



Instituto Infnet

Arquitetura de Computadores e Sistemas Operacionais

FABIANO GISBERT

Etapa 02

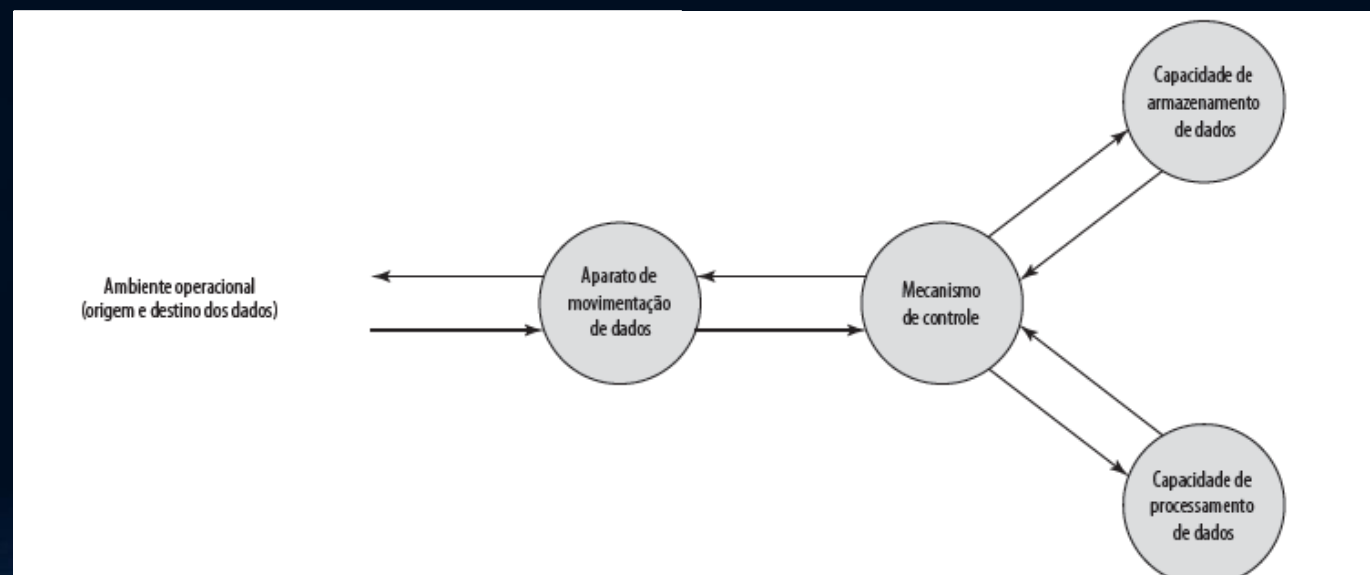
Estrutura do Computador



Instituto Infnet

Tanto a estrutura quanto o funcionamento de um computador são, essencialmente, simples. A figura representa as funções básicas que um computador pode realizar. Em termos gerais, existem apenas quatro:

- Processamento de dados.
- Armazenamento de dados.
- Movimentação de dados.
- Controle.



Estrutura do Computador



Instituto Infnet

O computador, naturalmente, precisa ser capaz de **processar dados**. Os dados podem assumir muitas formas e o intervalo de requisitos de processamento é amplo. Porém, veremos que existem apenas alguns métodos fundamentais ou tipos de processamento de dados.

Também é essencial que um computador **armazene dados**. Mesmo que o computador esteja processando dados dinamicamente (ou seja, os dados entram, são processados e os resultados saem imediatamente), o computador precisa armazenar temporariamente pelo menos as partes dos dados que estão sendo trabalhadas em determinado momento. Assim, existe pelo menos uma função de **armazenamento de dados a curto prazo**.

Igualmente importante, o computador realiza uma função de **armazenamento de dados a longo prazo**. Os arquivos de dados são armazenados no computador para subsequente recuperação e atualização.

Estrutura do Computador



Instituto Infnet

O computador precisa ser capaz de **movimentar dados** entre ele e o mundo exterior. O ambiente operacional do computador consiste em dispositivos que servem como suas *origens* ou *destinos* de dados. Quando os dados são recebidos ou entregues a um dispositivo conectado diretamente ao computador, o processo é conhecido como entrada/saída (E/S), e o dispositivo é referenciado como um periférico.

Quando os dados são movimentados por distâncias maiores, de ou para um dispositivo remoto, o processo é conhecido como **comunicações de dados**.

Finalmente, é preciso haver **controle** dessas três funções, e esse controle é exercido por quem fornece instruções ao computador. Dentro do computador, uma unidade de controle gerencia os recursos do computador e coordena o desempenho de suas partes funcionais em resposta a essas instruções.

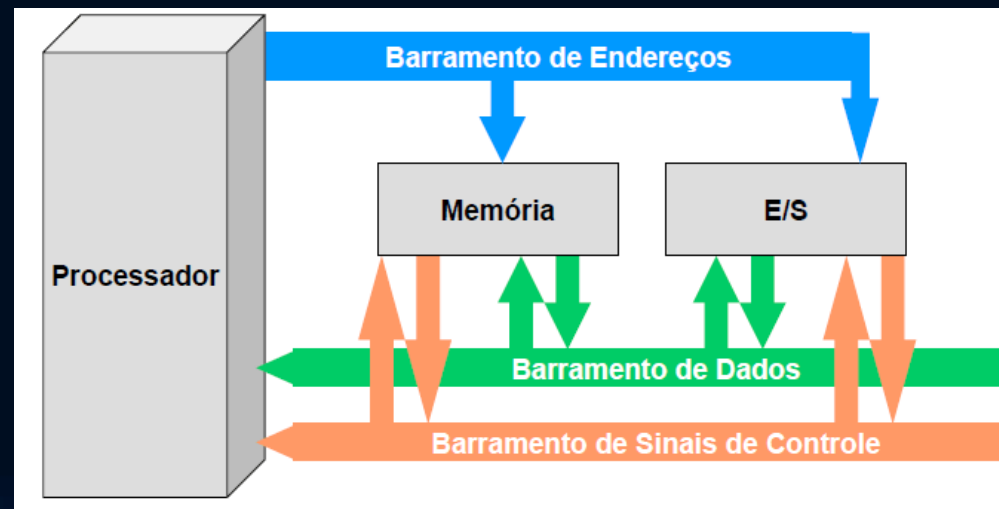
Estrutura do Computador



Instituto Infnet

Existem quatro componentes estruturais principais, na arquitetura atual de computadores:

- **Unidade central de processamento (cpu):** controla a operação do computador e realiza suas funções de processamento de dados; normalmente é chamado apenas de processador.
- **Memória principal:** armazena dados temporários.
- **Dispositivos de E/S:** move dados entre o computador e seu ambiente externo.
- **Interconexão do sistema:** algum mecanismo que oferece comunicação entre CPU, memória principal e E/S. Um exemplo comum de interconexão do sistema é por meio de um barramento do sistema.



UNIDADE CENTRAL DE PROCESSAMENTO

Microeletrônica



Instituto Infnet

Microeletrônica significa literalmente “pequena eletrônica”.

Desde os primórdios da eletrônica digital e da indústria da computação, tem havido uma tendência persistente e consistente em direção à redução no tamanho dos circuitos eletrônicos digitais.

Antes de examinarmos a estrutura de um processador, precisamos entender a natureza da eletrônica digital.

Os elementos básicos de um computador digital, como sabemos, precisam realizar funções de armazenamento, movimentação, processamento e controle.

Somente três tipos fundamentais de componentes são necessários:

Portas, células de memória e condutores.

Microeletrônica

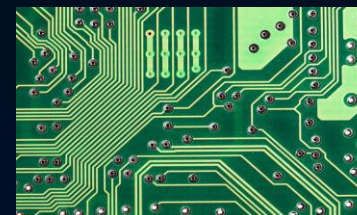
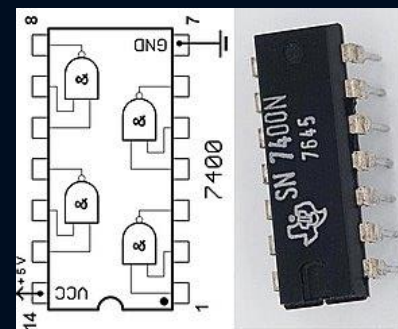


Instituto Infnet

Porta: Uma porta é um dispositivo que implementa uma função booleana ou lógica simples, como IF A AND B ARE TRUE THEN C IS TRUE (porta AND). Esses dispositivos são chamados de portas porque controlam o fluxo de dados de modo semelhante às portas de um canal.

Célula de Memória: é um dispositivo que pode armazenar um bit de dados, ou seja, o dispositivo pode estar em um de dois estados estáveis de cada vez.

Condutores: são os caminhos energizáveis entre os componentes cuja função é movimentar dados da memória para a memória ou da memória para as portas, e destas até a memória.



