

AeroTD

ESCOLA DE AVIAÇÃO CIVIL

Eletrônica II

Prof. Hélio Luis Camões de Abreu



| | |
|------------------------------|--|
| CNPJ | 72.443.914/0001-38 |
| Mantenedora | AERO TD ESCOLA DE AVIAÇÃO CIVIL LTDA - ME |
| Instituição | AERO TD Escola de Aviação Civil |
| Esfera Administrativa | Privada |
| Endereço (Rua, Nº.) | Rua Madalena Barbi nº 46. |
| Cidade UF CEP | Bairro: Centro - Florianópolis SC. CEP: 88.015-200 |
| Telefone Fax | (48) 32235191 |
| Eixo Tecnológico: | Infraestrutura |
| Curso: | Profissionalizante em Manutenção de Aeronaves - Habilitação Aviônicos |
| Carga Horária Total: | 1040 horas |

Sumário

| | |
|--------------------|---------|
| Apresentação | 4 |
| Módulo I | 6 a 67 |
| Módulo II | 70 a 88 |

Apresentação

Caro Aluno,

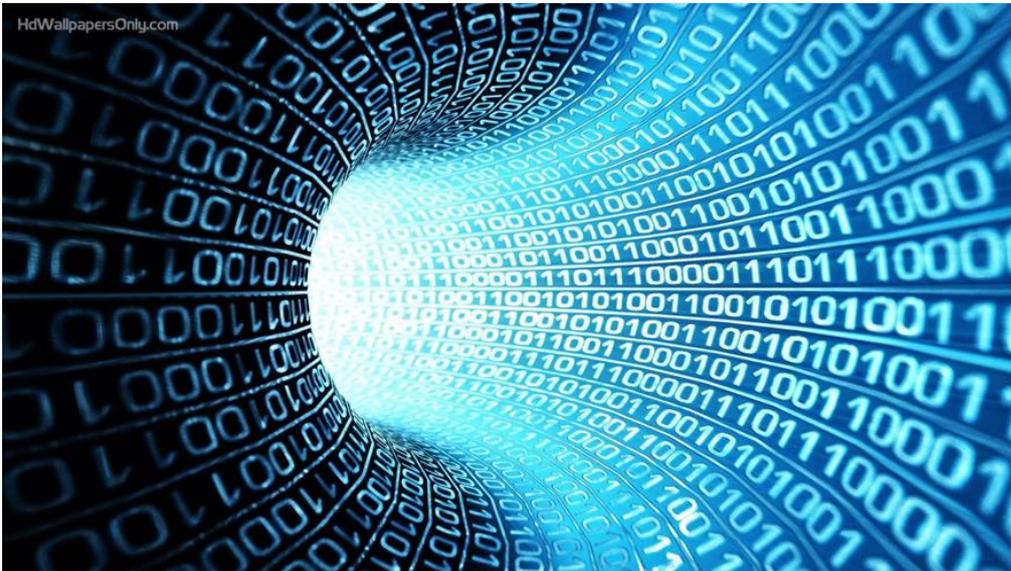
Ao optar pela obtenção da carteira de Aviônicos como especialidade técnica em manutenção de aeronaves você deu o primeiro passo para o ingresso em um mundo que oferece infinidade de opções e soluções para a aviação.

O termo “aviônicos” é derivado da expressão em inglês “aviation electronics” e designa todos os sistemas elétricos e eletrônicos embarcados em aeronaves.

À medida em que os avanços tecnológicos foram se estabelecendo na aviação, o que requer além da funcionalidade a garantia da segurança, mais e mais sistemas foram se automatizando com vistas a facilitar a vida da tripulação que, durante um voo comum, deve receber e processar milhares de informações e a partir daí tomar decisões e atuar nos comandos da aeronave adequadamente.

Hoje em dia, os aviônicos envolvem uma combinação de sensores, processadores, atuadores e instrumentos que devem trabalhar de forma coordenada e precisa para garantir um voo bem sucedido. Esses dispositivos cada dia mais estão inter-relacionados por processadores digitais que vieram simplificar os sistemas, tornando-os mais compactos e versáteis. É essencial para o técnico em manutenção aeronáutica dos dias de hoje conhecer a eletrônica digital e entender o funcionamento de sistemas microprocessados, que a cada dia se tornam mais comuns na aviação.

Prof. Hélio Luis Camões de Abreu



Fonte: <http://www.hdwallpapersonly.com/matrix-wallpapers-hd.html>

MÓDULO I

TÉCNICAS DIGITAIS

INTRODUÇÃO

Caro aluno,

A eletrônica digital tem evoluído a passos largos e à medida que vai adquirindo confiança e garantindo a segurança da operação, vai sendo implementada na aviação.

Não é raro hoje em dia aviões com verdadeiros computadores de bordo controlando as mais diversas funções.

Para entender o funcionamento desses computadores, inicialmente é necessário que se entenda a linguagem de que eles se utilizam para processar as informações. Essa linguagem é uma linguagem matemática que se relaciona com a nossa matemática convencional de uma forma simples, mas que requer especial atenção nas suas operações.

1.1 SISTEMAS DE NUMERAÇÃO

Os sistemas de numeração foram desenvolvidos na história da humanidade atendendo às crescentes necessidades de contagem e enumeração de objetos do dia a dia.

Por conveniência, o homem utilizou-se inicialmente dos dedos como auxiliares na contagem, criando o sistema decimal.

Com o advento do computador, outros sistemas vieram a ser criados, visando maior facilidade de representação interna codificada. Dentre os mais comuns podemos citar os sistemas: Binário, Octal e Hexadecimal, que adequam-se às necessidades ou funções internas de diversos equipamentos.

O sistema decimal, porém, sempre continuou sendo usado como forma de representação numérica convencional.

Sistema Decimal de Numeração

Genericamente, um sistema de numeração se utiliza de símbolos numéricos que, conforme a posição que ocupam na representação de um número, indicam quantas vezes o número contém determinada potência da base.

O sistema decimal é um sistema de base 10, no qual existem dez algarismos para representação de uma quantidade: 0, 1, 2, 3, 4, 5, , 9.

O menor algarismo de uma determinada base representa zero (0) e o maior representa o valor da base menos 1. No caso da base decimal, o maior algarismo representa nove (9), pois $(10 - 1 = 9)$.

No exemplo a seguir temos a representação de um número na sua base. O índice subscrito indica a base

Exemplo 1: $(583)_{10}$

Neste exemplo, os algarismos 5, 8 e 3 indicam quantas vezes a potência respectiva da base 10 correspondente a sua posição estão contidas no número representado. O valor mais à direita, ou seja 3, indica que o número possui 3 vezes 10 na potência 0, ou seja, 3×1 . O segundo valor, ou seja 8, indica que o número ainda possui 8 vezes 10 na potência 1, ou seja, 8×10 . E finalmente o terceiro valor, 5, indica que o número possui 5 vezes a base 10

na potência 2, ou seja, 5×100 . Realizando essas contas, teremos o valor do número representado.

$$(5 \times 100) + (8 \times 10) + (3 \times 1) = 500 + 80 + 3 = 583$$

A princípio, parece bastante intuitivo para nós que estamos acostumados com esse sistema e já vemos 583 como “quinhentos e oitenta e três”, mas imagine que uma criança que acaba de aprender o valor dos números, pode ver esse mesmo símbolo e achar que vale 16 ($5+8+3$)! Lembre-se que ainda não foi falado a ela que, por convenção, se o número 5 está na terceira casa da direita para a esquerda vale 500 e não 5, nem 50, nem 5000. É importante você notar essa diferença, pois ao se deparar com outras bases numéricas, terá de fazer algumas contas para descobrir o valor do número representado, pelo menos até se habituar com eles.

Exemplo 2: $(1592)_{10}$

Decompondo o mesmo teremos:

$$\begin{aligned} 1 \times 10^3 + 5 \times 10^2 + 9 \times 10^1 + 2 \times 10^0 &= 1592 \\ 1000 + 500 + 90 + 2 &= 1592 \end{aligned}$$

Exemplo 3: $(583,142)_{10}$

Notamos que no exemplo 3 temos um número com uma parte fracionária. Vejamos então sua decomposição em potência de dez:

$$\begin{aligned} 5 \times 10^2 + 8 \times 10^1 + 3 \times 10^0 + 1 \times 10^{-1} + 4 \times 10^{-2} \\ + 2 \times 10^{-3} \quad \text{ou} \end{aligned}$$

$$\begin{aligned} 500 + 80 + 3 + 1/10 + 4/100 \\ + 2/1000 \quad \text{ou ainda} \end{aligned}$$

$$\begin{aligned} 500 + 80 + 3 + 0,1 + 0,04 + 0,002 = \\ = 583,142 \end{aligned}$$

Sistema Binário de Numeração

No sistema binário a base é 2 ($b = 2$) e existem apenas dois algarismos para representar uma determinada quantidade: o algarismo 0 (zero) e o algarismo 1 (um).

Para representar a quantidade zero, utilizamos o algarismo 0, para representar a quantidade um, utilizamos o algarismo 1.

No sistema decimal, nós não possuímos o algarismo dez e representamos a quantidade de uma dezena utilizando o algarismo 1 (um) seguido do algarismo 0 (zero). Nesse caso, o algarismo 1 (um) significa que temos um grupo de uma dezena e o algarismo 0 (zero) nenhuma unidade, o que significa dez.

No sistema binário agimos da mesma forma, para representar a quantidade dois, utilizamos o algarismo 1 (um) seguido do algarismo 0 (zero). O algarismo 1 (um) significará que temos um grupo de dois elementos e o 0 (zero) um grupo de nenhuma unidade, representando assim o número dois.

Exemplo:

Seja o número $(1011)_2$ e façamos a sua decomposição em potência só que desta vez a base será dois:

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 1 \times (8)_{10} + 0 \times (4)_{10} + 1 \times (2)_{10} + 1 \times (1)_{10} = (8 + 2 + 1)_{10} = (11)_{10}.$$

Assim, temos que 1011 na base 2 representa o mesmo valor que 11 na base 10.

Apenas para reforçar o conceito, verifique que, na contagem na base 10, você utiliza os algarismos na ordem de valor, até o de maior valor absoluto, ou seja, o 9. Após utilizar o 9, se o número é acrescentado de uma unidade, o que você faz? Isso mesmo! Acrescenta uma unidade na casa da esquerda e substitui o 9 por 0 reiniciando a contagem.

No sistema binário, o procedimento é o mesmo, sendo que o maior valor é 1. Assim, em cada posição você conta 0, 1 e então acrescenta uma unidade na posição imediatamente à esquerda e substitui o 1 por zero, reiniciando a contagem. Vamos contar de 0 até 11 dessa forma:

0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011.

Observe que a representação obtida foi a mesma do exemplo anterior, o que era de se esperar.

Sistema Octal de Numeração

No sistema octal a base é oito e temos oito algarismos para representar qualquer quantidade. Esses algarismos são: 0, 1, 2, 3, ...7.

Para a formação de um número, utilizam-se esses algarismos e toda vez que tivermos uma quantidade igual ao valor da base, soma-se um (1) ao algarismo de valor posicional imediatamente superior como fazemos no sistema decimal e binário.

Notamos também que, em qualquer base o maior algarismo é igual ao valor da base menos um e o número de algarismos é sempre igual ao da base.

Exemplo:

Decompondo o número $(361)_8$ em potência de base oito temos:

$$3 \times 8^2 + 6 \times 8^1 + 1 \times 8^0 = 3 \times (64)_{10} + 6 \times (8)_{10} + 1 \times (1)_{10} = (192 + 48 + 1)_{10} = (141)_{10}.$$

Podemos escrever que, na representação em uma determinada base, o número correspondente a base elevada a uma determinada potência é igual a um 1 seguido de tantos zeros quantos forem os valores das potências, assim temos:

| | | | |
|---------------|--------------|-------------|------------|
| $2^3 = 1000$ | $2^2 = 100$ | $2^1 = 10$ | Na base 2 |
| $10^3 = 1000$ | $10^2 = 100$ | $10^1 = 10$ | Na base 10 |
| $8^3 = 1000$ | $8^2 = 100$ | $8^1 = 10$ | Na base 8 |

No sistema decimal, o número 100 aparece após o número 99 na ordem crescente.

No sistema binário, o número 100 aparece após o número 11 na ordem crescente.

No sistema octal, o número 100 aparece após o número 77 na ordem crescente.

Sistema Hexadecimal de Numeração

No sistema hexadecimal de numeração, a base é dezesseis e dispomos de dezesseis algarismos para representação de uma determinada quantidade de coisas. Como no nosso sistema numérico usual, de base 10, temos apenas 10 algarismos, lançamos mão do recurso

de usar letras do alfabeto para representar quantidades maiores que 9 no sistema hexadecimal. Portanto temos os seguintes algarismos:

0, 1, 2, 3,.....9, A, B, C, D, E e F.

| Hex | Dec |
|-----|-----|
| A | 10 |
| B | 11 |
| C | 12 |
| D | 13 |
| E | 14 |
| F | 15 |

Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Exemplo:

Tomemos o número $(2C0A)^{16}$ e façamos sua decomposição.

$$2 \times 16^3 + C \times 16^2 + 0 \times 16^1 + A \times 16^0 \text{ ou}$$

$$2 \times 4096 + 12 \times 256 + 0 \times 16 + 10 \times 1 =$$

$$= 8192 + 3072 + 0 + 10 =$$

$$= (11274)_{10}$$

Complemento de um Número

O complemento de um número é o número que somado a ele resulta na próxima potência da base. Por exemplo, se tomarmos o número 7, na base 10, a próxima potência da base é $10^1 = 10$. O número que somado a 7 resulta em 10 é 3 ($10-7$), assim, 3 é o complemento de 7 na base 10.

Vejamus outro exemplo na base 10. Qual o complemento de 123? A próxima potência de 10 é 1000 (10^3), pois 100 (10^2) e 10 (10^1) são menores que 123. Pois bem, o número que somado a 123 resulta em 1000 é 877 ($1000-123$), assim, o complemento de 123 na base 10 é 877.

Vimos que na representação em uma base, suas potências são representadas pelo algarismo 1 seguido de tantos zeros quanto for o valor da potência. Logo, as potências de um número, representadas na base daquele número, serão do tipo 100, 10000, 1000, 1000000.

No caso do sistema binário, ou seja, base 2, temos 2, 4 e 8, que são respectivamente as potências 2^1 , 2^2 e 2^3 , são representadas por 10, 100 e 1000 respectivamente.

Lembre-se também de que, na contagem no sistema binário, ao acrescentarmos uma unidade ao valor 1 em uma posição, substituímos esse 1 por 0 e acrescentamos uma unidade à posição imediatamente à esquerda. Assim, se quisermos obter uma potência de 2 a partir de um número qualquer representado na base 2, podemos usar um processo bem mais fácil que determinar a próxima potência e fazer a subtração, tal como fizemos para encontrar o complemento na base 10.

O processo consiste em encontrar qual o número que somado ao número que temos, terá uma representação apenas de 1's. Isso porque, ao somarmos uma unidade a um número com essa representação, teremos a próxima potência de 2.

Vamos a um exemplo, que tornará as coisas mais claras.

Seja o número 1001 na base 2. Qual o número que somado a ele, nos dará um resultado representado apenas por 1's? Isso mesmo! 110, pois $1001 + 0110 = 1111$.

E para o número 111000010101? Quanto devemos somar para obter uma representação só de 1's? 000111101010. Você percebeu que basta inverter dígito a dígito? Isso porque nas posições nas quais já temos 1, somamos 0 e nas posições nas quais temos 0, somamos 1.

O próximo passo é acrescentar uma unidade ao valor encontrado e teremos uma potência de 2.

Vamos começar com um número pequeno.

Para encontrar o complemento de 101, devemos encontrar o número que somado a ele nos dá a próxima potência de 2. Somando 010 a 101 temos 111 e somando 1 a este valor temos 1000. Logo, se somarmos $010 + 1$ ao nosso 101, teremos a próxima potência de 2, que é 1000. Assim, o complemento de 101 é 11 ($010+1$).

Sistematizando:

Para encontrar o complemento de um número binário, invertemos cada dígito e acrescentamos 1 ao resultado obtido.

Qual o complemento de 1011?

$(0100 + 1) = 101$.

Veremos adiante que o complemento pode ser usado para realizar operações de subtração utilizando um somador. Assim, ao invés de implementar um sistema de soma e um de subtração, podemos usar um sistema de soma e um de complemento, que é significativamente mais simples se já possuímos um somador.

Exemplo:

Complemento de $(7)_{10} \ 10 - 7 = 3$

No sistema binário para chegar-se ao complemento, obtém-se primeiramente o falso complemento.

$(1011)_2$
 $0100 \longrightarrow$ Complemento falso

Complemento verdadeiro consiste em somar-se 1 (um) ao complemento falso.

0100
 $+ 1$
 \hline
 0101

Conversão de Bases

Conversão para base decimal - Para convertermos um número representado em qualquer sistema numérico, para o sistema decimal usamos a notação posicional e resolvemos a expressão como na base decimal. Já fizemos isso anteriormente, não fizemos? Seja o número 1101 no sistema binário.

A notação posicional seria:

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

$$1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 =$$

$$8 + 4 + 0 + 1 = (13)_{10}$$

Portanto $(1101)_2 = (13)_{10}$

Como segundo exemplo o número 107 do sistema octal. A notação posicional seria:

$$1 \times 8^2 + 0 \times 8^1 + 7 \times 8^0 =$$

$$1 \times 64 + 0 \times 8 + 7 \times 1 =$$

$$64 + 0 + 7 = (71)_{10}$$

Portanto $(107)_8 = (71)_{10}$

Conversão do sistema decimal para outras bases - Para conversão da base 10 para outras bases, o método consiste em divisões sucessivas pela base desejada, até que o quociente seja nulo. Os restos das divisões indicarão o resultado da conversão, sendo o primeiro resto equivalente ao dígito menos significativo e o último ao mais significativo.

Exemplo 1

Façamos a conversão do número $(934)_{10}$ para base hexadecimal.

$$\begin{array}{r|l} \rightarrow & (10 \quad A) \\ 934 & \overline{16} \\ \hline 1^\circ \text{ resto } \underline{6} & 58 & \overline{16} \\ \hline 2^\circ \text{ resto } \underline{10} & 3 & \overline{16} \\ \hline & 3^\circ \text{ resto } \underline{3} & 0 \end{array}$$

Portanto $(934)_{10} = (3A6)_{16}$

Exemplo 2:

Conversão do número $(76)_{10}$ para a base 8.

$$\begin{array}{r|l} 76 & \overline{8} \\ \hline \underline{4} & 9 & \overline{8} \\ \hline & \underline{1} & 1 & \overline{8} \\ \hline & & \underline{1} & 0 \end{array}$$

Portanto $(76)_{10} = (114)_8$

Exemplo 3:

Conversão do número $(12)_{10}$ para a base 2.

$$\begin{array}{r|l}
 12 & 2 \\
 \hline
 \underline{0} & 6 \\
 & \underline{0} \\
 & | \\
 & 2 \\
 & \hline
 & 3 \\
 & \underline{1} \\
 & | \\
 & 2 \\
 & \hline
 & 1 \\
 & \underline{1} \\
 & | \\
 & 2 \\
 & \hline
 & 0
 \end{array}$$

Portanto $(12)_{10} = (1100)_2$

Lembre-se que o resto da divisão por um número é sempre menor que este número e, de fato, se vamos representar um valor em uma determinada base, deveremos usar algarismos menores que essa base.

