



ELETRÔNICA BÁSICA

SUMÁRIO

1-	CIRCUITOS DIGITAIS	3
2-	CONVERSORES ANALÓGICO-DIGITAL (A/D) E DIGITAL-ANALÓGICO (D/A)	8
3-	PRINCÍPIO DE OPERAÇÃO DOS CONVERSORES A/D	13
4-	APLICAÇÕES BÁSICAS: SENSORES E AQUISIÇÃO DE DADOS	24
5-	PORTAS LÓGICAS	36
6-	FLIP-FLOPS	46
	REFERÊNCIAS	

1- CIRCUITOS DIGITAIS

Os circuitos digitais ou circuitos lógicos são definidos como circuitos eletrônicos que empregam a utilização de sinais elétricos em apenas dois níveis de corrente (ou tensão) para definir a representação de valores binários.

Circuitos lógicos baseiam seu funcionamento na lógica binária, que consiste no fato de que toda informação deve ser expressa na forma de dois dígitos (tanto armazenada, como processada), sendo tais dígitos 0 (zero) ou 1 (um). A partir disto surge a nomenclatura “digital” (dois dígitos).

Este fato auxilia para a representação de estados de dispositivos que funcionam em dois níveis distintos, sendo estes: ligado/desligado (on/off), alto/baixo (high/low), verdadeiro/falso (true/false) entre outros.

Os computadores, telefones celulares e leitores de DVD ou blu-ray são alguns exemplos de aparelhos que baseiam parte do seu funcionamento em circuitos digitais.

Etimologia

A palavra digital vem do latim digitalis, que significa "relacionado aos dedos".

Desde que a humanidade desenvolveu o processo de contagem, os dedos foram os instrumentos mais simples e eficientes para contar pequenos valores. O sistema de numeração indo-arábico, o mais usado atualmente, é um sistema de base dez, pois são dez os dedos das duas mãos dos seres humanos. Muitos outros sistemas de numeração usam a base decimal, pois serviam para simbolizar a contagem com os dedos.

Normalmente com os dedos só é possível contar valores inteiros. Por causa dessa característica, a palavra digital também é usada para se referir a qualquer objeto que trabalha com valores discretos. Ou seja, entre dois valores considerados aceitáveis existe uma quantidade finita de valores aceitáveis.

Digital não é sinônimo de eletrônico: por exemplo, o computador eletrônico pode ser chamado de digital porque trabalha com o sistema binário, que é simbolizado por uma sequência finita de zeros e uns, qualquer que seja o tipo de dados.

Hoje em dia, porém, não se consegue desvincular a palavra "digital" do sistema informático e de tecnologias ligadas à computação, como, por exemplo, "transmissão digital".

Construção

Circuitos são constituídos pela associação de blocos lógicos.

Os blocos lógicos são divididos em 7 classes:

E (AND)

OU (OR)

NÃO (NOT)

NE (NAND)

NOU (NOR)

OU EXCLUSIVO (XOR)

NÃO-OU EXCLUSIVO (XNOR)

A partir destes blocos lógicos é possível construir praticamente todas as outras associações necessárias.

Exemplos:

Contadores Binários (flip-flops).

Unidades lógico-aritméticas (ULA, em inglês, ALU).

Circuitos digitais e circuitos analógicos

Circuitos digitais apresentam diversas vantagens se comparados aos circuitos analógicos, tais como:

Facilidade de projetar e armazenar informações

Extensa programabilidade

Maior exatidão e integração

São menos afetados por ruídos originários de flutuações de tensão de alimentação por não dependerem do valor exato da tensão elétrica recebida, e sim da diferença entre os níveis alto e baixo

Uma desvantagem dos circuitos digitais é que o mundo é de natureza analógica, melhor representada em valores contínuos com extensa variação de frequência, o que faz com que todo circuito digital que lide com variáveis físicas de natureza analógica necessite converter tal informação para o meio digital, para então processar, e posteriormente fazer o fluxo inverso, convertendo da natureza digital para a analógica.

Circuitos combinatórios

Um circuito digital é dito combinatório ou combina se a saída depende única e exclusivamente das combinações das variáveis de entrada recebidas em um dado momento, ou seja, o circuito combinacional não é capaz de armazenar valores em "memória", para uso posterior.

O funcionamento de todas as portas lógicas básicas e a lógica booleana que descrevem e analisam os circuitos feitos a partir da combinação de portas lógicas podem ser classificados como circuitos lógicos combinacionais porque, em qualquer instante de tempo, o nível lógico da saída do circuito depende da combinação dos níveis lógicos presente nas entradas. Um circuito combinacional não possui a característica de memória, portanto sua saída depende apenas dos valores atuais das entradas.

O circuito combinacional realiza um conjunto de equações booleanas realizando uma determinada operação de processamento da informação, ou seja, a combinação de valores de entrada é vista como uma informação distinta das outras e o conjunto de valores de saídas das operações representa o resultado da

operação. Em cada circuito pode ser usada a simplificação de circuitos lógicos pelo método de mapeamento ou teoremas da álgebra booleana.

Circuitos integrados em sistemas digitais

Os circuitos integrados permitiram a implementação de sistemas digitais mais simples e seguros em relação a sistemas implementados com componentes discretos pois exigem menor manutenção, consomem menos energia e têm menor custo, peso e tamanho.

Existem diversas tecnologias usadas atualmente para a produção de circuitos integrados, dentre elas, as mais comuns são:

Lógica Transistor-Transistor.

CMOS (Complementary Metal-óxido-semicondutor).

NMOS (N-type Metal-oxide-semiconductor Logic).

ECL (Emitter-Coupled Logic)

Transferência de informações

De maneira geral, existem duas formas de transferência de dados em CIs, sendo elas:

Paralela — os bits são transferidos simultaneamente, ou seja, várias filas de dados sofrem transição todas ao mesmo tempo, em um mecanismo de múltiplas filas

Serial — os bits são transferidos um de cada vez no mesmo fluxo de dados, em um mecanismo de única fila.[10]

Performance de circuitos digitais

Circuitos digitais possuem atrasos na propagação de seus sinais. Estes atrasos podem ocorrer devido ao tempo que cada porta lógica leva para gerar um sinal de saída, dada uma mudança no sinal de entrada, e devido à propagação dos sinais através dos fios do circuito.[11] :p.57

A velocidade de um circuito digital é sempre limitada de acordo com o maior atraso do circuito, causado pelas portas lógicas. O caminho com maior tempo de atraso entre dois pontos do circuito é chamado de caminho crítico.

2- CONVERSORES ANALÓGICO-DIGITAL (A/D) E DIGITAL-ANALÓGICO (D/A)

O conversor analógico-digital (frequentemente abreviado por conversor A/D ou AACD) é um dispositivo eletrônico capaz de gerar uma representação digital a partir de uma grandeza analógica, normalmente um sinal representado por um nível de tensão ou intensidade de corrente elétrica.

O processo de conversão A/D é mais complicado e mais demorado do que o processo de conversão D/A, havendo uma grande variedade de métodos para realizar tal conversão.

O circuito do conversor é composto por um contador de década que gera o código BCD 8421 nas saídas A', B', C' e D'. As saídas entram em um conversor D/A, fazendo com que apresente na saída uma tensão de referência, a qual é injetada em uma das entradas de um circuito comparador constituído por um amplificador operacional. Na outra entrada têm-se o sinal analógico a ser convertido.

Os ADCs são muito úteis na interface entre dispositivos digitais (microprocessadores, microcontroladores, DSPs, etc) e dispositivos analógicos e são utilizados em aplicações como leitura de sensores, digitalização de áudio e vídeo.

Por exemplo, um conversor A/D de 10 bits, preparado para um sinal de entrada analógica de tensão variável de 0V a 5V pode assumir os valores binários de 0 (0000000000) a 1023 (1111111111), ou seja, é capaz de capturar 1024 níveis discretos de um determinado sinal. Se o sinal de entrada do suposto conversor A/D estiver em 2,5V, por exemplo, o valor binário gerado será 512.

Anti-aliasing

Como os conversores são limitados em banda, ou seja, trabalham apenas em uma faixa específica de frequência (1 a 10kHz), normalmente $[0, f_N]$, onde f_N representa o dobro da frequência do maior sinal passível de ser adquirido ($f_N/2$ - frequência de Nyquist), normalmente utiliza-se um filtro passa-baixas com a finalidade de evitar

que amplitudes de harmônicas de alta frequência apareçam na entrada do conversor.

Estudo comparativo

Sinais gerados por circuitos analógicos são muitas vezes processados por circuitos digitais, por exemplo, por um microcontrolador ou por um microcomputador.

Para processar sinais analógicos usando circuitos digitais, deve-se efetuar uma conversão para essa última forma, a digital. Tal conversão é efetuada por um Conversor Analógico-Digital ("A/D converter" ou ADC).

O sinal recebido, depois de digitalizado, é processado e, na maioria das vezes, será utilizado para atuar sobre o circuito analógico que gerou o sinal original ou até mesmo sobre outro circuito.

Por isso, um sinal na forma digital, para ser processado por um bloco funcional analógico, deve ser previamente convertido (ou reconvertido) para a forma analógica equivalente.

Um sistema que aceita uma palavra digital como entrada e traduz ou converte o valor recebido para uma voltagem ou corrente analógicas proporcionais à entrada é chamado de Conversor digital-analógico ("D/A converter" ou DAC). Neste caso, quanto mais bits conter o sinal de entrada(digital), melhor será o sinal convertido(analógico) pois haverá maior precisão.

Tipos de conversores A/D

De um modo geral os conversores A/D podem ser divididos em alguns tipos técnicos, conforme as suas características fundamentais de conversão:

Conversor em Rampa

O conversor em rampa é um conversor A/D simples e que pode ser projetado para que tenha uma alta resolução. Quando a taxa de amostragem não necessita ser elevada mas não dispensa alta resolução, este tipo de conversor é a melhor opção. O funcionamento deste dispositivo basea-se na integração de sinais e seus componentes básicos são: um comparador, um contador e um integrador de Miller.

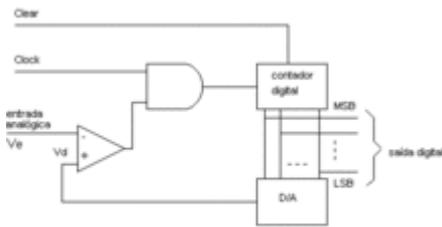


Figura 1. Conversor A/D do tipo rampa com contador.

Esse é seu princípio de funcionamento: primeiramente se integra o sinal de entrada (V_{in}) durante um tempo fixo, posteriormente move-se a chave presente na entrada do integrador para o ponto onde está conectada a tensão referênci conhecida (V_{ref}), de sinal contrário em relação à tensão de entrada, nesta fase faz-se a integração da V_{ref} que se volte ao estado inicial, ou seja, tensão nula na saída do integrador. O contador citado anteriormente como um dos componentes básicos deste tipo de conversor é responsável por determinar o tempo da rampa de subida (integração de V_{in}) e realizar a contagem do tempo de descida (integração de V_{ref}).

A inclinação da rampa de subida é dependente da amplitude do sinal de entrada, por isso, para diferentes sinais de entrada teremos um valor de tensão (V_m) diferente ao final da primeira etapa de integração. Já a rampa de descida irá apresentar sempre a mesma inclinação, pois o valor do sinal de entrada do integrador é sempre o mesmo nesta fase, apenas o tempo de duração desta rampa de descida irá alterar, e é justamente este tempo que é utilizado para a determinação da amplitude do sinal de entrada.

A simplicidade deste tipo de conversor, aliada a seu baixo consumo de energia, faz com que ele seja escolhido, geralmente, para aplicações de instrumentação de baixo custo, como multímetros digitais.

Conversor de Aproximação Sucessiva

O conversor SAR ADC (ou conversor de aproximação sucessiva) representam a maior parte de conversores de resoluções médio-grande. Eles possuem pouco consumo de energia, uma alta resolução e precisão, e o fato de serem fisicamente pequenos também ajuda. Devido a esses benefícios os ADCs SAR podem ser integrados com outras funções maiores.

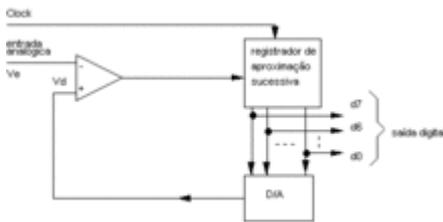


Figura 2. Conversor de aproximação sucessiva.

As suas maiores limitações são as pequenas taxas de amostragem, uma resolução limitada devido aos limites do DAC e comparador, e também o fato de seu tamanho aumentar com o número de bits. É ideal para sistemas com dados de multi-canais e frequências de amostras abaixo de 10MHz e resoluções entre 8 e 16 bits.

Seus circuitos são mais complexos do que os do conversor em rampa, porém seu tempo de conversão é muito menor, o que torna seu uso bastante atrativo. Além disso, os conversores A/D por aproximações sucessivas têm um tempo de conversão fixo, que não depende do sinal analógico presente em sua entrada. O esquema básico deste conversor é similar ao do conversor em rampa.

No entanto o conversor A/D por aproximações sucessivas não utiliza um contador para gerar a entrada do conversor D/A, usando, em seu lugar, um registrador comum. A lógica de controle modifica o conteúdo deste registrador bit a bit, até que o dado armazenado no registrador seja equivalente à entrada V_A , dentro da resolução do conversor.