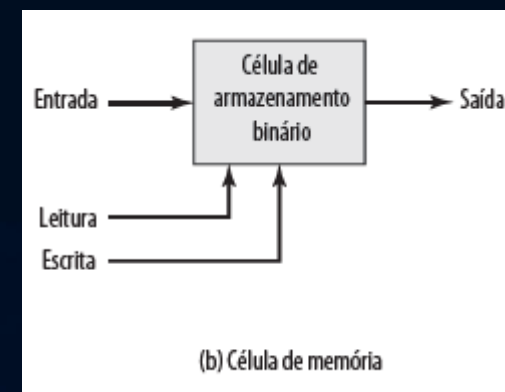
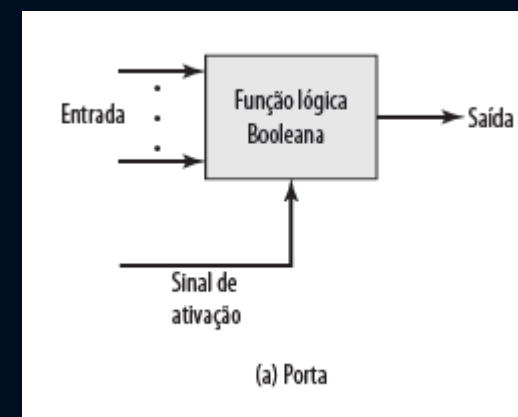
Microeletrônica



Instituto Infnet

Podemos relacionar isso com nossas quatro funções básicas da seguinte forma:

- **Armazenamento de dados:** fornecido por células de memória.
- **processamento de dados:** fornecido por portas.
- **Movimentação de dados:** condutores.
- **Controle:** os condutores entre os componentes podem transportar sinais de controle. Por exemplo, uma porta terá uma ou duas entradas de dados mais uma entrada de sinal de controle que ativa a porta. Quando o sinal de controle é ON, a porta realiza sua função sobre as entradas de dados e produz uma saída de dados. De modo semelhante, a célula de memória armazenará o bit que está em seu fio de entrada quando o sinal de controle ESCRITA for ON, e colocará o bit que está na célula em seu fio de saída quando o sinal de controle LEITURA for ON.



Microeletrônica

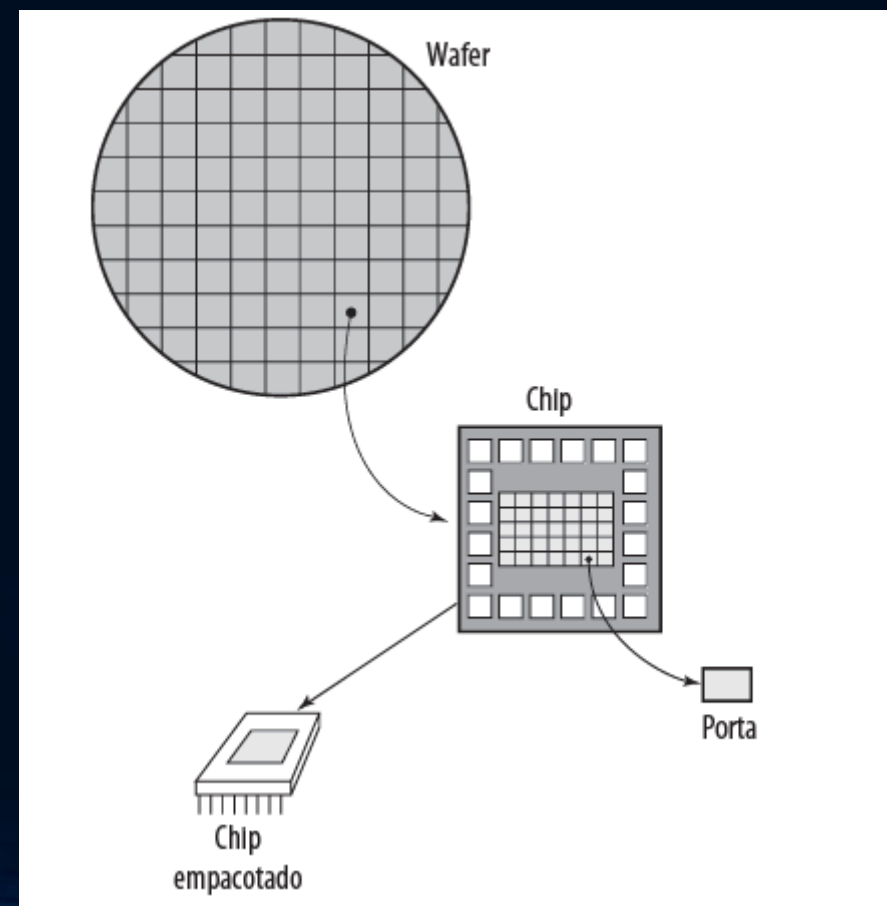


Instituto Infnet

Assim, um computador consiste em portas, células de memória e interconexões entre esses elementos. As portas e células de memória são, por sua vez, construídas de componentes eletrônicos digitais simples.

O circuito integrado explora o fato de que componentes como transistores, resistores e condutores podem ser fabricados a partir de um semicondutor como o silício.

Um wafer fino de silício é dividido em uma matriz de pequenas áreas, cada uma com poucos milímetros quadrados. O padrão de circuito idêntico é fabricado em cada área, e o wafer é dividido em chips. Cada chip consiste em muitas portas e/ou células de memória mais uma série de pontos de conexão de entrada e saída.



Portas Lógicas e Funções Booleanas



Instituto Infnet

Na álgebra de Boole, há somente dois estados (valores ou símbolos) permitidos:

- Estado 0 (zero)
- Estado 1 (um)

Em geral o estado zero representa não, falso, aparelho desligado, **ausência de tensão**, chave elétrica desligada, etc

O estado um representa sim, verdadeiro, aparelho ligado, **presença de tensão**, chave ligada, etc.

Portanto, em qualquer bloco (porta ou função) lógico somente esses dois estados (0 ou 1) são permitidos em suas entradas e saídas

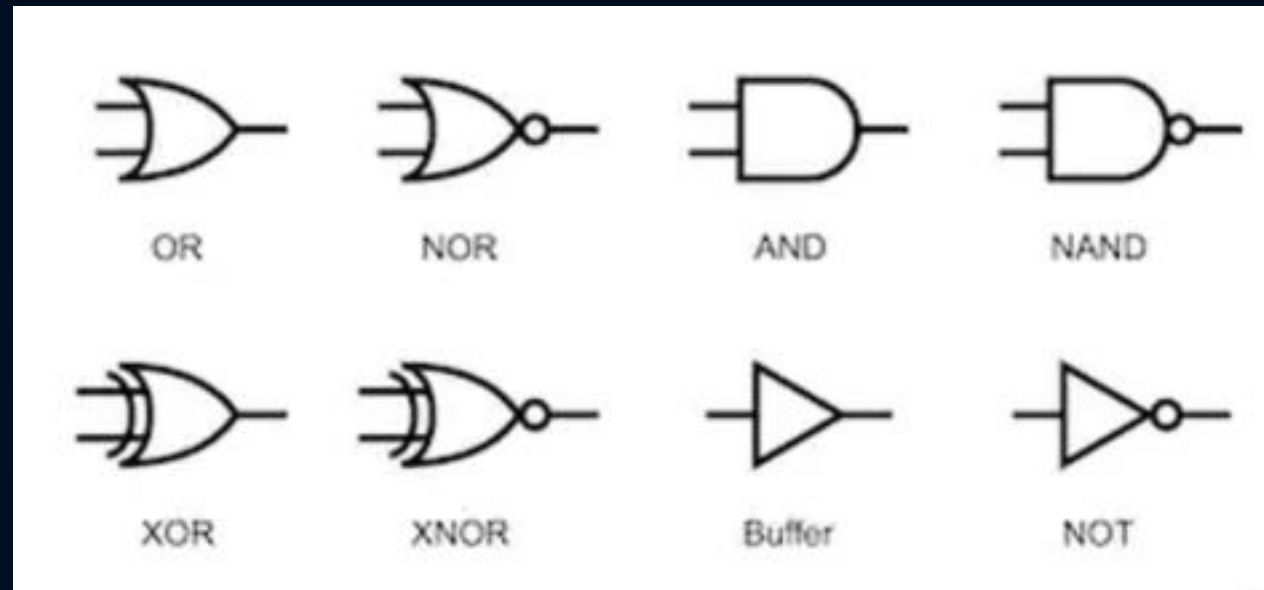
Portas Lógicas e Funções Booleanas



Instituto Infnet

As portas lógicas existentes numa arquitetura computacional, representam as seguintes expressões booleanas:

- E (AND)
- OU (OR)
- NÃO (NOT)
- NÃO E (NAND)
- NÃO OU (NOR)
- OU EXCLUSIVO (XOR)