Uma outra observação importante é que ao converter um número da base decimal para a hexadecimal, podemos encontrar restos 15, 14, 13, 12, 11 e 10 que nesse caso deverão ser substituídos pelas letras F, E, D, C, B e A, respectivamente, pois elas representam os referidos valores na base 16.

Para converter números de uma base diferente de 10 para outra diferente de 10, aconselhamos a converter inicialmente o número na primeira base para a base 10 e depois fazer a conversão da base 10 para a segunda base.

Contagem nas Diversas Bases

Na tabela de contagem nos sistemas de base decimal, binária, octal e hexadecimal observa-se que um número expresso num sistema de base menor exige maior quantidade de algarismos do que outro, de base maior, para representar a mesma quantidade.

| DECI-MAL | BINARIA | OCTAL | HEXA-DEC. |
|----------|-------------------|--------------|--------------|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | $10 = 2^1$ | 2 | 2 |
| 3 | 11 | 3 | 3 |
| 4 | $100 = 2^2$ | 4 | 4 |
| 5 | 101 | 5 | 5 |
| 6 | 110 | 6 | 6 |
| 7 | 111 | 7 | 7 |
| 8 | $1000 = 2^3$ | $10 = 8^1$ | 8 |
| 9 | 1001 | 11 | 9 |
| 10 | 1010 | 12 | A |
| 11 | 1011 | 13 | B |
| 12 | 1100 | 14 | C |
| 13 | 1101 | 15 | D |
| 14 | 1110 | 16 | E |
| 15 | 1111 | 17 | F |
| 16 | $10000 = 2^4$ | 20 | $10 = 16^1$ |
| - | - | - | - |
| 31 | 11111 | 37 | 1F |
| 32 | $100000 = 2^5$ | 40 | 20 |
| - | - | - | - |
| 63 | 111111 | 77 | 3F |
| 64 | $1000000 = 2^6$ | $100 = 8^2$ | 40 |
| - | - | - | - |
| 99 | 1100011 | 143 | 63 |
| 100 | 1100100 | 144 | 64 |
| - | - | - | - |
| 127 | 1111111 | 177 | 7F |
| 128 | $10000000 = 2^7$ | 200 | 80 |
| - | - | - | - |
| 255 | 11111111 | 377 | FF |
| 256 | $100000000 = 2^8$ | 400 | $100 = 16^2$ |
| - | - | - | - |
| - | - | $1000 = 8^3$ | . |

Fonte: IAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Códigos

Códigos são convenções usadas para indicarmos uma mensagem por meio de uma sequência de dígitos. Em computação, uma das utilidades dos códigos é a representação de caracteres alfanuméricos. Assim, podemos representar uma letra por uma sequência de bits ou também podemos representar um número grande, que em binário seria enorme, por uma sequência de bits que convencionalmente representam algarismos do sistema decimal. São vários os códigos existentes para esta finalidade, sendo que uns apresentam vantagens em relação a outros, de acordo com a aplicação ou funções internas do equipamento.

Código BCD 8421 - A sigla BCD representa as iniciais de "Binary Coded Decimal", que significa uma codificação no sistema decimal em binário. Os termos seguintes (8421) significam os pesos de cada coluna, isto é,

$$8 = 2^3, 4 = 2^2, 2 = 2^1 \text{ e } 1 = 2^0.$$

O valor corresponderá à soma dos pesos onde na coluna houver o "bit" um (1).

| DECIMAL | BCD 8 | 4 | 2 | 1 |
|---------|-------|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 |

Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

O número de "bits" de um código é o número de dígitos binários que este possui.

O código BCD 8421 é um código de 4 "bits".

Utilizando este código, podemos representar, por exemplo, o número 98 por 1001 1000 ao invés de 1100010 que apesar de utilizar menor número de bits é mais difícil de visualizar, além de ser pouco prático em determinadas aplicações, tal como a representação em um painel numérico, por exemplo.

Código excesso 3 - Consiste na transformação do número decimal, no binário correspondente, somando-se a ele três unidades.

Exemplo:

$$(0)_{10} = (0000)_2$$

Somando-se três unidades, teremos 0011

| DECIMAL | EXCESSO 3 |
|---------|-----------|
| 0 | 0 0 1 1 |
| 1 | 0 1 0 0 |
| 2 | 0 1 0 1 |
| 3 | 0 1 1 0 |
| 4 | 0 1 1 1 |
| 5 | 1 0 0 0 |
| 6 | 1 0 0 1 |
| 7 | 1 0 1 0 |
| 8 | 1 0 1 1 |
| 9 | 1 1 0 0 |

Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

O código Excesso 3 é utilizado em circuitos aritméticos.

Código Johnson - Baseia-se no deslocamento de "bits" e é utilizado na construção do Contador Johnson.

| DECIMAL | JOHNSON |
|---------|-----------|
| 0 | 0 0 0 0 0 |
| 1 | 0 0 0 0 1 |
| 2 | 0 0 0 1 1 |
| 3 | 0 0 1 1 1 |
| 4 | 0 1 1 1 1 |
| 5 | 1 1 1 1 1 |
| 6 | 1 1 1 1 0 |
| 7 | 1 1 1 0 0 |
| 8 | 1 1 0 0 0 |
| 9 | 1 0 0 0 0 |

Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Código Gray ou sistema de numeração refletido - Sua principal característica é que, em contagens sucessivas, apenas um "bit" varia.

A codificação Gray é mostrada na tabela a seguir, onde os campos em destaque representam um "espelho" a ser refletido para a contagem seguinte, acrescentando-se um "bit" 1 (um) imediatamente à esquerda.

| DECIM. | BINÁRIO | GRAY |
|--------|-----------|-----------|
| 0 | 0 0 0 0 0 | 0 0 0 0 0 |
| 1 | 0 0 0 0 1 | 0 0 0 0 1 |
| 2 | 0 0 0 1 0 | 0 0 0 1 1 |
| 3 | 0 0 0 1 1 | 0 0 0 1 0 |
| 4 | 0 0 1 0 0 | 0 0 1 1 0 |
| 5 | 0 0 1 0 1 | 0 0 1 1 1 |
| 6 | 0 0 1 1 0 | 0 0 1 0 1 |
| 7 | 0 0 1 1 1 | 0 0 1 0 0 |
| 8 | 0 1 0 0 0 | 0 1 1 0 0 |
| 9 | 0 1 0 0 1 | 0 1 1 0 1 |
| 10 | 0 1 0 1 0 | 0 1 1 1 1 |
| 11 | 0 1 0 1 1 | 0 1 1 1 0 |
| 12 | 0 1 1 0 0 | 0 1 0 1 0 |
| 13 | 0 1 1 0 1 | 0 1 0 1 1 |
| 14 | 0 1 1 1 0 | 0 1 0 0 1 |
| 15 | 0 1 1 1 1 | 0 1 0 0 0 |
| 16 | 1 0 0 0 0 | 1 1 0 0 0 |

Fonte: IAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

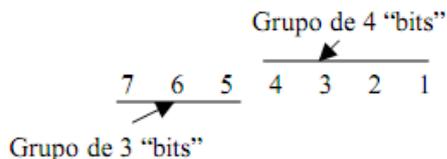
Este tipo de codificação garante que, com a variação de apenas um "bit" de uma contagem para outra, reduzam-se as consequências negativas geradas pela mudança de estado simultânea de registradores.

Código ASCII - O código ASCII é um tipo de codificação BCD, largamente utilizado em computadores digitais e em equipamentos de comunicação de dados. A sigla ASCII é formada pelas iniciais de American Standard Code for Information Interchange (Código Padrão Americano para Intercâmbio de Informações).

Consiste de um código binário de sete "bits" para transferir informações entre computadores e seus periféricos e em comunicação de dados a distância.

Com um total de sete "bits", podemos representar $2^7 = 128$ estados diferentes ou caracteres, que são usados para representar os números decimais de 0 a 9, letras do alfabeto e alguns caracteres especiais de controle.

É formado por dois grupos de "bits", sendo um de 4 "bits" e outro de 3 "bits".



Formato do caráter no Código ASCII

| CARACTER | ASCII | | | | | | |
|----------|-------|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| - | | | | | | | |
| 9 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| - | | | | | | | |
| A | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| B | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| - | | | | | | | |
| Z | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| - | | | | | | | |
| a | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| b | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| - | | | | | | | |
| z | 1 | 1 | 1 | 1 | 0 | 1 | 0 |

Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Exemplos de representações no código ASCII

1.2 OPERAÇÕES BINÁRIAS

A eletrônica em seus primórdios tinha seus cálculos baseados em álgebra convencional, através de sistemas analógicos ou lineares. Máquinas registradoras com sistemas de engrenagens são exemplos desses dispositivos.

Com o advento das portas lógicas e processadores eletrônicos, bem como de sistemas de comunicação e controle digitais, os problemas vieram a ser resolvidos baseados em uma álgebra especial, não linear, mas binária, isto é, baseada em dois valores lógicos V ou F, 0 ou 1, etc.

Aritmética Binária

As regras utilizadas em operações binárias no sistema decimal são também seguidas nas mesmas operações em outros sistemas de numeração. Neste capítulo trataremos de algumas técnicas que tornam mais simples a efetuação destas operações.

Adição no sistema binário - Para efetuarmos a adição no sistema binário, devemos agir como uma adição no sistema decimal, lembrando que no sistema binário temos apenas dois algarismos.

$$\begin{array}{r}
 1\ 0\ 0\ 1\ a \\
 + \\
 1\ 0\ 1\ 0\ b \\
 \hline
 0\ 0\ 1\ 1\ \text{soma} \\
 \leftarrow \text{vai um} \\
 1\ 0\ 0\ 0\ \text{transporte ou} \\
 \text{carreamento}
 \end{array}$$

A tabela mostra a operação soma e o transporte em separado. O símbolo + é o operador soma.

Como $1 + 1 = 10$ no sistema binário, o resultado é 0 (zero) e o transporte para a coluna imediatamente à esquerda é 1 (um). Esse transporte é idêntico ao do sistema decimal, pois quando tivermos uma soma igual ou maior que a base, haverá um "vai um" que será somado ao dígito de valor posicional imediatamente superior.

Exemplo 1:

$$\begin{array}{r}
 a + b \\
 (11)_2 + (10)_2 = (101)_2 \\
 3 + 2 = 5 \\
 \hline
 1\ 1\ a \\
 + \\
 1\ 1\ b \\
 \hline
 1\ 0\ 1\ \text{soma} \\
 \leftarrow \text{vai um}
 \end{array}$$

Exemplo 2:

$$\begin{array}{r}
 a + b \\
 (110)_2 + (111)_2 = (1101)_2 + \\
 6 + 7 = 13
 \end{array}$$

Exemplo 3

$$\begin{array}{r}
 a + b \\
 11001 + 1011 = 100100 + \\
 25 + 11 = 36
 \end{array}$$

Subtração no sistema binário- O método de resolução é análogo a uma subtração no sistema decimal:

$$\begin{array}{r}
 0 \ 0 \ 0 \ 1 \ \text{empréstimo} \\
 \swarrow \\
 0 \ 1 \ 1 \ 0 \ a \\
 - 0 \ 1 \ 0 \ 1 \ b \\
 \hline
 0 \ 0 \ 0 \ 1 \ \text{diferença}
 \end{array}$$

Exemplo 1

$$\begin{array}{r}
 a - b \\
 111 - 100 = 011 \\
 7 - 4 = 3
 \end{array}$$

Exemplo 2

$$\begin{array}{r}
 a - b \\
 1000 - 111 = 1 \\
 8 - 7 = 1
 \end{array}$$

Nos exemplos acima foram utilizados números tais que $a > b$. Consideremos agora um caso com $a < b$.

$$\begin{array}{r}
 a - b \\
 10110 - 11001 = -00011 \\
 22 - 25 = -3
 \end{array}$$

$$\begin{array}{r}
 1 \ 0 \ 1 \ 1 \ 0 \ a \\
 -1 \ -1 \ -1 \ \text{empréstimo} \\
 \hline
 1 \ 1 \ 0 \ 0 \ 1 \ b \\
 \hline
 1 \ 1 \ 1 \ 0 \ 1 \ \text{resultado parcial}
 \end{array}$$

Neste exemplo, seguindo-se as regras anteriores, observa-se que houve um empréstimo de uma posição que não havia no minuendo. Usou-se o algarismo 1 da próxima potência de 2. Nesta situação efetua-se a operação “complemento”, que consiste em inverterem-se os bits "0" por "1" e vice-versa, somando-se "1" em seguida. O resultado obtido é considerado “negativo”.

$$\begin{array}{r}
 1\ 1\ 1\ 0\ 1 \text{ resultado parcial} \\
 +\ 0\ 0\ 0\ 1\ 0 \text{ resultado invertido} \\
 \quad \quad \quad 1 \text{ (complemento)} \\
 \hline
 0\ 0\ 0\ 1\ 1 \\
 0\ 0\ 0\ 1\ 1 \text{ resultado final}
 \end{array}$$

Multiplicação no sistema binário - Procede-se como em multiplicações no sistema decimal, tendo-se como regra básica:

$$\begin{array}{l}
 0 \times 0 = 0 \\
 0 \times 1 = 0 \\
 1 \times 0 = 0 \\
 1 \times 1 = 1
 \end{array}$$

Exemplo 1

$$\begin{array}{r}
 a \times b \qquad \qquad 1\ 0\ 0\ 0\ a \\
 1000 \times 1 = 1000 \quad \times \underline{\qquad\qquad\qquad 1\ b} \\
 8 \times 1 = 8 \qquad \quad 1\ 0\ 0\ 0 \text{ produto}
 \end{array}$$

Exemplo 2

$$\begin{array}{r}
 a \times b \qquad \qquad \qquad 1\ 0\ 1\ 0\ 1\ a \\
 10101 \times 10 = 101010 \quad \times \underline{\qquad\qquad\qquad 1\ 0\ b} \\
 21 \times 2 = 42 \qquad \quad 0\ 0\ 0\ 0\ 0 \\
 \text{produto } 1\ 0\ 1\ 0\ 1\ 0 \quad \underline{\qquad\qquad\qquad 1\ 0\ 1\ 0\ 1}
 \end{array}$$

Divisão no sistema binário - Procede-se como em divisões no sistema decimal.

Exemplo:

$$\begin{array}{l} a : b \\ 111100 : 1100 = 101 \\ 60 : 12 = 5 \end{array}$$

$$\begin{array}{r|l} 1 & 1 & 1 & 1 & 0 & 0 & & 1 & 1 & 0 & 0 \\ - & 1 & 1 & 0 & 0 & & & 1 & 0 & 1 & \\ \hline 0 & 0 & 1 & 1 & 0 & & & & & & \\ - & 0 & 0 & 0 & 0 & & & & & & \\ \hline 0 & 1 & 1 & 0 & 0 & & & & & & \\ - & 1 & 1 & 0 & 0 & & & & & & \\ \hline 0 & 0 & 0 & 0 & & & & & & & \text{resto} \end{array}$$

1.3 ÁLGEBRA DE BOOLE

Em meados do século passado G. Boole desenvolveu um sistema matemático de análise lógica. Esse sistema é conhecido como Álgebra de Boole.

A álgebra booleana é baseada em apenas dois estados. Estes estados poderiam, por exemplo, ser representados por tensão alta e tensão baixa ou tensão positiva e tensão negativa.

Assim como na álgebra linear, encontramos vários tipos de funções, como veremos a seguir. Essas funções, basicamente têm como entrada dois valores, sendo que cada um deles pode ser 0 ou 1, e fornecem como saída um valor, também 0 ou 1. A definição da função irá determinar qual deve ser a saída de acordo com a entrada dada. Por exemplo, uma determinada função pode fornecer saída 1 caso as entradas sejam 0 e 1, mas uma outra função, para essas mesmas entradas, pode fornecer saída 0. É importante lembrar que a mesma função irá sempre fornecer a mesma saída para os mesmos conjuntos de entradas. Ou seja, se uma determinada função fornece saída 1 para as entradas 0 e 1, sempre que se der como entrada 0 e 1, a saída será 1.

Nesse contexto, temos algumas funções que são chamadas de básicas. As funções básicas são as mais simples e das quais se pode obter outras funções por associação. Podemos ter funções compostas, que são formadas pela associação de outras funções. Ainda existe a classe de funções universais que tem a característica de poderem implementar quaisquer das outras funções com apenas um tipo delas. As funções universais são importantes na indústria de componentes, pois pode-se fazer um chip com apenas um tipo de função e o usuário a associa conforme a sua necessidade, criando as funções de que necessita.

Funções Básicas

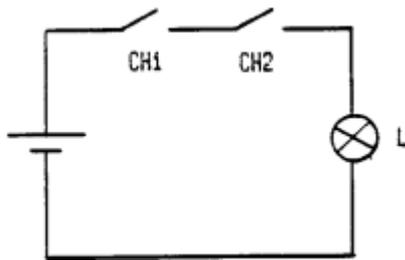
Função "E" ou "AND" -

E aquela que fornece saída 1 somente se ambas as entradas forem 1, caso contrário, a saída é 0.

A função "E" equivale à multiplicação das duas variáveis de entrada.

$S = A \cdot B$ (onde se lê A e B)

Para melhor entendimento veja a figura 1-1.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-1 Circuito da função E ou AND

Convenções:

Chave aberta = 0

Chave fechada = 1

Lâmpada apagada = 0

Lâmpada acesa = 1

Situações possíveis:

Ch 1 aberta e Ch 2 aberta = lâmpada apagada
 $0 * 0 = 0$

Ch 1 aberta e Ch 2 fechada = lâmpada apagada
 $0 * 1 = 0$

Ch 1 fechada e Ch 2 aberta = lâmpada apagada
 $1 * 0 = 0$

Ch 1 fechada e Ch 2 fechada = lâmpada acesa
 $1 * 1 = 1$

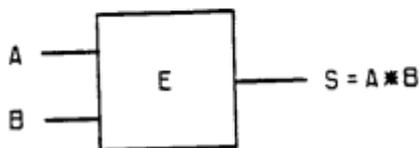
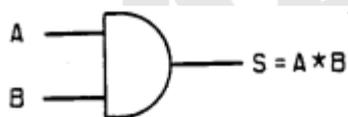
Concluimos que a lâmpada só acenderá quando a Ch 1 e a Ch 2 estiverem fechadas, correspondendo a equação $A \cdot B = S$

Podemos sintetizar o resultado da aplicação de diversas combinações de entrada em uma função através de uma tabela, na qual listamos todas as situações possíveis de entrada e as respectivas saídas.

| A | B | S = A . B |
|---|---|-----------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Tabela Verdade da função E ou AND

Costumamos também representar as funções através de simbologia própria. São as chamadas “portas lógicas”.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

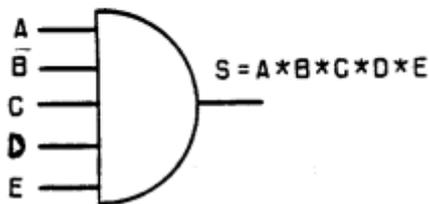
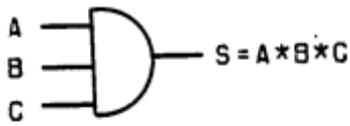
Figura 1-2 Simbologia da função E ou AND

O número de situações possíveis constante na tabela verdade é igual a 2^n , onde N é o número de variáveis de entrada.

