

SISTEMAS COMPUTACIONAIS

Rosana Ressa

E-book 2

FAM
ONLINE

Neste E-Book:

| | |
|---|----|
| INTRODUÇÃO | 3 |
| PORTAS LÓGICAS..... | 5 |
| HIERARQUIA DE MEMÓRIA | 13 |
| ARQUITETURAS CISC E RISC..... | 19 |
| PIPELINE | 22 |
| INTRODUÇÃO A SISTEMAS OPERACIONAIS | 26 |
| REDES DE COMPUTADORES: FUNDAMENTOS | 30 |
| CONSIDERAÇÕES FINAIS | 35 |
| SÍNTESE | 37 |

INTRODUÇÃO

O Módulo 1 foi praticamente uma iniciação à arquitetura e organização de computadores. Neste módulo, vamos começar a aprofundar no assunto. Dessa forma, examinaremos as portas lógicas e a álgebra booleana, pois o objetivo é que você consiga identificar os conceitos e os elementos básicos de organização das portas lógicas, bem como entender a representação matemática da lógica digital utilizada pela Álgebra de Boole.

Na sequência, abordaremos a Hierarquia de Memória, ou seja, você estudará inicialmente como a memória é organizada e suas características principais. O intuito é demonstrar que, para o funcionamento adequado de um computador, é necessário dispor, nele mesmo, de distintos tipos de memória e, com isso, você possa descrever o conceito e identificar os tipos.

Depois, você vai conhecer as arquiteturas RISC e CISC, pois o objetivo é mostrar que elas são diferentes, basicamente pela quantidade de instruções que podem executar. Em seguida, fazendo um link com esse tema, abordaremos a técnica de pipeline, a qual permitiu que dois ou mais processamentos fossem realizados ao mesmo tempo.

Dando continuidade, apresentaremos o conceito de Sistemas Operacionais de forma introdutória e teórica. O intuito é você entender a função e o que é implementado de acordo com as características de

cada equipamento. Embora o sistema operacional seja um assunto amplo, por ora mostraremos somente a relação de um sistema operacional com a arquitetura do computador.

Por fim, estudaremos as Redes de Computadores, momento em que você aprenderá o conceito básico de funcionamento de uma rede de computadores, os tipos e as classificações. A intenção deste tópico é reconhecer e qualificar a importância dos sistemas computacionais, das redes de computadores.

PORTAS LÓGICAS

As portas lógicas têm como base o sistema matemático de análise de circuitos lógicos, conhecido como **Álgebra de Boole**, que foi desenvolvida por George Boole (1815-1864), quem a publicou por volta de 1850. A álgebra de Boole expressa a operação de um circuito na forma de uma operação algébrica, em que suas constantes e variáveis podem assumir apenas dois valores: 0 ou 1.

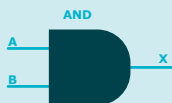
De acordo com Fávero (2011, p. 35), “esses circuitos, chamados de circuitos digitais, são formados por pequenos elementos capazes de manipular grandezas apenas binárias”. Tais elementos são conhecidos como portas lógicas, sendo que o termo *porta* é empregado para dizer que um circuito realiza uma operação lógica básica; os circuitos que incluem as portas lógicas são designados circuitos lógicos (CÓRDOVA JUNIOR; SANTOS; KISLANSKY, 2018).

Assim, a porta lógica é um alicerce para a constituição de qualquer sistema digital e, em geral, os circuitos lógicos são incorporados a um circuito integrado que pratica uma determinada função com a finalidade de desempenhar uma tarefa específica.

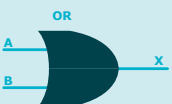
A álgebra booleana possui apenas três operações básicas: **AND** (e), **OR** (ou) e **NOT** (não). A porta AND, também chamada de conjunção binária, funciona como a multiplicação. Ela retorna um 1 se todas as entradas forem 1, senão retorna zero. Já a porta OR,

também conhecida como disjunção lógica, simula uma soma de binários, admitindo 1 sempre que uma ou mais de suas entradas forem 1. Na porta NOT, é representada a negação binária de uma informação, ou seja, será um bit 1 se o operando for zero, e será zero caso contrário. Essa porta é conhecida também como operador unário (FARIAS; MEDEIROS, 2013).

