

Guilherme Pina Cardim

*Relatório de Arquitetura de Computadores*

Presidente Prudente - SP, Brasil

30 de junho de 2010

Guilherme Pina Cardim

## *Relatório de Arquitetura de Computadores*

Relatório das aulas ministradas na disciplina  
de Arquitetura de Computadores na Faculdade de Ciências e Tecnologia - FCT|Unesp

Professor: Mauricio Araújo Dias

DEMEC - DEPARTAMENTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO  
FCT - FACULDADE DE CIÊNCIAS E TECNOLOGIA  
UNESP - UNIVERSIDADE ESTADUAL PAULISTA

Presidente Prudente - SP, Brasil

30 de junho de 2010

# *Resumo*

Este trabalho foi desenvolvido a partir das anotações feitas durante as aulas da disciplina de Arquitetura de Computadores, decorridas durante o primeiro semestre letivo do ano de 2010 na Faculdade de Ciências e Tecnologia - FCT|Unesp.

# *Sumário*

**Lista de Figuras**

**Lista de Tabelas**

<b>1</b>	<b>INTRODUÇÃO</b>	p. 9
1.1	Linguagem Binária . . . . .	p. 10
<b>2</b>	<b>Portas lógicas</b>	p. 11
2.1	Porta NOT . . . . .	p. 11
2.2	Porta AND . . . . .	p. 11
2.3	Porta OR . . . . .	p. 12
2.4	Porta XOR . . . . .	p. 12
<b>3</b>	<b>REPRESENTAÇÃO POR MEIO DE BITS</b>	p. 14
<b>4</b>	<b>ENDIANESS (ORDENAÇÃO/DIREÇÃO)</b>	p. 16
4.1	Big-Endian . . . . .	p. 16
4.2	Little-Endian . . . . .	p. 16
<b>5</b>	<b>FLIP-FLOPS</b>	p. 18
<b>6</b>	<b>ULA - UNIDADE LÓGICA ARITMÉTICA</b>	p. 20
<b>7</b>	<b>REGISTRADORES</b>	p. 21
<b>8</b>	<b>BARRAMENTOS</b>	p. 23

<b>9</b>	<b>MEMÓRIA ROM</b>	p. 24
<b>10</b>	<b>OUTROS COMPONENTES DE UMA ARQUITETURA</b>	p. 25
10.1	IC - <i>Instruction Counter</i> . . . . .	p. 25
10.2	Circuito Decodificador . . . . .	p. 25
10.3	Clock . . . . .	p. 26
<b>11</b>	<b>TIPOS DE CIRCUITOS</b>	p. 27
11.1	Circuitos Combinatórios . . . . .	p. 27
11.2	Circuitos Sequenciais . . . . .	p. 27
<b>12</b>	<b>FCTArq1</b>	p. 29
<b>13</b>	<b>DISPLAY DE 7 SEGMENTOS</b>	p. 30
<b>14</b>	<b>DISPOSITIVOS PERIFÉRICOS</b>	p. 33
14.1	Controladoras de Dispositivos Periféricos . . . . .	p. 33
14.1.1	Transmissão Paralela . . . . .	p. 33
14.1.2	Transmissão Serial . . . . .	p. 34
14.2	Scanner . . . . .	p. 35
14.3	Câmera Digital . . . . .	p. 36
14.4	Modem . . . . .	p. 36
14.5	Teclado . . . . .	p. 37
14.6	Mouse . . . . .	p. 38
14.6.1	Mouse Mecânico . . . . .	p. 38
14.6.2	Mouse Óptico . . . . .	p. 39
14.7	Monitor de Vídeo . . . . .	p. 39
14.7.1	Monitor de Tubo . . . . .	p. 39
14.7.2	Monitor de LCD . . . . .	p. 41

14.7.3 Monitor de Plasma . . . . .	p. 42
14.7.4 Monitor de LED . . . . .	p. 42
<b>15 DISPOSITIVOS DE ARMAZENAMENTO</b>	<b>p. 43</b>
15.1 Fita Magnética . . . . .	p. 43
15.2 CD-ROM . . . . .	p. 44
15.3 CD-R . . . . .	p. 44
15.4 CD-RW . . . . .	p. 44
15.5 DVD . . . . .	p. 45
15.6 Blue Ray . . . . .	p. 45
15.7 Discos Rígidos . . . . .	p. 45
15.7.1 Tempo de Acesso . . . . .	p. 46
15.7.1.1 Tempo de busca (SEEK) . . . . .	p. 46
15.7.1.2 Tempo de latência . . . . .	p. 47
15.7.1.3 Tempo de transferência . . . . .	p. 47
<b>16 DMA</b>	<b>p. 48</b>
<b>17 Memórias</b>	<b>p. 49</b>
<b>Referências</b>	<b>p. 51</b>

## *Lista de Figuras*

1	Porta NOT . . . . .	p. 11
2	Porta AND . . . . .	p. 12
3	Porta OR . . . . .	p. 12
4	Porta XOR . . . . .	p. 13
5	Circuito <i>Latch</i> . . . . .	p. 18
6	Circuito Flip Flop . . . . .	p. 18
7	Circuito JK . . . . .	p. 19
8	FCTArq1 . . . . .	p. 29
9	Display de 7 segmentos real e seus elementos de saída. . . . .	p. 30
10	Tabela de Validações . . . . .	p. 30
11	Mapa de Karnaugh do display de sete segmentos . . . . .	p. 31
12	Decodificador do display de sete segmentos . . . . .	p. 32
13	Transmissão Paralela . . . . .	p. 34
14	Transmissão Serial . . . . .	p. 35
15	Funcionamento de um Scanner . . . . .	p. 35
16	Funcionamento de uma Câmera Digital . . . . .	p. 36
17	Tipos de Modulação de Ondas . . . . .	p. 37
18	Matrix de Conexões de um Teclado . . . . .	p. 38
19	Funcionamento do Mouse Mecânico . . . . .	p. 39
20	Funcionamento do Monitor de Tubo . . . . .	p. 40
21	Funcionamento da Varredura . . . . .	p. 41
22	Estrutura de um Monitor LCD . . . . .	p. 42

23	Fita Magnética . . . . .	p. 43
24	Hierarquia de Memória . . . . .	p. 49
25	Tipos de memórias ROM . . . . .	p. 50

## *Lista de Tabelas*

1	Tabela Verdade da porta NOT . . . . .	p. 11
2	Tabela Verdade da porta AND . . . . .	p. 12
3	Tabela Verdade da porta OR . . . . .	p. 12
4	Tabela Verdade da porta XOR . . . . .	p. 13
5	Tabela de representações binárias . . . . .	p. 14

# 1 *INTRODUÇÃO*

O computador foi criado para imitar o ser humano, fazendo com que o computador o ajudasse na solução de alguns problemas ou tarefas.

No homem as informações são captadas através dos órgãos dos sentidos, como olhos e ouvidos, e são salvas na memória. Essas informações são processadas pela parte racional do cérebro que resolve os problemas mandando mensagens para órgãos periféricos, como os membros superiores e inferiores.

Pelo fato de ser desenvolvido para imitar o homem, o computador possui essas etapas de forma semelhante. Pensando nisso, as informações são captadas por periféricos de entrada, como mouse e teclado, depois são armazenadas na memória para então serem processadas pelo processador, o qual enviará a resolução do problema por meio de sinais para os periféricos de saída, como vídeo e impressora.

Ainda no sentido de imitar o ser humano o processador do computador é composto por portas lógicas, enquanto o cérebro humano é composto por neurônios.

As portas lógicas são compostas por transistores que agem como uma torneira. Eles possuem três pinos, onde um é responsável por abrir ou fechar o fluxo entre os outros dois pinos. Esses transistores são gravados em placas de silício. Essas placas estão armazenadas dentro de uma capa protetora, o que chamamos de encapsulamento do chip.