Uma porta "E" com duas entradas tem $2^n = 2^2 = 4$ situações possíveis.

Podemos encontrar portas lógicas com três ou mais entradas como mostrado na figura 1-3.

Essas portas são obtidas pela associação de portas de duas entradas.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-3 Portas E ou AND de três e de cinco entradas.

Repare que no caso das portas lógicas “E” com múltiplas entradas, a regra de formação da saída permanece a mesma: “a saída será 1 somente se TODAS as entradas forem 1, caso contrário, a saída será 0”.

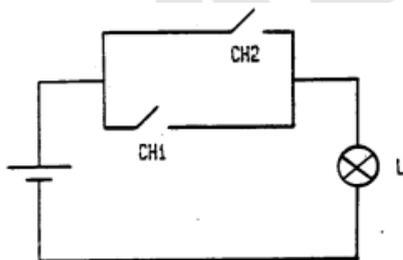
Função “OU” ou “OR” -

É aquela que assume o valor 1 na saída, quando uma ou mais variáveis na entrada forem iguais a 1, e assume o valor 0 se, e somente se, todas as entradas forem iguais a 0.

A saída da função “OU” equivale a soma das variáveis de entrada, considerando que só teremos soma 0 se ambas as parcelas forem 0. No caso de ambas serem 1, teríamos resultado 10, que nesse tipo de análise, é considerado como 1.

$$S = A + B \text{ (S igual a A ou B)}$$

Para melhor compreensão veja a figura 1-4



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-4 Circuito da função OU ou OR

Convenções:

Chave aberta = 0

Chave fechada = 1

Lâmpada apagada = 0

Lâmpada acesa = 1

Situações possíveis:

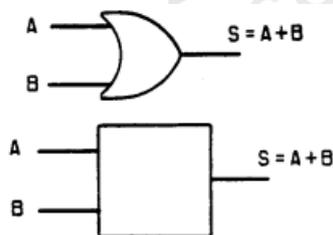
Ch 1 aberta e Ch 2 aberta = lâmpada apagada
0 + 0 = 0

Ch 1 aberta e Ch 2 fechada = lâmpada acesa
0 + 1 = 1

Ch 1 fechada e Ch 2 aberta = lâmpada acesa
1 + 0 = 1

Ch 1 fechada e Ch 2 fechada = lâmpada acesa
1 + 1 = 1

Concluimos que a lâmpada acenderá quando pelo menos uma das chaves estiver ligada, correspondendo à equação $A + B = S$.



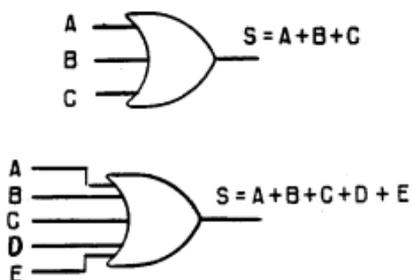
Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-5 Simbologia da função OU ou OR

Tabela Verdade da função OU ou OR

| A | B | S = A + B |
|---|---|-----------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Portas OR também podem ser encontradas com 3 ou mais entradas.

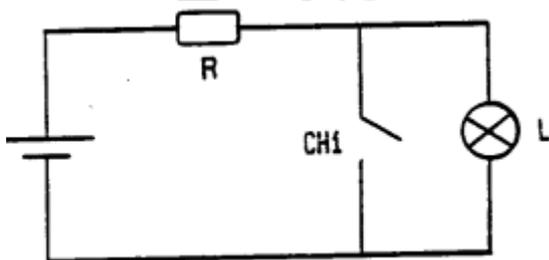


Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-6 Exemplos de portas OR

Função “NÃO” ou “NOT” -

A função NÃO, complemento ou inversão, é aquela que inverte o estado da variável, isto é, "0" inverte para "1" e "1" inverte para "0". Esta função tem exclusivamente apenas uma entrada. Veja a figura 1-7.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-7 Circuito da função NOT ou NÃO

Convenções:

| | |
|-----------------|-----|
| Chave aberta | = 0 |
| Chave fechada | = 1 |
| Lâmpada apagada | = 0 |
| Lâmpada acesa | = 1 |

Situações possíveis:

Chave 1 aberta = lâmpada acesa
0 = 1

Chave 1 fechada = Lâmpada apagada
1 = 0

Tabela verdade da função NOT ou NÃO

| A | \bar{A} |
|---|-----------|
| 0 | 1 |
| 1 | 0 |

Onde \bar{A} representa o inverso de A

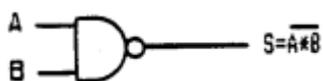
Funções Universais

Função “NÃO E” ou “NAND” -

A função “NÃO E” fornece saída 0 somente se ambas as entradas forem 1, e saída 1 em qualquer outro caso.

É uma combinação das funções "E" e "NÃO", que é representada da seguinte forma:

$S = \overline{A * B}$ (S igual a A e B barrados, ou A e B "not").



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-8 Simbologia NAND

Tabela Verdade da função NAND

| A | B | $\overline{A * B}$ (S) |
|---|---|------------------------|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Função "NÃO OU" ou "NOR" -

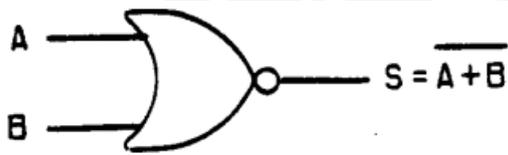
A função "NÃO OU" fornece saída 1 somente se a ambas as entradas forem 0, caso contrário, a saída será 1.

É a combinação das funções OU e NÃO, que é representada da seguinte forma:

$$S = \overline{A + B} \text{ (S igual a A ou B barrado, ou A ou B "not").}$$

Tabela Verdade NOR

| A | B | S = A + B |
|---|---|-----------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-9 Simbologia NOR

Funções XOR ou XNOR - As portas NAND e NOR são ditas portas universais, porque vários circuitos podem ser derivados, utilizando apenas estes tipos de portas.

Podemos criar diversas funções combinando os vários tipos de portas lógicas, dentre elas as denominadas XOR e XNOR.

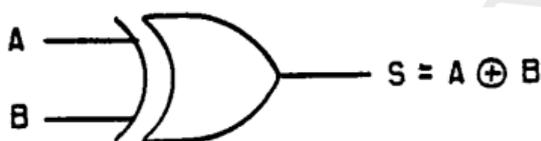
Funções Compostas

Função "OU EXCLUSIVO" ou "XOR"-

A função XOR é chamada de função comparadora de desigualdade, pois terá saída 1 se as entradas forem diferentes e saída 0 se as entradas forem iguais.

Tabela Verdade e Simbologia

| A | B | $S = A + B$ |
|---|---|-------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

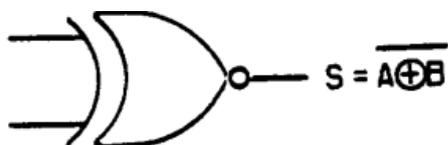
Figura 1-10 Simbologia XOR

Função “NÃO OU EXCLUSIVO” ou “XNOR”-

A função XNOR identifica a igualdade nas entradas, ou seja, gera saída 1 se as entradas forem iguais e saída 0 se forem diferentes.

Tabela Verdade e Simbologia

| A | B | $S = \overline{A + B}$ |
|---|---|------------------------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-11 Simbologia XNOR

As portas XOR e XNOR são denominadas portas COMPARADORAS.

A porta XOR é denominada comparadora de desigualdade e a porta XNOR comparadora de igualdade.

Formas Canônicas

As portas lógicas apresentadas podem ser associadas para produzirem uma ou mais saídas derivadas de uma ou mais entradas que não sigam necessariamente os padrões estabelecidos pelos circuitos padrão dessas portas. Associando adequadamente os diversos tipos de portas lógicas, pode-se implementar praticamente qualquer circuito que tenha uma representação na forma de tabela verdade. Estes circuitos poderão ser representados através de FORMAS CANÔNICAS, que são fórmulas matemáticas representativas da tabela verdade. Através da forma canônica de um circuito, pode-se facilmente descobrir que portas lógicas usar para implementá-lo na prática.

Pode-se representar a tabela verdade por um produto de somas (forma canônica conjuntiva) ou por uma soma de produtos (forma canônica disjuntiva) Existem técnicas regras básicas para se obter as formas canônicas, mas para nossa finalidade de estudo, basta verificar um exemplo.

Seja a tabela verdade a seguir:

| ENTRADAS | | | SAÍDA | |
|----------|---|---|-------|-----------------------------------|
| A | B | C | f | |
| 0 | 1 | 0 | 1 | $\overline{A} * B * \overline{C}$ |
| 0 | 1 | 1 | 1 | $A * B * C$ |
| 1 | 0 | 0 | 1 | $A * \overline{B} * \overline{C}$ |
| 1 | 1 | 0 | 1 | $A * B * \overline{C}$ |

Existem 8 entradas possíveis: 000, 001, 010, 011, 100, 101, 110 e 111. Dessas entradas, podemos perceber pela tabela que algumas devem produzir saída 1, são elas: 010, 011, 100 e 110. Como descobrir qual o circuito implementa esta saída?

Sabemos que a porta E, representada matematicamente pelo produto, fornece saída 1 se todas as entradas forem 1. Então, invertemos as entradas que são 0 em cada linha e as multiplicamos. Assim, temos quatro situações nas quais a saída será 1. Mas desejamos que a saída seja 1 se qualquer uma dessas situações ocorrerem, ou seja, se ocorrer uma OU outra. A função OU é representada pela soma matemática. Assim, temos a implementação do circuito pela soma dos produtos anteriormente encontrados.

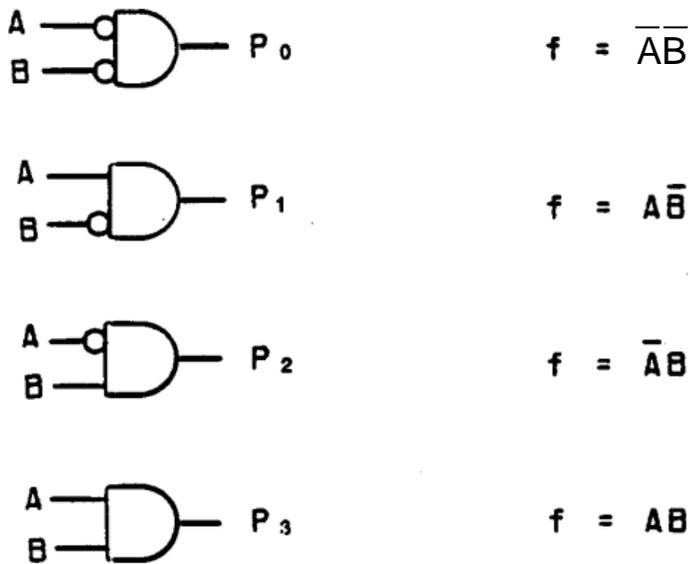
$$f = \overline{A}BC + A\overline{B}C + ABC + \overline{A}\overline{B}\overline{C} \quad \text{Forma Canônica}$$

Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Na função f acima, vemos que se qualquer dos produtos de três entradas for 1, a saída será 1. Mas cada produto corresponde a uma linha da tabela. Verifiquemos a primeira linha, pois o raciocínio para as demais é análogo. Na primeira linha temos que a saída é 1 se as entradas forem $A=0$, $B=1$ e $C=0$. Ou seja, se o complemento ou inversão de A for 1, B for 1 e complemento de C for 1. Logo, concluímos que a saída será o produto $\overline{A}B\overline{C}$. Onde a letra com a barra em cima representa o complemento ou inverso dela, ou seja, se A é 1, A com barra é 0. Vemos pela fórmula apresentada que quando $A=0$, $B=1$ e $C=0$, o resultado será 1 e que se qualquer dessas variáveis for alterada, o resultado será 0. Isso é a representação da primeira linha da tabela verdade. Mas se qualquer das linhas for 1, o resultado deve ser 1, então, somamos todas as fórmulas correspondentes a cada linha. A soma só será 0 se todos os resultados forem 0.

Circuitos geradores de produtos canônicos- São circuitos que geram as formas canônicas básicas, onde são estabelecidas e combinadas as entradas para todas as variações. Estabelecida a forma canônica de um circuito por fórmula matemática, basta implementar os complementos ou inversos com portas inversoras, os produtos com portas E, e as somas com portas OU.

Se quisermos gerar os produtos canônicos possíveis com " n " variáveis, necessitaremos de 2^n portas de " n " entradas.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-12 Exemplos com duas variáveis

1.4 CIRCUITOS DE COMUTAÇÃO

Os circuitos lógicos de um equipamento precisam ser compatíveis às necessidades do projeto. Na execução de funções lógicas, as entradas e saídas são variáveis, requisitando padrões de comutação.

Dentre as características dos circuitos de comutação, podemos citar as tensões de referência para os valores 0 e 1, o tempo de propagação do sinal, a potência dissipada pelo dispositivo, a imunidade à ruídos e a quantidade de saídas suportada pelo dispositivo.

Níveis Lógicos

Os níveis lógicos são as tensões designadas como estado "1" e estado "0" binários, para certo tipo de circuito digital.

Os valores nominais para os dois níveis são bem determinados mas, na prática, os valores obtidos podem variar, devido à tolerância dos componentes internos do circuito integrado, variações da fonte de alimentação, temperatura e outros fatores. Geralmente os fabricantes fornecem os valores máximos e mínimos admitidos para cada um dos níveis lógicos.

Por exemplo, pode ser definido que 5 Volts sejam interpretados como nível lógico 1 e 0 Volt como 0. Mas e se o circuito anterior, por questões de perdas não conseguir suportar 5 V e der uma entrada de 4,7 V? Essa entrada ainda será interpretada como 1?

É muito importante conhecer os níveis lógicos de um determinado tipo de circuito integrado pois, deste modo, ao trabalhar com equipamentos digitais, será fácil identificar os estados lógicos das entradas e saídas.

Tempo de Propagação

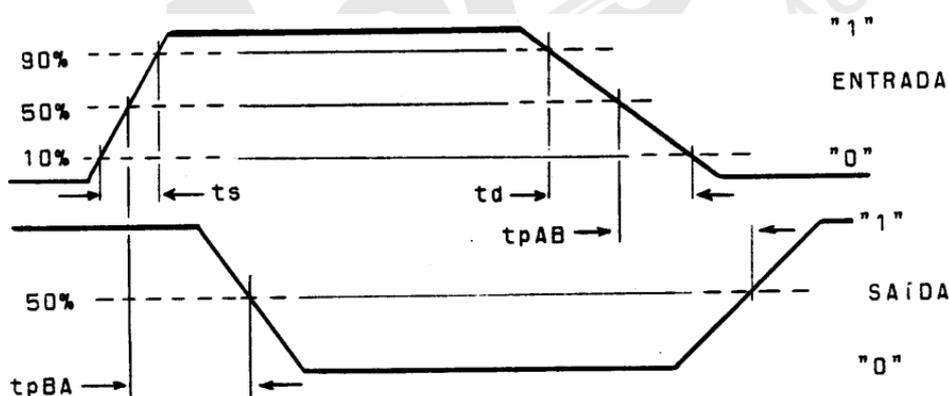
O tempo de propagação (Propagation Delay) é a medida do tempo de operação de um circuito lógico. A velocidade de operação é uma das características mais importantes e, para a maior parte das aplicações digitais, uma alta velocidade de operação, ou seja, um baixo tempo de propagação é desejável.

O tempo de propagação exprime o espaço de tempo necessário para que a saída de um circuito digital responda a uma mudança de nível de entrada. É composto pelo acúmulo de tempos de transição e retardo associados a qualquer circuito lógico.

Quando a tensão de entrada de um circuito digital muda de "0" para "1", ou vice-versa, a saída deste circuito responderá após certo período de tempo.

A figura 1-13 dá um exemplo de tempo de propagação. Temos aí representada a entrada de um circuito digital e, logo abaixo, a saída correspondente.

Veja que a transição de "0" para "1" na entrada ocasiona uma transição de "1" para "0" na saída e que a transição de saída ocorre certo tempo após a transição de entrada. Isto é que chamamos de tempo de propagação.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-13 Tempo de propagação

O tempo de propagação (t_p) é medido geralmente entre os pontos de 50% de amplitude, da transição inicial da entrada para a transição inicial da saída ou da transição final da entrada para a transição final da saída.

Observe ainda que existem dois tipos de tempo de propagação: um deles ocorre quando a entrada passa do nível baixo para o nível alto (t_{pBA}), e o outro quando a entrada passa de alto para baixo (t_{pAB}). Os dois tipos de tempos de propagação são geralmente diferentes, devido às características dos circuitos lógicos.

Os tempos de subida e descida dos pulsos de entrada e saída também são importantes. Define-se tempo de subida (t_s), como o período de tempo tomado pelo pulso para subir de 10% a 90% de sua amplitude máxima. O tempo de descida (t_d) é o necessário para o pulso descer de 90% a 10% dessa mesma amplitude.

Para a maioria dos circuitos integrados digitais, os tempos de subida e descida são bastante reduzidos. Podem ser conseguidos tempos de transição de 1 nanosegundo. Alguns tipos de circuitos digitais modernos apresentam tempos de propagação que chegam a apenas algumas dezenas de nanosegundos. Os tempos de transição são normalmente menores que os tempos de propagação.

Os tempos de propagação podem variar consideravelmente devido a tolerâncias de fabricação, fiação, etc. e são cumulativos. Quando portas e outros circuitos lógicos combinacionais são ligados uns aos outros, os tempos de propagação se somam.

Potência Dissipada

É a potência consumida por um circuito lógico operando em tempos iguais nos estados "0" e "1" isto é, num ciclo de carga de 50%.

A potência total dissipada por um circuito é uma consideração importante no projeto de um equipamento digital, pois uma elevada dissipação em potência significa um grande consumo de energia, o que irá determinar o tamanho e o custo da fonte de alimentação.

O calor liberado pelos circuitos também está relacionado à potência dissipada pelos mesmos e às vezes torna-se necessário o uso de sistemas de refrigeração, para garantir o bom funcionamento dos dispositivos.

A potência dissipada por uma porta pode variar da ordem de alguns microwatts até 100 miliwatts.

Compromisso Velocidade-Potência

As duas características descritas, velocidade potência dissipada, são diretamente interdependentes em todos os tipos de circuitos lógicos digitais. A relação entre elas é tal que a velocidade se apresenta proporcional à potência dissipada, ou seja, quanto mais rápida a comutação de um circuito lógico, maior será a potência dissipada.