Paralelo ou flash

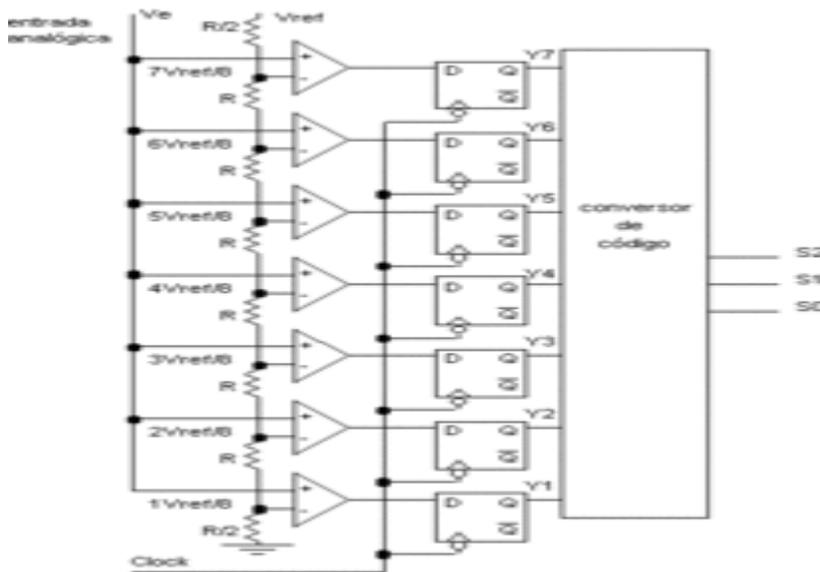


Figura 3. Conversor A/D de 7 comparadores.

Os conversores tipo paralelo têm como circuito básico de entrada um pré-amplificador e um latch, que atuam juntos em uma configuração de circuito comparador. Na saída dos comparadores é necessária a colocação de um circuito de codificação que irá receber os sinais dos comparadores e codificar o sinal de saída em código binário (ou “GRAY”).

A grande vantagem do conversor A/D paralelo é a grande rapidez na conversão, porque o sinal analógico de entrada é comparado diretamente e simultaneamente com cada nível de voltagem de referência em comparadores distintos.

A grande dificuldade ou desvantagem dos conversores A/D paralelo é o aumento do número de comparadores de latch e complexidade do codificador à medida que se aumenta a resolução, isso ocasiona um enorme aumento na área de silício e consumo de potência, devido ao grande número de componentes.

O conversor A/D paralelo é o mais rápidos dentre todos os tipos de conversores e normalmente é construído utilizando-se a versão mais rápida de uma determinada tecnologia, mas é expressivamente caro, visto que necessita de 2^n-1 comparadores para um conversor de n bits.

3- PRINCÍPIO DE OPERAÇÃO DOS CONVERSORES A/D

No mundo real as grandezas físicas raramente são de natureza elétrica. O primeiro passo para trazer esse mundo para o seu processador é o de transformar essas grandezas em sinais elétricos. Os equipamentos responsáveis por essa transformação são conhecidos por sensores ou transdutores. Esses transdutores estão em quase tudo ao nosso redor. São sensores de pressão, vazão, luz, temperatura, PH, etc. Todos esses transdutores transformam as grandezas físicas em sinais elétricos. Os sinais elétricos podem ser lineares e proporcionais à amplitude das grandezas medidas, ou então não lineares mas com curvas conhecidas, que podem ser compensadas de alguma maneira a posteriori.

Uma vez transformadas em sinais elétricos, a precisão das grandezas convertidas pelos transdutores fica limitada às características ou especificações desses transdutores. Sua natureza ainda é analógica e contínua no tempo. Para trazer essas grandezas para dentro do seu processador, será necessário realizar mais uma transformação do sinal analógico para digital, de forma que esse possa ser tratado e processado digitalmente. Essa transformação é realizada por um componente conhecido como Conversor A/D (Analógico/Digital).

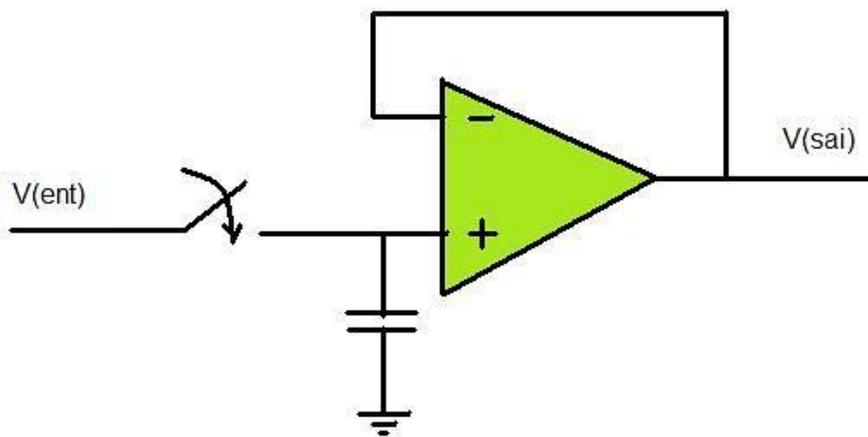
Conversor A/D

Um conversor A/D transforma um sinal analógico, contínuo no tempo, num sinal amostrado, discreto no tempo, quantizado dentro de um número finito de valores inteiros, determinado pela resolução característica do conversor em bits (8, 10, 12, 16 etc). Por exemplo, num conversor de 8 bits, o sinal de entrada é transformado em amostras com os valores entre 0 e 255.

O sinal a ser convertido por um conversor A/D dificilmente se acomoda diretamente à faixa de tensão de entrada do conversor. Ele precisa ser transformado

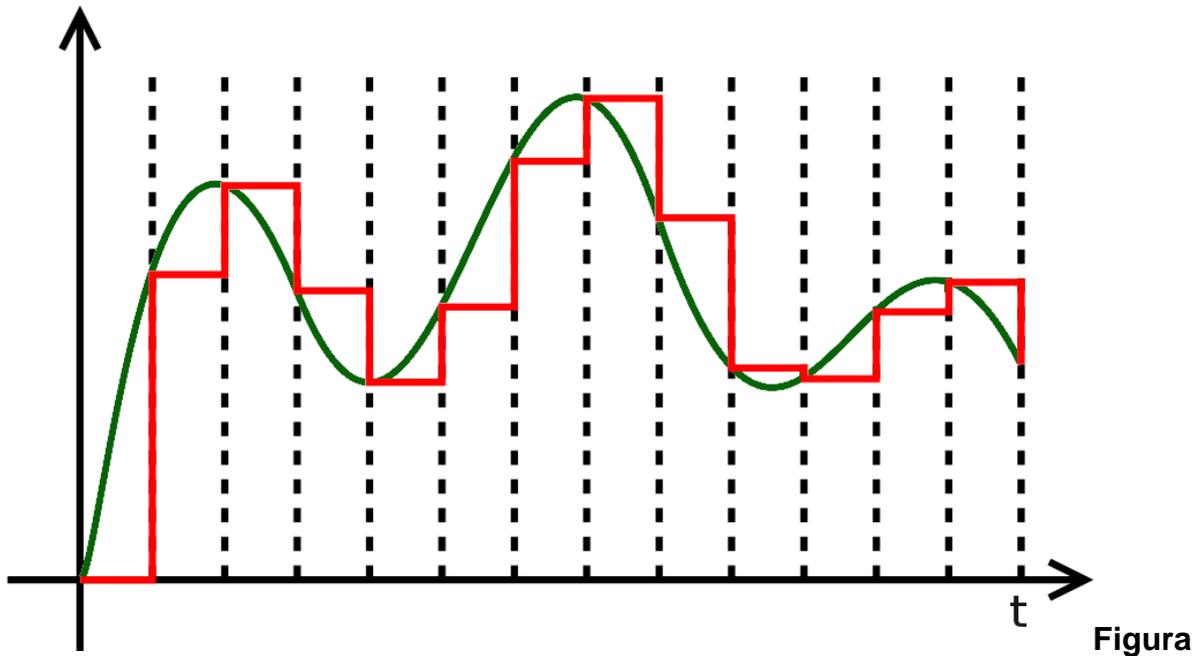
adequadamente para isso. Em geral a tensão de entrada de um conversor A/D é definida como a tensão de alimentação do conversor (+ 5 ou 3,3 V, por exemplo). Para realizar essa adaptação muitas vezes é necessário realizar um condicionamento do sinal, tipicamente com auxílio de circuitos analógicos passivos ou ativos.

Após o condicionamento do sinal existe um elemento na entrada do conversor A/D que realiza uma amostragem periódica do sinal analógico e o mantém estável até que o conversor propriamente dito possa convertê-lo para um código digital. Trata-se de um circuito de *Sample & Hold*. Um circuito ilustrativo de um S/H (*Sample and Hold*) pode ser visto na **Figura 1**. A ilustração do efeito dessa amostragem pode ser vista na **Figura 2**.



1 - Circuito *Sample & Hold* simplificado

Figura



2 - Saída de um circuito *Sample & Hold* quando estimulada por um sinal contínuo

Há diversas técnicas para a conversão A/D: conversão paralela, por contagem, de rampa simples, rampa dupla, sigma-delta etc. Recomendo a leitura do artigo "Como funcionam os Conversores A/D" de Newton C. Braga, dividido em duas partes, para aprofundar os seus conhecimentos sobre conversores A/D.

Quando for especificado um conversor A/D, é necessário verificar se o conversor atende às necessidades do seu projeto. É necessário verificar as seguintes especificações:

- Conversor interno ao processador ou externo?;
- Taxa de amostragem;
- Número de bits de resolução;
- Número de bits de precisão, ou bits efetivos;
- Relação sinal/ruído;
- Linearidade;
- Necessidade de se utilizar *Sample & Hold* externo;
- Necessidade de se utilizar um filtro analógico de *anti-aliasing*;
- Preço;

- etc.

No próximo tópico será abordado especificamente o tema da quantização e os efeitos decorrentes disso.

Quantização

Em processamento digital de sinais, chama-se de quantização de amostras a transformação das amostras de um sinal analógico numa representação dessas amostras em números inteiros. É o que ocorre nos conversores A/D. Por exemplo, um sinal real é transformado em números inteiros entre 0 (zero) e a potência de 2 correspondente ao número de bits do conversor (Por exemplo 4095, no caso de 12 bits). Esse efeito será ilustrado mais adiante.

Para gerar as figuras a seguir, foi utilizado o programa de matemática OCTAVE versão 4.0. As sequências de comando (*scripts*) também serão apresentadas para que o leitor possa reproduzir o que será apresentado ou simular outras situações.

Para a melhor compreensão do que será apresentado a seguir recomendo a leitura do artigo técnico Processamento Digital de Sinais - DSP - Parte 2.

Suponhamos uma senoide ideal, amostrada 16 vezes a cada ciclo (**Figura 3**).

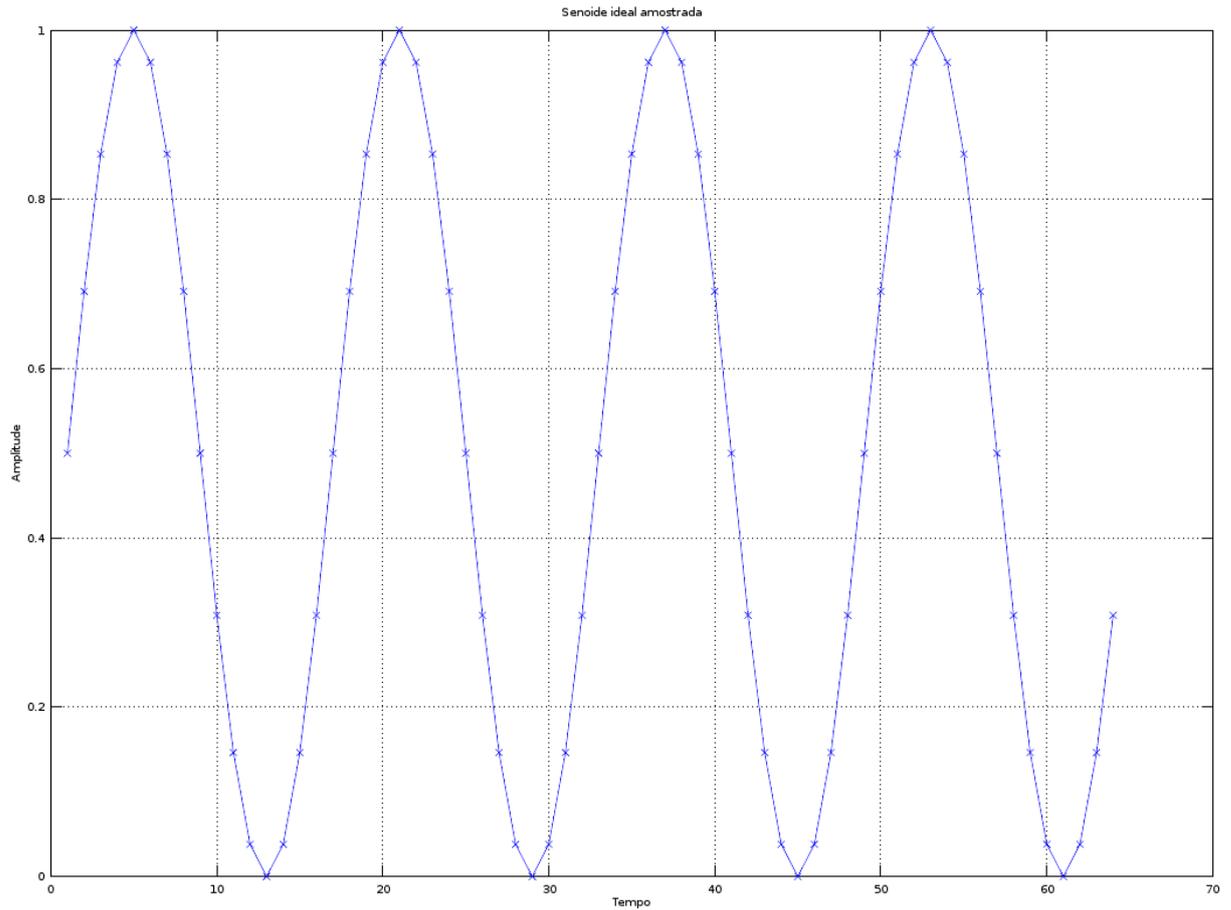


Figura 3 - Senoide ideal amostrada

A transformada de Fourier dessa senoide pode ser observada na **Figura 4**.
Observação: o OCTAVE utiliza números de ponto flutuante com precisão dupla.