Para gravar os transistores na placa de silício é utilizado o CMOS, um semiconductor de óxido metálico de simetria complementar. Além disso existem dois tipos de transistores que podem ser gravados nas placas de silício, são eles:

- NMOS - Nesse tipo de transistores é necessário aplicar força positiva para abrir o fluxo.
- PMOS - Esses transistores necessitam de força negativa para abrir o fluxo.

Na eletrônica analítica se trabalha com representações de transistores, já na eletrônica

---

digital as representações são feitas por portas lógicas. A eletrônica digital é uma abstração da eletrônica analítica.

## **1.1 Linguagem Binária**

A linguagem binária, é uma abstração da real linguagem da máquina para que o ser humano possa entender com maior facilidade o que está acontecendo no computador, contudo o que realmente acontece é o fato do computador interpretar diferentes voltagens como 2 sinais diferentes, ou seja o 0 e 1.

## 2 *Portas lógicas*

As portas lógicas são compostas por transistores que são unidos de forma a executarem algo pré determinado. Alguns exemplos de portas lógicas estão representados nas próximas seções.

### 2.1 Porta NOT

A porta NOT também é conhecida como porta inversora, isso pelo fato de que essa porta inverte o sinal que passar por ela, ou seja, se na entrada X ela receber valor *um*, produzirá uma saída Y igual a *zero*, e se X for igual a *zero* a saída Y será *um*. Observe a ilustração da porta NOT na Figura 1 e sua tabela verdade na Tabela 1.

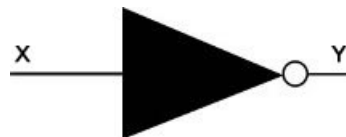


Figura 1: Porta NOT

X	Y
0	1
1	0

Tabela 1: Tabela Verdade da porta NOT

### 2.2 Porta AND

A porta AND trabalha recebendo dois sinais, um em X e outro em Y. Esses sinais são processados e o resultado é transmitido através da saída Z. O qual será de valor *um* se, e somente se, ambos os sinais recebidos em X e Y forem também de sinal *um*, caso contrário o valor de saída será igual a *zero*. Observe a ilustração da porta AND na Figura 2 e sua tabela verdade na Tabela 2.



Figura 2: Porta AND

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 2: Tabela Verdade da porta AND

## 2.3 Porta OR

A porta OR trabalha da mesma forma da porta AND recebendo dois valores através das entradas X e Y. O sinal de saída Z neste casos será de valor *zero* se, e somente se, ambos os sinais recebidos em X e Y também forem de sinal *zero*, caso contrário o valor de saída será igual a *um*. Observe a ilustração da porta OR na Figura 3 e sua tabela verdade na Tabela 3.



Figura 3: Porta OR

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

Tabela 3: Tabela Verdade da porta OR

## 2.4 Porta XOR

A porta XOR também funciona de forma semelhante a porta AND e a porta OR. Essa porta recebe dois sinais pelas entras X e Y, e após processados o resultado é transmitido através da saída Z. Essa saída será equivalente a valor *um* sempre que as entradas X e Y possuírem valores diferentes, já caso das entradas serem iguais, o valor de saída será igual

a zero. Observe a ilustração da porta XOR na Figura 4 e sua tabela verdade na Tabela 4.



Figura 4: Porta XOR

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 4: Tabela Verdade da porta XOR

### 3 *REPRESENTAÇÃO POR MEIO DE BITS*

Dentro do sistema de computação as sequências de *bits* funcionam como um nome de batismo dos objetos e funções computacionais, sendo que uma sequência de *bits* pode representar vários comandos e funções dentro desse sistema.

Pelo fato do código binário possuir apenas dois *bits* diferentes para a criação desses nomes, se torna comum a ocorrência de dois objetos computacionais possuírem a mesma representação binária. A diferenciação de objetos que possuem mesma representação binária depende de como a representação binária é interpretada, tanto pela arquitetura, como pelo aplicativo utilizado.

Sendo RB sigla de representação binária, ND números decimais e OA operadores aritméticos, observe a Tabela 5 que exemplifica possíveis representações binárias.

ND	RB	OA	RB
0	00	+	00
1	01	-	01
2	10	*	10
3	11	/	11

Tabela 5: Tabela de representações binárias

Cada linha de código binário representa uma instrução a ser executada pelo processador. Como exemplo pode-se utilizar a instrução 000110, onde os dois primeiros *bits* representam a operação aritmética escolhida, e os outros 4 *bits* representam dois a dois os números decimais que farão parte da operação determinada. Esse modo de possuir operandos ligados diretamente a instrução é chamado de endereçamento imediato.

Os fabricantes seguem um padrão de produção de códigos binários para que possam se comunicar e haver portabilidade de dados. Isso também é feito para que *softwares* também sejam padronizados e possam funcionar em todas essas arquiteturas que também seguem o padrão. Porém existe a possibilidade de se executar aplicativos que trabalhem de forma

diferente da arquitetura, desde que entre eles haja um interpretador das representações binárias.

Uma das vantagens de se trabalhar com um número maior de *bits* para a representação das palavras de memória, é o fato de que operações mais complexas poderão ser realizadas com menor esforço da arquitetura. Além disso, quanto maior a quantidade de *bits* na representação de palavras de memórias, mais objetos poderão ser representados.

As linguagens de programação foram criadas para facilitar a vida do ser humano. Todas as linguagens de programação devem ser traduzidas para a linguagem da máquina e é o compilador que faz essa tradução entre a linguagem humana e a linguagem binária.

O compilador é responsável não só por traduzir os programas, como também por realizar as análises sintática, semântica e léxica dos códigos desenvolvidos. O *link* editor é responsável por substituir a função da linguagem de programação pelo seu código binário existente dentro das bibliotecas do compilador.

## 4 *ENDIANESS* (*ORDENAÇÃO/DIREÇÃO*)

O conceito de *endianess* envolve a questão da direção com que as informações são interpretadas. Existe duas direções de leitura que são chamadas de *Big-endian* e *Little-endian*. Não há vantagens de uma forma para a outra, o que muda entre elas é somente a direção da leitura.

### 4.1 Big-Endian

No *Big-endian* a leitura é feita da esquerda para a direita, é desta forma que comumente fazemos leituras no dia a dia.

Exemplos:

- ordenação
- 00,123.4\$R
- *Bit* menos significativo -> 100101110001 <- *Bit* mais significativo

### 4.2 Little-Endian

No *Little-endian* a leitura acontece da direita para a esquerda, é desse modo que comumente fazemos leituras de numerais no dia a dia.

Exemplos:

- oãçanedro
- R\$4.321,00

- *Bit* mais significativo -> 100011101001 <- *Bit* menos significativo

## 5 FLIP-FLOPS

Para iniciar os estudos de *flip flops* vamos estudar um pouco sobre *latches*. *Latch* é o menor circuito digital capaz de armazenar um valor. A informação salva em *latch* só é perdida por substituição ou após acontecer o corte de energia nesse circuito. Um *latch* é a memória de armazenamento temporária mais cara existente, porém é a memória mais rápida. O *latch* possui duas entradas, R e S, quando recebe o sinal *um* através da entrada S, o circuito *latch* armazena o valor *um*, já no caso de receber o sinal *um* através da entrada R, o circuito armazena o valor *zero*. O circuito digital de um *latch* está representado na Figura 5.

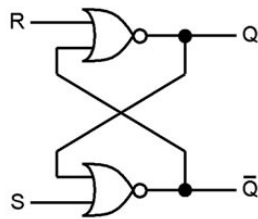


Figura 5: Circuito *Latch*

A partir da construção de um *Latch* podemos evoluí-lo para um *flip-flop*. Nesse circuito conseguimos encontrar mais uma entrada chamada *trigger*, que funciona como um semáforo para o armazenamento. Caso o valor de entrada do *trigger* for igual a *um* o sinal para armazenamento está "verde" e um valor poderá ser armazenado pelo circuito. Já no caso da entrada *trigger* for igual a *zero*, nenhum valor poderá ser armazenado dentro do circuito. Observe o circuito *flip-flop* apresentado pela Figura 6.