Portas Lógicas e Funções Booleanas

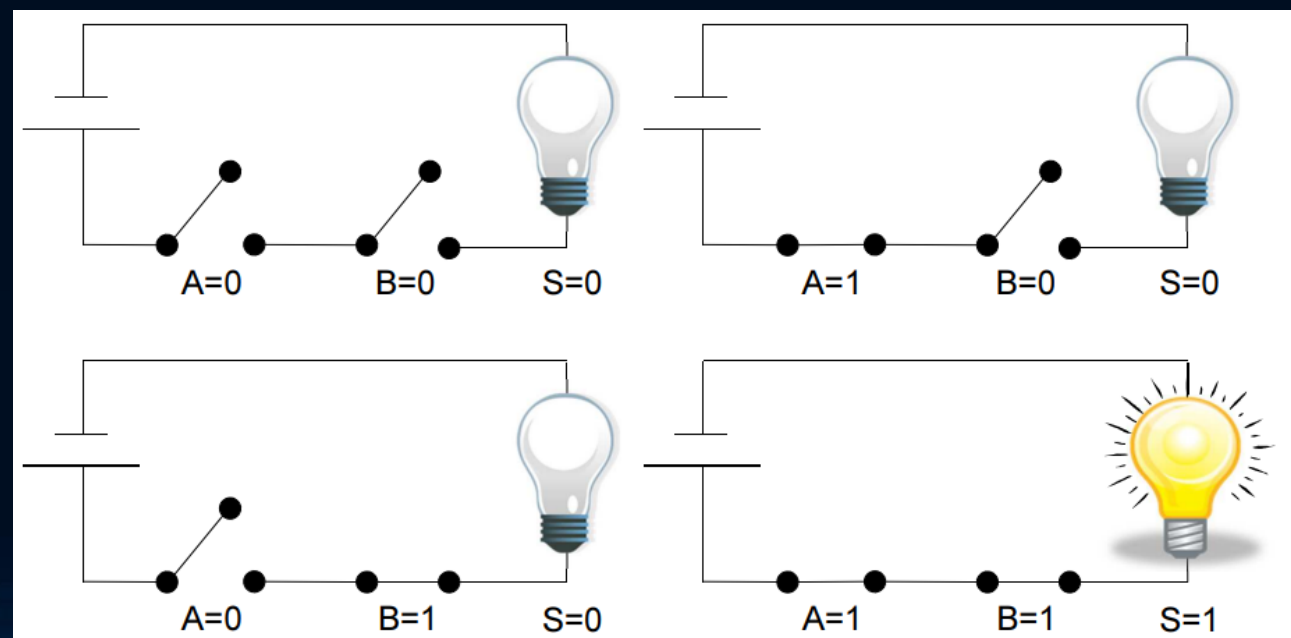


Instituto Infnet

Função E (AND)

Executa a multiplicação (conjunção) booleana de duas ou mais variáveis binárias. Por exemplo, assuma a convenção no circuito:

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



Portas Lógicas e Funções Booleanas

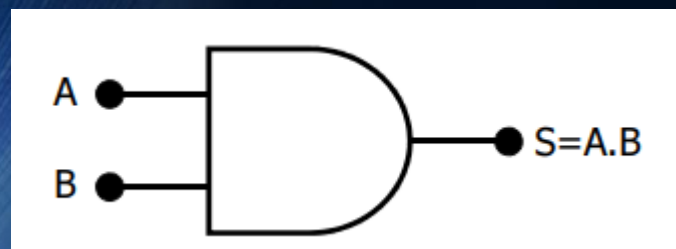


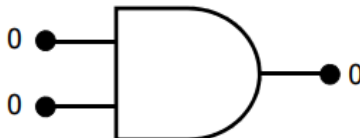
Instituto Infnet

Porta AND

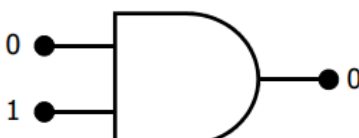
A porta AND é um circuito que executa a função E e executa a tabela verdade da função E.

Portanto, a saída será 1 somente se ambas as entradas forem iguais a 1; nos demais casos, a saída será 0

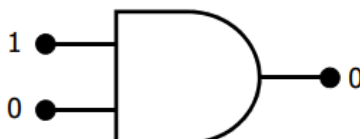




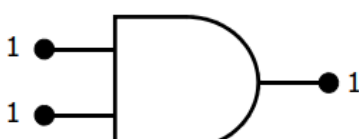
A	B	S=A.B
0	0	0
0	1	0
1	0	0
1	1	1



A	B	S=A.B
0	0	0
0	1	0
1	0	0
1	1	1



A	B	S=A.B
0	0	0
0	1	0
1	0	0
1	1	1



A	B	S=A.B
0	0	0
0	1	0
1	0	0
1	1	1

Portas Lógicas e Funções Booleanas



Instituto Infnet

Função OU (OR)

Executa a soma (disjunção) booleana de duas ou mais variáveis binárias

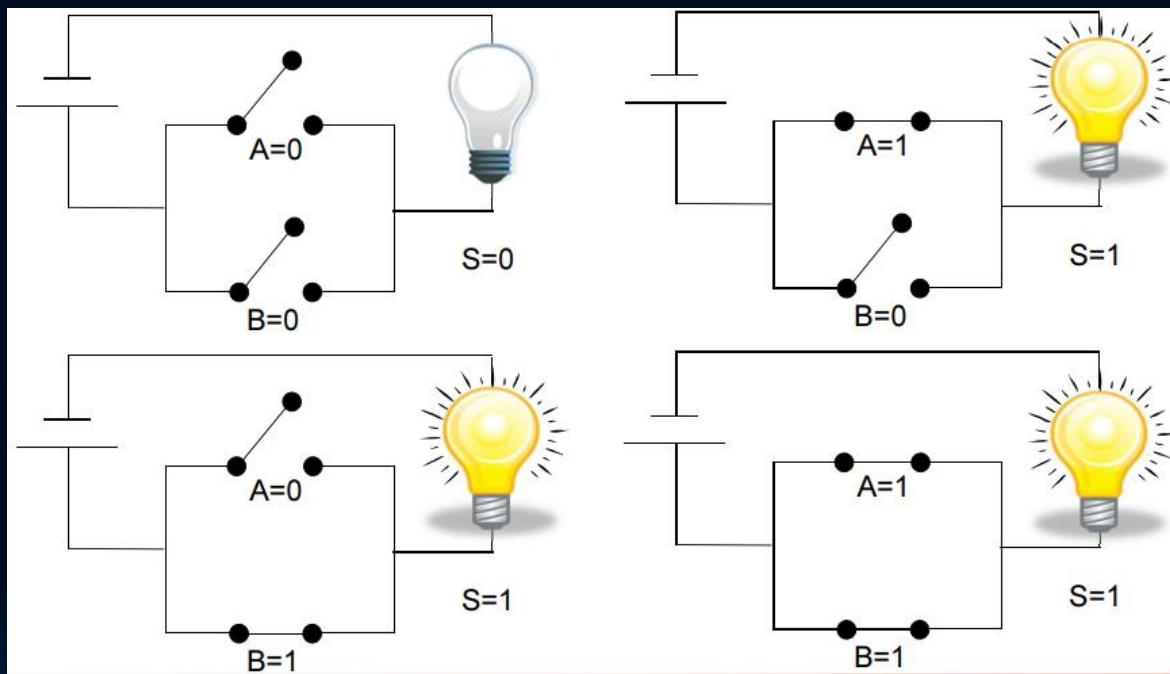
Por exemplo, assuma a convenção no circuito

Chave aberta = 0

Chave fechada = 1

Lâmpada apagada = 0

Lâmpada acesa = 1



Portas Lógicas e Funções Booleanas

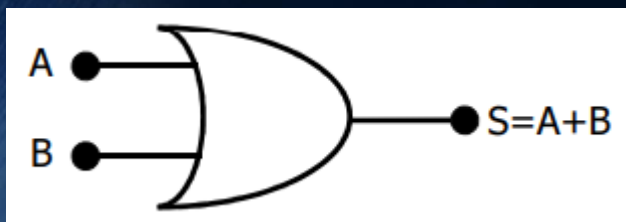


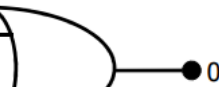
Instituto Infnet

Porta Lógica OR


A porta OU é um circuito que executa a função OU e executa a tabela verdade da função OU.