Os circuitos lógicos de alta velocidade empregam transistores bipolares não saturados que, associados a resistências internas de baixos valores, produzem um alto consumo de potência.

Os circuitos integrados do tipo MOS (Metal-Oxide-Semicondutor), consomem um mínimo de potência devido as altas impedâncias inerentes a esses componentes. No entanto, refletem em velocidades de comutação muito baixas, limitando sua operação a frequências também baixas. Pelo seu consumo bastante reduzido, adequam-se perfeitamente aos equipamentos portáteis operados a bateria, nos quais a alta velocidade não seja fator relevante.

Imunidade a Ruídos

A imunidade a ruídos é uma medida da característica do funcionamento do circuito não ser influenciado por sinais externos. Considera-se ruído qualquer sinal estranho, gerado externamente ou pelo próprio equipamento e que é acrescentado ou superposto aos sinais padrão do sistema.

Esse ruído pode ser um nível de tensão variando lentamente, picos de tensão, ou sinais de alta frequência e pequena duração. O ruído pode provocar uma comutação no circuito lógico, para um estado indesejável num momento impróprio.

A imunidade da maioria dos circuitos lógicos é de aproximadamente 10% a 50% do valor da tensão de alimentação. Isto significa que um pico será rejeitado, caso sua amplitude seja inferior a 10% ou 50% da tensão de alimentação.

A imunidade a ruídos é uma consideração de grande importância, porque a maioria dos sistemas digitais gera uma quantidade considerável de ruído em comutações de alta velocidade. Além disso, muitos equipamentos digitais são utilizados em ambientes

industriais de ruído intenso, onde interferências provenientes das linhas de força e de outros equipamentos elétricos podem causar falsas comutações nos circuitos lógicos.

"Fan-out"

"Fan-out" é uma característica que indica o quanto de carga pode ser ligado à saída de um circuito digital. É geralmente expresso em termos de número de cargas padrão que a saída de uma porta lógica aceita, sem afetar o nível lógico nominal, velocidade, temperatura ou outras características.

Uma porta lógica pode, por exemplo, apresentar um "Fan-out" igual a 10, o que indica que até dez entradas de portas poderiam ser ligadas à saída deste circuito lógico, sem afetar a sua operação.

1.5 FAMÍLIAS DE CIRCUITOS LÓGICOS

Como podem ser notados, os circuitos lógicos possuem características que deverão ser observadas durante o projeto, para que o mesmo utilize os componentes adequados à aplicação do equipamento. De acordo com estas características, os circuitos lógicos são agrupados em famílias.

Entende-se por famílias de circuitos lógicos, os tipos de estruturas internas que permitem a confecção dos blocos lógicos em circuitos integrados.

Dentre as famílias podemos destacar:

- RTL (Resistor-Transistor Logic).
- DTL (Diode-Transistor Logic).
- HTL (High Threshold Logic).
- TTL (Transistor-Transistor Logic).
- ECL (Emitter-Coupled Logic).
- C-MOS (Complementary MOS).

Tecnologia MOS

A família MOS (Metal Oxide Semiconductor) compõe-se de circuitos formados por MOSFETS, que são transistores de efeito de campo construídos a partir da tecnologia MOS, apresentando como características o baixo consumo e uma alta capacidade de integração, isto é, a colocação de uma grande quantidade de componentes lógicos num mesmo encapsulamento.

Comparação entre Famílias

Família RTL (Resistor-Transistor Logic)

Utiliza transistores e resistores, sendo das primeiras famílias utilizadas, formando portas NOR como principal bloco lógico.

Suas principais características são:

- Possui boa imunidade a ruídos;
- Tempo de propagação da ordem de 12 ns;
- Potência dissipada por bloco lógico, da ordem de 10 mW;
- Alimentação $3V \pm 10\%$.

Família DTL (Diode-Transistor Logic)

Utiliza diodos e transistores, sendo um desenvolvimento da lógica de diodos, permitindo a formação de blocos "E", "OU", "NAND" e "NOR".

Suas principais características são:

- Imunidade a ruídos da ordem de 0,8V;
- Tempo de propagação da ordem de 30ns;
- Potência dissipada da ordem de 10 mW por bloco lógico;
- Alimentação $5V \pm 10\%$.

Família HTL (High Threshold Logic)

Utiliza diodos e transistores como a DTL, acrescentando um diodo Zener, para aumento do nível de entrada, estabelecendo alta imunidade à ruídos.

Suas principais características são:

- Alta imunidade a ruídos;
- Alto tempo de propagação;
- Alta potência dissipada, da ordem de 60 mW.

Família TTL (Transistor-Transistor Logic)

É oriunda da família DTL, porém utilizando transistores multiemissores, que permitem a eliminação dos diodos e resistores de entrada, trazendo maior velocidade e menor custo, tornando-a das mais difundidas.

Suas principais características são:

- Boa imunidade a ruídos;
- Tempo de propagação da ordem de 10 ns;
- Potência dissipada da ordem de 20 mW por bloco lógico;
- Faixa de temperatura de 0° a 75° C.

* Identificação Comercial - série 74

Família ECL (Emitter Coupled Logic)

Utiliza nos circuitos, acoplamento pelo emissor dos transistores, o que os faz operar em regime de não saturação, permitindo a mais alta velocidade de comutação dentre as famílias.

Suas principais características são:

- Boa imunidade a ruídos;
- Muito baixo tempo de propagação, da ordem de 3 ns;
- Potência dissipada da ordem de 25 mW por bloco;
- Alimentação 5,2 V \pm 20%.

Família C-MOS (Complementary MOS)

É uma variação da família MOS, consistindo basicamente de pares de canais MOS complementares. Esta técnica tem como vantagem em relação ao MOS convencional, uma maior velocidade de comutação, da ordem de 80 ns, contra 300 ns.

Suas principais características são:

- Baixa dissipação de potência, da ordem de $10\mu\text{w}$;
- Alto índice de integração;
- Alta imunidade a ruídos;
- Ainda elevado tempo de propagação, da ordem de 60 a 70 ns;
- Larga faixa de alimentação de 3 a 18 V.

Métodos de Fabricação

Existem três formas básicas de se fabricar circuitos integrados. O método mais difundido é o chamado monolítico. Os outros são: o de película fina, ou de película espessa e o híbrido.

Método Monolítico - O circuito integrado monolítico é construído inteiramente de um único pedaço de silício semiconductor, chamado pastilha ou "chip". Materiais semicondutores são difundidos sobre esta base, dando origem a diodos, transistores e resistores. Como resultado, o circuito inteiro, com todos os componentes e interligações, forma-se sobre uma base única, dando origem ao termo "monolítico".

Os circuitos integrados monolíticos digitais se subdividem em dois tipos básicos: os Bipolares e os do tipo MOS, diferindo fundamentalmente no tipo de transistor utilizado. Os circuitos MOS, são mais fáceis de obter e ocupam menos espaço, desta forma é possível incluir muito mais circuitos num "chip" apresentando uma maior densidade de componentes e custo menor.

Método de película fina ou espessa - Neste método, os circuitos são obtidos depositando-se os materiais sobre uma base não condutora, como a cerâmica, formando

resistores, capacitores e indutores. Normalmente os dispositivos semicondutores não são obtidos por este processo.

Método Híbrido - O circuito integrado híbrido é formado pela combinação de circuitos monolíticos e circuitos de película. Os híbridos oferecem uma grande variedade de combinações entre circuitos integrados e componentes, resultando em várias funções que não poderiam ser obtidas com circuitos integrados específicos.

Classificação dos Circuitos Integrados Digitais

Os circuitos integrados digitais podem ser classificados basicamente em três grupos:

SSI - Small Scale Integration (Integração em Pequena Escala);

MSI - Médium Scale Integration (Integração em Média Escala);

LSI - Large Scale Integration (Integração em Grande Escala).

Os circuitos SSI representam a forma mais básica e simples dos circuitos integrados: são amplificadores ou portas, que realizam uma função elementar, devendo ser interligados externamente, caso queiramos formar circuitos funcionais completos.

Os circuitos MSI são mais complexos, formados por várias portas interligadas, compondo circuitos funcionais completos, a maioria contendo doze ou mais circuitos, desempenhando funções como um decodificador, um contador, um multiplexador.

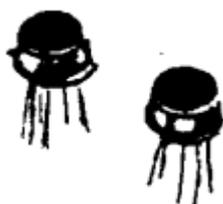
Os circuitos LSI contêm 100 ou mais portas ou dispositivos equivalentes, formando grandes circuitos funcionais, equivalentes a vários circuitos MSI. Seu maior campo de aplicação é o das memórias e microprocessadores.

Encapsulamento de Integrados

Atualmente há três tipos de encapsulamento para acomodar "chips":

- TO5 ou "caneca".
- FLAT PACK ou invólucro chato.
- DIP (Dual In-line Pack) ou em linha dupla.

Encapsulamento TO5 - Esta foi a primeira versão de encapsulamento usada em circuitos integrados, a partir de um invólucro padrão para transistores. Sua principal vantagem reside em seu grande poder de dissipação de calor, e por esta razão encontra maior aplicação nos circuitos lineares.

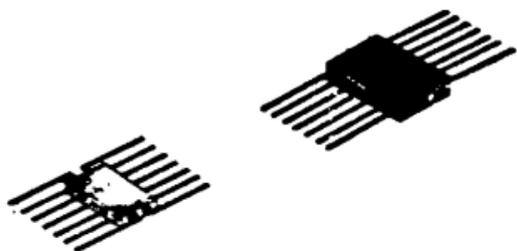


Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-14 Encapsulamento tipo caneca (TO5)

Encapsulamento chato (Flat Pack) - Apresenta o menor tamanho entre todos eles, sendo assim empregado onde se deseja uma elevada densidade de componentes na placa. Os invólucros têm um formato achatado e são apropriados para soldagem sobre circuitos impressos, podendo ficar muito próximos um dos outros.

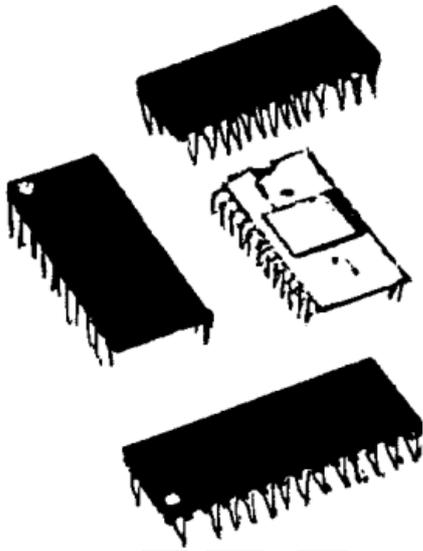
Encontram aplicações onde o espaço é crítico, como por exemplo, em aviação, sistemas militares de alta confiabilidade e equipamentos industriais especiais.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-15 Encapsulamento “chato” (Flat Pack)

Encapsulamento DIP (Dual In-Line Package) - O DIP ou encapsulamento em linha dupla, é assim chamado porque exhibe duas fileiras paralelas de terminais, tendo sido projetado para adaptar-se às máquinas de inserção automática de componentes em placas de circuitos impressos.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-16 Encapsulamento em linha dupla (DIP)

Pode ser encontrado desde o MINI-DIP de oito pinos, ao gigante de quarenta pinos. A maioria dos SSI apresenta-se em encapsulamentos de 8, 14 ou 16 pinos, enquanto o MSI com 14, 16 ou 24 pinos. Finalmente os LSI são encontrados mais frequentemente com 24, 28 ou 40 pinos.

1.6 CIRCUITOS COMBINACIONAIS

Conceitos

Circuitos digitais são circuitos que recebem entradas, processam os sinais da entrada por intermédio de portas lógicas e outros dispositivos e fornecem saídas, também em sinais digitais como produto. Os circuitos digitais podem ser combinacionais ou sequenciais.

Circuito lógico combinacional, ou simplesmente circuito combinacional, é aquele cujo estado de saída é uma função exclusiva das combinações possíveis das variáveis de entrada.

Os circuitos lógicos combinacionais que iremos estudar são constituídos pelas portas lógicas básicas, universais e comparadoras, associadas adequadamente para fornecer na saída o resultado do processamento desejado das variáveis de entrada.

Codificadores e Decodificadores

Um Codificador tem a função de "tradutor" de um código (linguagem) conhecido ou comum, para um código desconhecido ou incomum.

Um Decodificador tem a função de "tradutor" de um código (linguagem) desconhecido ou incomum, para um código conhecido ou comum.

Da relação dos "bits" 0 e 1 do sistema binário, com os estados lógicos 0 e 1, surgiu a aplicação de circuitos lógicos em calculadoras, com operações realizadas no sistema binário. Cabe aqui uma pergunta. Por que não empregar nas calculadoras eletrônicas circuitos que realizem operações diretamente no sistema decimal?

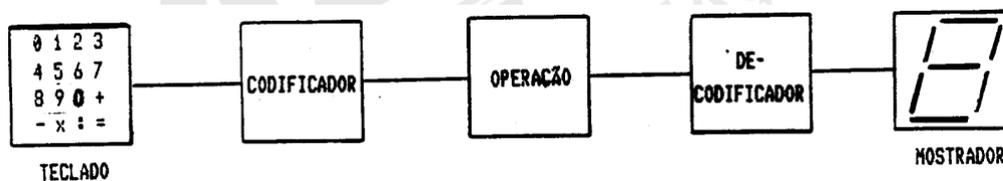
A resposta é simples: os circuitos teriam que discernir 1 entre 10 níveis diferentes, contra 1 entre 2, o que os tornaria complicados, caros e volumosos.

A máquina irá operar mais facilmente no sistema binário, mas para facilitar sua operação, a entrada dos dados a serem calculados e o resultado das operações, devem estar na forma decimal, que é o código comum aos humanos.

Vê-se, então, a necessidade de componentes lógicos conversores, dotados de circuitos codificadores e decodificadores que realizem as conversões decimal-binário ou binário-decimal.

Estes codificadores e decodificadores são na verdade circuitos lógicos combinacionais cujas saídas dependem dos estados lógicos das entradas.

Um número decimal pode ser codificado de tal maneira que a operação digital possa ser desempenhada utilizando-se números binários. A conversão de um sistema para o outro é realizada por circuitos codificadores. O circuito que tem a função inversa é denominado decodificador.

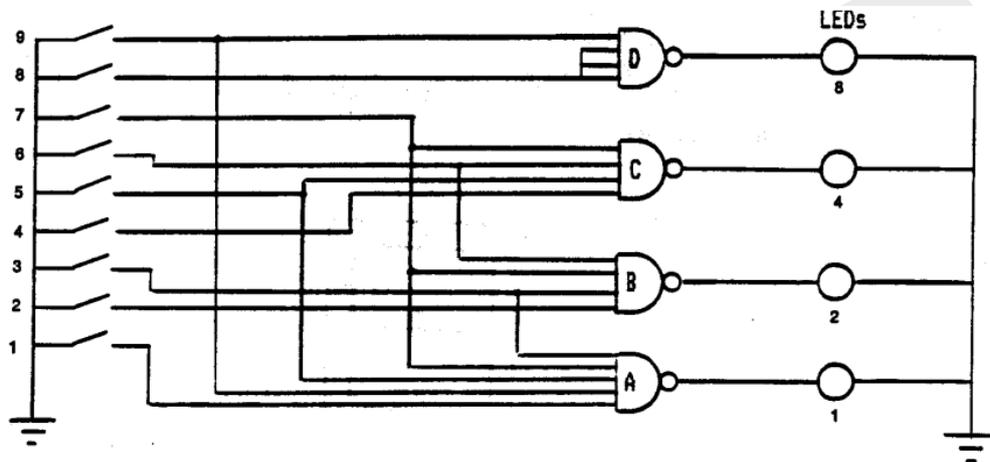


Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-17 Diagrama bloco básico de uma calculadora

Circuito Codificador - Um codificador binário consiste de um sistema de portas lógicas que convertem um número decimal em binário.

Na figura 1-18 a seguir, temos um circuito codificando um grupo de chaves que representam números decimais, para fornecimento de um código binário de 4 "bits".



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-18 Circuito codificador

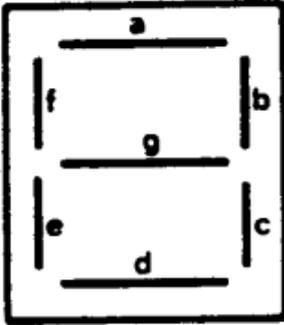
Quando todas as chaves estiverem abertas, teremos nível "1" (alto) na entrada de todas as portas NAND, ocasionando todas as saídas em nível "0" (baixo), gerando o binário "0000". Ao pressionarmos a chave "1", um nível baixo na entrada da porta "A", ocasionará um alto em sua saída, indicando o binário "0001".

Pressionando a chave "2", teremos um nível alto da saída da porta "B", acendendo o Led correspondente, indicando o binário "0010".

Acionando a chave "6", as portas B e C terão saída alta, ocasionando a indicação binária "0110".

"Display" de segmentos - A apresentação do resultado no sistema binário, não é propriamente adequada para o uso comum, pois nem todos compreendem esta representação. São necessários então, componentes que nos proporcionem uma forma simples de representação.

Os "displays" de sete segmentos são componentes mais comuns para representação numérica. Estes "displays" possibilitam representarmos números decimais e alguns outros símbolos. São compostos por segmentos que podem ser ativados individualmente, permitindo combinações.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-19 "Display" de sete segmentos

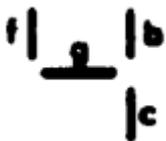
Para formação do algarismo zero (0), necessitamos ativar os segmentos "a", "b", "c", "d", "e" e "f", desativando o segmento "g".



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-20 Representação do zero (0)

A representação do algarismo quatro (4) requer a ativação dos segmentos "b", "c", "f" e "g".



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-21 Representação do quatro (4)

Circuito Decodificador - Como o código interno normalmente utilizado é o binário, torna-se necessário um decodificador que permita a ativação individual dos segmentos. Este decodificador possui a seguinte tabela verdade:

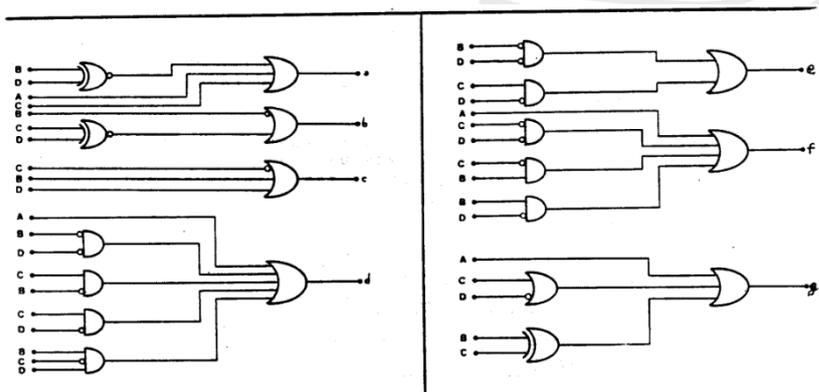
| DECIMAL | BCD 8421 | | | | CÓDIGO DE 7 SEGMENTOS | | | | | | |
|---------|----------|---|---|---|-----------------------|---|---|---|---|---|---|
| | A | B | C | D | a | b | c | d | e | f | g |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |

Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Você pode perceber aí dois conceitos já estudados: códigos e funções lógicas.