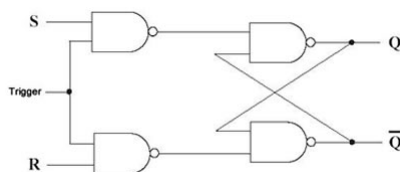


Figura 6: Circuito Flip Flop

Ao abrir o *trigger*, ou seja, atribuir o valor *um*, leva um determinado tempo para o sinal de armazenamento chegar ao circuito. Esse tempo é chamado de atraso de propagação. Esse atraso é o tempo que o *bit* leva para percorrer o circuito digital e influencia diretamente na velocidade da arquitetura.

Pode-se melhorar ainda mais o *flip flop* apresentado de tal forma que este obtenha duas realimentações fazendo com que o circuito seja mais estável. Esse circuito é chamado de *flip flop JK* e está ilustrado pela Figura 7.

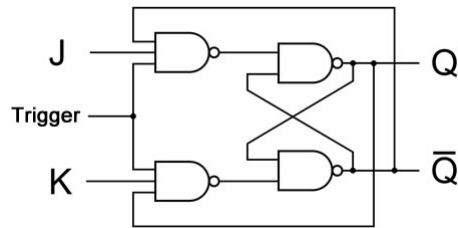


Figura 7: Circuito JK

No *flip flop JK*, para que ele funcione corretamente como um registrador suas entradas J e K devem ser inversas.

## ***6 ULA - UNIDADE LÓGICA ARITMÉTICA***

A ULA, ou do inglês ALU (*Aritmethic Logical Unit*), é a unidade responsável pela realização dos cálculos aritméticos e lógicos da arquitetura.

## 7 REGISTRADORES

Um registrador é um tipo de memória composta por *flip flops*, e estes, por sua vez, são compostos por portas lógicas que são conjuntos de transistores. Sabe-se que cada *flip flop* tem a capacidade de armazenar somente um *bit*, sendo assim, se houver a necessidade de um armazenamento maior pode-se fazer a união de vários *flips flops*.

Nesse trabalho será utilizado o *flip flop JK* para demonstrar o funcionamento de um registrador. Por padrão, o *trigger* do JK deve receber valor *um* para permitir uma gravação, e valor *zero* para que as entradas J e K não alterem o conteúdo interno do flip flop.

A entrada *Clear* é responsável por zerar o *flip flop* independentemente do valor do *trigger*, enquanto que a entrada *preset* é responsável por armazenar o valor *um* no *flip flop* independentemente da entrada *trigger*. Como exemplo de função *clear* pode-se pensar no *reset* de um computador, já um exemplo da função *preset* pode-se imaginar a ação de inicializar um contador decrescente para deixar o valor dos registradores igual a *um*.

A seguir temos a definição de alguns tipos específicos de registradores.

- **Acumulador (ACC):** É um registrador responsável por armazenar o valor proveniente da ULA.
- **MDR (*Memory Data Registrar*):** É um registrador de dados da memória. É o primeiro registrador onde as informações provenientes da memória, seja memória RAM ou ROM, são armazenadas.
- **MAR (*Memory Adress Registrar*):** É um registrador de endereços de memória.
- **MIR (*Micro Instruction Registrar*):** É um registrador de micro instrução, ou sinais de controle.
- **IR (*Instruction Registrar*):** É um registrador de instruções responsáveis por informar como a arquitetura funcionará naquele momento.

*Triggers* estão presentes em todos os registradores e servem como semáforo para permitir o armazenamento de informações. Se existirem *triggers* com informações independentes, eles poderão ser abertos em paralelo o que ajuda na fluxo correto das instruções entre as entidades da arquitetura, pois a abertura dos *triggers* deve ser feita no tempo adequado para cada registrador.

Se o *trigger* de um registrador ficar aberto por muito tempo, pode acontecer de um valor não desejado seja armazenado no lugar de outro. Esse semáforo garante o registro apenas das informações desejadas.

## 8 *BARRAMENTOS*

Barramentos são conjuntos de fios que tem como objetivo a transmissão de dados específicos. Normalmente em uma arquitetura é possível encontrar três barramentos com funcionalidades diferentes, são eles:

- **Barramento de Endereço:** Responsável por transmitir os endereços de memória entre o MAR e a memória onde estão armazenados os códigos a serem executados pelo processador.
- **Barramento de Controle:** Responsável por transmitir os sinais de controle do MIR para os *triggers* presentes na arquitetura.
- **Barramento de Dados:** Responsável por transmitir os dados presentes na memória para o MDR.

## 9 MEMÓRIA ROM

A memória ROM (*Read Only Memory*) é um tipo de memória somente de leitura, pois é gravada fisicamente pela fábrica no ato de sua produção. É responsável por dar início a arquitetura executando suas instruções básicas e logo ao fim delas chamando o Sistema Operacional do computador para entrar em execução. Uma memória é sempre composta por palavras de memórias. Estas funcionam como se fossem gavetas que irão armazenar informações, ou instruções para a arquitetura.

Endereços da memória, ou *adress*, são capazes de acessar diretamente um elemento, ou seja, uma palavra da memória, sendo que cada endereço acessa um único elemento da memória.

Normalmente os elementos da memória acessados pelos endereços são códigos responsáveis pelas instruções a serem executadas pelo processador, sendo que esses códigos possuem origem nos programas escritos pelos desenvolvedores.

# 10 OUTROS COMPONENTES DE UMA ARQUITETURA

## 10.1 IC - *Instruction Counter*

IC (*Instruction Counter*) é um circuito capaz de efetuar uma contagem de valores. Ele possui *flip flops JK* unidos em série que funcionam de forma a efetuar uma contagem, ou seja, esse circuito é utilizado na arquitetura de forma a gerar o endereço da próxima instrução a ser executada do programa que está na memória. Deve receber valor igual a *um* nas entradas J e K de todos os *flip flops*.

Cada instrução do programa é endereçável, sendo que cada código gerado pelo IC e serve como endereço para busca na memória da próxima instrução a ser executada pela arquitetura.

## 10.2 Circuito Decodificador

O circuito decodificador possui 2 valores de entradas e 4 valores de saída. Ele recebe esse nome por decodificar, ou traduzir, o código recebido pelos valores de entrada em instruções, onde somente uma de suas 4 saídas possuirão valor igual a *um*, e a saída que transmitirá esse valor irá depender dos valores de entrada do circuito.

A CU (*Control Unit*), ou unidade de controle, é composta por um circuito decodificador, e tem por responsabilidade controlar a arquitetura por meio da abertura dos *triggers*, ou seja, por sinais de controle. Isso é realizado por meio das ordens recebidas pelos códigos que são enviados pelos desenvolvedores por meio de seus programas.

Os códigos de entrada *in\_0* e *in\_1* da unidade de controle possuem origem nos programas escritos pelos programadores e salvos na memória. O código da operação a ser realizada é os valores que irão entrar nesse circuito.

## 10.3 Clock

O *clock* é uma fina e pequena lâmina de cristal de quartzo. Quando essa lâmina recebe energia, ela vibra de forma a gerar as chamadas ondas quadráticas que variam de 0 a 1. Dessa forma o *clock* se torna responsável por definir quando cada fato deve acontecer dentro da arquitetura. É o *clock* que define quando os *triggers* serão abertos, determinando o fluxo das instruções pela arquitetura. Em geral, quem gera as instruções de controle dos *triggers* é a CU, contudo é o *clock* que diz quando isso irá acontecer, pois é este que controla os tempos adequados dentro da arquitetura através do MIR e do IR.

Se o *clock* for mais rápido do que o atraso de propagação a arquitetura perderá informações e não funcionará corretamente.

# 11 TIPOS DE CIRCUITOS

Existem dois tipos de circuitos, os circuitos combinatórios e os circuitos sequenciais.

## 11.1 Circuitos Combinatórios

Circuitos combinatórios não utilizam registradores, os resultados da saída não depende de valores armazenados na arquitetura. Sua vantagem é ser rápido, contudo possui a desvantagem de se tornar um circuito muito grande para executar cada instrução em parte separada do chip.