Portanto, a saída será 0 somente se ambas as entradas forem iguais a 0; nos demais casos, a saída será 1







A	B	S=A+B
0	0	0
0	1	1
1	0	1
1	1	1



A	B	S=A+B
0	0	0
0	1	1
1	0	1
1	1	1



A	B	S=A+B
0	0	0
0	1	1
1	0	1
1	1	1



A	B	S=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Portas Lógicas e Funções Booleanas



Instituto Infnet

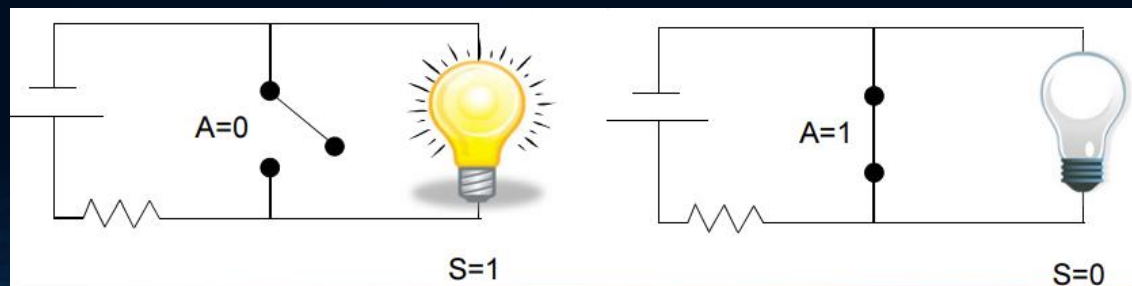
Função NÃO (NOT)

Executa o complemento (negação) de uma variável binária. Se a variável estiver em 0, o resultado da função é 1. Se a variável estiver em 1, o resultado da função é 0. Essa função também é chamada de Inversora.

Usando as mesmas convenções dos circuitos anteriores, tem-se que:

Quando a chave A está aberta ($A=0$), passará corrente pela lâmpada e ela acenderá ($S=1$)

Quando a chave A está fechada ($A=1$), a lâmpada estará em curto-circuito e não passará corrente por ela, ficando apagada ($S=0$)



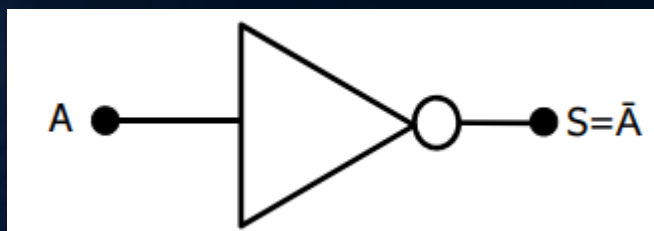
Portas Lógicas e Funções Booleanas

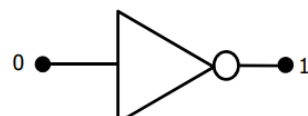


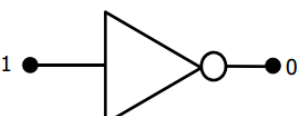
Instituto Infnet

Porta Lógica NOT

A porta lógica NÃO, ou inversor, é o circuito que executa a função NÃO. O inversor executa a tabela verdade da função NÃO. Se a entrada for 0, a saída será 1; se a entrada for 1, a saída será 0



	A	$S = \bar{A}$
	0	1
	1	0

	A	$S = \bar{A}$
	0	1
	1	0

Portas Lógicas e Funções Booleanas



Instituto Infnet

Função NÃO E (NAND)

Composição da função E com a função NÃO, ou seja, a saída da função E é invertida

$$\begin{aligned} S &= \overline{(A.B)} = \overline{A.B} \\ &= (A.B)' \\ &= \neg(A.B) \end{aligned}$$

□ Tabela verdade

A	B	$S = \overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

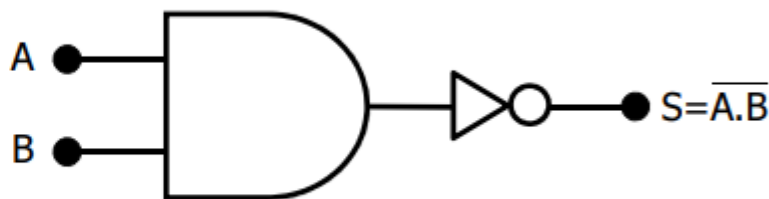
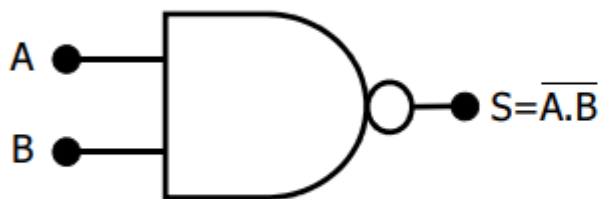
Portas Lógicas e Funções Booleanas



Instituto Infnet

Porta NAND

A porta NAND é o bloco lógico que executa a função NÃO E, ou seja, sua tabela verdade



Portas Lógicas e Funções Booleanas



Instituto Infnet

Função NÃO OU (NOR)

Composição da função OU com a função NÃO, ou seja, a saída da função OU é invertida

$$\begin{aligned} S &= \overline{(A+B)} = \overline{A+B} \\ &= (A+B)' \\ &= \neg(A+B) \end{aligned}$$

□ Tabela verdade

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

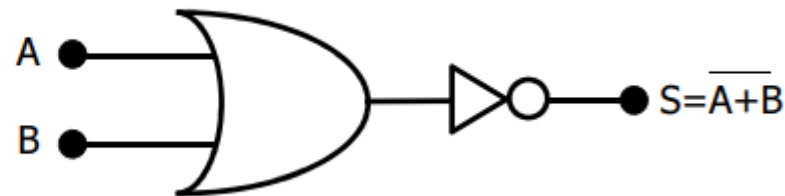
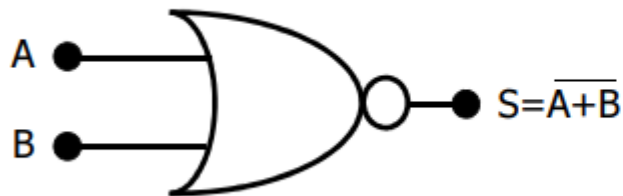
Portas Lógicas e Funções Booleanas



Instituto Infnet

Porta NOR

A porta NOR é o bloco lógico que executa a função NÃO OU, ou seja, sua tabela verdade



Portas Lógicas e Funções Booleanas



Instituto Infnet

Função OU Exclusivo (XOR)

A função OU Exclusivo fornece 1 na saída quando as entradas forem diferentes entre si e caso contrário receberá 0.

$$S = A \oplus B = \bar{A}.B + A.\bar{B}$$

Tabela verdade:

Tabela verdade

A	B	$S=A\oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

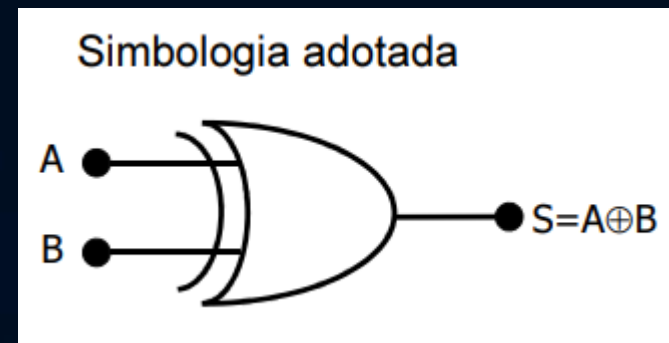
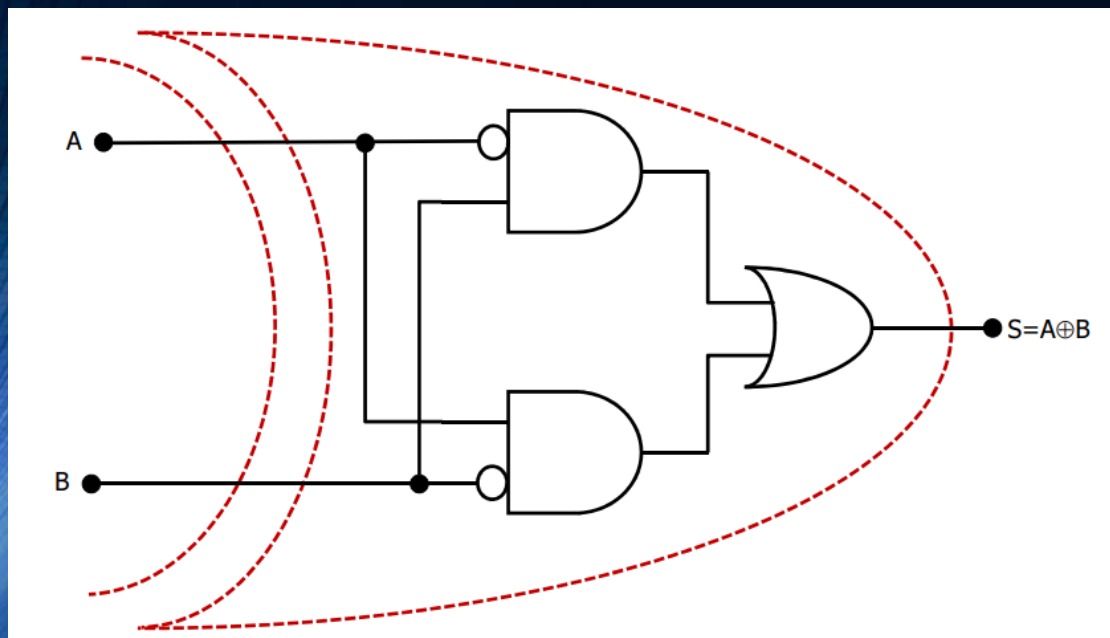
Portas Lógicas e Funções Booleanas