O código é a representação do número no sistema binário em uma cadeia de zeros e uns que irá acender o display de forma a desenhar o número decimal. Para fazer essa transformação, usamos uma função lógica para cada bit de saída que depende de todo o conjunto de bits de entrada. Ao implementarmos essa função com portas lógicas, temos um circuito decodificador.

As funções da tabela poderão ser obtidas através do circuito da figura 80.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-22 Decodificador para "Display" de sete segmentos

Somadores e Subtratores

Tal como os codificadores e os decodificadores, os circuitos somadores e subtratores também fornecem resultados que dependem unicamente das variáveis de entrada.

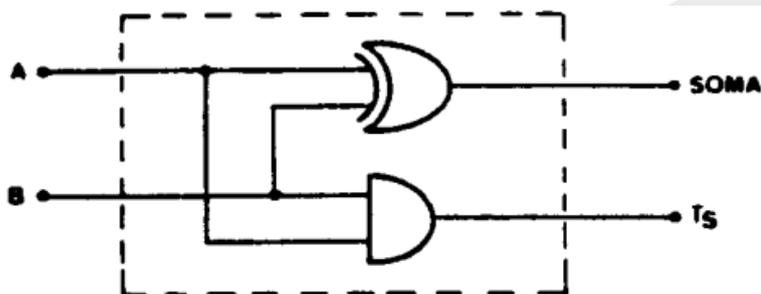
Somador - Se quisermos somar dois dígitos binários, teremos duas entradas para o circuito de soma, havendo quatro combinações para estas entradas: (0 + 0), (0 + 1), (1 + 0) e (1 + 1).

Na aritmética binária, "1" mais "1" (1+1) é igual a 0 (zero) e um dígito 1 é transportado para a coluna da esquerda.

| A + B = S | T |
|-----------|---|
| 0 + 0 = 0 | 0 |
| 0 + 1 = 1 | 0 |
| 1 + 0 = 1 | 0 |
| 1 + 1 = 0 | 1 |

Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

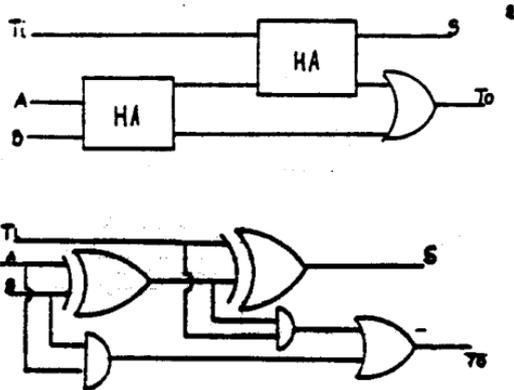
De acordo com a tabela verdade, a função soma (S) pode ser executada por uma porta XOR (OU EXCLUSIVA), e a função transporte (T) por uma porta AND.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-23 Meio Somador (Half Adder)

Para somar as colunas menos significativas, será suficiente o circuito acima, com duas entradas, que é denominado Meio Somador (Half Adder), porém ao somarmos as demais colunas teremos que considerar uma terceira entrada, o transporte da coluna anterior.

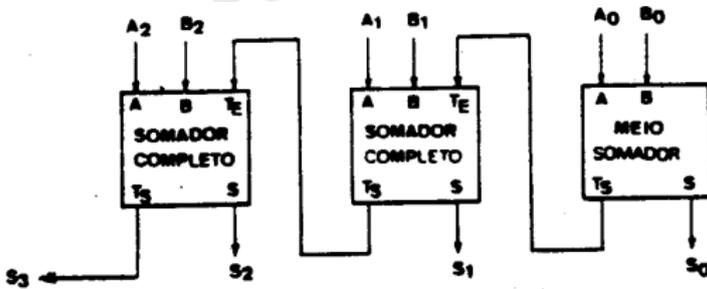


Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-24 Somador Completo (Full Adder)

O somador que executa a soma dos dígitos mais significativos e que possui uma terceira entrada para o transporte, é denominado Somador Completo (Full Adder), sendo formado por dois "Half Adders" (HÁ) e uma porta OR.

Um somador será composto de vários "Full Adder" (HÁ), para a coluna menos significativa.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-25 Somador para dois dígitos de três "bits"

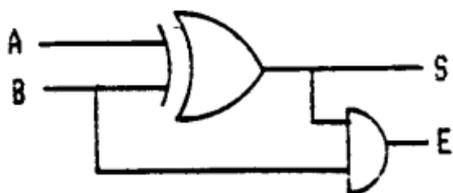
Subtrator - Na aritmética binária, "0" menos "1" (0 - 1) é igual a "1" e um dígito 1 é tomado emprestado da coluna da esquerda.

| A | B | = | S | E |
|---|---|---|---|---|
| 0 | 0 | = | 0 | 0 |
| 0 | 1 | = | 1 | 1 |
| 1 | 0 | = | 1 | 0 |
| 1 | 1 | = | 0 | 0 |

Toma 1
emprestado

Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

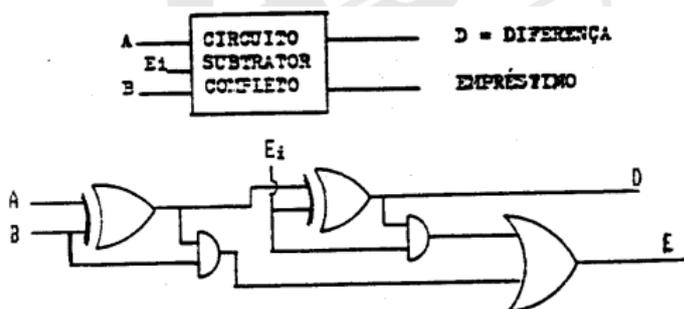
Analogamente ao somador, para subtrairmos dígitos na coluna menos significativa, fazemos uso de um Meio Subtrator (Half Subtractor) e, para as demais colunas, utilizamos o Subtrator Completo (Full Subtractor).



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

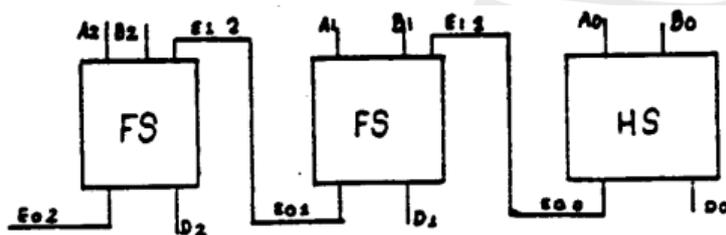
Figura 1-26 Meio Subtrator (Half Subtractor)

Um subtrator será composto de vários "Full Subtractors" (FS), para as colunas mais significativas e um "Half Subtractor" (HS), para a coluna menos significativa.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-27 Subtrator Completo (Full Subtractor)



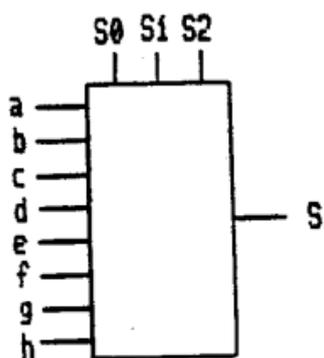
Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-28 Subtrator para dois dígitos de três "Bits"

A subtração também pode ser realizada utilizando-se o somador e um circuito que calcule o complemento do subtraendo. Essa modalidade de subtração é particularmente útil quando o subtraendo é maior que o minuendo e o resultado é negativo.

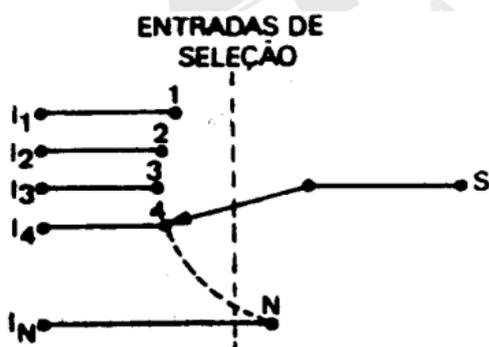
Multiplexadores e Demultiplexadores

Os Multiplexadores são componentes que permitem selecionar um dado, dentre diversas fontes, como uma chave seletora de várias posições.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-29 Multiplexador



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

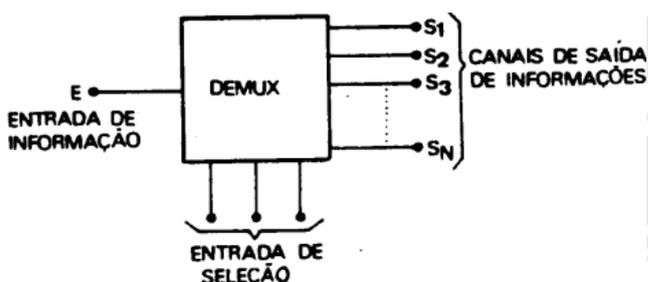
Figura 1-30 Circuito equivalente de um multiplexador

Através do circuito equivalente verificamos que a saída poderá estar ligada a qualquer das entradas, bastando posicionar a chave. No multiplexador, a seleção é feita de acordo com o valor digital das entradas de seleção (S0), (S1) e (S2), com pesos binários 1, 2 e 4, respectivamente. As entradas de "A" a "H", corresponderão a valores decimais de 0 a 7. Na saída, teremos o nível da entrada, cujo valor decimal corresponde ao valor binário das entradas seletoras.

Podemos exemplificar a aplicação de um multiplexador em um sistema de áudio que recebe sinais de duas ou três frequências de comunicação externa, um sistema fechado de

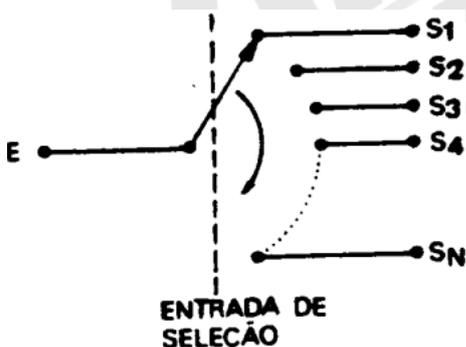
comunicação interna e de mais dois ou três aparelhos de navegação, por exemplo. O piloto irá selecionar apenas um de cada vez para ouvir nos fones ou autofalantes.

Os Demultiplexadores são componentes que distribuem o nível de uma única entrada, para uma, dentre as várias saídas, de acordo com o valor binário das entradas seletoras.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-31 Demultiplexador



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-32 Circuito equivalente de um demultiplexador

Analogamente ao exemplo dado para o multiplexador, o demultiplexador poderá ser usado para direcionar o sinal de voz que o piloto emite em seu microfone para a transmissão por dois ou mais dos rádios comunicadores, para o interfone ou para os autofalantes da cabine de passageiros.

É interessante notar que os circuitos equivalentes foram apresentados para mostrar a mecânica de funcionamento dos multiplexadores e demultiplexadores (também conhecidos como mux e demux). Os circuitos, de fato, são implementados com portas lógicas.

Você seria capaz de escrever as funções canônicas de um multiplexador de duas entradas, um controle e uma saída? As entradas podem assumir qualquer valor e a saída assumirá o valor da primeira entrada caso o controle seja 0 e o valor da segunda entrada caso o controle seja 1. Vamos lá!

1.7 CIRCUITOS SEQUENCIAIS

Os circuitos combinacionais vistos anteriormente apresentam as saídas dependentes unicamente das variáveis de entrada.

Os circuitos sequenciais têm as saídas dependentes de variáveis de entrada e de seus estados anteriores que foram armazenados. Logo, necessitaremos de dispositivos que tenham a capacidade de armazenar dados para podermos implementar circuitos sequenciais.

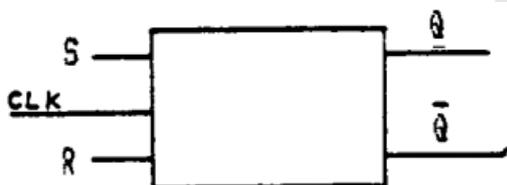
Outra característica dos circuitos sequenciais é que eles são normalmente sistemas pulsados, isto é, operam sob o comando de pulsos denominados "Clock". A cada mudança de nível do clock os estados do circuito mudam.

Dentre os componentes utilizados em circuitos sequenciais, o "Flip-Flop" é um dispositivo fundamental, que permite, por suas características, o armazenamento de estados lógicos anterior.

Flip-Flop

Flip-Flop é um dispositivo que possui dois estados estáveis. Um pulso em suas entradas poderá ser armazenado, e transformado em nível lógico estável.

Há vários tipos de Flip-Flop, que podem ser representados basicamente conforme a figura 17-36.



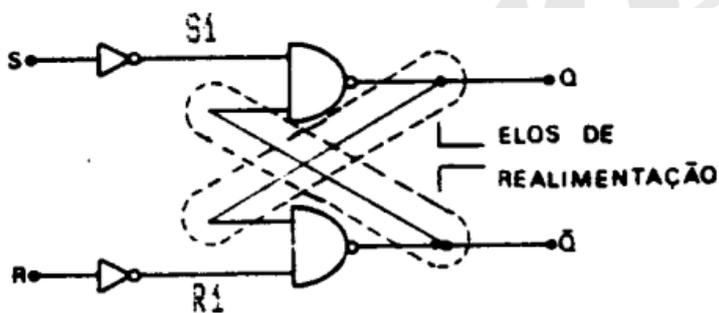
Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-33 Flip-Flop

Um pulso na entrada "S" será armazenado, tornando "Q" verdadeiro e "Q" falso. Um pulso na entrada "R", será armazenado, tornando "Q falso e "Q" verdadeiro. A mudança só ocorrerá quando houver a mudança de nível do clock.

Flip-Flop tipo "RS" (Latch)

| | S | R | S1 | R1 | Q, | \bar{Q} |
|---|---|---|--------|----|----|-----------|
| A | 1 | 0 | 0 | 1 | 1 | 0 |
| B | 0 | 0 | 1 | 1 | 1 | 0 |
| C | 0 | 1 | 1 | 0 | 0 | 1 |
| D | 0 | 0 | 1 | 1 | 0 | 1 |
| E | 1 | 1 | ilegal | | | |

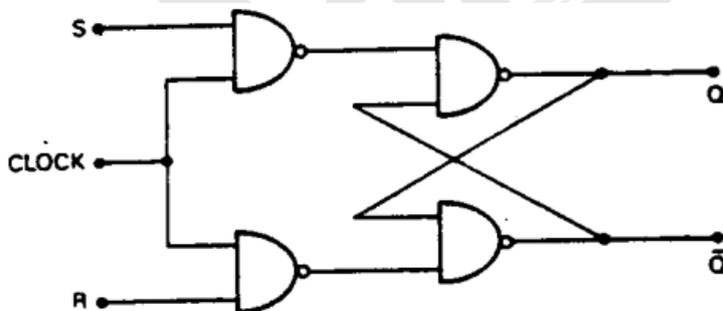


Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-34 Flip-Flop tipo "RS"

Flip-Flop "RS" comandado por Clock - Substituem-se os inversores na entrada do RS básico, por portas NAND.

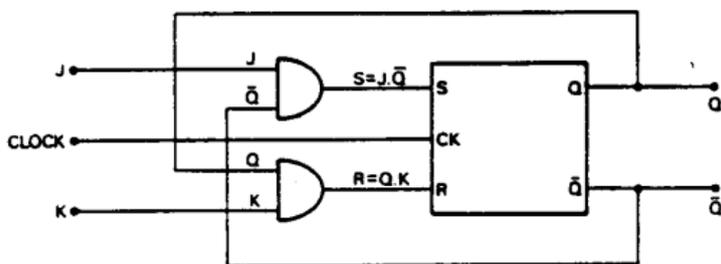
| | S | R | CLK | S1 | R1 | Q | \bar{Q} |
|---|---|---|-----|--------|----|---|-----------|
| A | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| B | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| C | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| D | 1 | 1 | 1 | ilegal | | | |



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-35 Flip-Flop "RS" comandado por CLOCK

Flip-Flop JK - Os Flip-Flop "RS" possuem um estado não permitido, quando as entradas "R" e "S" são iguais a "1" acarretando uma saída indeterminada. O Flip-Flop "JK" resolve este problema, utilizando um "RS" realimentado.



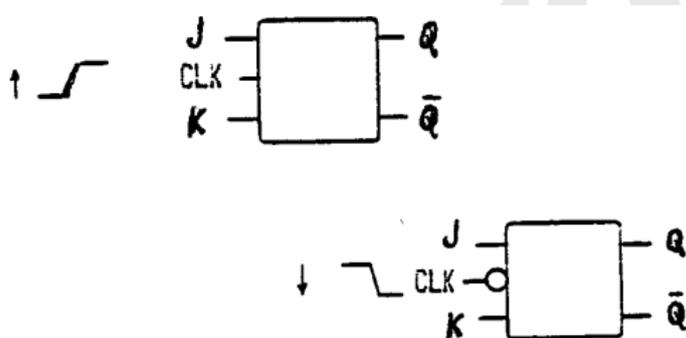
Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-36 Flip-Flop "JK"

De acordo com o circuito, o FF JK, com as entradas J e K no estado "1", terá seu estado complementado a cada "clock", isto é, se estiver "setado" (saída Q = 1), complementar-se-á (Q -> 0 e Q -> 1), se estiver "ressetado" (saída Q = 0), complementar-se-á (Q -> 1 e Q -> 0).

Flip-Flop "JK" Mestre-Escravo - No FF JK, no momento em que o Clock for igual a "1", o circuito funcionará como um combinacional, passando o estado das entradas J e K diretamente para a saída.

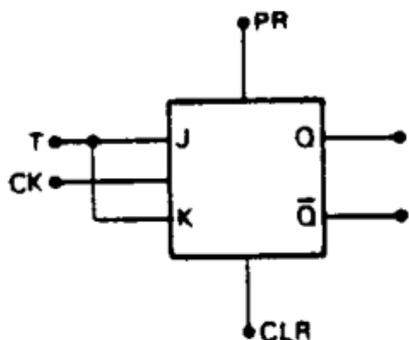
Para evitar este inconveniente, criou-se o Flip-Flop JK Mestre-Escravo (Master-Slave), que consiste basicamente de dois FF JK, permitindo a comutação do FF, apenas na transição positiva ou negativa do Clock. 1



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-37 Flip-Flop JK Mestre-Escravo

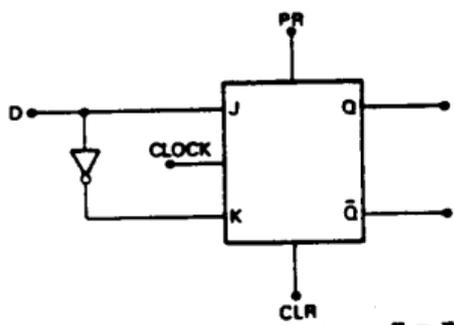
Flip-Flop tipo "T" - Consiste de um FF JK com as entradas J e K interligadas. Sua característica é de complementar-se toda vez que a entrada estiver igual a "1", mantendo-se no último estado quando a entrada for igual a "0".