## 11.2 Circuitos Sequenciais

Circuitos sequenciais utilizam registradores para armazenar valores e funcionar em *pipeline*. *Pipeline* funciona como uma linha de produção de Fordismo, o qual surgiu quando várias pessoas trabalhavam na montagem de um único carro até que Ford percebeu que seria mais rápido e vantajoso se cada pessoa tivesse sua função específica e trabalhassem ao mesmo momento em carros diferentes seguindo uma ordem.

O mesmo pode ser feito em um arquitetura de computador. Enquanto informações estão sendo armazenadas em uma entidade, outras informações já podem estar sendo armazenadas ou processadas em outras entidades, o que agiliza o processo de execução do programa no total.

Nos circuitos sequenciais existe um ciclo de instrução, o qual é responsável por toda a execução de uma instrução pela arquitetura. O ciclo de instrução por sua vez é dividido em 2 partes:

- **Ciclo de Busca:** Ato de buscar a instrução na memória e trazer para dentro do processador.

- **Ciclo de Execução:** Conjunto de etapas necessárias para executar, ou processar, a instrução dentro do computador.

Encerrando o ciclo de execução, encerra-se também o ciclo de instrução.



## 13 DISPLAY DE 7 SEGMENTOS

O display de 7 segmentos é um dispositivo de saída capaz de mostrar o resultado no formato decimal. Ele não possui lógica interna, sendo assim, a lógica deve ser implementada fora deste. Na Figura 9 pode-se observar um display de 7 segmentos e suas saídas.

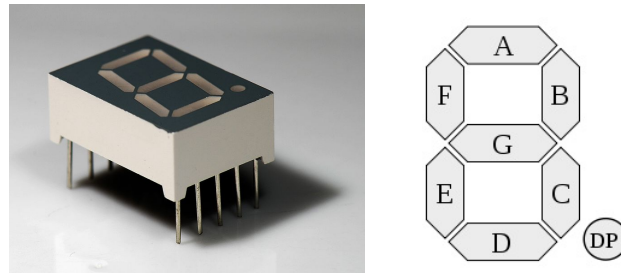


Figura 9: Display de 7 segmentos real e seus elementos de saída.

Cada segmento é um *led* independente que se acende para representar os números decimais, ou o que se desejar.

Observe a Figura 10 que representa a tabela de validação para cada um dos números decimais possíveis de se representar em um display de 7 segmentos.

Entradas				Saídas Segmentos							display
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	0
0	0	1	1	1	1	1	1	0	0	1	0
0	1	0	0	0	1	1	0	0	1	1	0
0	1	0	1	1	0	1	1	0	1	1	0
0	1	1	0	1	0	1	1	1	1	1	0
0	1	1	1	1	1	1	0	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1	0
1	0	0	1	1	1	1	1	0	1	1	0

Figura 10: Tabela de Validações

Através do mapa de Karnaugh, apresentado pela Figura 11, podemos simplificar a construção dos circuitos deste display.

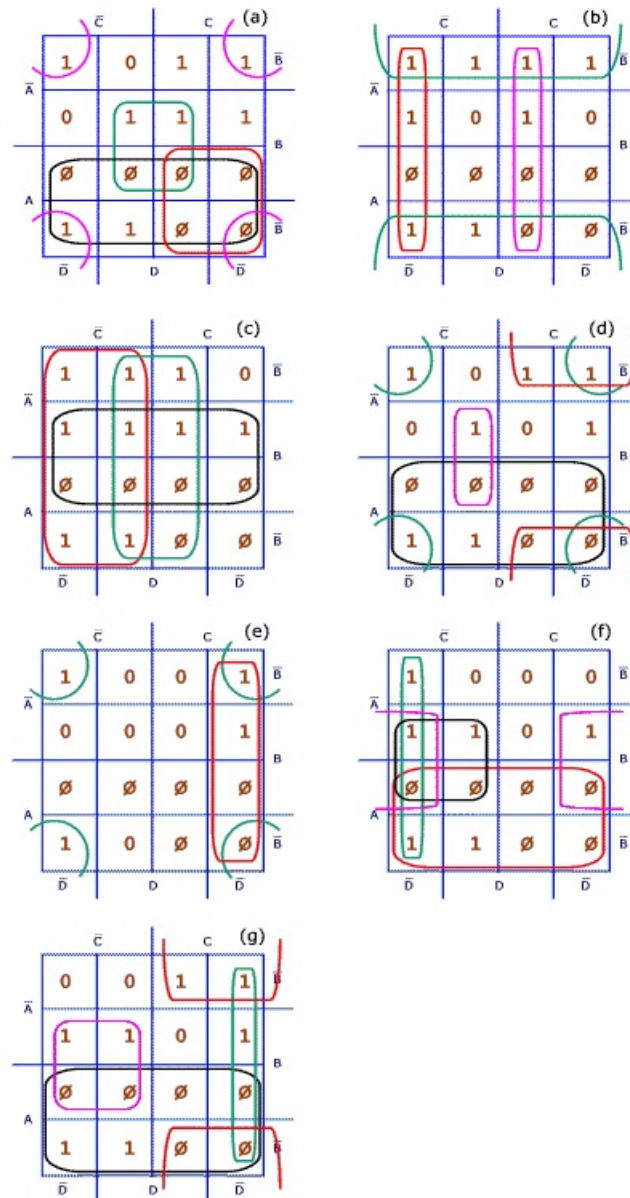


Figura 11: Mapa de Karnaugh do display de sete segmentos

- Saída a:  $A + C + BD + \bar{B}\bar{D}$
- Saída b:  $\bar{B} + CD + \bar{C}\bar{D}$
- Saída c:  $B + \bar{C} + D$
- Saída d:  $A + \bar{B}\bar{D} + C\bar{B} + C\bar{D} + B\bar{C}D$
- Saída e:  $\bar{B}\bar{D} + C\bar{D}$

- Saída f:  $A + \bar{C}\bar{D} + \bar{C}B + B\bar{D}$
- Saída g:  $A + B\bar{C} + \bar{B}C + C\bar{D}$

Com as informações das saídas adquiridas pelo mapa de Karnaugh, é possível criar o circuito decodificador de forma mais simples. Observe o circuito decodificador do display de sete segmentos na Figura 12.

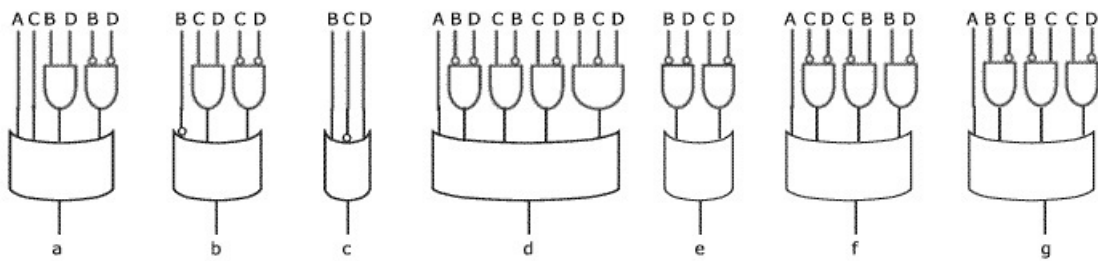


Figura 12: Decodificador do display de sete segmentos

# 14 DISPOSITIVOS PERIFÉRICOS

Dispositivos periféricos são aparelhos ou placas que enviam ou recebem informações do computador. Podemos referir a periférico como qualquer equipamento que seja ligado a CPU (*Central Process Unit*). Os periféricos servem para colocar informações dentro da arquitetura, assim como colocar informações para fora da arquitetura. Em analogia com o ser humano um periférico seria algo como órgãos de sentidos (olhos, ouvidos, língua, boca, nariz, ...). Exemplos de periféricos: impressora, mouse, teclado, monitor, webcam, microfone, etc.

O subsistema de memória e o processador se comunicam com os periféricos por meio de diferentes tipos de transmissões.

## 14.1 Controladoras de Dispositivos Periféricos

A memória e o CPU se comunicam com os dispositivos periféricos através de controladores de dispositivos periféricos. Estes, por sua vez, não estão ligados diretamente ao barramento de sistema (união dos barramentos de controle, dados e endereço), isso se faz por intermédio dos controladores.