Instituto Infnet

Porta Lógica XOR

A porta XOR é um combinado de portas AND e OR, que de acordo com a organização apresentada abaixo, fornece a função OU EXCLUSIVO


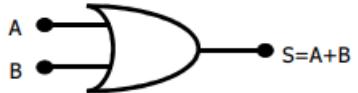
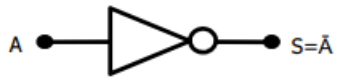
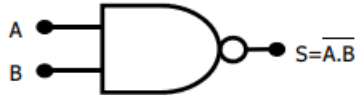




Portas Lógicas e Funções Booleanas



Instituto Infnet

Portas Lógicas

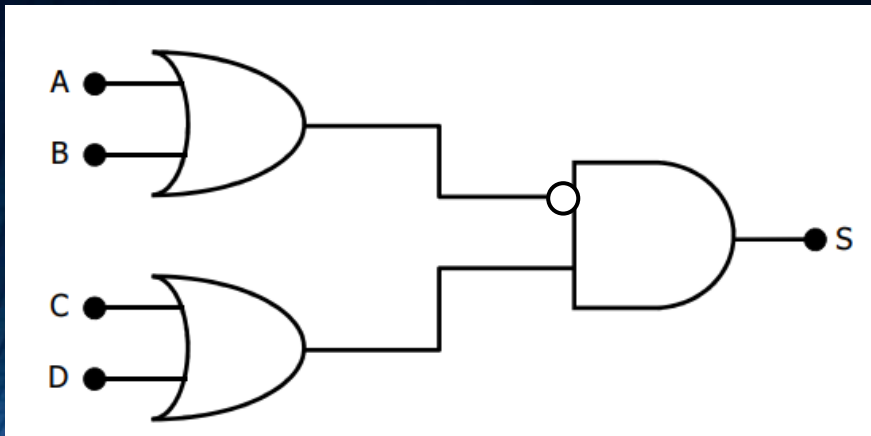
Nome	Símbolo Gráfico	Função Algebrica	Tabela Verdade															
E (AND)		$S=A.B$ $S=AB$	<table><tr><th>A</th><th>B</th><th>$S=A.B$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	$S=A.B$	0	0	0	0	1	0	1	0	0	1	1	1
A	B	$S=A.B$																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OU (OR)		$S=A+B$	<table><tr><th>A</th><th>B</th><th>$S=A+B$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	$S=A+B$	0	0	0	0	1	1	1	0	1	1	1	1
A	B	$S=A+B$																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NÃO (NOT) Inversor		$S=\bar{A}$ $S=A'$ $S=\neg A$	<table><tr><th>A</th><th>$S=\bar{A}$</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	$S=\bar{A}$	0	1	1	0									
A	$S=\bar{A}$																	
0	1																	
1	0																	
NE (NAND)		$S=\overline{A.B}$ $S=(A.B)'$ $S=\neg(A.B)$	<table><tr><th>A</th><th>B</th><th>$S=\overline{A.B}$</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	$S=\overline{A.B}$	0	0	1	0	1	1	1	0	1	1	1	0
A	B	$S=\overline{A.B}$																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOU (NOR)		$S=\overline{A+B}$ $S=(A+B)'$ $S=\neg(A+B)$	<table><tr><th>A</th><th>B</th><th>$S=\overline{A+B}$</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	$S=\overline{A+B}$	0	0	1	0	1	0	1	0	0	1	1	0
A	B	$S=\overline{A+B}$																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR		$S=A\oplus B$	<table><tr><th>A</th><th>B</th><th>$S=A\oplus B$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	$S=A\oplus B$	0	0	0	0	1	1	1	0	1	1	1	0
A	B	$S=A\oplus B$																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

Portas Lógicas e Funções Booleanas



Instituto Infnet

Composição de Portas



Processador

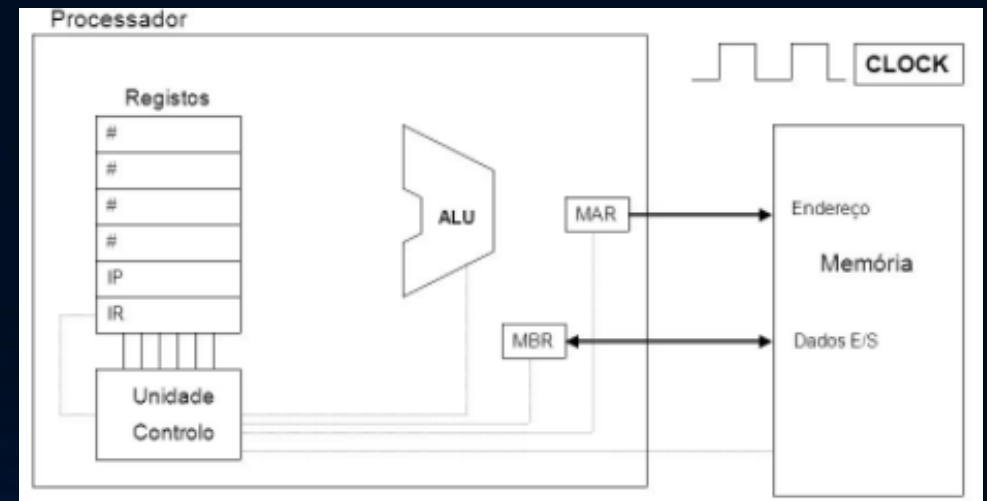


Instituto Infnet

A **unidade central de processamento**, ou mais comumente conhecido apenas por processador ou CPU (central processor unit), controla a operação do computador e realiza suas funções de processamento de dados; normalmente é chamado apenas de processador. Tradicionalmente, havia apenas um único processador em cada arquitetura. Nos anos recentes, aumentou o uso de múltiplos processadores em um único computador.

Seus principais componentes estruturais são os seguintes:

- **Unidade aritmética e lógica** (ALU, do inglês arithmetic and logic unit): realiza as funções de processamento de dados do computador.
- **Unidade de controle**: controla a operação da CPU e, portanto, do computador.
- **Registradores**: oferece armazenamento interno à CPU.
- **Interconexão da cpu**: algum mecanismo que oferece comunicação entre unidade de controle, ALU e registradores.



Processador



Instituto Infnet

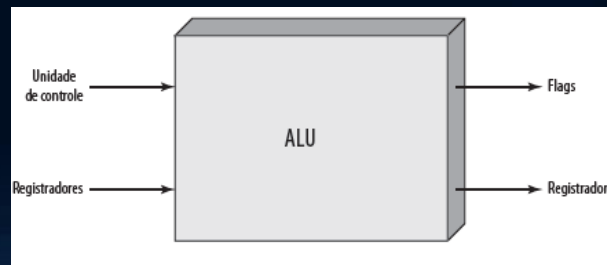
Unidade Lógica e Aritmética (ALU)

A ALU é a parte do computador que realmente realiza operações lógicas e aritméticas sobre os dados.

Todos os outros elementos do sistema de computação (unidade de controle, registradores, memória, E/S) existem principalmente para trazer dados para a ALU processar, e depois levar os resultados de volta.

De certa forma, chegamos ao núcleo ou essência de um computador quando consideramos a ALU.

Uma ALU e, na realidade, todos os componentes eletrônicos no computador, são baseados no uso de dispositivos lógicos digitais simples, que podem armazenar dígitos binários e realizar operações lógicas booleanas simples.



Aritmética do Computador



Instituto Infnet

As duas principais questões para a aritmética do computador são:

- O modo como os números são representados (o formato binário) ;
- Os algoritmos usados para as operações aritméticas básicas (adição, subtração, multiplicação, divisão). Essas duas considerações se aplicam à aritmética de inteiros e de ponto flutuante.



Aritmética do Computador



Instituto Infnet

Ponto Flutuante

Os números de ponto flutuante são expressos como um numero (significando) multiplicado por uma constante (base) elevada a alguma potencia inteira (expoente). Os números de ponto flutuante podem ser usados para representar números muito grandes e muito pequenos.

Base 10:

$$\begin{array}{rclcl} 0,0000586 & = & 0,586 & \times & 10^{-4} \\ -120000000000000 & = & -0,12 & \times & 10^{14} \end{array}$$

A maioria dos processadores implementa o padrão IEEE 754 para a representação e a aritmética de ponto flutuante. O IEEE 754 define um formato de 32 bits e um de 64 bits.

Aritmética do Computador



Instituto Infnet

Sistema Binário

No sistema numérico binário, números quaisquer podem ser representados apenas com os dígitos zero e um, o sinal de menos e a vírgula, ou vírgula fracionada.

$$-1101,0101_2 = -13,3125_{10}$$

Para as finalidades de armazenamento e processamento no computador, porém, não temos o benefício dos sinais de menos e vírgulas.