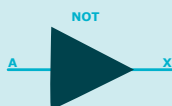
A Figura 1 apresenta um conjunto básico de portas lógicas e suas respectivas tabelas-verdade.



| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



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

Figura 1: Portas lógicas básicas e tabelas-verdade. **Fonte:** Adaptada de Farias e Medeiros (2019).

REFLITA

Existem várias formas de aplicação para a lógica binária. Uma aplicação interessante da porta AND seria na transferência de bits de dados de um lugar para outro, como da memória para a CPU. Diante disso, o objetivo seria o de garantir que um bit de origem seja o próprio bit de destino.

Existem portas lógicas distintas que resultam das portas lógicas apresentadas anteriormente. São elas a **NAND** (não e) e a porta **NOR** (não ou). Para realizar operações com essas portas, primeiramente é necessário realizar a operação AND ou OR e, em seguida, inverter seu resultado (Figura 2):

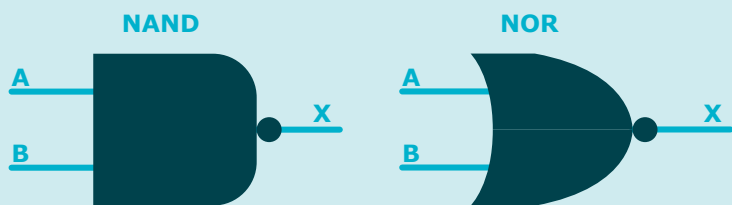
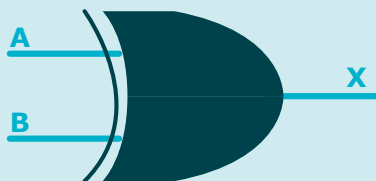


Figura 2: Símbolo gráfico NAND e NOR. **Fonte:** Adaptada de Fávero (2011).

Há também as portas lógicas especiais **XOR** e **XNOR**. A porta XOR (ou exclusivo) tem apenas duas entradas que, em sua saída, produz o nível lógico 1 quando suas entradas apresentarem valores distintos entre si, e o nível lógico zero quando as entradas forem iguais (Figura 3):

XOR

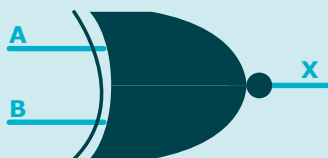


| A | B | $X = A \oplus B$ |
|---|---|------------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Figura 3: Porta XOR e tabela-verdade. **Fonte:** Adaptada de Fávero (2011).

Com a porta XNOR (ou-não-exclusivo), ocorre o inverso da XOR. Assim, teremos o resultado 1 unicamente se as duas entradas forem iguais (Figura 4):

XNOR



| A | B | $X = A \oplus B$ |
|---|---|------------------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Figura 4: Símbolo típico da porta XNOR e tabela-verdade.
Fonte: Adaptada de Corrêa (2016).

FIQUE ATENTO

Uma tabela-verdade é, essencialmente, formada por um conjunto de colunas, em que são listadas todas as combinações prováveis entre as variáveis de entrada (à esquerda) e o resultado da função (à direita).

Não obstante, um circuito lógico pode ter diferentes portas lógicas e, conseqüentemente, suas tabelas-verdade poderão ter muitas entradas e muitas saídas, as quais podem ser concebidas por suas relativas equações booleanas (FÁVERO, 2011). A Tabela 1 apresenta uma síntese das funções algébricas de portas lógicas.

| Função lógica | Função algébrica |
|---------------|----------------------------|
| AND | $X = A \times B$ |
| OR | $X = A + B$ |
| NOT | $X = \bar{A}$ |
| NAND | $X = \overline{A \cdot B}$ |
| NOR | $X = \overline{A + B}$ |
| XOR | $X = A \oplus B$ |
| XNOR | $X = A \odot B$ |

Tabela 1: Resumo das funções algébricas. **Fonte:** Adaptada de Stallings (2010, p. 708).