Para efetuar essa comunicação entre controlador e periférico existem dois tipos de transmissão de dados entre eles, a transmissão paralela, e a transmissão serial.

### 14.1.1 Transmissão Paralela

A transmissão paralela é a transmissão mais adequada, pois transmite em paralelo a quantidade de *bits* em que a arquitetura trabalha, isto é, se uma arquitetura trabalha com 8 *bits*, seus barramentos transmitem 8 *bits* em paralelo, assim como a transmissão dos dados entre a controladora e o periférico. Observe a imagem do funcionamento de uma transmissão paralela demonstrada pela Figura 13.



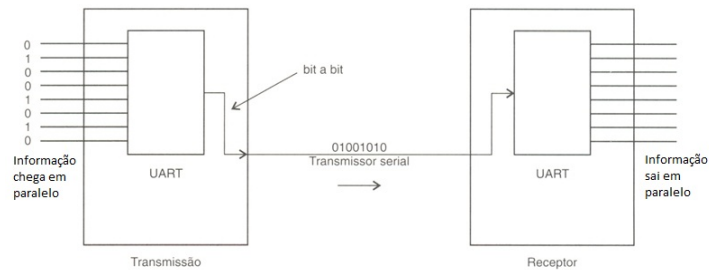


Figura 14: Transmissão Serial

## 14.2 Scanner

O *scanner* é um periférico que serve para fazer cópias digitais de um documento. Você deve colocar o documento no vidro do scanner, onde logo abaixo existe um cilindro que percorre de um lado ao outro do documento com um feixe de luz que irá varrer o documento. Durante essa varredura, a luz é refletida pelo documento e espelhos dentro do scanner direcionam a luz refletida para o CCD (*Charge-Coupled Device*). O CCD é um dispositivo composto por células fotossensíveis e a capacidade do *scanner* esta associada a quantidade dessas células presentes no CCD. Dessa forma o DPI (*Dots Per Inch*) é a quantidade de células fotossensíveis que o CCD vai possuir em uma polegada quadrada. Quanto maior o número do DPI melhor será a imagem captada pelo scanner. Contudo, antes de chegar ao CCD a luz ainda passa por filtros e por lentes com o intuito de melhorar a imagem a ser captada pelo CCD.

Cada cor possui uma ondulação diferente em sua propagação. É por esse comprimento de onda que uma fotocélula consegue diferenciar as cores. Observe a imagem apresentada pela Figura 15 que representa o funcionamento de um scanner.

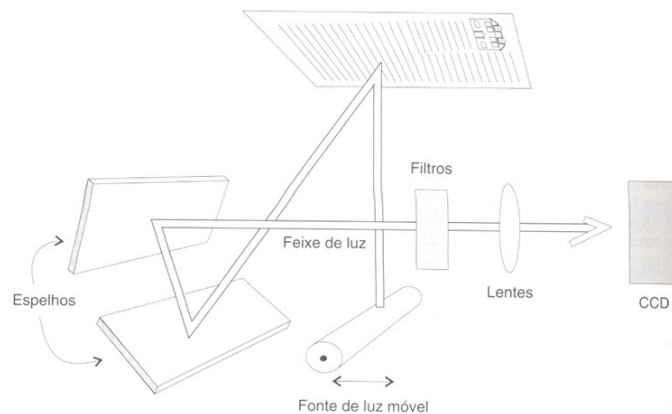


Figura 15: Funcionamento de um Scanner

## 14.3 Câmera Digital

Dentro de uma câmera a parte mais interessante é a matriz de CCD, mais uma vez é assim que a câmera captura as imagens. Contudo uma câmera faz a captura de uma imagem de forma invertida, como no olho humano. Cada pixel é composto por CCD's de cores vermelha, azul, e verde. Pois estas cores são as cores primárias de luz, sendo possível o armazenamento de qualquer cor. Pode-se visualizar o funcionamento de uma câmera digital através da imagem apresentada pela Figura 16.

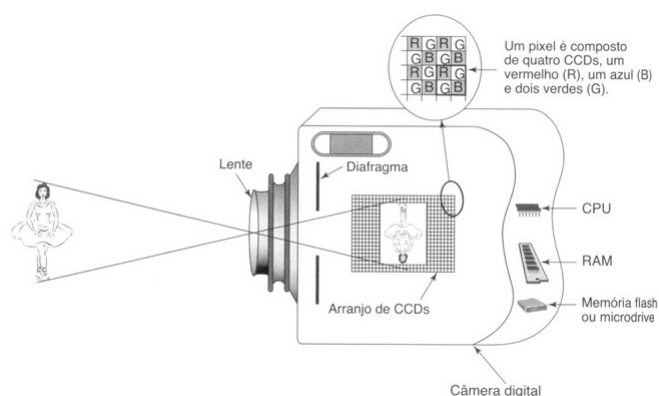


Figura 16: Funcionamento de uma Câmera Digital

## 14.4 Modem

As linhas telefônicas foram criadas para transmitir voz que se propaga por meio de ondas senoidais. No modem, essa transmissão de voz é feita de forma analógica. No caso do computador, a transmissão deve ser feita por ondas quadráticas, necessitando dessa forma de uma tradução para serem transmitidas pelo sistema de telefonia instalado em todo o mundo. Sendo assim o modem surgiu para traduzir a linguagem do computador para a linguagem humana para ser transmitida através das linhas telefônicas. Existem três formas de tradução das ondas quadráticas para ondas senoidais e vice e versa, são elas:

- **Modulação por amplitude:** se em um espaço de tempo não houver ondas (baixa amplitude) o modem traduz essa onda como valor 0, se houver ondas (alta amplitude) o modem traduz o valor como 1.
- **Modulação por frequência:** Em um determinado espaço de tempo podemos ter um número menor ou maior de ondas. O Modem traduzirá as baixas frequências

como 0 e as altas frequências como 1.

- **Modulação por fase:** A onda se quebra quando se quer mudar de 0 para 1, ou vice e versa. A onda para suas propagação e começa outra propagação diferente mostrando a mudança de estado do *bit*.

Observe os tipos de modulação das ondas por meio da imagem apresentada pela Figura 17.

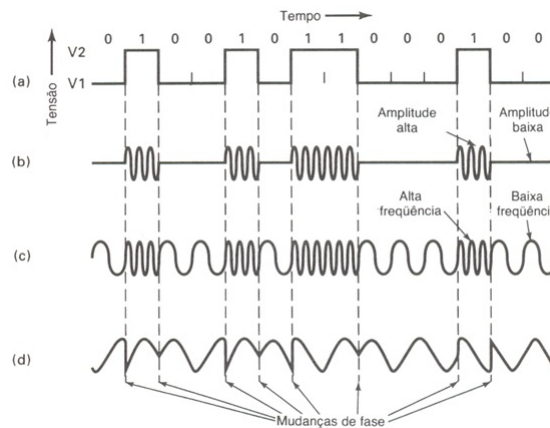


Figura 17: Tipos de Modulação de Ondas

## 14.5 Teclado

Dentro de um teclado existe uma matriz de conexões, contudo essa matriz não está interligada, ou seja, as linhas não estão em contato com as colunas. A única maneira de se fazer conexão entre uma linha vertical e uma horizontal da matriz é por meio das linhas diagonais quebradas que entram em contato ao apertar determinada tecla do teclado.

Cada corte nas pequenas linhas diagonais equivale a uma tecla do teclado. Ao pressionar uma tecla você estará fechando o circuito e a eletricidade passará de uma linha para outra. O teclado gera um código para cada tecla pressionada que será transmitido ao computador e poderá ser utilizado pelos softwares, inclusive o SO presente no computador. Contudo esse código não é o código ASCII (*American Standard Code for Information Interchange*), ele é um código gerado pelo teclado que deverá ser interpretado e traduzido para o código ASCII. Esse código fica salvo na memória ROM e é acessado pelos *softwares*, pois estes utilizam o código da tabela ASCII traduzindo então o código gerado pelo teclado para o código desejado. Observe a imagem da matriz de conexões do teclado apresentada pela Figura 18.