Somente dígitos binários (0 e 1) podem ser usados para representar os números. Se estivermos limitados a inteiros não negativos, a representação é direta.

$$00000000 = 0$$

$$00000001 = 1$$

$$00101001 = 41$$

$$10000000 = 128$$

$$11111111 = 255$$

Aritmética do Computador



Instituto Infnet

Sistema Binário

Existem várias convenções alternativas usadas para representar números inteiros negativos e também positivos, todas envolvem o tratamento do bit mais significativo (mais à esquerda) na palavra como um bit de sinal. Se o bit de sinal for 0, o número é positivo; se o bit de sinal for 1, o número é negativo.

A forma de representação mais simples que emprega um bit de sinal é a representação sinal -magnitude. Em uma palavra de n bits, os $n - 1$ bits mais à direita representam a magnitude do inteiro

+ 18	=	00010010	
- 18	=	10010010	(sinal-magnitude)

Aritmética do Computador



Instituto Infnet

Sistema Binário

Representação em complemento a dois

Por ser mais difícil de implementar através de portas lógicas, a representação sinal -magnitude raramente é usada na implementação da parte inteira da ALU. Em vez disso, o esquema mais comum é a representação de complemento a dois.

Assim como sinal -magnitude, a representação de complemento de dois utiliza o bit mais significativo como um bit de sinal, tornando mais fácil testar se um inteiro é positivo ou negativo. Ela difere do uso da representação sinal -magnitude no modo como os outros bits são interpretados.

Aritmética do Computador



Instituto Infnet

Sistema Binário

Representação em complemento a dois

A representação por trás dele é muito simples. Para obter a representação negativa de um número você deve:

- 1) Inverter todos os bits do seu número;
- 2) Somar 1.

E para o sistema expressar um valor negativo ao invés de positivo, basta que o bit mais significativo (MSB, ou bit mais à esquerda) seja 1. Caso contrário, o número é positivo.

Binário	Hexadecimal	Decimal	Comp. de 2
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	8	8	-8
1001	9	9	-7
1010	A	10	-6
1011	B	11	-5
1100	C	12	-4
1101	D	13	-3
1110	E	14	-2
1111	F	15	-1

Aritmética do Computador



Instituto Infnet

Sistema Binário

Conversão de Base – Decimal para Binário

O sistema de numeração que utilizamos é o decimal, e consiste em dez valores possíveis (0, 1, 2, 3, 4, 5, 6, 7, 8 e 9) para cada casa numérica.

Para converter um número decimal para a base binária, devemos dividi-lo pelo número 2. Na sequência, continuaremos dividindo os resultados até atingir o resultado 0.

o número binário resultante, será representado pelos restos da divisão, escritos debaixo (o último resto) para cima (o primeiro resto).

Aritmética do Computador



Instituto Infnet

Sistema Binário

Conversão de Base – Decimal para Binário

Vamos realizar a conversão do número 500. O primeiro passo é realizar a divisão do número pelo número de sua base. Na sequência, continuaremos dividindo os resultados até atingir o resultado 0.

Feito isso, devemos agora escrever o número binário, pegando a sequência correta dos restos da divisão.

Assim: $500_{10} = 111110100_2$

Valor inicial	Dividir pela base		Resultado	Resto
500	2	=	250	0
250	2	=	125	0
125	2	=	62	1
62	2	=	31	0
31	2	=	15	1
15	2	=	7	1
7	2	=	3	1
3	2	=	1	1
1	2	=	0	1

Aritmética do Computador



Instituto Infnet

Sistema Binário

Conversão de Base – Binário para Decimal

Podemos também realizar o processo inverso, ou seja, transformar o número binário em um no sistema decimal.

O primeiro passo é escrever as potências do número 2, da direita para a esquerda, começando em 0 e aumentar o expoente até $n-1$ números do binário.

Em seguida, devemos reescrever o binário e realizar a multiplicação do valor do binário pela sua potência respectiva. Assim, os valores correspondentes ao número 0 se tornarão 0 e os correspondentes ao número 1 se manterão iguais.

E o valor do número no sistema decimal será igual ao somatório dos valores obtidos na multiplicação.

Aritmética do Computador



Instituto Infnet

Sistema Binário

Conversão de Base – Binário para Decimal

Como exemplo, vamos transformar o número 11010111 para o sistema de medida decimal.

Como nosso exemplo tem 8 dígitos, o maior expoente será 7. Assim:

Binário	1	1	0	1	0	1	1	1
Potência	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
a								
Potência	128	64	32	16	8	4	2	1
a								



Binário	1	1	0	1	0	1	1	1
Potência	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Potência	128	64	32	16	8	4	2	1
Binário	1	1	0	1	0	1	1	1
Multiplicação	128	64	0	16	0	4	2	1
o								

E o valor do número no sistema decimal será igual ao somatório dos valores obtidos na multiplicação. Portanto: $11010111_2 = 215_{10}$

Componentes do Processador



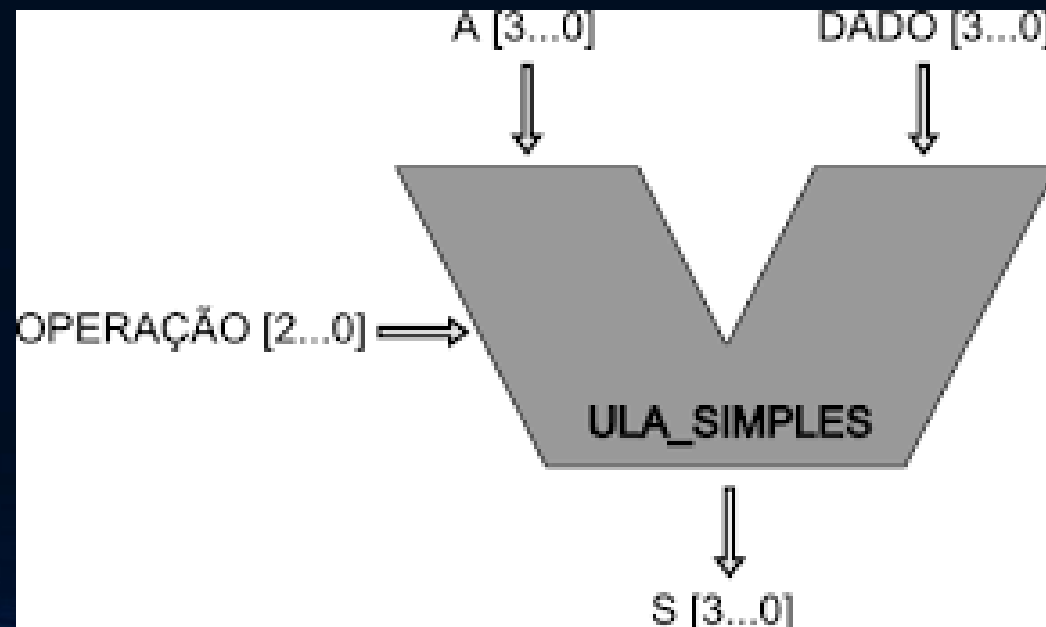
Instituto Infnet

Unidade lógica e aritmética

A ULA é parte do computador que realmente realiza operações lógicas e aritméticas sobre os dados. Todos os outros elementos do sistema de computação (unidade de controle, registradores, memória, E/S) existem principalmente para trazer dados para a ULA processar, e depois levar os resultados de volta.

De certa forma, chegamos ao núcleo ou essência de um computador quando consideramos a ULA.

Uma ULA e, na realidade, todos os componentes eletrônicos no computador, são baseados no uso de dispositivos lógicos digitais simples, que podem armazenar dígitos binários e realizar operações lógicas booleanas simples



Componentes do Processador



Instituto Infnet

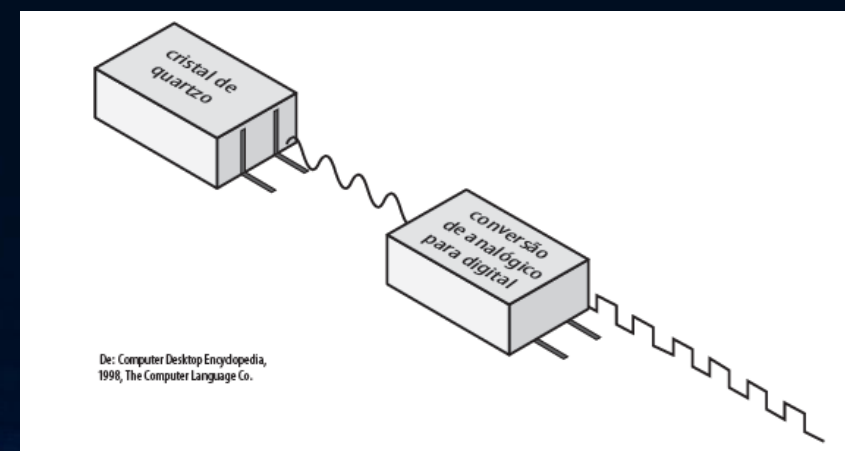
Clock do Sistema

As operações realizadas por um processador, como busca e decodificação de uma instrução, realização de uma operação aritmética e assim por diante, são controladas por um clock do sistema.