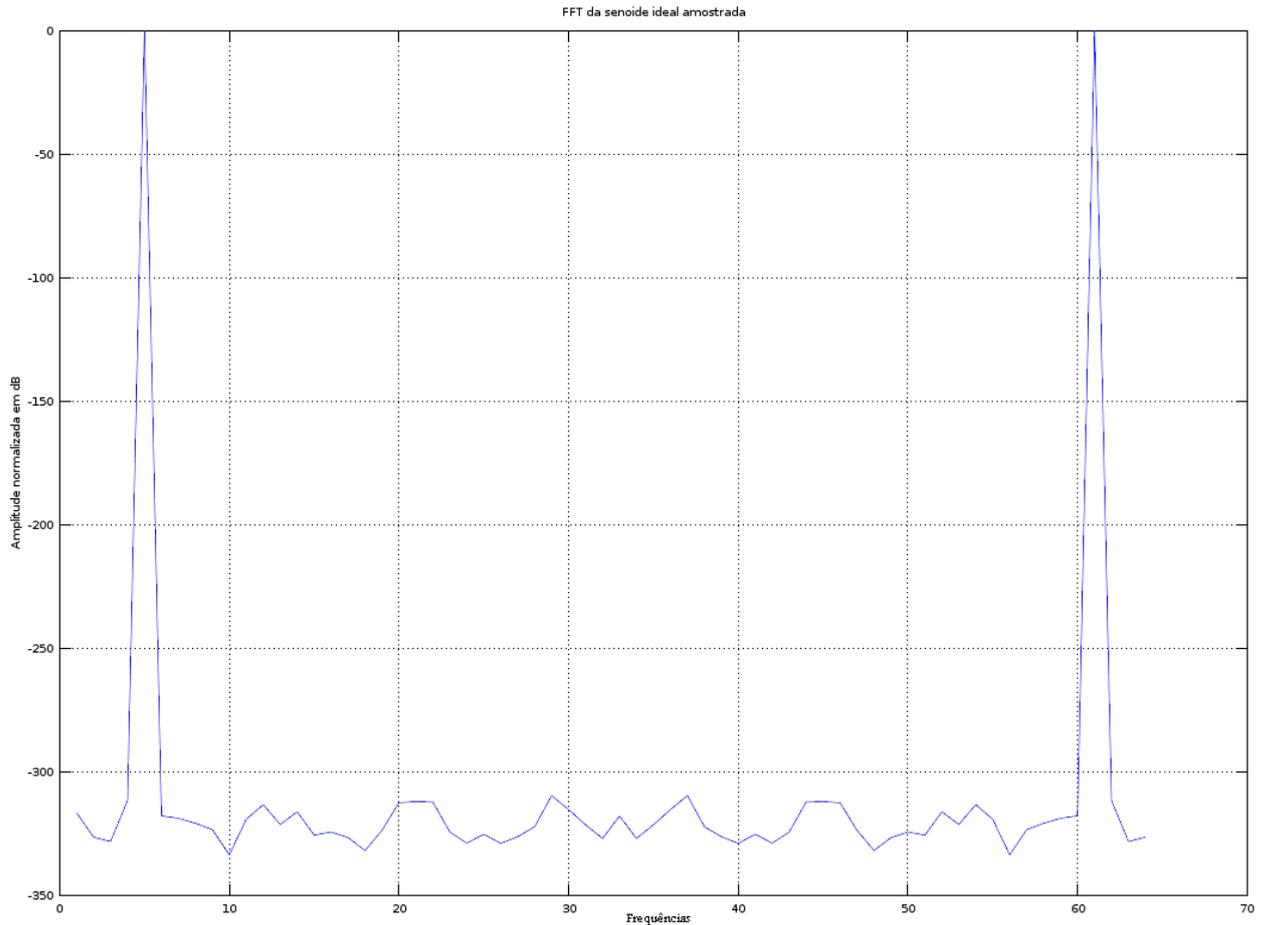


Figura 4 - Transformada de Fourier da senoide ideal

Observe que o ruído está abaixo de - 300 dB. É desprezível. O *script* do OCTAVE para gerar essas figuras encontra-se a seguir:



```

1 % Gera e plota uma senoide ideal amostrada
2
3 Num_Amostras = 16;           % Define o numero de amostras por ciclo
4 Num_Ciclos = 4;             % Define o numero de ciclos
5 Delta_t = 1/Num_Amostras;   % Define o delta t entre as amostras
tempo = 0:Delta_t:Num_Ciclos - Delta_t; % Define os tempos para gerar a

```

```
6 senoide
7 seno1 = 0.5 * (1 + sin(2 * pi * tempo)); % gera a senoide
8 close all; % Fecha todas as figuras
9 figure(1); % Abre a figura 1
10 plot(seno1); % Plota o seno na figura 1
11 hold; % Retém a figura
12 grid; % Gera a grade
13 xlabel('Tempo');
14 ylabel('Amplitude');
15 title('Senoide ideal amostrada');
16 plot(seno1, 'x'); % Marca os pontos amostrados
17
18 seno1 = seno1 - 0.5; % Suprime a componente CC
19
20 figure(2); % Abre a figura 2
21 fftseno1 = abs(fft(seno1)); % prepara a FFT
22 plot(20 * log10(fftseno1/max(fftseno1))); % Plota a FFT normalizad e em dB
23 grid; % Gera a grade
24 xlabel('Frequencias');
25 ylabel('Amplitude normalizada em dB');
26 title('FFT da senoide ideal amostrada');
27 %
```

A seguir serão mostradas as transformadas de Fourier da mesma senoide ideal mostrada na **Figura 3** quantizada respectivamente para 8 , 12 e 24 bits.

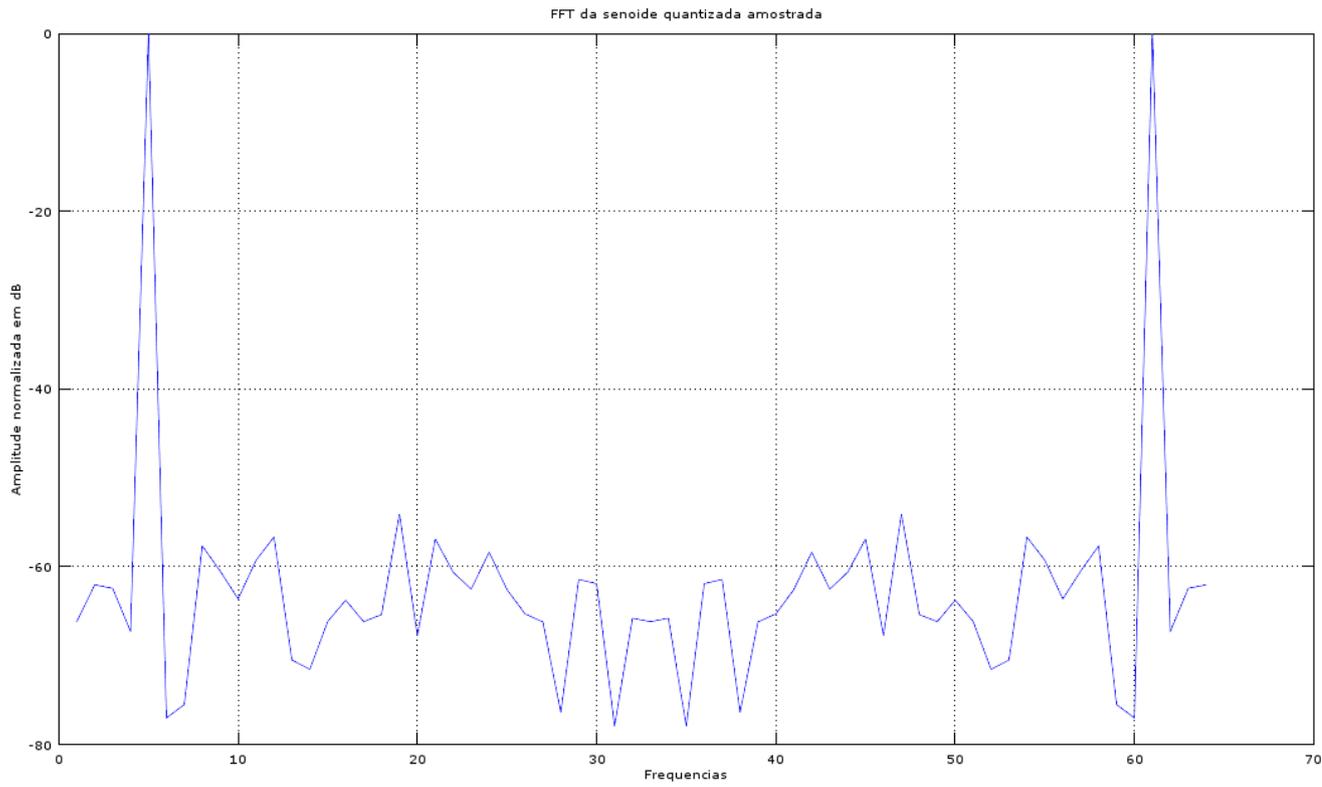


Figura 5 - FFT da senoide ideal quantizada para 8 bits

Observe na **Figura 5** como aparecem picos artificiais de ruído no espectro da senoide amostrada decorrentes da quantização.