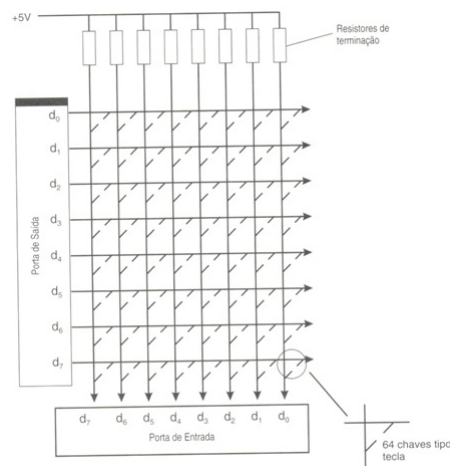


Figura 18: Matrix de Conexões de um Teclado

## 14.6 Mouse

O objetivo do mouse é movimentar o cursor na tela do computador. O princípio básico se baseia no fato de que a tela do computador pode ser representado por um plano cartesiano, onde cada posição da tela pode ser representada por um ponto  $(x,y)$ .

### 14.6.1 Mouse Mecânico

A peça fundamental dentro do mouse mecânico é a existência de um disco perfurado com um eixo no centro, pelo qual o disco gira. De um dos lados do disco existe uma fonte de luz enquanto do outro lado existe uma única célula fotossensível. Quando a luz passa por um dos buracos do disco a fotocélula registra a movimentação. A quantidade de vezes que a luz pisca indica a quantidade pontos na tela que mouse deve percorrer.

Esse dispositivo aparece duplicado dentro do mouse, para que se possa detectar tanto movimentos horizontais como movimentos verticais.

Resumidamente o piscar da luz na fotocélula indica o movimento a ser feito pelo ponteiro do mouse. Esse sistema indica a direção do movimento e sua quantidade, contudo não indica o sentido do movimento. Esse sentido de movimento é indicado mecanicamente pelo sentido de rotação do disco perfurado. Observe o mecanismo de funcionamento de um mouse mecânico por meio da imagem apresentada pela Figura 19.

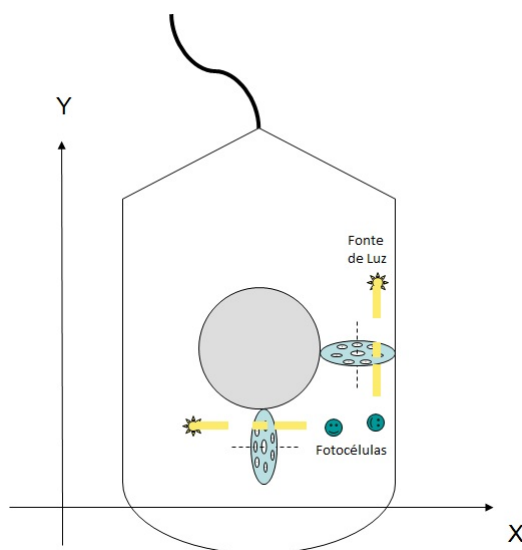


Figura 19: Funcionamento do Mouse Mecânico

### 14.6.2 Mouse Óptico

O mouse óptico emite uma luz (*lazer*) que será refletida pela base onde o mouse está apoiado, e uma fotocélula irá captar a reflexão dessa luz na base conseguindo diferenciar irregularidades na base para identificar a movimentação do mouse.

Para fazer essa identificação do movimento do mouse, o sensor funciona como uma câmera rústica pela qual a imagem da superfície iluminada é capturada. Ao deslocar o mouse, o sensor capta várias imagens, ou sinais. Como a cada nova imagem, parte da imagem anterior é capturada novamente. Dessa forma, o mouse faz uma análise entre as imagens identificando a direção e o sentido que o mouse está percorrendo.

## 14.7 Monitor de Vídeo

Um monitor de vídeo é um periférico que tem por função transmitir informações ao usuário por meio de imagens. Os monitores são classificados de acordo com a tecnologia de amostragem de imagens utilizada, os mais comuns são os monitores de tubo, LCD, Plasma e LED.

### 14.7.1 Monitor de Tubo

Monitores de Tubo recebem esse nome por utilizarem um tubo (válvula) de raio catódico. No começo do tubo existe um canhão cuja função é atirar elétrons contra a tela

do monitor. No final do tubo existe uma película de fósforo, onde os elétrons são atirados, antes do vidro de proteção pelo qual o usuário verá as imagens. Essa película é feita de fósforo pois este material brilha quando recebe eletricidade.

A ideia é direcionar cada elétron atirado pelo canhão para a posição desejada na película. Mas como direcionar os elétrons, se o canhão é fixo e atira muitos elétrons por segundos?

Os elétrons são direcionados através de um sistema existente entre o canhão e a película de fósforo. Esse mecanismo é composto pelo sistema de foco que acelera o elétron, pois quanto maior a pancada deste na película, maior será a iluminação, logo depois existe uma deflexão vertical, onde duas chapas direcionam o elétron verticalmente e depois a deflexão horizontal composto novamente por duas chapas funcionando como imã direcionando o elétron horizontalmente.

Outro ponto importante, é que além da película e do vidro, existe uma grade metálica cuja função é recolher os elétrons atirados.

Cada película de fósforo possui somente uma cor para reflexão. Sendo assim, antigamente havia a existência de muitos monitores onde tudo era visualizado em uma única cor, normalmente a cor verde. Para mostrar imagens coloridas foram criados monitores com películas de fósforos com os chamados *pixels*, esses, por sua vez, são subdivididos em três pequenos pontos das cores vermelha, verde e azul. Dessa forma o mecanismo de direcionamento dos elétrons direciona um elétron para cada cor desse *pixel*, sendo que cada parte terá seu impacto diferente, produzindo a cor desejada por meio da mistura dos três brilhos existentes em cada *pixel*. Observe a imagem apresenta pela Figura 20 que ilustra o funcionamento de um monitor de tubo.

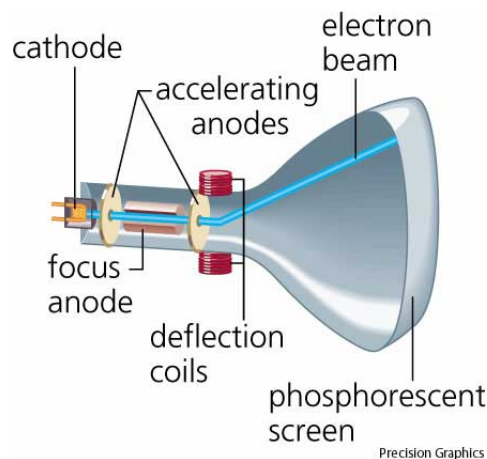


Figura 20: Funcionamento do Monitor de Tubo

O monitor transmite os elétrons para a tela nas linhas ímpares e pares intercaladamente em cada varredura da tela. Uma varredura da tela feita pelo monitor corresponde ao ato dos elétrons serem transmitidos a tela de forma a começar do canto superior esquerdo e fazer uma movimentação horizontal até o canto superior direito, de onde ele volta para a lateral esquerda, porém sem transmitir elétrons, de forma diagonal para atingir uma linha horizontal abaixo da ultima linha desenhada. O sistema faz essa movimentação para transmitir os elétrons até chegar ao canto inferior esquerdo, de onde ele voltará para o canto superior esquerdo de onde começará uma nova varredura de tela.

Este deslocamento da direção dos elétrons é muito rápida, fazendo com que o ser humano não perceba essa modificação, induzindo-o a ver a tela como um todo. Observe a imagem apresenta pela Figura 21 que mostra o funcionamento da varredura em um monitor.