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-38 Flip-Flop tipo "T"

Flip-Flop tipo "D" - Consiste de um FF JK com as entradas interligadas através de um inversor, permitindo que seja "setado" (colocado no estado "1") quando, no momento do Clock a entrada estiver igual a "1", e que seja "ressetado" (colocado no estado "0"), quando, no momento do Clock a entrada estiver igual a "0".



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

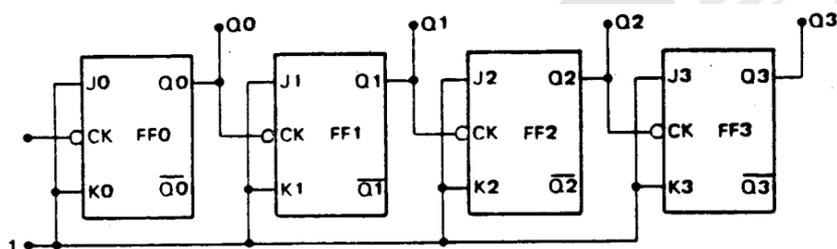
Figura 1-39 Flip-Flop tipo "D"

Contadores

São circuitos digitais compostos de FlipFlops, que variam seus estados, sob comando de um Clock, de acordo com uma sequência pré-determinada. Assim, se a sequência pré-determinada for a sequência de contagem em binário, o contador estará contando a quantidade de pulsos do clock.

O que determinará a capacidade de um contador, será o número de Flip-Flop utilizados.

Contador de pulsos - Consiste de um grupo de FF Master-Slave de comutação na transição negativa do Clock, configurados em série, de tal modo que a saída de cada estágio terá a metade da frequência do estágio anterior. Essa relação determina a sequência de números binários.

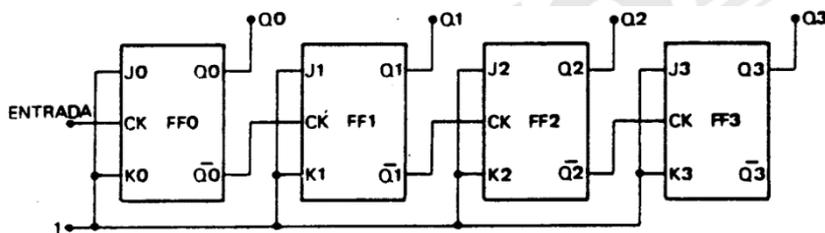


Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-40 Contador de pulsos

Contadores Decrescentes

O circuito que efetua a contagem crescente é o mesmo para contagem decrescente, com a diferença de utilizar as saídas "Q" dos FFs. A tabela verdade de um contador crescente corresponderá ao complemento da tabela de um contador decrescente.

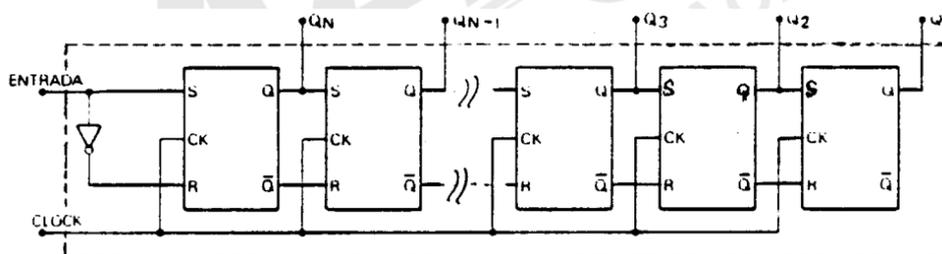


Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-41 Contador decrescente

Registadores (Shift Registers)

O flip-Flop tem a característica de armazenar o valor de um "bit", mesmo que sua entrada não esteja mais presente. Se necessitarmos guardar informações com uma quantidade de "bits" maior que um, o FlipFlop será insuficiente. Para isso utilizamo-nos de um componente denominado Registrador de Deslocamento (Shift Register), que compõe-se de um certo número de Flip-Flops, de forma que as saídas de um alimentem as entradas do FF seguinte. Cada estágio do registrador armazenará o sinal de entrada no momento do Clock. Serão necessários tantos "Clocks", quantos forem os "bits" a serem armazenados.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-42 Registrador de Deslocamento (Shift Register)

Este tipo de registrador é bastante utilizado nas conversões de sistemas seriais para sistemas paralelos, onde a entrada recebe os sinais serialmente, recebendo ao final a informação completa paralela.

1.8 MEMÓRIAS

Memórias são dispositivos que armazenam informações. Essas informações poderão ser números, letras, ou caracteres quaisquer.

Tipos de Memórias

Podemos classificar as memórias quanto a:

- Acesso;
- Volatilidade;

c) Possibilidade de regravação;

d) Retenção.

Acesso - Se refere a forma como os dados são acessados segundo os endereços na memória. Endereços são a localização dos dados armazenados. Dependendo da codificação utilizada, cada endereço conterá um conjunto de "bits", ao qual chamamos "palavra". Cada endereço conterá uma palavra de memória.

Podemos acessar palavras de memória de duas maneiras:

- Acesso Sequencial;
- Acesso Aleatório.

No acesso Sequencial, o endereçamento será feito em sequência, isto é, para uma dada posição de memória todos os endereços precisam ser acessados desde o primeiro endereço. Em virtude disto, o tempo de acesso dependerá do lugar onde a informação estiver armazenada.

No acesso Aleatório, o endereçamento é feito diretamente na palavra desejada, sem necessidade de passar-se pelas posições intermediárias. Estas memórias são conhecidas por RAM (Random Access Memory). Como principal vantagem tem o tempo de acesso, que é reduzido e idêntico para qualquer endereço.

Volatilidade - As memórias podem ser voláteis e não voláteis. As memórias voláteis são aquelas que perdem a informação armazenada quando da interrupção da sua alimentação.

As memórias não voláteis são aquelas que mantêm armazenadas as informações, mesmo na ausência de alimentação.

Possibilidade de regravação - As memórias RW (Read/Write) de leitura e escrita, permitem o acesso a qualquer endereço, para consulta da informação ou para alteração da informação.

São utilizadas em processos onde é necessária a constante alteração das informações. São normalmente identificadas como RAM (Random Access Memory).

As memórias apenas de Leitura (Read Only Memory ou ROM) são aquelas cuja informação uma vez gravada somente estará disponível para Leitura e não mais para alteração.

São utilizadas em processos onde a informação é necessária para consulta ou inicialização de uma rotina. Possuem capacidade de armazenamento, isto é, quantidade de endereços, inferior às RAM's.

Quanto a esta classificação podemos citar:

- a) PROM (Programmable Read Only Memory) - São memórias apenas para leitura, que permitem que a sua programação, isto é, a gravação inicial seja feita pelo usuário. Esta gravação é permanente, não permitindo alterações, passando ela a operar como uma ROM.
- b) EPROM (Eraseble / Programmable Read Only Memory) - São memórias que funcionam como PROM's, que permitem, porém o seu apagamento e posterior regravação. O processo de apagamento é possível por meio de um "banho" ultravioleta, através de janelas no seu encapsulamento.
- c) EEPROM - São EPROM's que permitem sua regravação por meios elétricos, sem necessidade de banhos Ultravioleta.

É conveniente lembrar que, embora as EEPROM's permitam regravações, a sua aplicação é diferente das RAM's. As EEPROM's, assim como as EPROM's, PROM's e ROM's, são utilizadas para armazenamento de informações que durante um processo são apenas consultadas, como as instruções para sequência de um programa. A característica de regravação em alguns tipos de ROM tem por finalidade permitir alterações nestas instruções, sem a necessidade de substituição de componentes.

Retenção - Classificam-se em Estáticas e Dinâmicas.

As memórias de armazenamento Estático, retém os dados inseridos enquanto a alimentação estiver presente.

As memórias Dinâmicas, por outro lado, possuem um efeito capacitivo, isto é, perdem as informações carregadas, após um determinado tempo, necessitando de ciclos periódicos de "recarga" (Refresh Cycle). As memórias Estáticas são mais caras e de menor capacidade.

Aplicação

Memórias são aplicadas de formas diversas, mas sempre que for necessário o armazenamento temporário ou permanente de informações.

Uma informação poderá ser um valor a ser processado, o resultado de uma operação, ou mesmo a própria sequência com as instruções da operação.

Valores fixos ou variáveis, em processamento, são chamados "DADOS". Sequências de instruções de operação são chamadas de "PROGRAMAS".

As instruções de um programa são normalmente armazenadas em memórias do tipo "ROM", pois são informações fixas. Dados são armazenados normalmente em memórias do tipo "RAM".

1.9 CONVERSÃO DE SINAIS

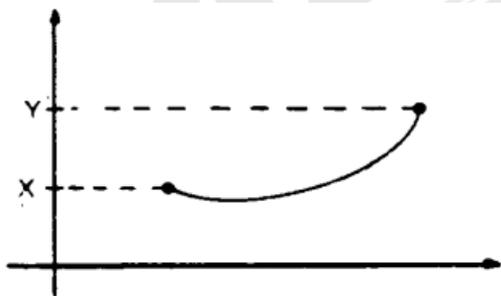
Existem basicamente dois tipos de sinais: Analógicos e Digitais

Sistemas digitais e analógicos não são compatíveis entre si, necessitando de conversores.

Os conversores têm por finalidade transformar sinais digitais em analógicos e vice-versa.

Sistemas Analógicos e Digitais

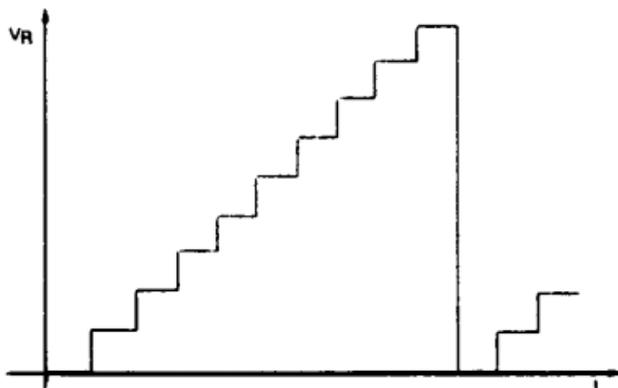
Entende-se por ANALÓGICA, toda variação linear ou contínua de um sinal. Entre dois valores distintos, podemos encontrar um valor intermediário. Grandezas físicas como temperatura, pressão, tensão, resistência, variam de forma analógica.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-44 Gráfico de variação Analógica

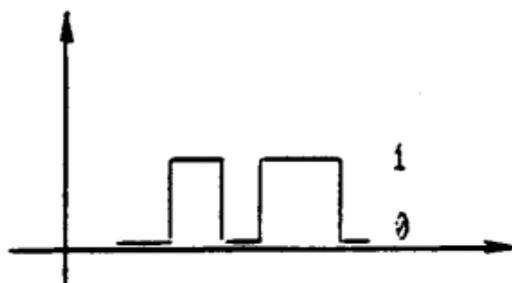
Entende-se por DIGITAL, toda variação discreta, isto é, em degraus definidos ou "steps".



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-45 Gráfico de variação Digital

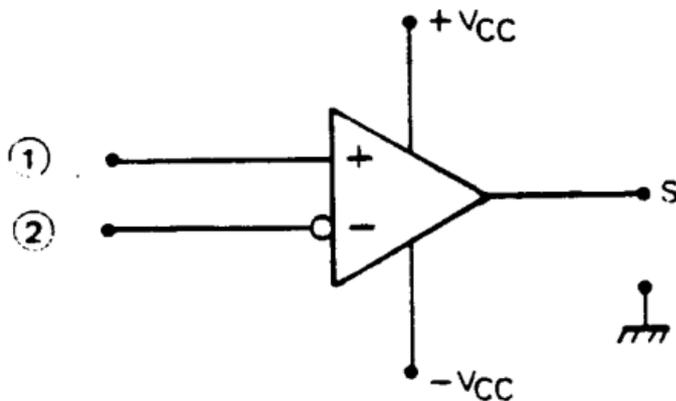
Os sistemas digitais, como internamente se utilizam de valores binários, somente reconhecem duas variações discretas, o zero (0) e o um (1). Não se concebe nesse sistema um valor que esteja entre 0 e 1. Aos sinais utilizados por estes sistemas chamamos digitais binários.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-46 Gráfico de variação digital binária

Amplificadores operacionais- São componentes lineares cuja finalidade é amplificar uma diferença entre dois sinais, possuindo ganho controlável. Diferenças de amplitude entre dois sinais são amplificadas gerando uma saída proporcional a entrada.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

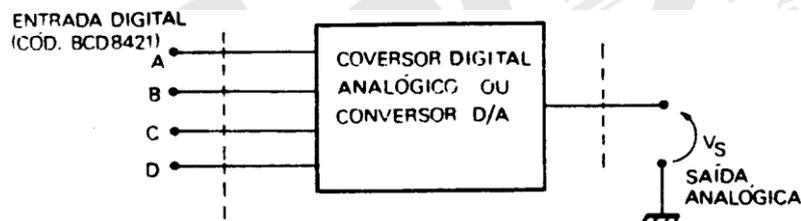
Figura 1-47 Amplificador operacional

A existência de duas entradas, sendo uma inversora e a outra não inversora permite que, dependendo de sua utilização, o sinal de saída seja normal ou invertido.

O limite de amplificação, isto é, o valor máximo de amplitude de saída, dependerá das alimentações do amplificador, limitando-se aos seus valores. A partir daí, o amplificador estará saturado, mantendo a saída fixa até que a diferença entre as entradas seja reduzida.

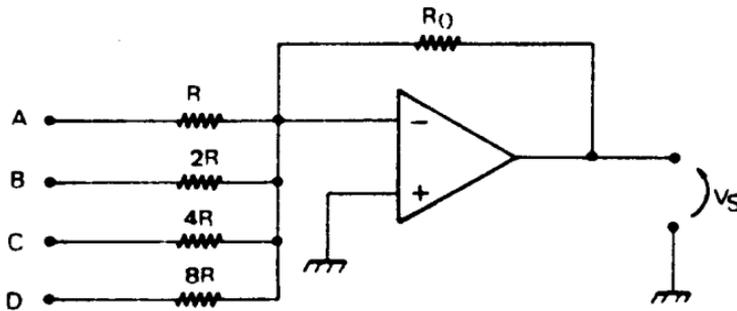
Conversor Digital-Analógico

É utilizado quando for necessária a conversão de uma variável digital em variável analógica. A variável digital é normalmente codificada em BCD 8421. A saída analógica assumirá valores de grandeza correspondentes às variações digitais da entrada.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-48 Conversor D / A



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-49 Circuito básico

O circuito da figura 1-49 amplifica diferencialmente os valores dos bits do sinal de entrada. O sinal na entrada D é amplificado 8 vezes, o da C, 4 vezes, o da B, 2 vezes e o da A, apenas 1 vez. Esses valores correspondem ao peso de cada posição na representação binária. Assim, o sinal de 1 V em D e 0 V nas outras posições representa o número 1000, que é 8 em binário. Mas como o sinal de D é amplificado 8 vezes, na saída teremos 8 V. Se o sinal 1 V estiver em B, por exemplo, com os outros em zero, teríamos a representação de 0010, que é 2 em binário e como o sinal de B é amplificado 2 vezes, teremos 2 V na saída. Verifique outros valores de entrada e a saída correspondente.

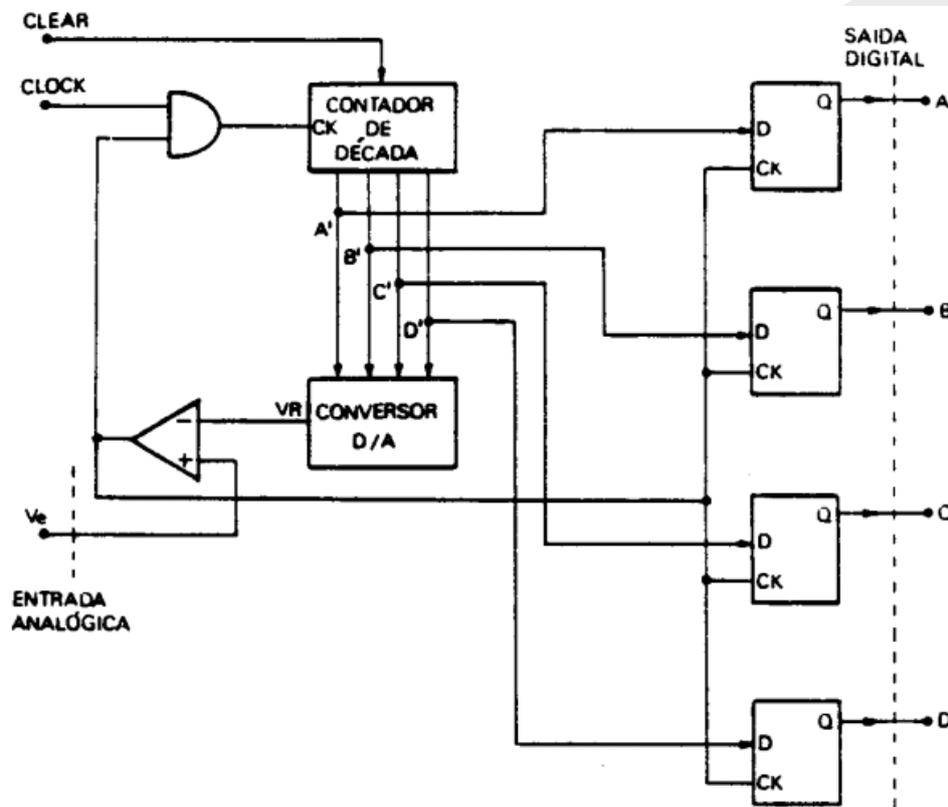
Sistemas de computação digital, não são capazes de gerar sinais analógicos linearmente, mas dependendo da precisão desejada, poderão ser utilizados mais "bits", que gerando mais "steps", darão condições de geração de sinais bastante próximos dos analógicos.

Conversor Analógico-Digital

É utilizado quando for necessária a conversão de uma variável analógica em variável digital. O conversor efetua vários passos até a conversão final, utilizando-se de um contador, um conversor D/A, um amplificador operacional atuando como comparador e Flip-Flops.

O circuito é basicamente constituído por um contador cuja saída é aplicada ao conversor D/A. Assim, o valor do contador é transformado em sinal analógico. Este sinal analógico, da saída do contador, é comparado com o sinal de entrada, aquele que se deseja transformar em digital. Se o sinal proveniente do contador é menor que o sinal de entrada, o contador é incrementado e a comparação é feita novamente. Isso se repete até que o contador atinge o valor do sinal de entrada, quando então o contador para de ser

incrementado e o seu valor digital (antes de ser transformado em analógico) é fornecido como saída.



Fonte: IAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 1-50 Conversor A / D



Referência Bibliográfica

BRASIL. IAC – Instituto de Aviação Civil. Divisão de Instrução Profissional Matérias Básicas, tradução do AC 65-9A do FAA (Airframe & Powerplant Mechanics-General Handbook). Edição Revisada 2002.