Dada uma expressão algébrica qualquer, é possível implementar o circuito lógico que a representa, uma vez que o circuito lógico é a combinação das portas lógicas pertinentes às operações que se concretizarão a respeito das variáveis de entrada. Por exemplo, a expressão algébrica: $X = (A + B) \odot C$, é esquematizada da seguinte maneira (Figura 5):

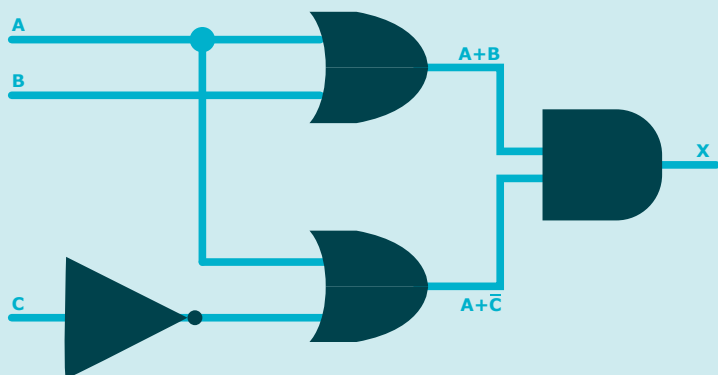


Figura 5: Exemplo de circuito lógico. **Fonte:** Elaborada pela autora (2019).

Pode-se observar que o resultado obtido das operações é conduzido por fios, os quais, no desenho, são representados por linhas simples.

SAIBA MAIS

Saiba mais sobre este conteúdo, assistindo a **Portas Lógicas (AND, OR, XOR, NOT)**, disponível em: <http://www.youtube.com/watch?v=4ENGYy68JqM&feature=related>. Acesso em: 10 jul. 2019.

HIERARQUIA DE MEMÓRIA

Uma das partes mais importantes de um sistema computacional (ou computador) e que merece destaque é a memória. Em linhas gerais, o processador apenas recebe os dados e os executa segundo determinada programação; em seguida, ele os devolve, não importando de onde saíram ou aonde vão.

Os programas a serem executados e os dados a serem processados ficam na memória, posto que o espaço para armazenamento de dados do processador é restrito. Nesse sentido, a memória é um instrumento que possibilita ao computador armazenar dados temporária ou permanentemente.

Segundo Rebonatto (2003), existem essencialmente dois tipos de memória que compõem a memória principal:

- ROM (*Read-Only Memory*): permite a leitura de dados, porém, é lenta. Em contrapartida, não perde seu conteúdo quando desligada.
- RAM (*Random Access Memory*): este tipo é ágil, admite a leitura e escrita, contudo, seu conteúdo é perdido assim que desligada.

A memória ROM é particularmente não volátil, pois é sempre empregada para armazenar programas de controle durante o processo de fabricação do chip. Não há como modificar ou apagar seu conteúdo,

visto que, dentro da memória ROM, existem necessariamente três programas: BIOS, POST e SETUP. Esses programas podem ser compreendidos como procedimentos básicos de inicialização de uma máquina.

Já a memória RAM tem como característica principal a volatilidade, ou seja, as informações nela armazenadas são perdidas na falta de energia elétrica. Contudo, o processador acessa a memória RAM praticamente o tempo todo, pois ela tem um tempo constante de acesso a qualquer endereço. A esse respeito, Patterson e Hannessy (2000) afirmam que a CPU manipula um dado em 5 nanosegundos, enquanto a memória dilata um dado em 60 nanosegundos.

FIQUE ATENTO

O Basic Input/Output System (BIOS) inicializa todos os dispositivos básicos do sistema (processador, placa de vídeo, unidades de disco, mouse etc.), logo depois, confere a gerência ao sistema operacional. O Power On Self Test (POST) é uma sequência de testes ao hardware de um computador, efetivada pela BIOS, com a finalidade de averiguar se o sistema está em estado operacional. Já o SETUP é empregado para configurar alguns parâmetros do BIOS, ou seja, é um sistema operacional bem incipiente que tem como responsabilidade colocar em funcionamento o computador assim que você o liga.

Para o funcionamento apropriado de um sistema computacional, é indispensável dispor, nele mesmo, distintos tipos de memória. Por isso, primeiramente a memória precisa ser organizada para que o processador saiba onde procurar um dado e onde alocar outro já processado. Decompondo as memórias em um subsistema, o objetivo almejado é um sistema de memória com performance muito próximo ao da memória mais rápida e o custo por bit associado ao da memória mais barata (CORRÊA, 2016).

A Figura 6 apresenta uma pirâmide com a hierarquia das memórias existentes em um computador. Ao avaliar os diferentes tipos de memória, que variam bastante em função de sua tecnologia de fabricação, capacidade de armazenamento, velocidade e custo, podemos dizer que é muito complexo projetar um computador usando somente um tipo de memória.