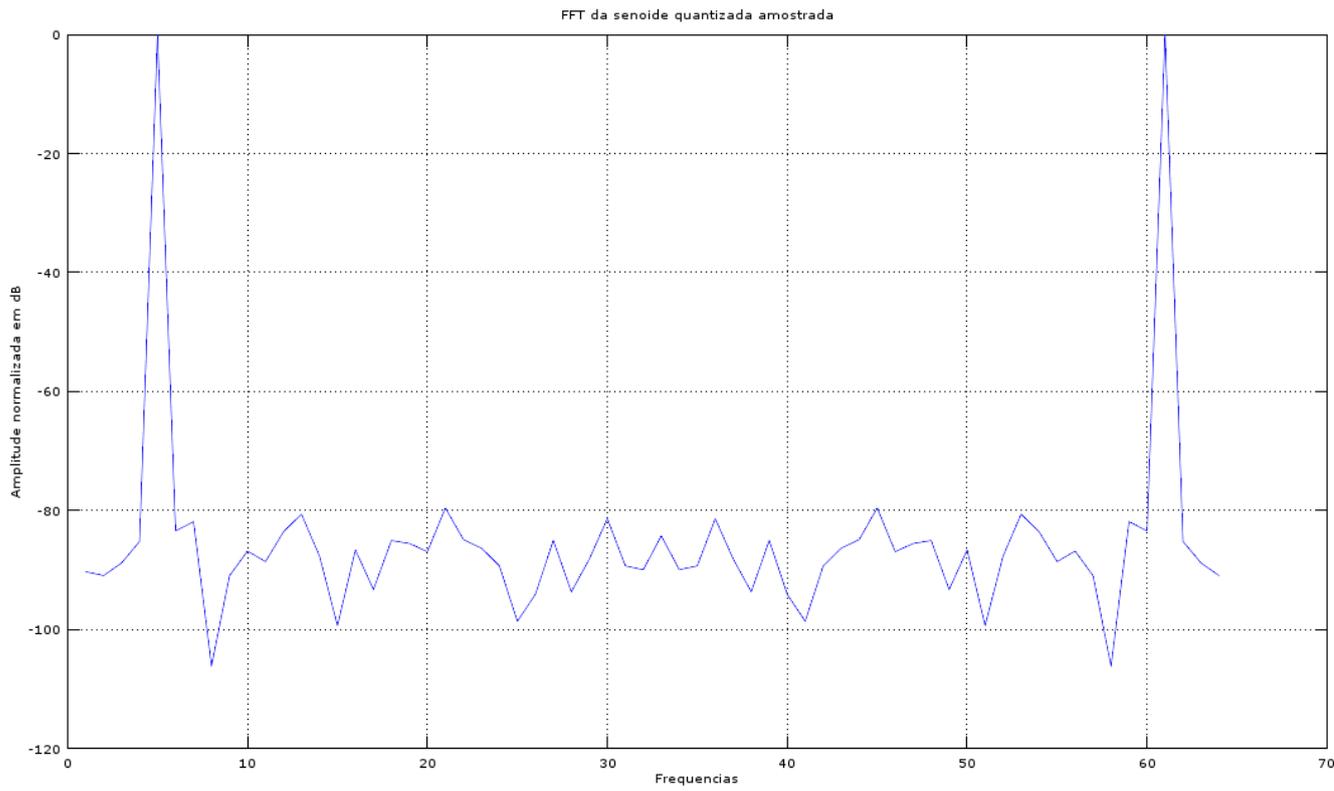


Figura 6 - FFT da senoide ideal quantizada para 12 bits

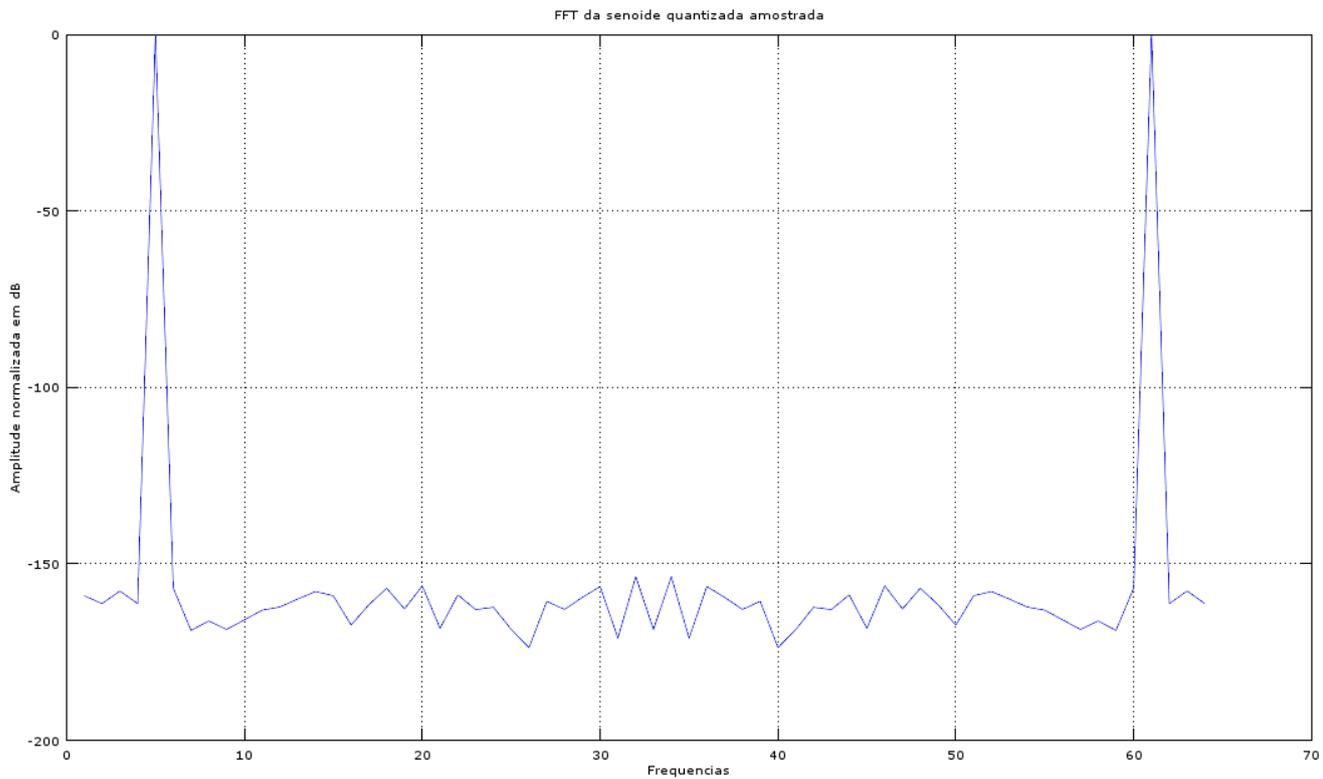


Figura 7 - FFT senoide ideal quantizada para 24 bits

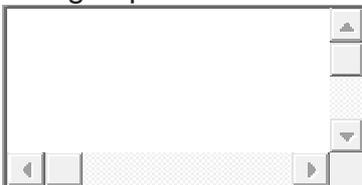
Por causa da quantização apareceram raias de ruído que "sujam" o sinal original. Quanto maior o número de bits, menor a amplitude do ruído. Por isso deve-se, sempre que possível, condicionar o sinal de entrada para abranger a amplitude máxima do conversor A/D e ter consciência que o processo de conversão e quantização pode inserir ruído indesejado. O ruído indesejado, se necessário, pode ser atenuado por meio de filtros digitais. Alguns conversores já possuem estes filtros digitais internamente para a redução dos ruídos.

O ruído de quantização é apenas uma das fontes de ruídos que são transferidos para os sinais amostrados. Por isso que é sempre importante observar, além dos parâmetros mais comuns, se a relação sinal ruído, tipicamente definida em dB, está de acordo com a precisão desejada. Veja a **Tabela 1**.

Tabela 1 - Relação mínima de sinal ruído para cada resolução do conversor A/D em bits

Bits do A/D	SNR Mínimo em dB
8	-60
10	-72
16	-96
20	-120
24	-144

A seguir pode-se ver o *script* utilizado para gerar esses dados.



- ```

1 % Gera a quantizacao da senoide ideal
2
3 Numero_de_Bits = 8; % Especifica o numero de bits
4
5 % Gera ruido de quantizacao aleatorio

```

```

6
7 ruido = (rand(length(seno1),1) -0.5) / 2^Numero_de_Bits;
8
9 senoq = seno1 + ruido'; % Gera a senoide com ruído de quantização
10
11 Cte = (2^Numero_de_Bits) - 1; % Calcula a potência de 2
12 seno1_quantizado = round(senoq * Cte); % gera o seno quantizado sem a
 componente CC
13
14 figure(3); % Abre a figura 3
15 fftseno1_quant = abs(fft(seno1_quantizado)); % Prepara a FFT
16 plot(20 * log10((fftseno1_quant/max(fftseno1_quant))+ 10^-16)); % Plota a FFT
17 grid; % Gera a grade
18
19 xlabel('Frequencias');
20 ylabel('Amplitude normalizada em dB');
21 title('FFT da senoide quantizada amostrada');
 %

```

A utilização de conversores A/D é o principal recurso para trazer para dentro do seu processador os sinais analógicos do mundo real. É necessário, porém, ter um bom conhecimento sobre o funcionamento, limitações e aplicações do conversor A/D para que se possa especificá-lo corretamente e garantir que ele atenda às necessidades do seu projeto.

#### 4- APLICAÇÕES BÁSICAS: SENSORES E AQUISIÇÃO DE DADOS

Sistemas de aquisição de dados e controle de dispositivos vêm sendo desenvolvidos para diferentes áreas de atuação, tanto industriais como científicas. O seu objetivo é apresentar ao observador os valores das variáveis ou parâmetros que estão sendo medidos.

Nos últimos anos tem-se visto um crescente desenvolvimento de sistemas de aquisição e tratamento digital de sinais, sendo que vários fatores têm contribuído para a evolução nesta área:

- O avanço da microeletrônica que tem possibilitado o aumento das capacidades e velocidades dos DSP's, elemento central numa aplicação de tratamento digital de sinal;
- A crescente performance dos computadores pessoais, tal como a sua relação qualidade/preço e a sua confiabilidade;
- A existência de cada vez mais e melhores ferramentas de desenvolvimento de software que permitem criar aplicações de alto nível com avançadas interfaces gráficas;
- O desenvolvimento de novas tecnologias de comunicação que permitem o controle remoto de instrumentos usando a internet e o wireless como veículos de transmissão de dados.

##### **Por que uma empresa precisa de um sistema de aquisição e supervisão?**

Simple...para proporcionar níveis maiores de qualidade, redução dos custos operacionais, maior desempenho de produção e fundamentalmente, para facilitar a excelência operacional.

Vale citar um pouco da história dos avanços tecnológicos e que culminaram com vários benefícios em desenvolvimento de sistemas de aquisição, controle e automação.

## **Um pouco de história**

Os primeiros sistemas de automação foram desenvolvidos no final do século XIX durante a revolução industrial.

O trabalho que era manual passou a ser realizado por máquinas dedicadas e customizadas a uma determinada tarefa visando cada vez mais o aumento da produtividade e eficiência. As funções de controle eram implementadas através de dispositivos mecânicos que automatizavam algumas tarefas críticas e repetitivas. Estes dispositivos eram desenvolvidos para cada tarefa e devido à natureza mecânica dos mesmos, tinham vida útil reduzida e alta manutenção.

Posteriormente, com o advento dos relés e contadores, estes dispositivos foram substituídos e apareceram dispositivos automáticos em linhas de montagens, dando um grande passo na época. A lógica a relés viabilizou o desenvolvimento de funções de controle mais complexas e sofisticadas.

Após a segunda guerra mundial, houve um avanço tecnológico e apareceram as máquinas por comando numérico e os sistemas de controle na indústria de processo, assim como o conceito de referência de tensão para instrumentação analógica. Aparecem os primeiros circuitos integrados, os CIs, que proporcionaram o desenvolvimento de uma nova geração de sistemas de automação. Vale lembrar que em 1947, Willian Shockley, John Barden e Walter Brattain descobriram o transistor, que é um componente eletrônico amplamente utilizado nos processadores modernos, de forma integrada.

No início dos anos 70, os primeiros computadores comerciais começaram a ser utilizados como controladores em sistemas de automação de grande porte, porém estes computadores eram grandes, ocupando muito espaço, de alto custo, difíceis de programar e muito sensíveis ao ambiente industrial. Mas tinham a vantagem de manipular a aquisição e controle de várias variáveis.

Ainda na década de 70 tivemos um grande avanço em termos de automação.

A partir de uma demanda existente na indústria automobilística norte-americana, foi desenvolvido o *Programmable Logic Controller* (PLC), ou Controlador Lógico Programável (CLP). O CLP é um computador dedicado e projetado para trabalhar no

ambiente industrial, onde sensores e atuadores são conectados a cartões de entradas e saídas. Os primeiros CLPs tinham um conjunto de instruções reduzido; normalmente somente condições lógicas e não possuíam entradas analógicas, podendo manipular apenas aplicações de controle discreto. Os CLPs substituíram os painéis de controle com relés, diminuindo assim, o alto consumo de energia, a difícil manutenção e modificação de comandos e também as onerosas alterações de fiação.

Atualmente, devido à demanda das plantas industriais, os CLPs manipulam tanto controle discreto quanto malhas analógicas. Estes sistemas são usualmente chamados de Controladores Programáveis, por não serem limitados a operações com condições lógicas. As atuais funções de controle existentes em uma planta industrial são em geral distribuídas entre um número de controladores programáveis, os quais são montados próximos aos equipamentos a serem controlados. Os diferentes controladores são usualmente conectados via rede local a um computador supervisor central, o qual gerencia os alarmes, receitas e relatórios.

Entramos em uma fase onde a tecnologia e conectividade industrial eram proprietárias e um “casamento” entre cliente e fornecedor acontecia. No mercado apareceram os SDCSs (Sistemas Digitais de Controle Distribuídos).

Na década de 90, o mundo começou a presenciar enormes avanços na área tecnológica, em que os circuitos eletrônicos passaram a proporcionar maior eficiência, maiores velocidades, mais funcionalidades, maiores MTBFs (Mean Time Between Failures, maior confiabilidade), consumos menores, espaços físicos menores e ainda, com reduções de custos. Ao mesmo tempo em que impulsionou o desenvolvimento de computadores, interfaces e periféricos mais poderosos, com alta capacidade de processamento e memória e o mais interessante, dando vazão a alta escala de produção com custos reduzidos e o que foi uma vantagem de forma geral, pois aumentou a oferta de microcontroladores, CIs e ASICs para toda a indústria.

E se não bastasse esta revolução eletrônica, os sistemas mecânicos também passaram e vêm passando por inovações e modificações conceituais com a

incorporação da capacidade de processamento, tornando-os mais rápidos, eficientes e confiáveis, com custos de implementação cada vez menores. Ao longo dos últimos anos é cada vez mais freqüente a utilização de componentes eletrônicos para acionamento e controle de sistemas mecânicos.

Um grande desenvolvimento verificou-se na interface gráfica com o advento de ferramentas baseadas no Windows. A simplicidade da operação juntou-se à crescente capacidade de processamento estabelecendo a união entre os computadores e a instrumentação. Tornou-se habitual os instrumentos serem embutidos em computadores de aplicação geral, permitindo medidas diversas e manipulações complexas, e ainda grandes capacidades de armazenamento em memória ou disco, monitorização inteligente, apresentação gráfica de fácil compreensão e controle dos processos avançados.

Não resta dúvida que hoje não é somente a condição de controle que importa. A gestão da informação, a inteligência da instrumentação, a tecnologia verdadeiramente aberta e não proprietária, os benefícios da tecnologia digital são o que agregam valores ao usuário.

### **Objetivo de um sistema de aquisição de dados**

Os sistemas de aquisição têm por finalidade obter dados digitais através de medições de grandezas físicas (analógicas), tais como temperatura, pressão, densidade, pH, umidade, posição, etc, e vêm sendo incorporados por diversas vertentes, tanto industriais quanto científicas.

