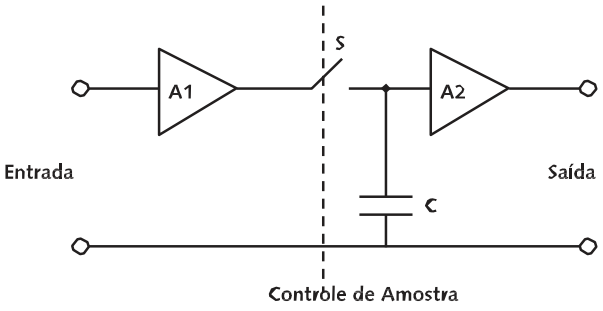


FIGURA 8.7: Circuito de Amostra e Retenção (Sample-and-Hold)

REFERÊNCIAS BIBLIOGRÁFICAS

DATAPOOL ELETRÔNICA. **Módulo 8810 : teoria e prática**. Itajubá, [199-], 1 v.

MALVINO, Albert Paul. **Microcomputadores e microprocessadores**. São Paulo : Makron Books. 1985. 578 p.

MALVINO, Albert Paul. **Eletrônica digital : princípios e aplicações**. São Paulo, Makron Books do Brasil, 1987. 2 v.

NATALE, Ferdinando. **Tecnologia digital**. São Paulo : Atlas, 1992. 376 p.

TAUB, Herbert. **Circuitos digitais e microprocessadores**. São Paulo, Makron Books, 1984. 510 p.