Fonte: <http://dc96.4shared.com/doc/9kocAS7f/preview.html>

MÓDULO II

INTRODUÇÃO AOS COMPUTADORES

APRESENTAÇÃO

Caro aluno,

Hoje em dia vemos computadores por todos os lados. A visão do empresário da área de informática que queria colocar um microcomputador em cada residência se expandiu para um ou mais computadores para cada pessoa. Usamos os computadores no trabalho, no entretenimento, na comunicação e nas transações bancárias. E, como não podia deixar de ser, na aviação.

Para o técnico em manutenção aeronáutica, não basta apenas saber utilizar o computador como usuário final, tal como faz o correntista de um banco ao operar o terminal de autoatendimento. Mesmo que não vá fazer reparos a nível de hardware ou software, o conhecimento dos princípios de funcionamento de um computador, no que se refere a transmissão e processamento de dados, é fundamental para que o mecânico de voo possa fazer julgamentos adequados e até mesmo saber o que necessita para o seu trabalho.

Neste módulo final, vamos ver como um microprocessador faz uso dos dados que nele são inseridos para gerar informação útil de acordo com a programação que foi anteriormente feita.

Vamos lá!?

HISTÓRICO

Um microprocessador é um circuito eletrônico muito complexo. Consiste em milhares de transistores microscópicos compactados em uma minúscula pastilha de silício, que na maioria das vezes não ocupa mais que um oitavo de polegada quadrada. A pastilha é colocada num invólucro contendo dezenas de pinos.

Os milhares de transistores que compõem o microprocessador são arranjados para formar muitos circuitos diferentes dentro da pastilha. Do ponto de vista de aprendizagem de como o micro processador opera, os circuitos mais importantes são os registradores, contadores e decodificadores.

Um microprocessador, cujo nome pode ser abreviado por " μp " é uma das partes de um computador, a porção responsável pelo controle e processamento dentro de um sistema. Para implementar um computador, é necessário acrescentar memória para o programa de controle e circuitos de entrada e saída (I/O) para a comunicação com equipamentos periféricos.

Desde a construção dos primeiros computadores a válvula, como o UNIVAC I (1950) o desenvolvimento de sistemas de processamento de dados tem sofrido uma evolução acelerada. Enquanto esses computadores primitivos só podiam ser justificados como objeto de pesquisa (sem questionar a validade econômica), os modernos sistemas revolucionaram praticamente todos os campos de atividade do homem moderno. Isto se deveu, unicamente, à evolução da tecnologia eletrônica do estado-sólido, que reduziu o tamanho e o custo dos sistemas de computação, entre inúmeros equipamentos.

Em 1960, o baixo preço dos computadores justificava o aparecimento dos computadores de propósito geral, para processamento de dados.

Atualmente, o advento de microprocessadores permite a aplicação de métodos computacionais de custo extremamente baixo ao controle e processamento de sistemas em

geral. Isto representa um passo decisivo em direção a uma disseminação extensiva do processamento de dados nos aspectos mais triviais da vida moderna.

2.1 APLICAÇÕES

Computador de Escritório

O baixo custo de um microcomputador permite a sua utilização em escritórios comerciais de pequeno porte. O sistema básico compreende, geralmente, um console de processamento, um monitor de vídeo, um teclado, uma unidade de disco rígido e impressora. Aumentando-se o número destes periféricos, pode-se acompanhar o crescimento das exigências da automação.

A finalidade deste equipamento é controlar as folhas de pagamento, fazer controle de estoque, manipular informações de contabilidade, fazer processamento de texto, tudo isto aliado à possibilidade de se disseminar a informação simultaneamente através de diversos terminais.

Atualmente esses sistemas estão ligados em rede com outros computadores da empresa e na rede mundial de computadores, de onde podem receber e transmitir informações.

Computador Pessoal

Atualmente em fase de rápida expansão, esta aplicação possibilita trazer a revolução da informática para o lar e os dispositivos portáteis permitem que o usuário tenha acesso aos serviços praticamente em qualquer lugar. Além de usar o microcomputador, para jogos eletrônicos, pode-se fazer o controle conexão à rede mundial, o computador pessoal pode ter acesso a informações tais como cotação de ações na Bolsa, jornais ou bancos de dados. Já podemos fazer encomendas num supermercado pelo microcomputador, por tablets ou smartphones, consultando os preços dos artigos em estoque e até mesmo trabalhar em casa, enviando e recebendo informações do computador da empresa.

Computador de Bordo

Microcomputadores são empregados em sistemas de computação para automóveis, barcos e aeronaves.

Além de fornecerem informações sobre navegação, consumo, condições do veículo etc, poderão receber dados de outros computadores. Isto permitirá que um piloto receba instruções de controle para pouso ou decolagem através de um monitor de vídeo situado no painel, agilizando e aumentando a capacidade do controle de tráfego aéreo.

Equipamentos Automáticos de Teste

Para o controle de qualidade é possível elaborar sistemas capazes de fazer o teste de equipamentos, numa rapidez e precisão impossível para o ser humano, a um custo reduzido.

Máquinas com "Inteligência"

O uso de processadores em equipamentos de uso geral permite sofisticá-los com funções até então inviáveis economicamente. Balanças e Caixas Eletrônicas, por exemplo, são aplicações recentes de microprocessadores. Deve-se lembrar também que a manutenção desses equipamentos é simplificada por programas de diagnóstico e até de calibração automática.

Robôs

Recentemente, a indústria japonesa desenvolveu uma variedade de máquinas capazes de executar tarefas repetitivas, como robôs industriais. Este é um dos campos mais promissores e polêmicos de aplicação de microprocessadores.

Armamento - Míssil Auto Dirigido

O primeiro passo de Neil Armstrong na superfície da Lua foi possível, em grande parte, em decorrência dos sistemas de orientação computadorizados.

Evidentemente, a engenharia de foguetes interplanetários apoia-se em uma tecnologia muito precisa, mas, sem o "hardware" e o "software" de computadores, jamais seria possível executar cálculos de posição com rapidez e exatidão suficientes para permitir o acoplamento de dois objetos a uma grande distância, mesmo que um desses objetos tenha o tamanho da Lua.

Quando se levam em conta as técnicas militares modernas, que exigem a colocação de ogivas com limite de erro de poucos metros após um voo transcontinental, é enorme a capacidade de processamento de dados necessária para os cálculos.

As primeiras experiências militares mostraram que o problema fundamental da tecnologia de mísseis estava no fato de que eram impossíveis correções no seu trajeto após ter sido feito o lançamento. A primeira grande conquista deu-se com o desenvolvimento de sistemas de orientação capazes de calcular a posição do foguete em relação a um ponto na superfície (local de lançamento) pela dedução da distância percorrida e de sua direção. Mas até mesmo os equipamentos modernos de alta qualidade estão sujeitos a erros graves. Outro método mais preciso utiliza satélite em órbita geoestacionária como ponto de referência. A principal desvantagem desses sistemas é que a linha de voo do míssil - e provavelmente seu alvo - pode ser calculada pelo inimigo imediatamente após o lançamento, dada a capacidade dos modernos radares de longo alcance. Para eliminar essa vulnerabilidade, projetou-se um míssil capaz de voar a baixa altura, provido de radar de varredura horizontal, que avalia dados para o cálculo do percurso até o alvo. Assim nasceu o míssil "Cruise".

2.2 TERMOS E CONVENÇÕES

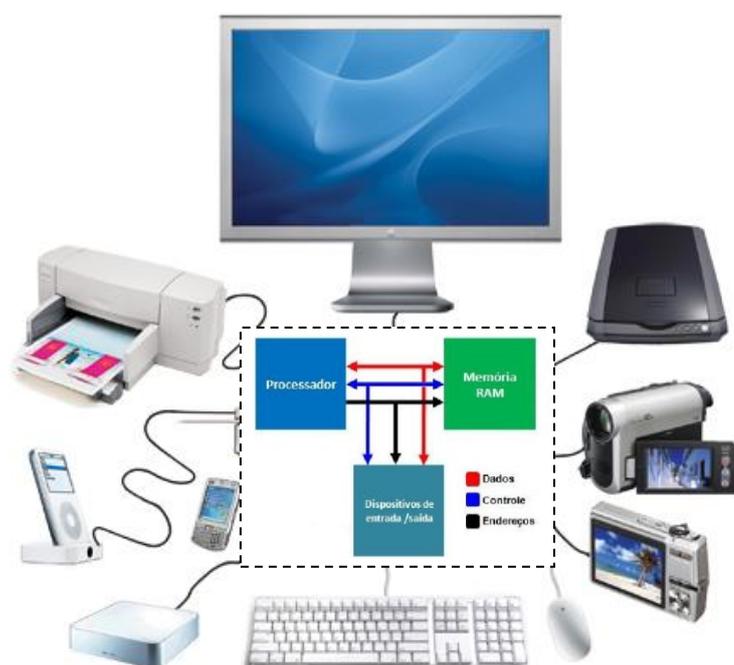
Um microprocessador é um dispositivo lógico que é usado em sistemas eletrônicos digitais. Também é usado como passatempo, como computador de uso geral de baixo custo, para técnicos e grupos de pesquisa com baixo nível orçamentário. Mas, uma distinção deverá ser feita entre o microprocessador e o microcomputador.

Um microcomputador contém um microprocessador, mas também contém outros circuitos como um dispositivo de memória para armazenar informação e adaptadores de interface para conectá-lo com o mundo externo.

A figura 2-1 mostra um microcomputador típico no qual esses circuitos adicionais são acrescentados. As setas representam condutores nos quais seguem as informações binárias.

As setas largas representam vários condutores conectados em paralelo. Um grupo de condutores paralelos, que transportam informação, é chamado barramento (bus).

O computador possui dois barramentos principais: o "ADDRESS BUS" (ADD BUS) e o "DATA BUS". O ADD BUS é unidirecional, isto é, possui um único sentido para o fluxo. O DATA BUS é bidirecional permitindo, por exemplo, o fluxo de dados da CPU para a unidade de entrada e saída (I/O) ou desta para a memória.



Fonte: <http://dc401.4shared.com/doc/XEIZsGM6/preview.html> e <http://marcobruno.com.br/ads-unip/conceitos-basicos-perifericos/>

Figura 2-1 Computador básico

O programa do computador é um conjunto ordenado de instruções que são executadas uma a uma, sequencialmente, na ordem estipulada. Instrução é uma palavra chave (ordem) que diz ao computador qual a tarefa específica que deve executar.

O microcomputador é composto por tudo o que está dentro da linha pontilhada na figura 2-1. Tudo o que está fora da linha pontilhada refere-se ao mundo externo e todos os microcomputadores precisam ter alguns meios de comunicação com ele.

A informação recebida do mundo externo pelo microcomputador é chamada de entrada de dados. A informação que transmite do microcomputador para o mundo externo é chamada de saída de dados.

O computador pode ser definido como um sistema complexo capaz de receber informações, processá-las e fornecer resultados. A entrada de informações poderá ser gerada de dispositivos de memória (disco rígido, cartão de memória, pendrive) ou de outros computadores via rede.

A saída de informações poderá ser enviada aos monitores, memórias, impressoras ou outros computadores. O ponto no qual o dispositivo de I/O conecta-se ao microcomputador é chamado de "porta".

2.3 MEMÓRIA

O conjunto de dados e de instruções necessários à operação de um computador fica localizado numa unidade chamada memória.

Podemos imaginar a memória como sendo um conjunto de escaninhos, cada qual com um endereço e contendo uma unidade de informações.

Existem diversos tipos de memória, classificados segundo suas características. De um modo geral, temos memórias voláteis ou não voláteis. A memória volátil é aquela cuja informação se perde quando a alimentação é interrompida. As memórias RAM são um exemplo deste tipo. Memórias não voláteis, por outro lado, retêm a informação mesmo após interrupção da alimentação. Um exemplo deste tipo é a memória ROM.

- ROM ("Read Only Memory") - Como o nome indica, este tipo de memória não permite realizar operações de escrita, apenas de leitura. Também chamada de memória morta, é gravada durante a fabricação, retendo sempre esta informação.
- RAM ("Random Access Memory") - Esta memória se caracteriza por permitir tanto a leitura como a escrita, sendo, entretanto volátil.

Memórias Magnéticas

Devido às suas propriedades este tipo de memória é sempre não-volátil. Além disso, geralmente são capazes de armazenar grandes quantidades de dados, embora a velocidade de leitura/escrita seja baixa. Por isso são mais usadas como memória de massa, isto é, uma espécie de armazém de programas e de dados, de onde a CPU os retira para processamento em RAM.

Os tipos mais comuns são discos magnéticos e fitas magnéticas.

2.4 ENTRADA E SAÍDA

São unidades que permitem ao microcomputador comunicar-se com o mundo externo. É através de operações de entrada e saída que um operador utiliza um terminal de vídeo-teclado para "conversar" com um computador.

O elemento que serve de ligação entre o microcomputador e o periférico é chamado "interface".

A transferência pode ser feita de vários modos, segundo as necessidades ou limitações do sistema. Transmissões por via telefônica, por exemplo, são feitas no modo serial, isto é, um "bit" por vez.

Por outro lado, entre um teclado e a CPU, a transmissão pode ser em paralelo, com 7 ou 8 "bits" transmitidos de uma só vez.

2.5 PALAVRAS DO COMPUTADOR

Na terminologia de computação, a palavra é um conjunto de dígitos binários que pode ocupar um local de armazenamento. Embora a palavra seja constituída de vários dígitos binários, o computador manipula cada palavra como se ela fosse uma simples unidade. Portanto, a palavra é a unidade fundamental de informação usada no computador.

Uma palavra pode ser um número binário que está sendo manipulado como um dado. Ou, a palavra pode ser uma instrução que diz ao computador que operação deve executar. A palavra poderá representar, também, um endereço.

Tamanho da Palavra

Nos últimos anos, uma ampla variedade de microcomputadores tem sido desenvolvida. Seu custo e sua capacidade variam grandemente. Uma das mais importantes características de um microprocessador é o tamanho da palavra que ele pode manipular que se refere ao tamanho em "bits" da maior unidade fundamental de informação.

Um tamanho da palavra comum para o microprocessador há alguns anos era o de 8 "bits". Números, endereços, instruções e dados são representados por números binários de 8 "bits".

O menor número binário de 8 "bits" é 0000 0000₂ ou 00₁₆. O maior é 1111 1111₂ ou 11₁₆. Em decimal, o alcance é de 0 a 255₁₀. Então, um número binário de 8 "bits" pode ter algum dos 255₁₀ possíveis valores.

Uma palavra de 8 "bits" pode especificar números positivos ou negativos.

2.6 PRINCIPIOS DE FUNCIONAMENTO

Código de Máquina

O código de máquina é a linguagem entendida pelo microprocessador (a CPU), que constitui o cerne do computador e pode apenas executar funções muito simples (adiciona dois dígitos a um número, por exemplo, mas não os multiplica). Faz isso, no entanto, a velocidades muito altas. Cada operação do microprocessador é especificada de acordo com o número de "ciclos de relógio" empregados. Se a CPU em seu computador funcionar a 1 MHz, o "ciclo de relógio" será de 1 microssegundo e uma operação que requer quatro "ciclos de relógio" será realizada em 4 milionésimos de segundo, pois $F = 1 / T$.

Como consequência, um programa desenvolvido em código de máquina vai requerer grande quantidade de instruções e qualquer função deverá ser elaborada "à mão", a partir de operações simples. Toda a programação consistirá na manipulação de "bits" (unidade de informação) ou "bytes" (conjunto de 8 bits) isolados de memória, empregando-se funções lógicas simples como AND, OR e NOT, além de aritmética elementar.

Esse é um dos motivos por que o desenvolvimento de programas nessa linguagem se torna uma tarefa lenta. O outro reside no fato de o programador ser obrigado a saber a localização de tudo que está armazenado na memória.

Ao iniciar a programação em código de máquina, você descobre que tem de especificar um endereço (uma posição de memória) para cada conjunto de dados a armazenar. E cabe também a você garantir que não haja superposição acidental com outros conjuntos de dados.

Examinemos em que consiste o código de máquina. Todos os exemplos a seguir serão relacionados a CPUs de 8 "bits" de capacidade, como a do Z80 e do 6502. O microprocessador conecta-se à memória do computador por meio de dois "buses" (vias): o "bus" de endereços e o de dados. Há também um elemento denominado "bus" de controle que fornece apenas sinais de cronometragem à CPU e não é utilizado pelo programador.

O "bus" de endereços tem capacidade de 16 "bits" e a atribuição de um padrão de "bits" a esse "bus" possibilita à CPU selecionar qualquer dos 65.536 "bytes" em seu "mapa de memória". Em microcomputadores comuns, algumas dessas posições estarão na RAM, outras na ROM, algumas nos "chips" especiais de entrada-saída e ainda existirão as que não serão utilizadas.

Se a CPU quiser ler determinada posição de memória (uma das linhas no "bus" de controle indica se deve haver leitura ou registro), o "byte" selecionado coloca seus conteúdos no "bus" de dados, na forma de um padrão de 8 "bits". De modo semelhante, a CPU pode registrar um padrão de 8 "bit" em qualquer posição escolhida. A CPU não sabe quais as partes da memória em que estão a ROM e a RAM. Desse modo, determinar o endereço correto é outra responsabilidade importante do programador.

No interior do microprocessador, há talvez meia dúzia de "registros", semelhantes a posições individuais de memória, utilizados para o armazenamento de resultados temporários e execução de funções aritméticas, lógicas e binárias. A maior parte desses registros corresponde a 1 "byte" de memória, embora algumas tenham 02 "bytes" de capacidade.

Outro registro muito importante (mas, agora, com apenas 8 "bits" de extensão) é o "acumulador". Como o nome indica, esse registro acumula totais, isto é, "bytes" que podem se somar ou subtrair. Na verdade, esse é, em geral, o único registro que pode executar qualquer tipo de procedimento aritmético.

O motivo comum para o emprego do código de máquina é a velocidade: ao endereçar diretamente o processador, o programa não precisa ser traduzido ou interpretado. Com a eliminação desse estágio intermediário, reduz-se bastante o tempo de execução do programa. No entanto, o processo de codificação, teste, depuração, modificação e manutenção de um código de máquina exigirá, com certeza, duas vezes mais tempo do que levaria num programa em linguagem de alto nível (BASIC, por exemplo).

A falta de interação com o programador e a dificuldade de se lidar com o código de máquina foi o principal estímulo para a criação das linguagens de alto nível, como COBOL e BASIC. Vimos que o conjunto de instruções em código de máquina equivale ao conjunto de operações do processador.

Execução do Programa Mnemônico

É a forma de representação de tal modo que haja facilidade de retenção na memória, isto é, que haja memória.

Como exemplos de mnemônicos no nosso cotidiano, podemos considerar os termos: "SAMPA", "BELZONTE", etc.

Em se tratando de computação e programação em linguagem de máquina, os mnemônicos mais usados são:

- LDA (LOAD ACCUMULATOR) - Carregar Acumulador;
- STA (STORAGE ACCUMULATOR) - Armazenar acumulador;
- ADC (ADD WITH CARRY) - Somar com transporte.