Estas grandezas são “sentidas” por sensores ou transdutores e são convertidas em quantidades elétricas por acondicionadores de sinais que levam os sinais até hardwares dedicados e os transformam em valores digitais. Estes valores são processados por controladores e segundo lógicas de controles devolvem a resposta processada aos atuadores (elementos finais de controle), como na figura 1.

Em sua grande parte, os valores medidos não são iguais ao da variável real, ou seja, o sistema de aquisição tem como entrada o valor real da variável e sua saída é o valor medido.

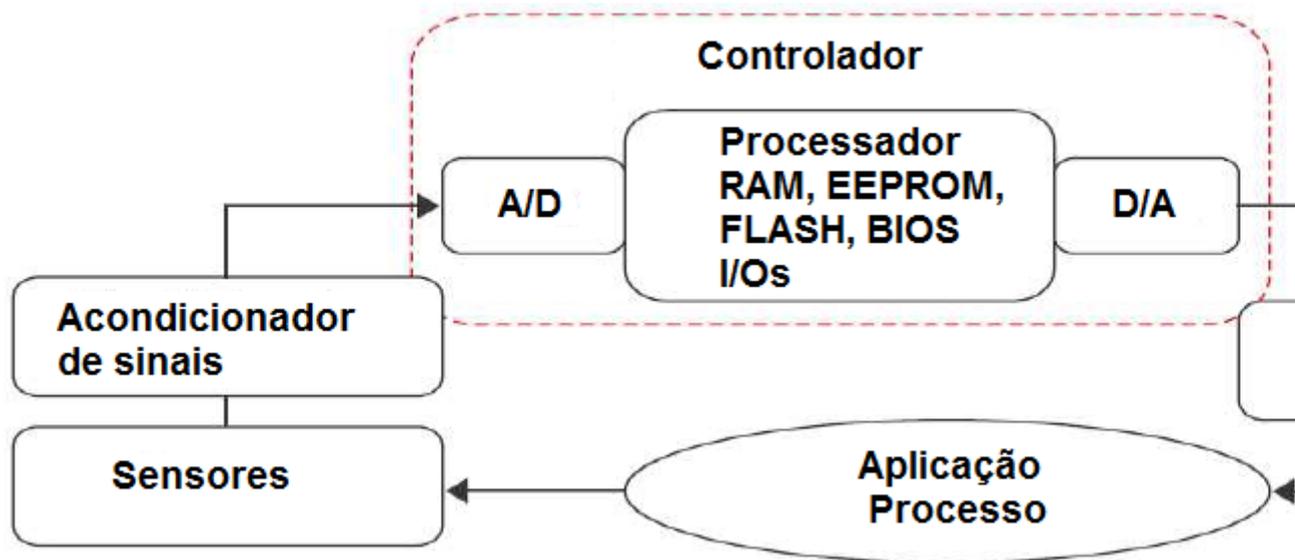


Figura 1 – Componentes de um sistema de aquisição de dados

Os sensores transformam um fenômeno físico em sinais elétricos que uma vez detectados/produzidos variam de acordo com os parâmetros físicos que estão sendo monitorados, e devem ser acondicionados para fornecer sinais apropriados ao hardware de aquisição de dados. Os circuitos e ou elementos de acondicionamento de sinais amplificam, isolam, filtram e excitam sinais para que estes sejam apropriados aos hardwares de aquisição. Uma vez acondicionados e trabalhados na forma desejada, os sinais podem ser lidos em controladores ou mesmo computadores, com placas específicas instaladas, e armazenados em diferentes formas, como arquivos de texto, excel, banco de dados, etc.

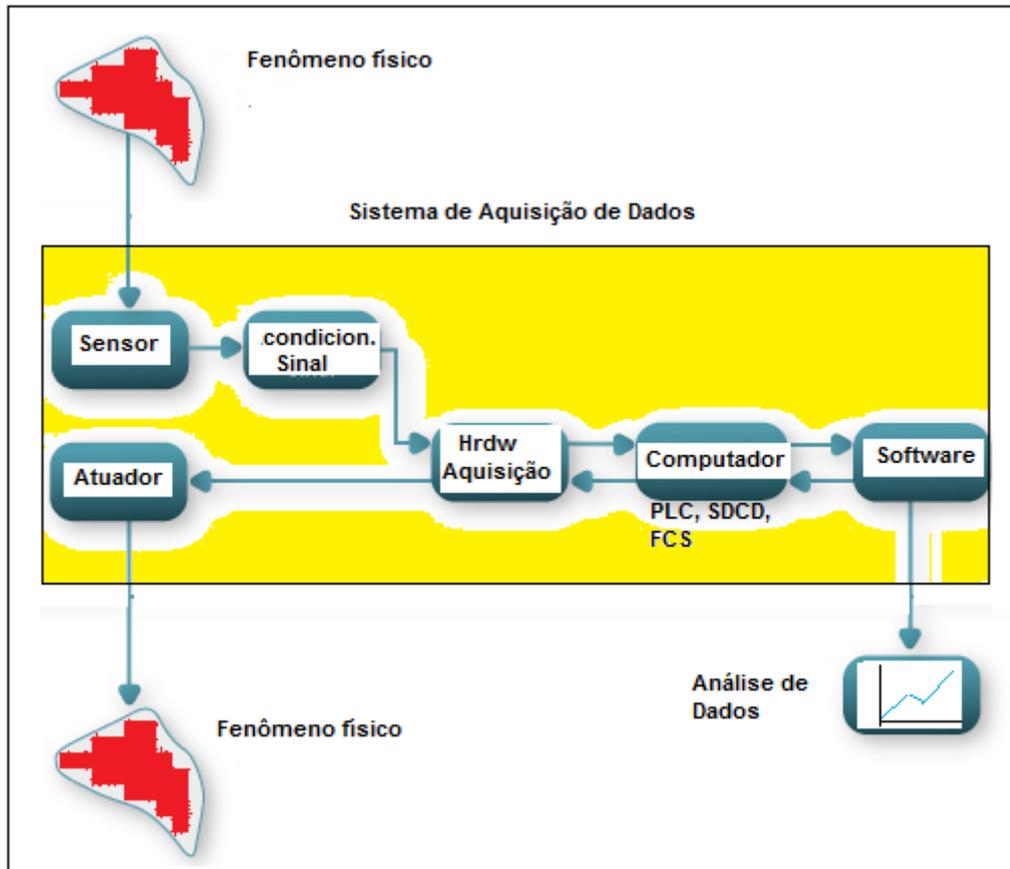


Figura 2 - Componentes de um sistema de aquisição de dados e fenômeno físico

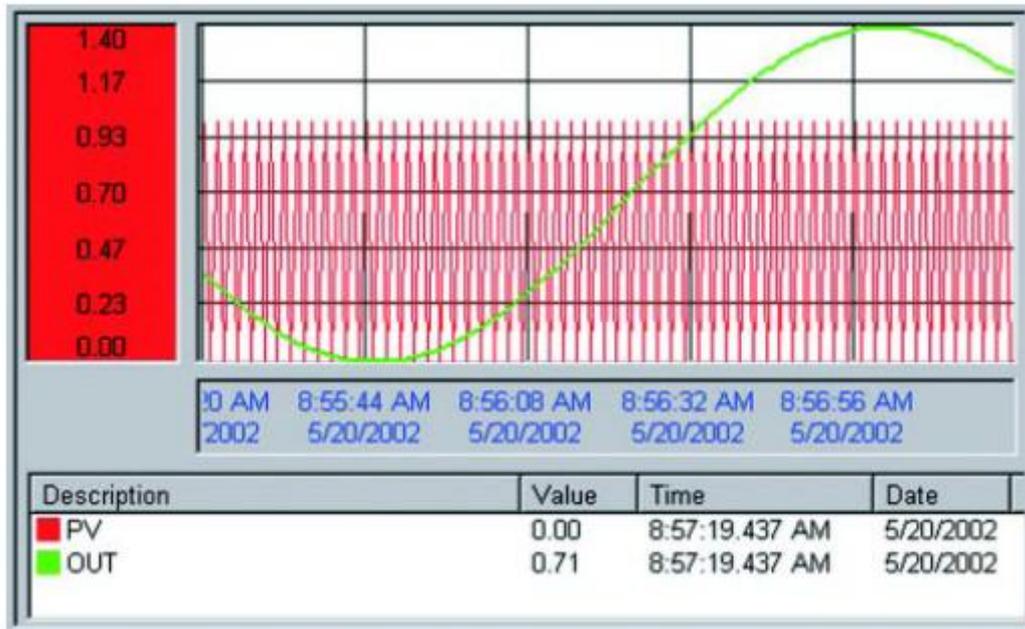


Figura 3- Exemplo de um Trend - ProcessView

#### Tipo de endereçamento em Sistemas de Aquisição de Dados

Os dispositivos responsáveis por manipular os dados utilizam endereços que podem ser de memória ou de registradores, dependendo do tipo de dispositivo usado.

Assim, tem-se entrada e saída mapeada em memória (*Memory-Mapped IO*) e entrada e saída mapeada em espaço de entrada e saída (*IO Mapped IO*).

#### Sistemas de Supervisão e Aquisição de Dados

Abreviadamente SCADA (proveniente do seu nome em inglês *Supervisory Control and Data Aquisition*) são sistemas que utilizam software para monitorar e supervisionar as variáveis e os dispositivos de sistemas de controle. São conectados através de drivers específicos. Estes sistemas podem assumir topologia simples, cliente-servidor ou múltiplos servidores-clientes. Atualmente, com o advento de

sistemas de automação e controle baseados em redes digitais abertas permitem arquiteturas cliente-servidor OPC (OLE for Process Control).

De forma genérica, um sistema de supervisão é uma ferramenta de software que permite monitorar e controlar partes ou todo um processo industrial. Têm 2 módulos básicos: o desenvolvedor e o executável ("run-time").

A maioria dos passos de programação é automatizada, suprimindo a maior parte das necessidades de um projeto sem a real necessidade de conhecimento de linguagem de programação. Em casos mais complexos e específicos, onde os passos não estão automatizados, algumas ferramentas incorporam módulos de programação em VBA (Visual Basic For Applications) ou VBS (Visual Basic Script).

### **Data Mining**

Prospecção de dados ou mineração de dados (também conhecida *data mining*) é o processo de explorar grandes quantidades de dados à procura de padrões consistentes, com regras de associação ou sequências temporais, para detectar relacionamentos sistemáticos entre variáveis, detectando assim novos subconjuntos de dados.

Esse é um tópico recente em ciência da computação mas que utiliza várias técnicas da estatística, recuperação de informação, inteligência artificial, sistemas de aquisição de dados e reconhecimento de padrões. A mineração de dados é formada por um conjunto de ferramentas e técnicas que através do uso de algoritmos de aprendizagem ou classificação baseados em redes neurais e estatística, são capazes de explorar um conjunto de dados, extraíndo ou ajudando a evidenciar padrões nestes dados e auxiliando na descoberta de conhecimento. Esse conhecimento pode ser apresentado por essas ferramentas de diversas formas: agrupamentos, hipóteses, regras, árvores de decisão, grafos, ou dendrogramas.

Um passo fundamental para um processo de data mining bem sucedido é ter um sistema de aquisição de dados que possa gerar um banco de dados, relatórios, logs, etc. Disto nascem os repositórios organizados (*Data Marts* e *Data Warehouses*).

A ferramenta de aquisição de dados

Este tipo de ferramenta exige a transferência rápida de dados entre dispositivos, software de processamento e de aplicação com interface gráfica avançada, sensores e controladores de elevada precisão. Com a velocidade dos avanços tecnológicos, cada vez mais equipamentos e instrumentação vêm com hardware poderosos. Na vertente do software, as tecnologias vão-se tornando cada vez mais normalizadas, permitindo um desenvolvimento mais rápido e eficaz dos produtos.

O campo de estudos da instrumentação virtual ainda está a dar os primeiros passos. Nos próximos anos serão desenvolvidos vários equipamentos avançados em recursividades, os chamados instrumentos virtuais. Serão os blocos para a construção da nova geração de instrumentação e medida. As ferramentas devem ser simples e de fácil uso pelo usuário. Deve permitir o acesso a toda a funcionalidade proporcionada pela aplicação, maximizando a funcionalidade e mantendo a simplicidade da interface.