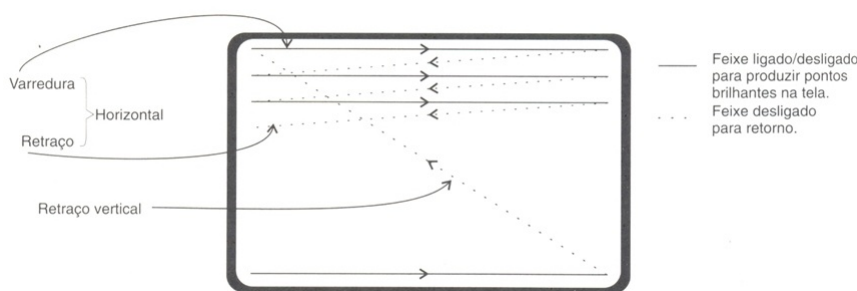


Figura 21: Funcionamento da Varredura

## 14.7.2 Monitor de LCD

Do inglês, *Liquid Crystal Display*, o monitor de cristal líquido possui dois vidros, pelo qual existe um material viscoso chamado cristal líquido. Na verdade, os vidros funcionam como um aquário para esse material com propriedades de um cristal em relação a luz.

Atrás do aquário existe pontos onde é possível transmitir pulsos elétricos, o que faz com que as propriedades físicas do cristal sejam alteradas, possibilitando a passagem ou não da luz.

No LCD simples, a luz ambiente entra pelo aquário e bate ao fundo no último vidro. Quando está desligado nada acontece e essa luz ambiente é refletida. Quando um pulso elétrico é transmitido em determinada posição as propriedades físicas do cristal são alteradas, fazendo com que o cristal retenha essa luz.

A ideia do LCD Colorido é igual a do LCD simples, mas tem alguns detalhes especiais.

Ao invés de trabalhar somente com luz externa, o monitor tem uma fonte de luz branca interna que poderá ou não chegar aos olhos de quem está trabalhando com o monitor. No ponto desejado é jogado um pulso de eletricidade de forma a permitir a passagem de apenas uma frequência de luz. Dessa forma, ao regular diferentes voltagens em cada pixel do monitor, fica determinada a cor formada naquela posição. A estrutura do monitor de LCD está representado pela Figura 22.

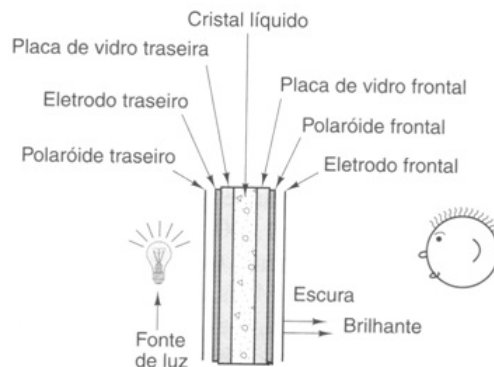


Figura 22: Estrutura de um Monitor LCD

### 14.7.3 Monitor de Plasma

Segue o mesmo princípio do monitor de LCD, porém ao invés de possuir o aquário com o cristal líquido, o aquário contém um "gás plasma", onde os pontos recebem impulsos elétricos fazendo o gás emitir luz formando os *pixels* da imagem. Normalmente, o gás utilizado nesse equipamento é o gás NEON.

### 14.7.4 Monitor de LED

Do inglês, *Light Emitting Diode*, os monitores de LED, funcionam de forma idêntica aos monitores de LCD. A única diferença existente entre eles, é o fato dos monitores de LED possuírem uma matriz de pequenos *leds* emitindo luz branca, ao invés de uma única fonte de luz branca como é feito no monitor LCD.

# 15 DISPOSITIVOS DE ARMAZENAMENTO

## 15.1 Fita Magnética

A Fita Magnética é uma memória que serve tanto para entrada como para a saída de dados no computador. Por esse motivo, a fita magnética também é considerada um periférico.

A fita funciona como uma matriz, sendo que cada casa dessa matriz armazena um *bit*. Cada linha da matriz é chamada de trilha, sendo que cada fita possui nove trilhas. Cada coluna da matriz, atingindo todas as trilhas da fita, é responsável por armazenar um *byte*, ou seja, oito *bits*. Dessa forma utilizaremos as 8 primeiras trilhas da fita para o armazenamento do *byte*, e a última trilha serve para armazenar algum valor de controle necessário.

Cada registro físico é separado por um GAP (espaço) do próximo registro. A representação de um segmento de uma fita magnética está apresentado pela Figura 23.

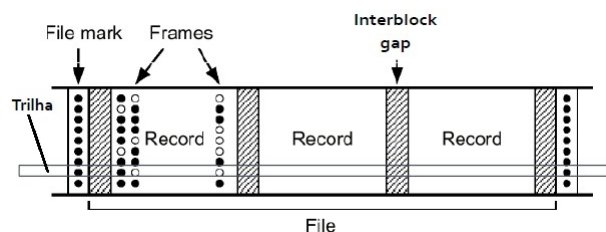


Figura 23: Fita Magnética

As fitas magnéticas possibilitam o armazenamento de um grande número de informações, contudo é considerada muito lenta por ser um tipo de memória sequencial. A cabeça de leitura da fita deve ler todos os frames existentes até o frame desejado.

A fita é feita de um material, cuja a superfície é capaz de conter elétrons, os quais podem modificar seus estados de alguma forma. A cabeça de gravação transmite por

indução uma certa energia para aquele ponto magnético alterando o estado do elétron.

## 15.2 CD-ROM

O CD-ROM (Compact Disc - Read Only Memory) grava as informações de forma espiral, ou seja, a cabeça de leitura e gravação começa no centro do disco e vai fazendo a espiral até as bordas do disco. Denominamos de *pits* as covas (0), e *lands* são as montanhas (1). Esses *bits* são reconhecidos por tempo.

O CD é normalmente feito por um policarbonato revestido por uma camada de alumínio. A cabeça de leitura projeta sobre essa superfície um feixe de infra-vermelho que bate na superfície e volta para o receptor que reconhece quando o *bit* é um *pit* ou um *land*. No CD-ROM a informação é gravada fisicamente, a superfície lisa representa o *land* e juntamente você terá os *pits* fisicamente (pequenos buracos). Esses pits físicos são criados em fábrica e um usuário comum não tem mais como gravar ou adicionar informações nesse tipo de CD.

## 15.3 CD-R

No CD-R (*Compact Disc - Recordable*), além da superfície refletora e do policarbonato, você tem no meio deles uma camada de corante. Quando se quer gravar um *bit* de valor *um*, a cabeça de de gravação emite um laser exatamente no ponto onde deve ser gravado fazendo alterar as propriedades desse corante de modo a abrir um buraco nesse corante. Dessa forma a leitura vai conseguir verificar onde a luz é refletida por passar por esse buraco criado, e quando a luz não é refletida por não ter buracos no corante.

## 15.4 CD-RW

O CD-RW (*Compact Disc - ReWritable*), possui uma liga metálica que quando recebe o feixe de laser altera sua propriedade física. Na verdade essa liga faz o papel ou não da reflexão. Normalmente o papel da liga metálica é refletir, porém quando o laser bate em algum ponto, ele perde a capacidade de refletir a luz da cabeça de leitura. O interessante é que a liga se altera de acordo com a intensidade do laser. Um ponto que já recebeu o laser e teve as propriedades alteradas para não refletir, pode receber o laser novamente com outra intensidade de forma a voltar a sua propriedade física original refletindo a luz

novamente.

## 15.5 DVD

Do inglês, *Digital Video Disc*, o DVD funciona da mesma maneira de um CD. A diferença entre eles é o fato das espirais de gravação serem mais densas no DVD, possibilitando um maior armazenamento de dados. Porém para ler e gravar dados no DVD, o feixe de luz a ser usado é o laser vermelho, garantindo uma maior precisão na leitura dos dados.

Na gravação do DVD se utiliza duas camadas, dentre as quais existe uma camada semi-reflectiva. Para ler ambas as informações basta alterar a intensidade do laser, de forma que para ler as informações da camada inferior você necessita de um feixe de laser mais intenso para atravessar a camada semi-reflectiva e de um laser menos intenso para ler acima da camada semi-reflectiva.