Consideremos que o programa já tenha sido adequadamente escrito no computador pelo processo tradicional: o programador lê o programa, no papel e introduz, via teclado, no computador.

O programa é armazenado, em linguagem de máquina, numa área de memória, embora os dados que ele opera possam estar em alguma outra posição.

Observe que os operandos, por exemplo, \$3F80 estão armazenados em dois "bytes", com o "byte" menor (\$80) situado antes do maior (\$3F).

O símbolo Dólar (\$), usado na frente de uma sequência, indica que a representação está no sistema de base hexadecimal.

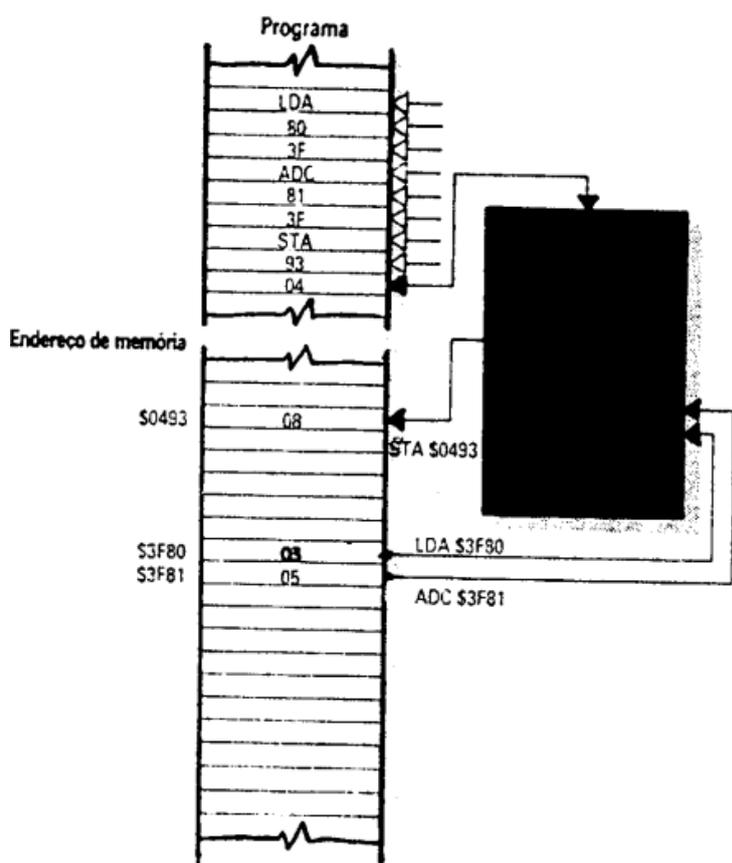
Todos os programas em código de máquina são constituídos por operações simples que transportam "bytes" de memória para os registros internos da CPU, realiza seu processamento e, a seguir, os remetem de novo a uma posição de memória.

A figura 2-2 apresenta o programa necessário para somar os conteúdos de duas posições de memória e armazenar o resultado de uma terceira.

O contador do programa (PC) é um registro no interior da CPU que indica a instrução que está sendo executada.

A primeira instrução fornece os conteúdos da posição \$3F80, isto é, o valor "5" ao acumulador.

O terceiro armazena o conteúdo do acumulador, agora o "08", na posição de memória \$0493.

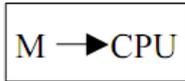


Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 2-2 Execução de um programa

Funcionamento com o Programa

O conteúdo das posições de memória \$00, \$01 e \$02 (LDA, \$3F80), é colocado na CPU.



A CPU decodifica essa instrução e a interpreta como: carregar o acumulador com o conteúdo da posição de memória \$3F80.



Após a execução dessa operação, o contador de programa (PC) será incrementado (PC + 1, PC + 2 e PC + 3) acessando os conteúdos das próximas posições de memória (endereços): \$03, \$04 e \$05. Esses conteúdos são enviados para a unidade central de processamento.



Em seguida a CPU interpreta os códigos e conclui que deverá "somar" o conteúdo atual do Acumulador com o conteúdo da posição de memória \$3F81.



Nota: Como o próprio nome indica o Registrador acumulador "acumula" resultados das operações lógicas ou aritméticas. Em consequência o resultado das operações será enviado para esse registrador.

Como a operação anterior já foi completada, o PC é devidamente incrementado (PC + 1, PC + 2 e PC + 3), e os conteúdos das posições subsequentes serão enviados à CPU e devidamente interpretados como: armazenar o conteúdo do acumulador na posição de memória \$0493.



Nota: O efeito real desta transferência é de cópia onde o conteúdo origem não é apagado. Daí conclui-se que o conteúdo final do registrador acumulador será o último obtido.

Com isso chega-se ao objetivo final do programa proposto que foi: somar os conteúdos de duas posições de memória e colocar o resultado em uma terceira. Pelo exposto, também podemos escrever:

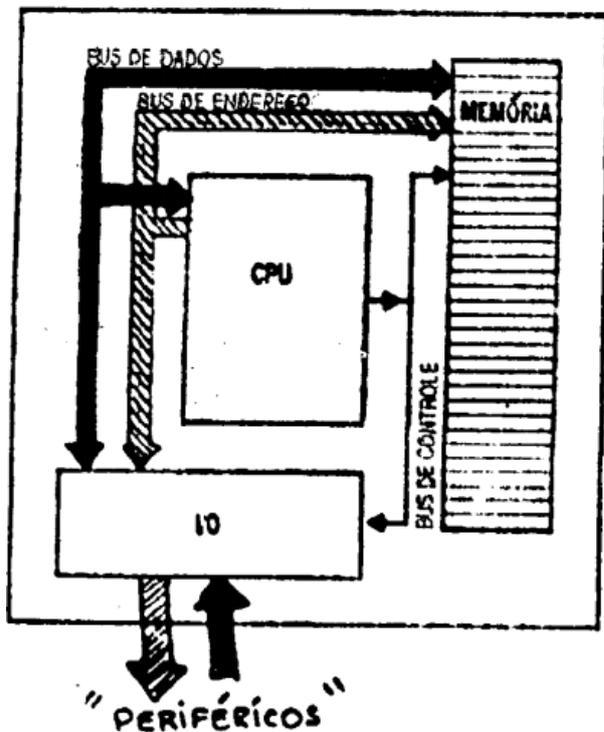
$M \rightarrow CPU; M \rightarrow ACC; M \rightarrow CPU; ACC +$
 $M \rightarrow ACC; ACC \rightarrow M$

2.7 UNIDADE CENTRAL DE PROCESSAMENTO (CPU)

Em termos simples, a CPU ("Central Processing Unit", ou Unidade Central de Processamento) do computador não passa de um interruptor que controla o fluxo de corrente num sistema de computação. Compõe esse sistema a ALU ("Arithmetic and Logic Unit", ou Unidade Aritmética e Lógica), o PC (Contador de Programa), ACC (Acumulador) e outros registradores.

Ao acionar uma tecla, você introduz alguma informação na máquina por meio de uma configuração de voltagem gerada na unidade de teclado. A CPU transfere essa configuração de voltagem para uma posição da memória. Em seguida, transfere uma configuração correspondente, proveniente de algum outro lugar da memória, para a tela, de modo a gerar um determinado padrão de caracteres.

Esse processo é semelhante ao funcionamento de uma máquina de escrever, mas com a diferença de que nesta há uma conexão mecânica entre o acionamento de uma tecla e a impressão do caractere, enquanto num computador essa ligação ocorre porque a CPU transfere configurações corretas de voltagem de um lugar para outro.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 2-3 O Centro do Sistema

Nem sempre o acionamento de uma tecla faz aparecer um caractere no vídeo: pode também destruir um asteroide, gravar um programa, apagar um arquivo em disco, ou imprimir uma carta. O resultado da operação depende do modo e da finalidade com que a CPU transfere a corrente elétrica.

Modo de Funcionamento da CPU

Os procedimentos executados pela CPU classificam-se, para nossos objetivos, nas operações: **aritméticas**, **lógicas**, de **memória** e de **controle**. Todas resultam de transferência de informações através de diferentes trajetos, no sistema e na CPU, ou seja, para esta todas as operações se assemelham.

Operações **aritméticas**, como adição e subtração constituem a característica mais importante da máquina. Ela subtrai por meio da representação negativa de um dos números e sua subsequente adição com o outro.

Por exemplo,

$7 + 5 = 12$ significa:

+ 7 somado a + 5 é igual a + 12;

enquanto $7 - 5 = 2$ equivale a:

+ 7 somado a - 5 é igual a + 2.

A multiplicação e a divisão são consideradas adições ou subtrações repetidas, de modo que também é possível programar a CPU para simular essas operações. Se a CPU consegue realizar as quatro operações aritméticas, então pode efetuar qualquer outro cálculo matemático. No entanto, lembre-se de que todo o seu potencial matemático depende simplesmente da capacidade de somar dois números.

As operações **lógicas** efetuam a comparação de dois números não apenas em termos de quantidades relativas, mas também em termos da configuração de seus dígitos. É fácil ver que sete é maior que cinco porque extraímos cinco de sete e obtemos um resultado positivo. Além de fazer esse tipo de comparação, a CPU também verifica que, por exemplo, 189 e 102 têm o mesmo dígito na coluna das centenas.

As operações de **memória** envolvem tanto a cópia de informações de uma posição qualquer da memória externa para sua própria memória (registro), como de seu registro para outra posição da RAM.

Executando essas operações em sequência, a CPU transfere informações de uma parte qualquer da memória para outra. Para que a memória do computador tenha alguma utilidade, é absolutamente necessário que a CPU seja capaz de realizar essas duas operações. Só assim torna-se possível um controle completo da memória.

As operações de **controle** consistem, na verdade, em decisões quanto à sequência pela qual a CPU executa as outras operações descritas. Por enquanto, tudo o que precisamos saber sobre as operações de controle é que a CPU pode tomar determinadas decisões a respeito de sua própria atividade.

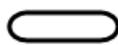
Portanto, a CPU executa operações aritméticas, compara números, desloca informações na memória e decide sobre sua própria sequência de operações. Essa lista de procedimentos é suficiente para definir uma máquina de computação ideal.

2.8 CONCEITOS DE FLUXOGRAMA

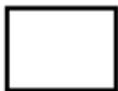
O fluxograma é uma representação gráfica das tarefas de um programa, por meio de símbolos que fornecem uma visualização imediata do significado da tarefa.

Definição dos Elementos de Fluxograma

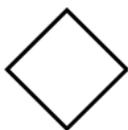
Abaixo se seguem os símbolos mais usados nas representações gráficas.



Terminal: Início, término ou interrupção de um programa.



Processamento: Uma ação que deve ser tomada.



Decisão: Desvio para diversos pontos do programa de acordo com uma situação testada.



Entrada / Saída: Qualquer função relacionada com dispositivos de entrada ou saída em geral.



Monitor: Terminal de vídeo ou Display.



Cartão: Entrada ou saída através de cartão.



Teclado: Entrada de informação através do teclado.



Impressora: Saída de informações através da impressão em papel.

Exemplos do Uso de Fluxograma

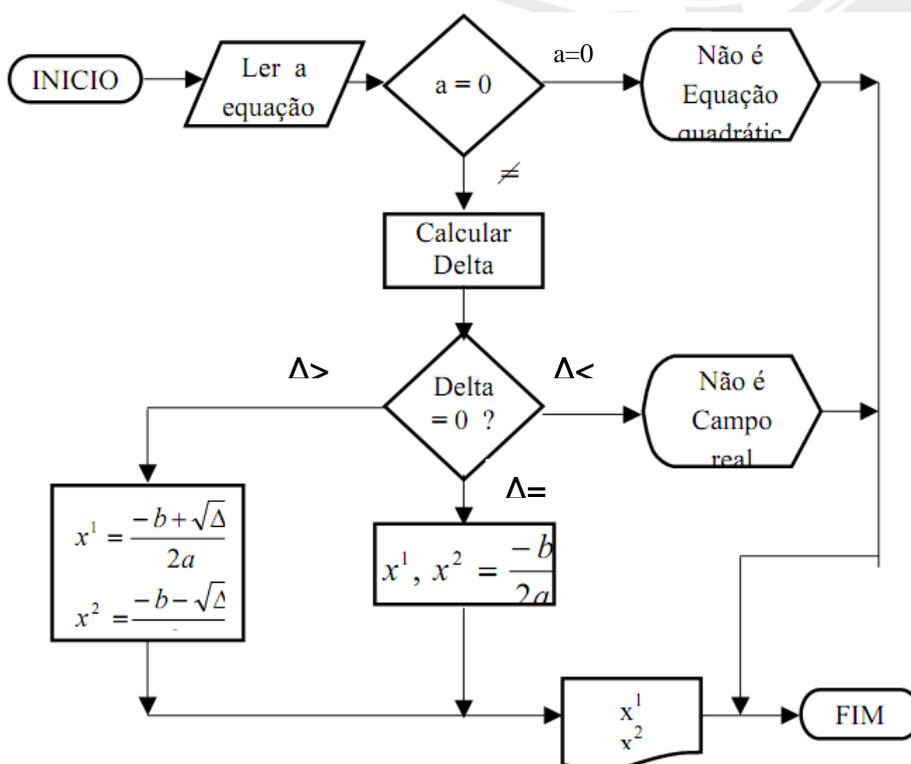
Equação Quadrática - Vamos examinar alguns exemplos simples, para termos uma visualização de fluxograma.

O passo inicial é examinar um fluxograma genérico que represente os passos do programa para calcular as raízes da equação quadrática: $ax^2 + bx + c = 0$

Também podemos escrever que:

$$x^1, x^2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-b \pm \sqrt{\Delta}}{2a}$$

A solução deste problema pode ser representada como no fluxograma da figura 112.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

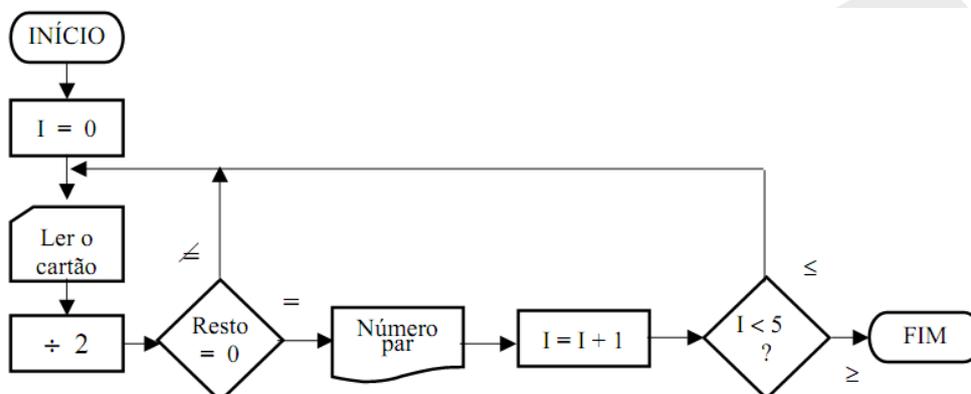
Figura 2-4 Fluxograma de Equação Quadrática

Impressão de Números

O fluxograma da figura 2-5 é um programa para imprimir cinco números pares lidos de um cartão de memória.

Observe que a instrução $I = I + 1$, não significa que I seja igual a $I + 1$, o que seria um absurdo, mas que a variável I passa a assumir o valor que ela tinha acrescido de 1, ou seja, se I era igual a 2, passa a ser 3 e assim por diante. Assim, a variável I , que iniciou com o

valor 0, é incrementada de 1 sempre que um número par é impresso, até que atinja um valor que não seja menor que 5, quando o programa é instruído a parar.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 2-5 Fluxograma dos números pares

2.9 LINGUAGEM DO COMPUTADOR

Desde o princípio da era do computador, a principal preocupação foi de como poderia ser feita a comunicação entre o homem e a máquina.

Como se pode notar, tanto o computador como a máquina, utilizam-se apenas de níveis de tensão, chamados níveis lógicos.

A grande preocupação do homem é utilizar o computador em todas as áreas e, para isso, necessita de uma comunicação de fácil acesso com o meio externo. Isto, em outras palavras, significa transformar níveis de tensão em informações, com o tipo de linguagem empregada pela maioria dos homens.

Uma das primeiras comunicações com o computador foi a linguagem de máquina ou objeto. Esta linguagem é muito cansativa por lidar diretamente com códigos binários, isto é, níveis lógicos "zero" e "um".

Em função da dificuldade apresentada por esta linguagem, surgiu a linguagem "Assembly", que transforma códigos binários em mnemônicos, isto é, nomeia cada código para que este possa ser utilizado mais facilmente.

Com o surgimento desta linguagem, houve a necessidade de transformar os mnemônicos em códigos binários. Esta transformação é chamada de "Compilador Assembler".

Mesmo com o surgimento da linguagem Assembly, o operador continuava encontrando muitas dificuldades para manusear este grupo de instruções. Mediante estas dificuldades, outras linguagens foram sendo desenvolvidas.

Com o aparecimento de outras linguagens, houve uma divisão entre as linguagens de baixo nível e as de alto nível. As de alto nível surgiram especificamente para cada área, como "Fortran" para a área científica, "Cobol" para a área comercial e outras linguagens para cada finalidade como "Basic", "Pascal", "PL/1", etc.

As linguagens de alto nível deram margem ao surgimento dos programas tradutores. Como o próprio nome está dizendo, o computador necessita de um programa que traduza tais linguagens para que tenha condições de executar as instruções a ele designadas.

A linguagem de máquina pode ser escrita em octal, hexadecimal ou binária. O programa fonte recebe a denominação de linguagem de programação e o programa resultante da conversão em linguagem de máquina recebe a denominação de programa objeto.

Programa de processamento é aquele que traduz a linguagem de programação para linguagem de máquina.



Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 2-6 Representação em blocos do fluxo das linguagens.

| Programa | Fonte | Processamento | Observações |
|---------------|-----------------------|-----------------------------------|---|
| Máquina | Máquina | Não há necessidade | Este programa é escrito na linguagem binária |
| | Hexadecimal | Hexadecimal loader | É usado o carregador hexadecimal |
| | Assembly | Assembler | É a linguagem de programação escrita em Mnemônicos |
| Processamento | Alto nível (Compiler) | Tradutor-Interpretador/Compilador | Converte linha a linha e a tradução é feita de uma só vez |
| | Hand Assembly | Hexadecimal Loader | O operador faz a tradução do Assembly para o Hexa, isto é, age como programa de processamento |

Fonte: LAC – Instituto de Aviação Civil – Divisão de Instrução Profissional

Figura 2-7 Quadro geral de linguagens.



Referência Bibliográfica

BRASIL. IAC – Instituto de Aviação Civil. Divisão de Instrução Profissional Matérias Básicas, tradução do AC 65-9A do FAA (Airframe & Powerplant Mechanics-General Handbook). Edição Revisada 2002.



Encerrando a Disciplina

Caro aluno,

Chegamos ao final de nossos estudos com o entendimento de como funciona um sistema computacional completo. Esperamos que toda informação fornecida seja útil para sua carreira de técnico em aviônicos e que você possa contribuir com eficiência, conforto e segurança nos voos.

Sucesso!

Prof. Hélio Luis Camões de Abreu