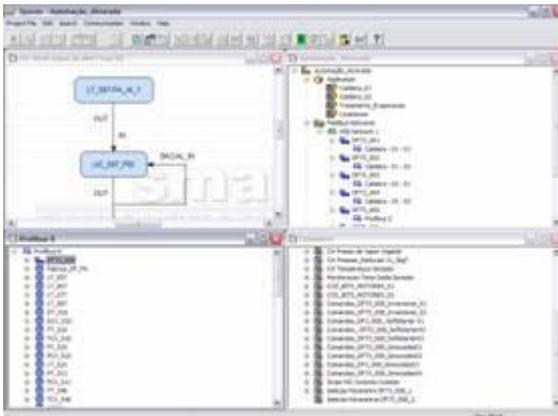




Figura 2 - Típica tela de representação de processos

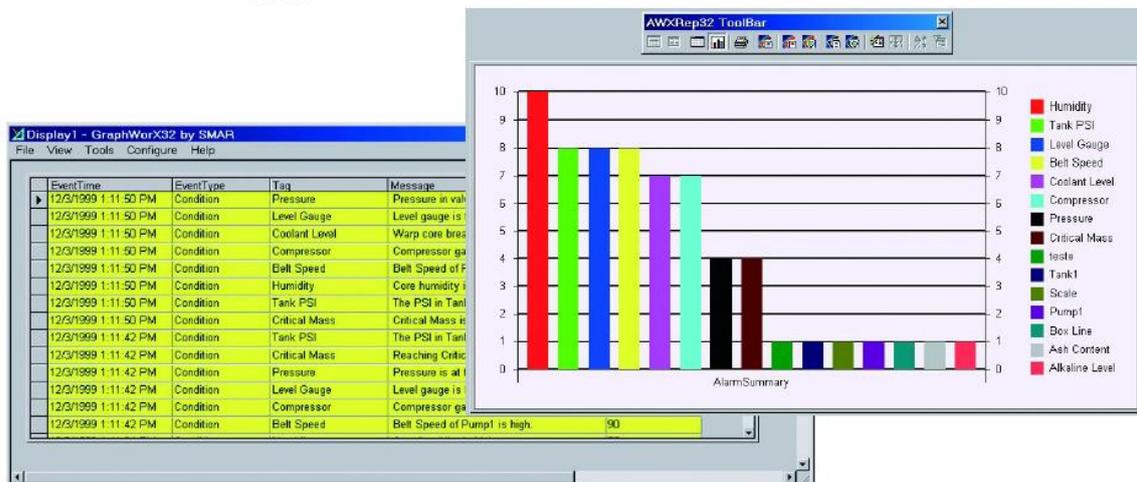
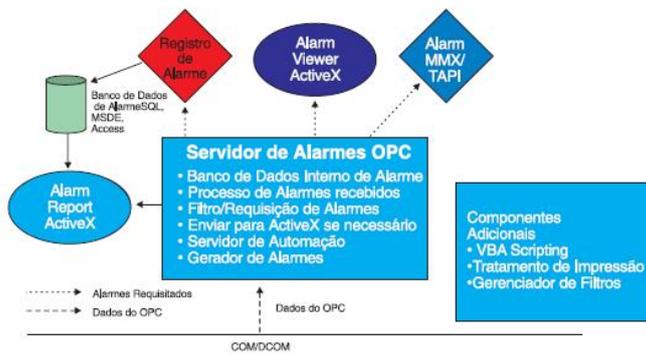


Figura 3 - Alarmes e Eventos

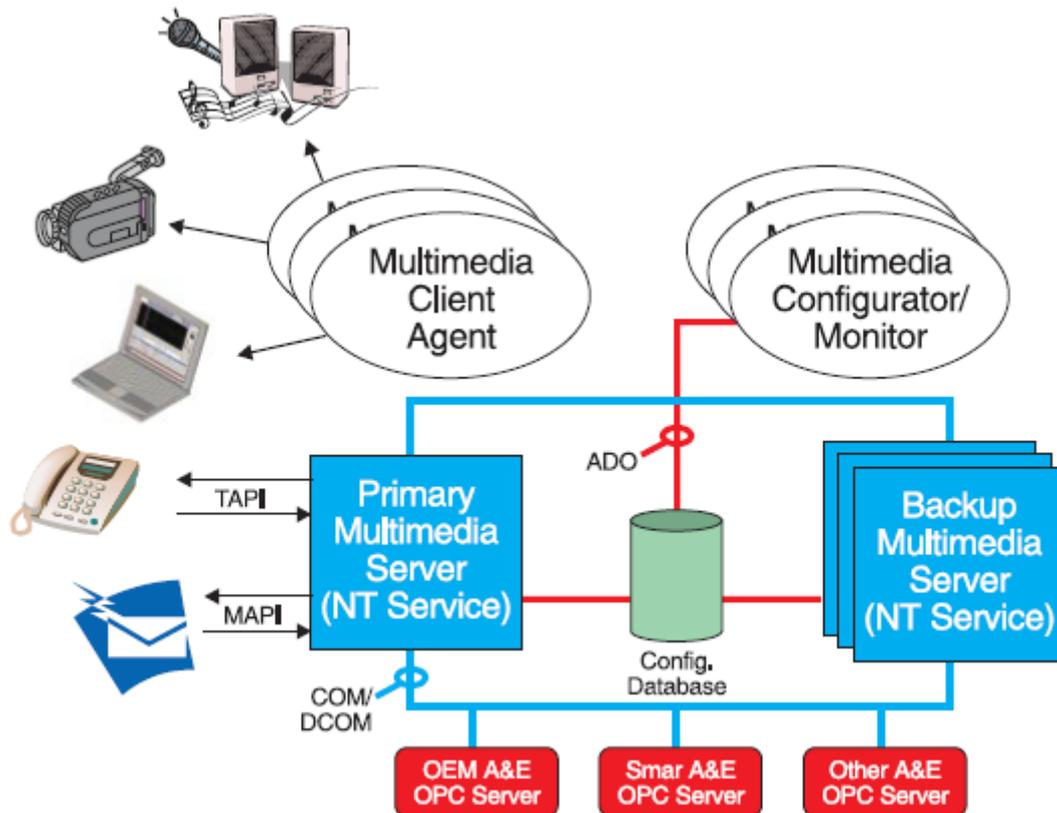


Figura 4 – Anúncios Avançados Multimídia

### Conclusão

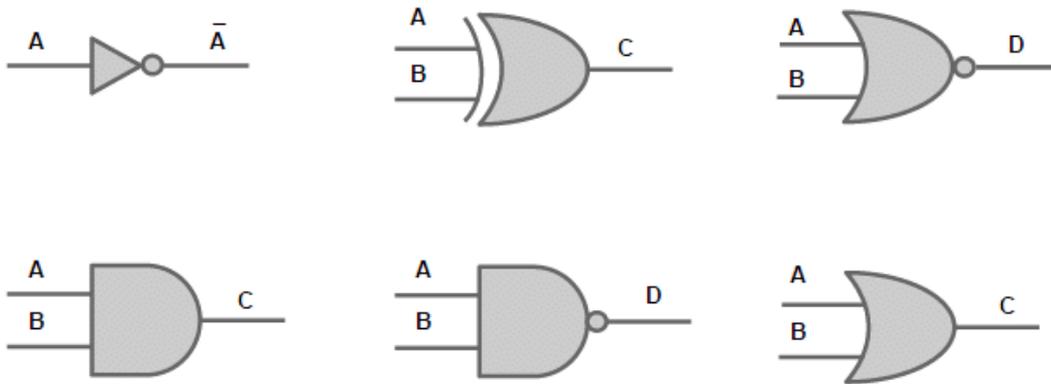
A aplicação da computação para automatização de diversos processos industriais tem crescido de forma significativa, e mais certo são os benefícios advindos da introdução da computação no processo produtivo que tem incrementado a produtividade e reduzido as possibilidades de falhas durante o processo.

Vimos neste artigo, de forma breve que uma das aplicações que se destaca é a utilização sistemas de aquisição de dados e supervisão para monitorar os mais diversos sensores que integram uma determinado processo e a partir das informações obtidas enviar sinais de controle para outros dispositivos, ou gerar relatórios técnicos para posterior análise.

Usuários vem colhendo benefícios com sistemas avançados de aquisição de dados e supervisão. Essa mudança é encarada como um processo natural demandado pelos novos requisitos de qualidade, confiabilidade e segurança do mercado. A sua

utilização traz uma vantagem competitiva, no sentido que essa nova tecnologia traz aumentos de produtividade pela redução das variabilidades dos processos e redução dos tempos de indisponibilidade das malhas de controle.

## 5- PORTAS LÓGICAS

**Sistema Digital**

Um sistema digital é um sistema matemático que define informações como valores numéricos. Dessa forma, é possível definir operações digitais como cálculos matemáticos. Comumente trabalhamos com valores numéricos na base decimal, mas um sistema digital trabalha de maneira diferente. Em analogia ao sistema decimal, onde cada dígito possui 10 valores possíveis, um sistema digital é um sistema binário, onde cada dígito possui apenas 2 valores possíveis. Esses dois valores são definidos como "níveis lógicos" e adota-se o valor de 0 (zero) ou 1 (um) apenas.

Transportando esse sistema para um sistema eletrônico, é necessário apresentar esses dois valores como sinais elétricos. Para tanto, podemos entendê-los como:

- Ligado ou desligado;
- Nível alto ou nível baixo;
- Alimentado ou em zero;

- VCC ou Terra.

As operações observáveis para esses níveis lógicos são definidas como operações lógicas. Todas as possíveis operações lógicas são baseadas em apenas 3 operações primárias, que são:

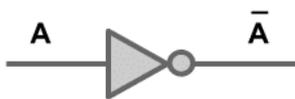
- Inversão;
- Soma lógica;
- Produto lógico.

## Portas Lógicas

### Porta Lógica NOT

A porta NÃO ou inversora (NOT) utiliza o operador de inversão. Para um determinado valor na entrada, a saída possui um valor contrário ao da entrada. Se a entrada for 1, a saída será 0. Se a entrada for 0, a saída será 1. Ou seja, para um valor na entrada a saída será seu complemento, ou o inverso do valor na entrada.

## PORTA NÃO (NOT)



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

Porta Lógica NOT

### Porta NOT em Verilog:

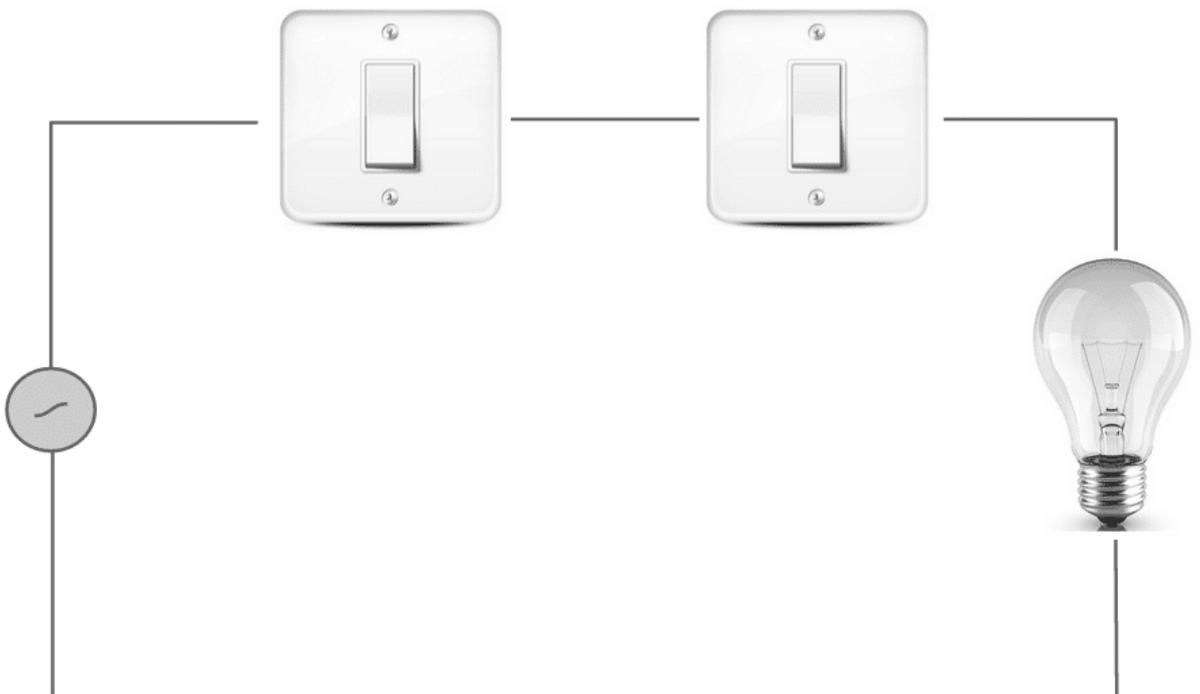


```
1 module exemplo_INVERSOR (a, b);
```

2 input wire a;  
3 output wire b;  
4  
5 assign b = ~a;  
6  
7 endmodule

### Porta Lógica AND

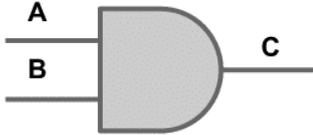
Para se explicar o funcionamento da porta E (AND) pode-se fazer um paralelo com um circuito com interruptores, como na figura abaixo. Para que a lâmpada acenda é preciso que os dois interruptores estejam ligados.



### Analogia com a porta lógica AND

A porta lógica E (AND) utiliza-se do operador de produto lógico. A saída é igual a 1 se todas as entradas for 1. A saída é igual a zero se ao menos uma entrada for 0, se todas entradas não forem 1.