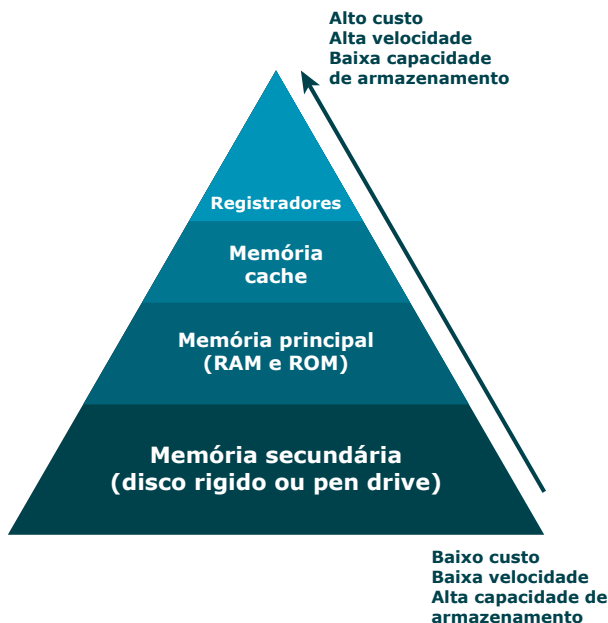


Figura 6: Hierarquia de memória. **Fonte:** Adaptada de Stallings (2010).

A base da pirâmide simula, então, os dispositivos de armazenamento de massa (memória secundária), de baixo custo por **bit** armazenado, mas ao mesmo tempo com baixa velocidade de acesso. A seta em direção ao topo sugere que quanto mais velozes ficam as memórias, mais alto será seu custo e menor sua capacidade de armazenamento em um computador.

A memória secundária é, na maioria das vezes, memória em disco magnético; a memória cache é tradicionalmente uma memória estática (SRAM), enquanto a principal, ou simplesmente RAM, é uma memória dinâmica (DRAM). A disposição da memória com

essa hierarquia é apropriada porque os programas utilizam o princípio da localidade, que são:

a) Temporal – acessam periodicamente os mesmos conjuntos de dados. Se um local de dados é acessado, ele tenderá a ser acessado novamente em breve.

b) Espacial – acessam dados que ficam próximos uns dos outros. Se um local de dados é acessado, os dados com endereços próximos tenderão a ser acessados em breve (CRISTO; PREUSS; FRANCISCATTO, 2013, p. 51).

Conforme discorreremos sobre a memória principal, é importante enfatizar que ela é o sistema de memória interna mais importante do computador, uma vez que “cada posição da memória principal tem um endereço único, e a maioria das instruções de máquinas refere-se a um ou mais endereços da memória principal” (CRISTO; PREUSS; FRANCISCATTO, 2013, p.127).

Os registradores são circuitos digitais centrados no interior do processador que são capazes de armazenar e deslocar dados binários, frequentemente utilizados como dispositivo de armazenamento volátil. Na maior parte dos computadores modernos, quando falamos da execução das instruções de um programa, os dados são levados da memória principal para os registradores.

Dessa forma, as instruções que utilizam esses dados são executadas pelo processador e, finalmente, os dados são movidos de volta para a memória principal, o que representa sua taxa de transferência de

dados dentro do processador é bastante elevada e consequentemente de alto custo.

A memória cache é um tipo de memória que é volátil e fica situada perto do processador. A memória cache tem basicamente duas funções: “Obter velocidade de memórias próxima das memórias mais rápidas; Disponibilizar uma memória de grande capacidade ao preço de semicondutoras mais baratas” (CORRÊA, 2016, p. 135). Ela contém os dados mais utilizados pelo processador, o que reduz o número de operações em que é necessário trazer dados diretamente da lenta memória RAM.

A memória secundária, também chamada de externa, é empregada para armazenar arquivos de programas e dados de forma mais permanente (não volátil). É formada por distintos tipos de dispositivos, alguns diretamente interligados ao sistema para acesso imediato (discos rígidos) e outros que podem ser conectados quando desejado (pen-drive), funcionando como suplemento da memória principal para guardar dados, não podendo ser endereçada diretamente pela CPU.