Normalmente, todas as operações começam com o pulso do clock. Assim, no nível mais fundamental, a velocidade de um processador é ditada pela frequência de pulso produzida pelo clock, medida em ciclos por segundo, ou Hertz (Hz). Os sinais de clock são gerados por um cristal de quartzo, que gera uma onda de sinal constante enquanto a alimentação é aplicada. Essa onda é convertida em um *stream* de pulsos de voltagem digital, que é fornecido em um fluxo constante aos circuitos do processador.

Por exemplo, um processador de 1 GHz recebe 1 bilhão de pulsos por segundo. A taxa de pulsos é conhecida como ***taxa de clock***, ou ***velocidade de clock***. Um incremento (ou pulso) do clock é conhecido como um ***ciclo de clock***, ou um ***clock tick***.

O tempo entre os pulsos é o ***tempo de ciclo***.



Componentes do Processador



Instituto Infnet

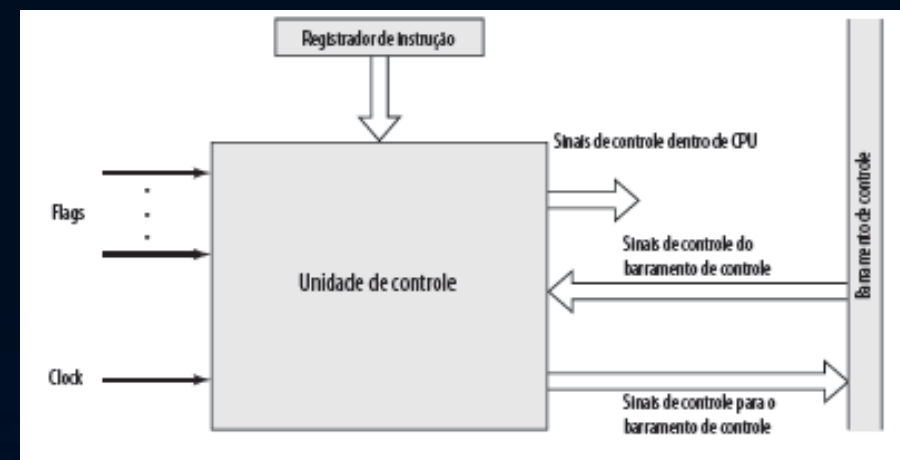
Unidade de Controle

A execução de uma instrução envolve a execução de uma sequência de subpassos, geralmente chamados de ciclos. Por exemplo, uma execução pode consistir de ciclos de busca, indireto, execução e interrupção. Cada ciclo é, por sua vez, feito de uma sequência de operações mais básicas chamadas de micro-operações. Uma única micro-operação geralmente envolve uma transferência entre registradores, uma transferência entre um registrador e um barramento externo ou uma simples operação da ALU.

A unidade de controle de um processador efetua duas tarefas:

- (1) ela faz com que o processador execute uma série de micro-operações na sequência correta, com base no programa que está sendo executado e
- (2) ela gera os sinais de controle que fazem com que cada micro-operação seja executada.

Os sinais de controle gerados pela unidade de controle causam a abertura e o fechamento de portas lógicas, resultando na transferência de dados para e de registradores e na operação da ALU.



Componentes do Processador



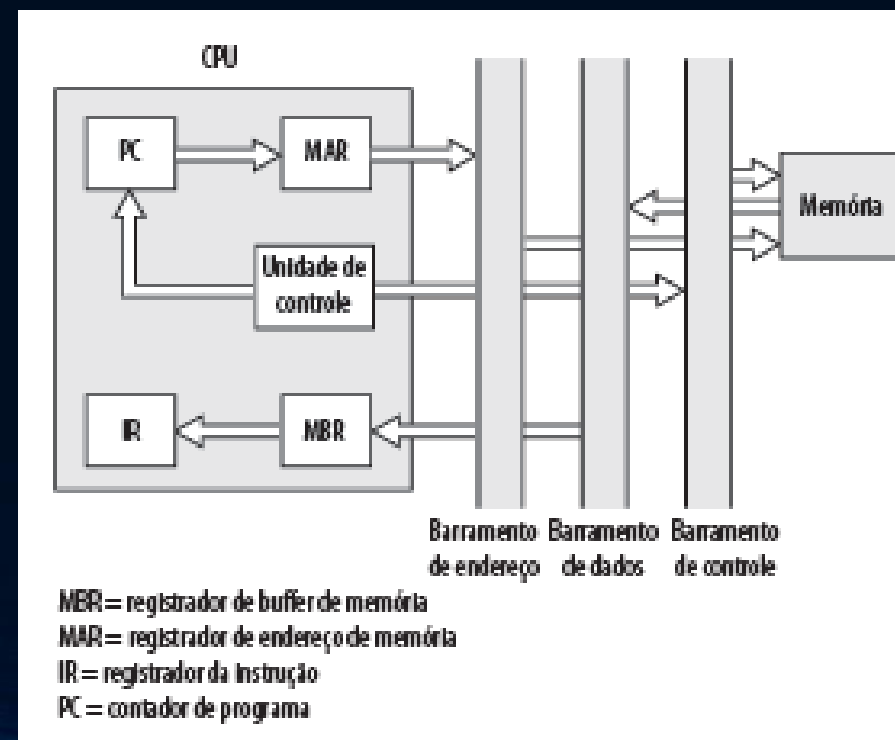
Instituto Infnet

Registradores

Para realizar a análise do ciclo de busca, o qual ocorre no início de cada ciclo de instrução e faz com que uma instrução seja obtida da memória, a Unidade de controle utiliza Registradores de memória. Para propósitos de entendimento, assumimos a organização ilustrada na Figura ao lado.

Quatro registradores estão envolvidos:

- **Registrador de endereço de memória (MAR):** conectado às linhas de endereço do barramento de sistema. Ele especifica o endereço na memória para uma operação de leitura ou escrita.
- **Registrador de buffer de memória (MBR):** conectado às linhas de dados do barramento de sistema. Ele contém o valor a ser guardado na memória ou o último valor lido da memória.
- **Contador de programa (PC):** guarda o endereço da próxima instrução a ser lida.
- **Registrador de instrução (IR):** guarda a última instrução lida.



Processadores CISC e RISC



Instituto Infnet

Uma instrução de um programa de alto nível é realizada por diversas instruções de processador de baixo nível. Por exemplo, uma instrução de um programa que imprime um conjunto de caracteres na tela é implementado no nível de processador por um conjunto de instruções.

De acordo com o número de instruções, pode-se classificar um processador como:

CISC - Complex Instruction Set Computing. Os processadores construídos na arquitetura CISC possuem uma grande quantidade de instruções complexas, vários modos de endereçamento, poucos registradores de dados e processamento controlado por microprograma. A filosofia é que o hardware é mais rápido que o software, então um conjunto poderoso de instruções produziria programas executáveis pequenos, ou seja, com poucas instruções.

Exemplo: Arquitetura x86

RISC - Reduced Instruction Set Computer . Tem como característica que as atribuições de dados simples, instruções condicionais, loops, e mecanismo de controle sequencial deveriam ser otimizados. Portanto, a arquitetura RISC reduz o conjunto de instrução ao mínimo, e as instruções não implementadas diretamente são realizadas por uma combinação de instruções existentes. Mas em compensação, um programa é implementado por um número maior de instruções. Exemplo: Arquitetura ARM

Arquitetura X86



Instituto Infnet

Em informática, x86 ou 80x86 é o nome genérico dada à família (arquitetura) de processadores baseados no Intel 8086, da Intel Corporation, que possui arquitetura de instruções CISC.

A arquitetura X86 foi introduzida pela Intel em 1978 com o lançamento do processador Intel 8086. Desde então, a arquitetura X86 evoluiu com o lançamento de várias gerações de processadores, como o Intel 80286, Intel 80386, Intel Pentium e Intel Core.

A arquitetura x86 possui um **comprimento variável de instrução**, onde basicamente são modelados em endereços CISC com ênfase em compatibilidade com versões anteriores.



Evolução dos processadores X86



Instituto Infnet

8080: o primeiro microprocessador de uso geral do mundo. Esta era uma máquina de 8 bits, com um caminho de dados de 8 bits para a memória. A partir deste processador, a família x86 foi desenvolvida.



8086: uma máquina muito mais poderosa, de 16 bits. Além de um caminho de dados mais largo e registradores maiores, o 8086 ostentava uma cache de instruções, ou fila, que fazia a pré-busca de algumas instruções antes que fossem executadas. Uma variante desse processador, o 8088, foi usado no primeiro computador pessoal da IBM, assegurando o sucesso da Intel. O 8086 é o primeiro aparecimento da arquitetura x86.



Evolução dos processadores X86



Instituto Infnet

80286: esta extensão do 8086 permitia o endereçamento de uma memória de 16 MB, em vez de apenas 1 MB.