## PORTA E (AND)



$$C=A \cdot B$$

| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Porta Lógica AND

## Porta AND em Verilog:



```
1 module exemplo_AND (A, B, C);
```

```
2 input wire A, B;
```

```
3 output wire C;
```

```
4
```

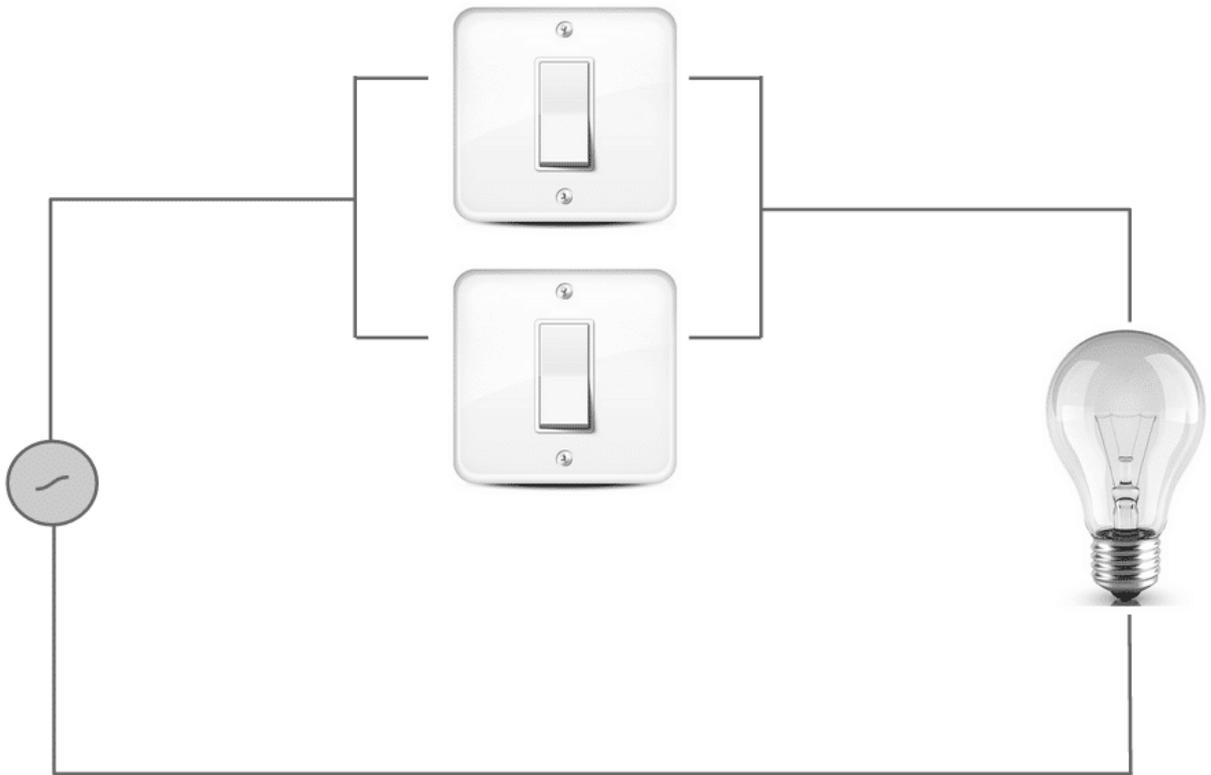
```
5 assign C = A & B;
```

```
6
```

```
7 endmodule
```

**Porta Lógica OR**

Uma forma simples de se entender o funcionamento da porta OU (OR) é pensar em um circuito com interruptores em paralelo, como na figura abaixo. Para que a lâmpada acenda é preciso que um dos dois interruptores esteja ligado.

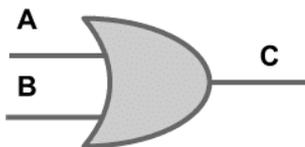


**Analogia com a porta lógica OR**

A porta lógica OU (OR) utiliza-se do operador de soma lógica. A saída é igual a 1 se pelo menos uma das entradas for 1. A saída é igual a zero se nenhuma entrada for 1, todas forem zero.

**PORTA OU (OR)**

$C=A+B$



| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Porta Lógica OR

**Porta OR em Verilog:**



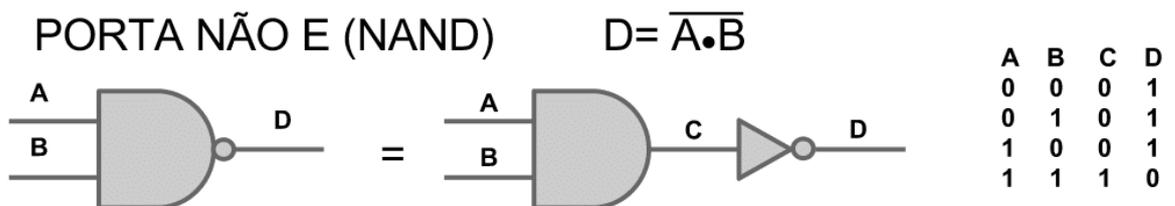
```

1 module exemplo_OR (a, b, c);
2 input wire a, b;
3 output wire c;
4
5 assign c = a | b;
6
7 endmodule

```

**Porta Lógica NAND**

A porta lógica NÃO E (NAND) utiliza-se do operador de produto lógico e o de inversão. A saída é igual a 0 se todas as entradas for 1. A saída é igual a 1 se ao menos uma entrada for 0, se todas entradas não forem 1.



Porta Lógica NAND

**Porta NAND em Verilog:**



```

1 module exemplo_NAND (a, b, d);
2 input wire a, b;
3 output wire d;
4

```

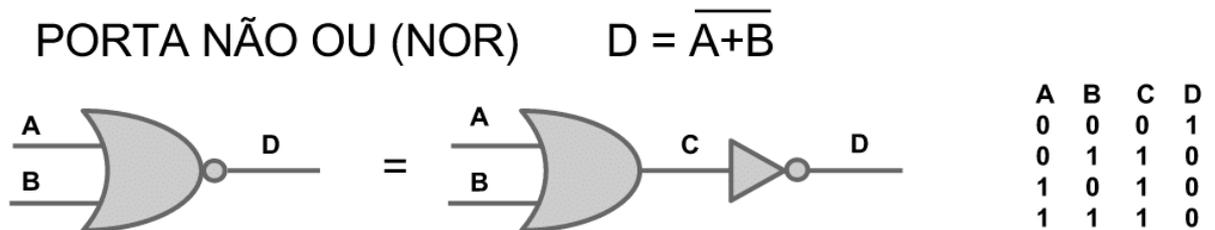
5 assign d = a ~& b;

6

7 endmodule

### Porta Lógica NOR

A porta lógica NÃO OU (NOR) utiliza-se do operador de soma lógica e o de inversão. A saída é igual a 0 se pelo menos uma das entradas for 1. A saída é igual a 1 se nenhuma entrada for 1, todas forem zero.



Porta Lógica NOR

### Porta NOR em Verilog:



1 module exemplo\_NOR ( a, b, d );

2 input wire a, b;

3 output wire d;

4

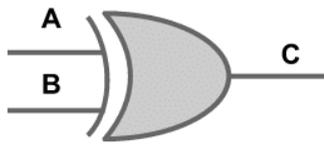
5 assign d = a ~| b;

6

7 endmodule

## Porta Lógica XOR

A porta lógica OU EXCLUSIVO (XOR) utiliza-se do operador de soma lógica, com um círculo. A saída é igual a 0 se as entradas forem iguais. A saída é igual a 1 se as entradas não forem iguais, se uma delas diferirem das outras.

PORTA OU EXCLUSIVO (XOR)  $C=A\oplus B$ 

| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Porta Lógica

XOR

## Porta XOR em Verilog:



```

1 module exemplo_XOR (a, b, c);
2 input wire a, b;
3 output wire c;
4
5 assign c = a ^ b;
6
7 endmodule

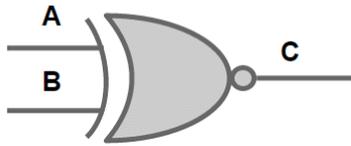
```

## Porta Lógica XNOR

A porta lógica NÃO OU EXCLUSIVO (XNOR) utiliza-se do operador de soma lógica, com um círculo e o de inversão. Tem as saídas inversas da operação XOR. A saída

é igual a 1 se as entradas forem iguais. A saída é igual a 0 se se as entradas não forem iguais, se uma delas diferirem das outras.

PORTA NÃO OU EXCLUSIVO (XNOR)  $C = \overline{A \oplus B}$



| A | B | C |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Porta

Lógica XNOR

**Porta XNOR em Verilog:**



```
1 module exemplo_XNOR (a, b, c);
```

```
2 input wire a, b;
```

```
3 output wire c;
```

```
4
```

```
5 assign c = a ~^ b;
```

```
6
```

```
7 endmodule
```

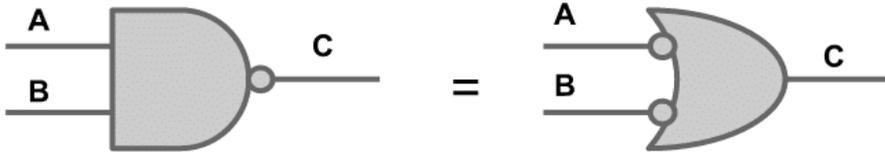
Teorema de De Morgan

O Teorema de De Morgan diz respeito às seguintes afirmações:

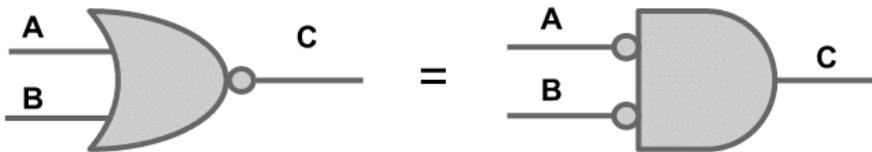
- Uma operação NAND é igual a uma operação OR com todas as entradas invertidas;
- Uma operação NOR é igual a uma operação AND com todas as entradas invertidas.

Com essas duas afirmações podemos fazer diversas simplificações em expressões lógicas, referentes a circuitos digitais.

$$C = \overline{A \cdot B} = \overline{A} + \overline{B}$$



$$C = \overline{\overline{A} + \overline{B}} = \overline{\overline{A}} \cdot \overline{\overline{B}}$$



de deMorgan

Teorema

## 6- FLIP-FLOPS

Em eletrônica e circuitos digitais, um **flip-flop, multivibrador biestável**, ou simplesmente **biestável**, é um circuito digital que pulsado capaz de servir como uma memória de um bit. Um *flip-flop* tipicamente inclui zero, um ou dois sinais de entrada, um sinal de relógio, e um sinal de saída, apesar de muitos flip-flops comerciais proverem adicionalmente o complemento do sinal de saída. Alguns flip-flops também incluem um sinal da entrada *clear*, que limpa a saída atual. Como os flip-flops são implementados na forma de circuitos integrados, eles também necessitam de conexões de alimentação. A pulsação ou mudança no sinal de relógio faz com que o flip-flop mude ou retenha seu sinal de saída, baseado nos valores dos sinais de entrada e na equação característica do flip-flop.

De forma simplificada a representar de um circuito flip-flop como um bloco obtendo 2 saídas: **Q1** e **Q2**, entrada para as variáveis e uma entrada de controle (Clock). A saída Q1 será a principal do bloco. Este sistema é composto por basicamente dois estados de saída. Podendo assumir um destes estados sendo necessário que haja uma combinação das variáveis e do pulso de controle (Clock). Alternando seu valor por pulsos de clock, permanece neste estado até a chegada de um novo pulso de clock e, então, de acordo com as variáveis de entrada, alternará ou não de estado anterior.

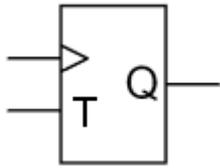
Quatro tipos de flip-flops possuem 8 aplicações comuns em sistemas de clock não-sequencial: **flip-flop T** ("toggle"), **flip-flop S-R** ("set-reset"), **flip-flop J-K** e o **flip-flop D** ("data"). O comportamento de um flip-flop é descrito por sua equação característica, que prevê a "próxima" (após o próximo pulso de clock) saída,

em termos dos sinais de entrada e/ou da saída atual,

O primeiro flip-flop eletrônico foi inventado em 1919 por William Eccles e F. W. Jordan. Ele foi inicialmente chamado de *circuito de disparo Eccles-Jordan*. O nome flip-flop posterior descreve o som que é produzido em um alto-falante conectado a uma saída de um amplificador durante o processo de chaveamento do circuito

**Flip-flop Tipo T**

Se T estiver em estado alto, o flip-flop T (*toggle*) muda o estado sempre que a entrada de clock sofrer uma modificação. Se a entrada T for baixa, o flip-flop mantém o valor do seu estado. Seu comportamento é descrito pela seguinte equação característica:



O símbolo do flip-flop T, onde > é a entrada de clock, T é a entrada de toggle e Q é informação de saída armazenada.

(expandindo o operador XOR)

e pela tabela verdade<sup>[1]</sup>:

| T | Q | Q*       |
|---|---|----------|
| 0 | 0 | <b>0</b> |
| 1 | 1 | <b>0</b> |
| 0 | 1 | <b>1</b> |
| 1 | 0 | <b>1</b> |

Q\* → Estado seguinte do Q

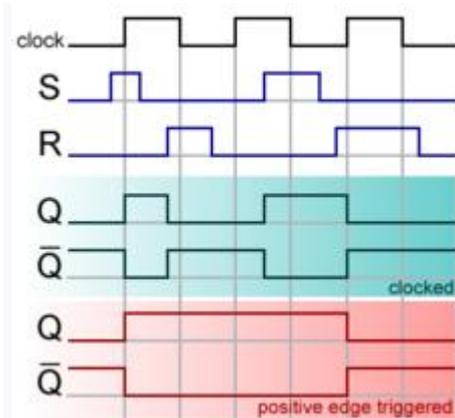
**Flip-flop SR Síncrono**

Diagrama de tempos de um flip-flop SR

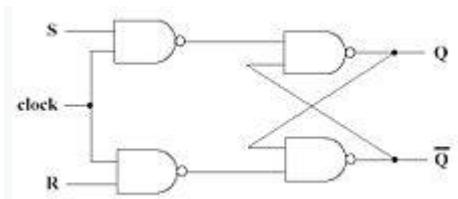
Um flip-flop S-R Síncrono depende da habilitação de suas entradas por um sinal de clock para que essas possam alterar o estado dele. Este sinal pode operar de duas formas: mantendo as entradas ativas durante todo o período do pulso ou apenas no instante da mudança de estado do sinal de clock. Essas duas formas de operação podem ser denominadas como modo clocked e modo triggered, respectivamente.