## 15.6 Blue Ray

O *Blue Ray* funciona de forma idêntica ao DVD. A diferença entre eles é o laser utilizado para a leitura e gravação de dados. Enquanto no DVD é utilizado o laser vermelho, no Blue Ray utiliza o laser azul, tornando, dessa forma, a espiral de dados ainda mais densa, possibilitando o armazenamento de um número maior de informações.

## 15.7 Discos Rígidos

No HD (*Hard Disk*), as trilhas são feitas de forma circular seguindo o formato do disco. Cada trilha possui seu conjunto de setores, que são a menor unidade de armazenamento de informação em um disco, ou seja, ao gravar uma informação pelo menos um setor deve ser gravado, ou ao efetuar uma leitura, pelo menos um setor deve ser lido.

A leitura e gravação de dados são feitas por meio de blocos de informações. Esse bloco pode conter de 1 a n setores. Do mesmo modo que na fita magnética, tem-se espaçamentos entre os setores chamados de gaps. Além disso, existe um espaçamento entre as trilhas do disco.

Na verdade, o HD possui vários discos unidos por um eixo central que gira todos os discos ao mesmo tempo. Cada disco possui duas faces onde as informações são ar-

mazenadas.

Cada face dos discos possui uma cabeça de leitura e gravação própria, sendo que normalmente todas estão unidas por um único braço. Isso permite que todas as cabeças gravem e leiam informações paralelamente. As informações são divididas entre todos os discos na hora da gravação para aproveitar o posicionamento das cabeças de leitura. Pelo fato da informação ser dividida entre todos os discos, pode-se imaginar as informações em cilindros.

O número de *bits* na trilha mais próxima do centro do disco é idêntico ao número de *bits* na trilha mais perto da borda. Fazendo com que a densidade no centro do disco seja maior do que nas bordas do mesmo. Isso acarreta em perda de espaço de armazenamento, contudo melhora a performance do disco diminuindo o tempo de acesso a informação, uma vez que a cabeça de leitura e gravação pode estar fixada horizontalmente sendo necessário a movimentação apenas para escolher a trilha desejada.

### 15.7.1 Tempo de Acesso

Tempo de acesso é o tempo que o disco demora para devolver a informação para o sistema após este ter pedido essa informação para o disco.

Esse tempo é uma soma simples que envolve três variáveis:

#### 15.7.1.1 Tempo de busca (SEEK)

É o tempo que a unidade de disco leva para colocar a cabeça de leitura até a trilha desejada.

Para calcular o tempo de busca, deve-se possuir dois componentes, o tempo inicial de partida, e o tempo requerido para percorrer as trilhas depois que o braço de leitura está pronto para se movimentar. O tempo de busca pode ser aproximado pela seguinte fórmula linear:

$$T_s = m \times n + s, \text{ onde:}$$

$T_s$  = tempo estimado de busca

$m$  = constante que depende da unidade de disco

$n$  = número de trilhas percorridas

$s$  = tempo de partida

### 15.7.1.2 Tempo de latência

É o tempo de atraso rotacional. Após posicionar o braço na trilha correta, o disco deve girar até que a informação chegue na cabeça de leitura.

Discos giram tipicamente a 3600 rpm, ou sejam, efetuam uma revolução a cada 16.7 ms. Dessa maneira, o atraso rotacional é de 8,3 ms.

### 15.7.1.3 Tempo de transferência

É o tempo que o sinal, ou a informação, leva para ser copiada do disco e chegar ao sistema, ou seja é o tempo que a informação leva para percorrer os fios e chegar na unidade controladora.

O tempo de transferência do disco depende da sua velocidade de rotação, de acordo com a seguinte relação:

$$T = \frac{b}{rN}, \text{ onde}$$

$T$  = tempo de transferência

$b$  = números de *bytes* transferidos

$N$  = número de *bytes* na trilha

$r$  = velocidade de rotação em número de revoluções por segundo

## ***16 DMA***

O dispositivo DMA (*Direct Memory Access*) possui a função de fazer a movimentação entre as informações do disco rígido para a memória principal e vice e versa.

## 17 Memórias

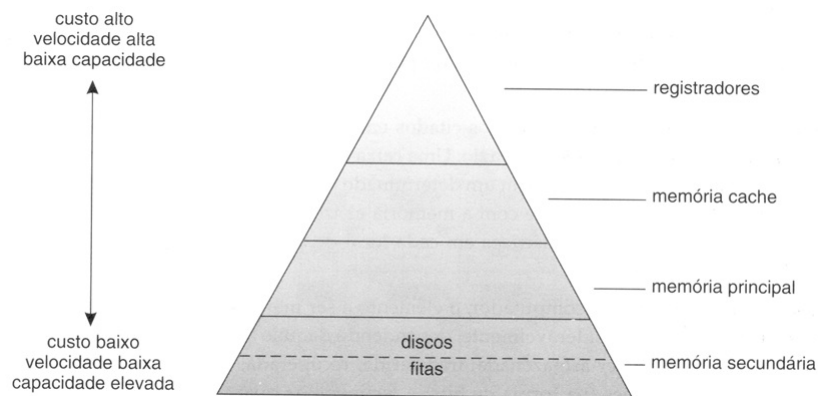


Figura 24: Hierarquia de Memória

Para acessar um endereço de memória, o código é interpretado pelo decodificador de endereço da memória, para então poder acessar a informação desejada. Essa informação fica armazenada nas unidades responsáveis por armazenar os *bits*.

Existem vários circuitos que podem ser utilizados para armazenar os *bits* na memória. Caso essas unidades forem *flip flops*, a memória será denominada RAM estática (SRAM). Já no caso das unidades serem capacitores, a memória será um tipo de memória RAM dinâmica (DRAM).

Capacitor é um elemento da eletrônica que possui a capacidade de armazenar energia, porém funciona como um balde furado, pois perde energia constantemente. Por esse motivo, essa memória deve receber um *refresh* de energia periodicamente para continuar armazenando as informações presentes em seu interior.

Já no caso da unidade de armazenamento for um fusível ou um diodo, a memória será denominada ROM, pois os valores *zeros* são representados por essas unidades queimadas, o que impossibilita uma nova gravação de dados.

Existem outros tipos de memórias, alguns deles são apresentados a seguir:

- EDRAM - Uma DRAM melhorada por possuir um registrador gigante ou seja, existe

uma SRAM dentro da EDRAM.

- CDRAM - Além da memória RAM, ela possui uma memória cache dentro do chip.
- SDRAM - É uma DRAM síncrona. Ela trabalha em sincronismo com o processador. A busca pela informação é feita em tempo pré-determinado. O processador pede a informação dentro de  $x$  milissegundos. Ao invés do processador ficar esperando a resposta da RAM, ele vai fazer outra coisa e volta a executar o programa no tempo  $x$  para receber a informação.

Tipo de memória	Categoria	Mecanismo de apagamento	Mecanismo de escrita	Volatilidade
Memória de acesso aleatório (RAM)	Memória de leitura e de escrita	Eletricamente em nível de bytes	Eletricamente	Volátil
Memória apenas de leitura (ROM)	Memória apenas de leitura	Não é possível	Máscaras	Não-Volátil
ROM programável (PROM)			Eletricamente	
PROM apagável (EPROM)	Luz UV, em nível de pastilha			
Memória flash	Eletricamente, em nível de blocos			
PROM eletricamente apagável (EEPROM)	Memória principalmente de leitura	Eletricamente, em nível de bytes		

Figura 25: Tipos de memórias ROM

## *Referências*

<<http://www.mspc.eng.br/eledig/eldg0610.shtml>>. Acesso em 23/06/2010.

<<http://pt.wikipedia.org/wiki/Perif%C3%A9rico>>. Acesso em 23/06/2010.

<<http://pt.wikipedia.org/wiki/UART>>. Acesso em 23/06/2010.

<[http://pt.wikipedia.org/wiki/Monitor\\_de\\_v%C3%ADdeo](http://pt.wikipedia.org/wiki/Monitor_de_v%C3%ADdeo)>. Acesso em 24/06/2010.