80386: a primeira máquina de 32 bits da Intel e uma reformulação geral do produto. Com uma arquitetura de 32 bits, o 80386 competia em complexidade e potência com os minicomputadores e mainframes introduzidos alguns anos antes. Esse foi o primeiro processador da Intel a aceitar multitarefa, significando que poderia executar vários programas ao mesmo tempo.

80486: o 80486 introduziu o uso de tecnologia de cache muito mais sofisticada e poderosa, e pipeline sofisticado de instrução. O 80486 também ofereceu um coprocessador matemático embutido, tirando da CPU principal operações matemáticas complexas. Empresas como a AMD e Cyrix comercializam clones deste processador.



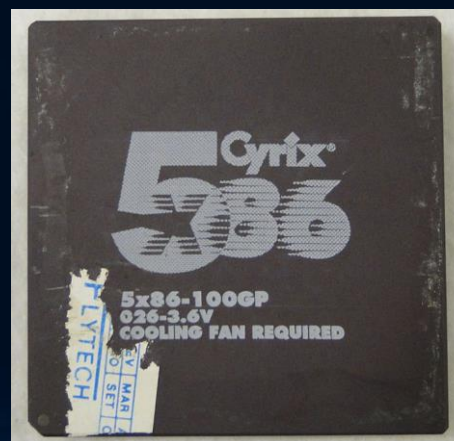
Evolução dos processadores X86



Instituto Infnet

Pentium: com o Pentium, a Intel introduziu o uso de técnicas superescalares, que permitem que múltiplas instruções sejam executadas em paralelo com instruções de 32 bits.

No mercado concorrente, a Cyrix e Amd apostavam numa melhoria do 486, denominada 586. Embora o "5x86" e o clock de 133 MHz dessem a entender que se tratava de um processador com um desempenho similar a um Pentium 133, o desempenho era muito inferior, mal concorrendo com um Pentium 66.



Evolução dos processadores X86

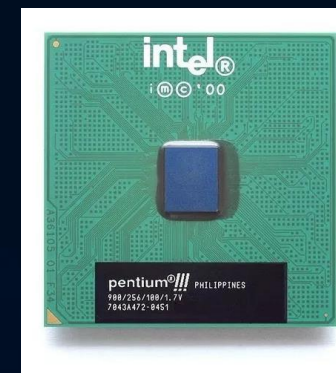


Instituto Infnet

Pentium MMX: Em 1996 a Intel lançou o Pentium MMX, que foi o integrante final da família Pentium. As instruções MMX permitiam empacotar determinadas instruções, o que multiplicava o poder do processador, permitindo que ele processasse até quatro instruções de 16 bits, ou até oito instruções de 8 bits como se fossem uma única instrução.

Pentium II: o Pentium II incorporou a tecnologia MMX da Intel, que foi projetada especificamente para processar dados de vídeo, áudio e gráfico de forma eficiente.

Pentium III: o Pentium III incorpora instruções adicionais de ponto flutuante para dar suporte ao software gráfico 3D.



Evolução dos processadores X86

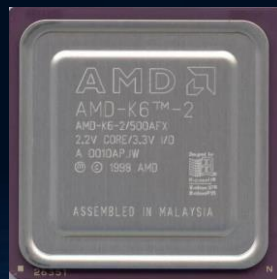
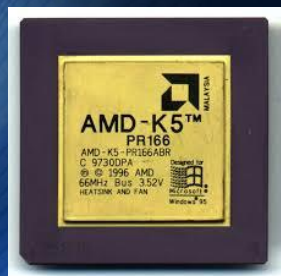


Instituto Infnet

K5 - Depois do 5x68 a AMD lançou o K5, um processador tecnicamente superior ao Pentium, mas que era caro e não era capaz de atingir frequências de operação competitivas.

K6 - Ele foi seguido pelo K6 e mais tarde pelo K6-2, que fez muito sucesso, servindo como uma opção de processador de baixo custo e, ao mesmo tempo, como uma opção de upgrade para quem tinha um Pentium ou Pentium MMX. Porém possuía um problema de arquitetura que levava a um aquecimento demasiado.

Athlon - As coisas voltaram aos trilhos com o Athlon, que se tornou a arquitetura de maior sucesso da AMD. A primeira versão usava um formato de cartucho (slot A) similar ao Pentium II, mas incompatível com as placas para ele. Ele foi sucedido pelo Athlon Thunderbird, que passou a usar o formato de soquete utilizado (com atualizações) até os dias de hoje.



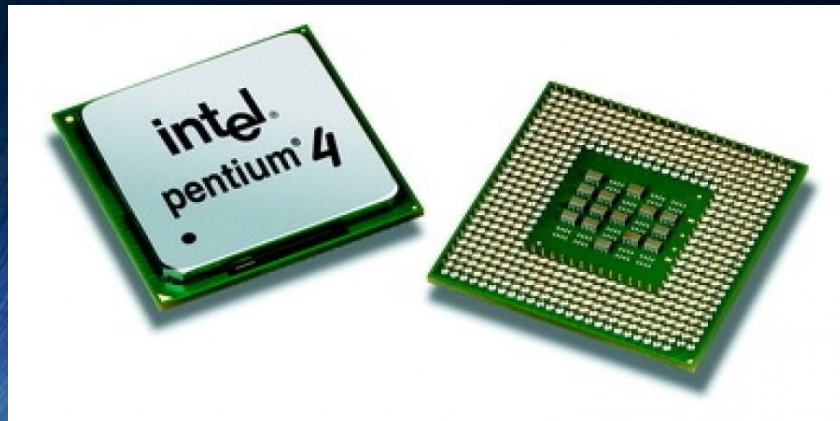
Evolução dos processadores X86



Instituto Infnet

Pentium 4: o Pentium 4 inclui ponto flutuante adicional e outras melhorias para multimídia. Introduz o conceito multithread

Core : estende a arquitetura para 64 bits. Oferece vários núcleos de processadores em um único chip.



Arquitetura ARM



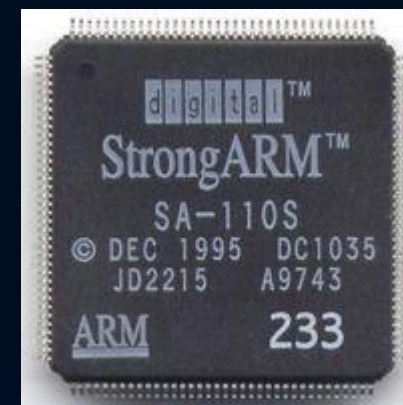
Instituto Infnet

ARM, originalmente Acorn RISC Machine, e depois Advanced RISC Machine, é uma família de microprocessadores e microcontroladores baseados em RISC, projetados pela ARM Inc., Cambridge, Inglaterra. A empresa não fabrica processadores, mas projeta arquiteturas de microprocessador.

O projeto ARM combinou com uma necessidade comercial crescente por um processador de alto desempenho, baixo consumo de energia, pequeno tamanho e baixo custo para aplicações embarcadas.

Microprocessadores com uma arquitetura RISC em geral necessitam de menos transistores do que microprocessadores CISC, como os da arquitetura x86, comumente encontrada em computadores pessoais.

Essa característica permite um consumo menor, custo menor, e dissipação de calor menor, o que faz dessa arquitetura algo desejado por fabricantes de dispositivos pequenos, portáteis, e movidos a bateria, como smartphones, laptops, e outros sistemas embarcados.



A ARM publica periodicamente atualizações para suas arquiteturas, praticamente todas elas com espaço de endereçamento de 32 bits

Evolução dos Processadores ARM



Instituto Infnet

Família	Recursos notáveis	Cache	MIPS típico @ MHz
ARM1	RISC 32 bits	Nenhuma	
ARM2	Instruções de multiplicação e swap; unidade de gerenciamento de memória integrada, processador gráfico e de E/S	Nenhuma	7 MIPS @ 12 MHz
ARM3	Primeira a usar cache de processador	4 KB unificada	12 MIPS @ 25 MHz
ARM6	Primeira a aceitar endereços de 32 bits: unidade de ponto flutuante	4 KB unificada	28 MIPS @ 33 MHz
ARM7	SoC integrado	8 KB unificada	60 MIPS @ 60 MHz
ARM8	Pipeline de 5 estágios; previsão estática de desvio	8 KB unificada	84 MIPS @ 72 MHz
ARM9		16 KB/16 KB	300 MIPS @ 300 MHz
ARM9E	Instruções DSP melhoradas	16 KB/16 KB	220 MIPS @ 200 MHz
ARM10E	Pipeline de 6 estágios	32 KB/32 KB	
ARM11	Pipeline de 9 estágios	Variável	740 MIPS @ 665 MHz
Cortex	Pipeline superescalar de 13 estágios	Variável	2 000 MIPS @ 1 GHz
XScale	Processador de aplicações; pipeline de 7 estágios	32 KB/32 KB L1 512KB L2	1 000 MIPS @ 1,25 GHz

Fim da Etapa 02