O flip-flop "set/reset" *ativa* (set, muda sua saída para o nível lógico 1, ou retém se este já estiver em 1) se a entrada S ("set") estiver em 1 e a entrada R ("reset") estiver em 0 quando o clock for mudado. O flip-flop *desativa* (reset, muda sua saída para o nível lógico 0, ou a mantém se esta já estiver em 0) se a entrada R ("reset") estiver em 1 e a entrada S ("set") estiver em 0 quando o clock estiver habilitado. Se ambas as entradas estiverem em 0 quando o clock for mudado, a saída não se modifica. Se, entretanto, ambas as entradas estiverem em 1 quando o clock estiver habilitado, nenhum comportamento particular é garantido. Isto é comumente escrito na forma de uma "tabela verdade"

| Q | Q* | S | R |
|---|----|---|---|
| 0 | 0  | 0 | X |

|   |   |   |   |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | X | 0 |

$Q^* \rightarrow$  Estado seguinte do Q



Estrutura de um Flip-Flop S-R

Exemplo com a tabela verdade mostrando o estado anterior:

| S | R | Qant | Qfim | Qfim* | Descrição        |
|---|---|------|------|-------|------------------|
| 0 | 0 | 0    | 0    | 1     | Fixa Qfim = Qant |
| 0 | 0 | 1    | 1    | 0     | Fixa Qfim = Qant |
| 0 | 1 | 0    | 0    | 1     | Fixa Qfim = 0    |
| 0 | 1 | 1    | 0    | 1     | Fixa Qfim = 0    |
| 1 | 0 | 0    | 1    | 0     | Fixa Qfim = 1    |

|   |   |   |   |   |                      |
|---|---|---|---|---|----------------------|
| 1 | 0 | 1 | 1 | 0 | <b>Fixa Qfim = 1</b> |
| 1 | 1 | 0 | 1 | 1 | <b>Não Permitido</b> |
| 1 | 1 | 1 | 1 | 1 | <b>Não Permitido</b> |

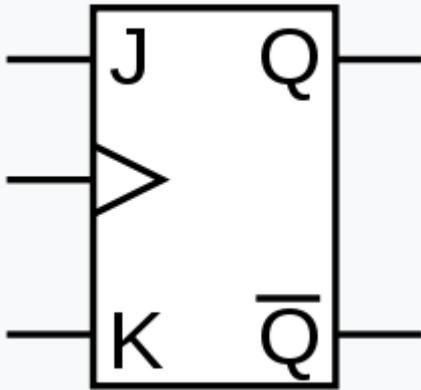
Legenda: S (Set), R (Reset), Qant (Estado anterior da saída Q), Qfim (Estado em que a saída deve assumir "estado futuro" após a aplicação das entradas), Qfim\* (Qfim Linha) De: Adilson José Ngonga

### **Flip-flop J-K**



Diagrama de tempos do flip-flop J-K

O flip-flop J-K tem a prioridade de aprimorar o funcionamento circuito flip-flop R-S interpretando a condição  $S = R = 1$  como um comando de inversão. Especificamente, a combinação  $J = 1, K = 0$  é um comando para ativar (set) a saída do flip-flop; a combinação  $J = 0, K = 1$  é um comando para desativar (reset) a saída do flip-flop alternando a condição inicial; e a combinação  $J = K = 1$  é um comando para inverter o flip-flop, trocando o sinal de saída pelo seu atualizado. Fazendo  $J = K$  o flip-flop J-K se torna um flip-flop T(Toggle).



O símbolo do flip-flop J-K, onde > é a entrada de clock, J e K são as entradas de dados, Q é a saída de dados armazenada e Q' é o complemento de Q.

A equação característica do flip-flop J-K é:

e sua tabela verdade é:

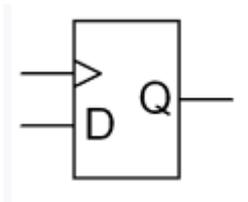
| Operação do Flip Flop JK |   |                       |                      |
|--------------------------|---|-----------------------|----------------------|
| Tabela Verdade           |   |                       |                      |
| J                        | K | Q <sub>próx</sub>     | Comentário           |
| 0                        | 0 | Q <sub>anterior</sub> | mantém (hold)        |
| 0                        | 1 | 0                     | reestabelece (reset) |
| 1                        | 0 | 1                     | estabelece (set)     |
| 1                        | 1 | Q <sub>anterior</sub> | alterna (Toggle)     |

$Q^* \rightarrow$  Estado anterior do Q

O flip-flop J-K recebeu este nome em homenagem a Jack Kilby, o homem que inventou o circuito integrado, em 1958, pelo qual ele recebeu o prêmio Nobel em Física no ano 2000. "Jump-kill", também é utilizado como analogia a "set-reset".

### **Flip-flop D (Data)**

O flip-flop D ("Data" ou dado, pois armazena o bit de entrada) possui uma entrada, que é ligada diretamente à saída quando o clock é mudado. Independentemente do valor atual da saída, ele irá assumir o valor 1 se  $D = 1$  quando o clock for mudado ou o valor 0 se  $D = 0$  quando o clock for mudado. Este flip-flop pode ser interpretado como uma linha de atraso primitiva ou um hold de ordem zero, visto que a informação é colocada na saída um ciclo depois de ela ter chegado na entrada.



O símbolo esquemático de um flip-flop D, onde  $\triangleright$  é a entrada de clock, D é a entrada de dados e Q é a saída de dados.

A equação característica do flip-flop D é:

A sua tabela verdade é:

| D | Q | $Q^*$ |
|---|---|-------|
| 0 | 0 | 0     |
| 0 | 1 | 0     |

|   |   |   |
|---|---|---|
| 1 | 0 | 1 |
| 1 | 1 | 1 |

$Q^*$  → Estado posterior do

O flip-flop pode ser utilizado para armazenar um bit, ou um dígito binário de informação. A informação armazenada em um conjunto de flip-flops pode representar o estado de um sequenciador, o valor de um contador, um caractere ASCII em uma memória de um computador ou qualquer outra parte de uma informação.

Um uso é a construção de máquinas de estado finito a partir da lógica eletrônica. O flip-flop lembra o estado anterior de máquina, e a lógica digital utiliza este estado para calcular o próximo estado.

### ***Flip-flop T (toggle)***

O flip-flop "T" é útil para contagens. Sinais repetidos à entrada de clock farão com que o flip-flop mude seu estado a cada transição de nível alto-para-baixo da entrada de clock. Se sua entrada T for "1", a saída de um flip-flop pode ser ligada à entrada clock de um segundo flip-flop e assim por diante até a saída final do circuito, considerada com o conjunto de todas as saídas dos flip-flops individuais. A esta montagem formada, caracterizamos como uma contagem, em sistema binário, do número de ciclos da primeira entrada de clock, até um limite máximo de  $2^n - 1$ , onde  $n$  é o número de flip-flops utilizados no circuito.

Um dos problemas com este tipo de contador (chamado de *contador de ripple* ou *contador de pulsos*) é que a saída é brevemente inválida conforme ocorre a mudança de pulso através da lógica. Existem duas soluções para este problema. A primeira é retirar uma amostra da saída apenas quando a mesma for válida. A segunda, mais utilizada, é montar um tipo diferente de contador, chamado de *contador síncrono*. Este utiliza uma lógica mais complexa para garantir que as saídas do contador mudem todas a um mesmo período de tempo.

Divisão de frequência: um conjunto de flip-flops "T" utilizados da maneira descrita acima irá funcionar de modo a dividir a frequência da entrada por  $2^n$  na saída do último flip-flop, aonde  $n$  é o número de flip-flops utilizados entre a entrada e a saída.

Os registradores podem ser utilizados para armazenar dados nos computadores. Um flip-flop "D" pode representar um dígito de um número binário. A unidade de controle do computador envia o sinal de clock no momento certo para poder capturar estes dados.

Pela família CMOS o flip-flop D é representado pelo integrado 4013 na série 4000 e pelos integrados versão HC da série 74XX.

### **Temporização e metaestabilidade**

---

Um flip-flop em combinação com um Schmitt Trigger pode ser utilizado para a implementação de um arbitro em circuitos assíncronos.

Os flip-flop com clock estão predispostos a um problema chamado de metaestabilidade, que ocorre quando um dado ou uma entrada de controle está mudando no momento do pulso de clock. O resultado é que a saída pode se comportar imprevisivelmente, levando muito tempo mais que o seu normal para se estabilizar no seu estado correto, ou mesmo oscilando uma série de vezes antes de se estabilizar. Gerando por exemplo a queima de um equipamento dentro de um sistema de um computador, isto pode levar a uma corrupção dos dados ou travamento.

Em muitos casos, a metaestabilidade nos flip-flops pode ser evitada garantindo-se que as entradas de dados e controle sejam mantidas constantes para períodos especificados antes e após o pulso de clock, estes períodos são chamados de tempo de setup ( $t_{su}$ ) e tempo de hold ( $t_h$ ) respectivamente. Estes tempos são especificados na documentação (data sheet) do dispositivos, e são tipicamente entre alguns nanosegundos e algumas centenas de picosegundos nos dispositivos modernos.

Infelizmente, não é sempre possível atingir os critérios de setup e hold, pois o flip-flop pode estar conectado a um sinal em tempo real que pode mudar a

qualquer momento, fora do controle do projetista. Neste caso, o melhor que se pode fazer é reduzir a probabilidade de erro a um certo nível, dependendo da fidelidade requerida do circuito. Uma técnica para reduzir a metaestabilidade é conectar-se dois ou mais flip-flops em uma corrente, de modo que a saída de um alimenta a entrada de dados do outro, e todos os dispositivos compartilham um clock comum. Com este método, a probabilidade de um evento metaestável pode ser reduzida a um valor desprezível, mas nunca a zero.

Existem flip-flop com metaestabilidade reduzida, os quais trabalham reduzindo os tempos de setup e hold o máximo possível, porém mesmo estes não podem eliminar o problema completamente. Isto ocorre porque a metaestabilidade é mais que uma consequência do projeto do circuito. Quando as transições no clock e nos dados estão em um intervalo de tempo próximo, o flip-flop é forçado a escolher qual dos eventos ocorrerá primeiro. Entretanto devido às altas velocidades de processamento, existe sempre a possibilidade de que os eventos da entrada estejam tão próximos que ele não possa detectar qual ocorreu primeiro. Desta forma é logicamente impossível construir um flip-flop totalmente livre de metaestabilidade.

Outro valor importante para um flip-flop é o atraso de clock-a-saída (clock-to-output delay, o símbolo comum é  $t_{CO}$ ) ou atraso de propagação ( $t_P$ ), que é o tempo que o flip-flop leva para mudar a sua saída após o sinal de clock. O tempo de uma transição de alto-para-baixo (high-to-low transition,  $t_{PHL}$ ) é algumas vezes diferente do tempo de uma transição de baixo-para-alto (low-to-high transition,  $t_{PLH}$ ).

Quando se conectam flip-flop em uma corrente, é importante se assegurar que o  $t_{CO}$  do primeiro flip-flop é maior que o tempo de hold (hold time,  $t_H$ ) do segundo flip-flop, caso contrário o segundo flip-flop não irá receber os dados confiavelmente. A relação entre  $t_{CO}$  e  $t_H$  é normalmente garantida se ambos os flip-flops são do mesmo tipo.

### **Circuitos integrados de *Flip-Flops***

---

Podem ser encontrados circuitos integrados (CIs) com um ou dois flip-flops na mesma pastilha. Como exemplo, temos o 7473, internamente com dois Flip-Flops J-K Master-Slave, na série 7400 (TTL).

## REFERÊNCIAS

[https://pt.wikipedia.org/wiki/Circuito\\_digital](https://pt.wikipedia.org/wiki/Circuito_digital)>acesso em 24 de outubro de 2019

[https://pt.wikipedia.org/wiki/Conversor\\_anal%C3%B3gico-digital](https://pt.wikipedia.org/wiki/Conversor_anal%C3%B3gico-digital)>acesso em 24 de outubro de 2019

<https://www.embarcados.com.br/conversor-a-d/>>acesso em 24 de outubro de 2019

<http://www.smar.com/brasil/artigo-tecnico/sistemas-de-supervisao-e-aquisicao-de-dados>>acesso em 24 de outubro de 2019

<https://www.embarcados.com.br/portas-logicas/>>acesso em 24 de outubro de 2019

<https://pt.wikipedia.org/wiki/Flip-flop>>acesso em 24 de outubro de 2019