SAIBA MAIS

Saiba mais sobre os detalhes que caracterizam as especificidades de cada memória, como módulos, frequência, taxa de transferência e latência. Leia **Arquitetura e Organização de Computadores** (p. 31-41), disponível em: <https://bit.ly/2JOfm7x>. Acesso em 28 ago. 2019.

ARQUITETURAS CISC E RISC

Segundo atributos do conjunto de instruções, os processadores são dispostos em duas grandes famílias: Complex Instruction Set Computer (CISC), ou computador com conjunto de instruções complexo; e Reduced Instruction Set Computer (RISC), ou computador com conjunto de instruções reduzido. A arquitetura CISC ostenta um grande conjunto de instruções, abrangendo tanto as simples quanto as complexas; e por suportar mais instruções, sua execução fica mais lenta.

Para Fernandez (2015), trata-se de uma arquitetura em que cada instrução pode efetuar várias operações de baixo nível, como uma leitura da memória, uma operação aritmética e uma escrita na memória, empregando uma única instrução, o que a transfigura como extremamente versátil. No entanto, ela torna os circuitos da CPU e da unidade de controle bastante complicados. Um exemplo famoso dessa arquitetura é a família Intel x86, que remonta ao final da década de 1970.

Por sua vez, a arquitetura RISC tem como filosofia a transferência da complexidade das operações para o software, conservando no hardware somente as operações primitivas (RICARTE, 1999). Um computador RISC caracteriza-se por contar com um número pequeno de instruções, as quais fazem o mínimo

possível de operações simples e, por lidar com menos instruções, as executa com mais rapidez. Um exemplo de uso da arquitetura RISC é em celulares e videogames por serem processadores menores, mais baratos e que tendem a gastar menos energia.

REFLITA

Qual então é a melhor arquitetura?

Essa é uma questão difícil e sem resposta categórica. A melhor resposta, no entanto, é a de que depende do uso que se quer fazer do processador.

Uma diferença básica entre esses processadores é que os modelos CISC utilizam mais memória principal e cache, enquanto que os processadores RISC utilizam mais registradores. Embora a arquitetura CISC pareça menos vantajosa frente à arquitetura RISC (do ponto de vista da performance), atualmente podemos observar tecnologias híbridas que abrigam as características de ambas as arquiteturas, a fim de se obter o melhor desempenho possível.

Segundo Stallings (2010), reconheceu-se que os projetos RISC podem ser favorecidos com a inclusão de alguns aspectos da arquitetura CISC e vice-versa. A exemplo do projeto PowerPC, que não é composto “puramente” de RISC, do mesmo modo que o Pentium II incorpora algumas características do RISC.

SAIBA MAIS

Saiba mais das arquiteturas RISC e CISC, ao ler o artigo **Comparação entre as arquiteturas de processadores RISC e CISC**, de Luís Filipe Silva e Vítor José Marques Antunes, que está disponível em: <http://www.inf.unioeste.br/~guilherme/oac/Risc-Cisc.pdf>.

PIPELINE

Sabe-se que o processo de buscar instruções na memória é um gargalo relacionado à velocidade de execução de instruções. Para contornar essa restrição em sistemas computacionais, aplica-se a técnica de pipeline. Os pipelines são utilizados especialmente no processamento de instruções e operações aritméticas, que podem ser favoravelmente subdivididas em etapas menores de execução, por ser uma técnica de aceleração de execução de operações inicialmente pertinente a processadores RISC, mas que hoje em dia é empregada de forma ampla em todos processadores modernos (REBONATTO, 2003; RICARTE, 1999).

O princípio da técnica de pipeline é poder iniciar uma nova tarefa antes que o resultado da tarefa anterior na sequência de tarefas tenha sido gerado, ou seja, os processamentos inerentes a um pipeline são executados paralelamente, de maneira que todos sejam finalizados ao mesmo tempo. Contudo, o pipeline não suprime totalmente o tempo inativo em um processador, visto que executar paralelamente a função daqueles módulos inativos melhora de forma significativa a execução de um programa.

Um ponto importante no projeto de um pipeline é o equilíbrio entre várias unidades, também chamadas de estágios, em que a instrução é desmembrada em diversas partes e cada uma é manipulada por uma parte dedicada do hardware, e todas elas podem ser realizadas em paralelo (CORRÊA, 2016). A Figura 7

esboça um pipeline de 5 estágios (a) e o estado de cada estágio como uma função de tempo (b).

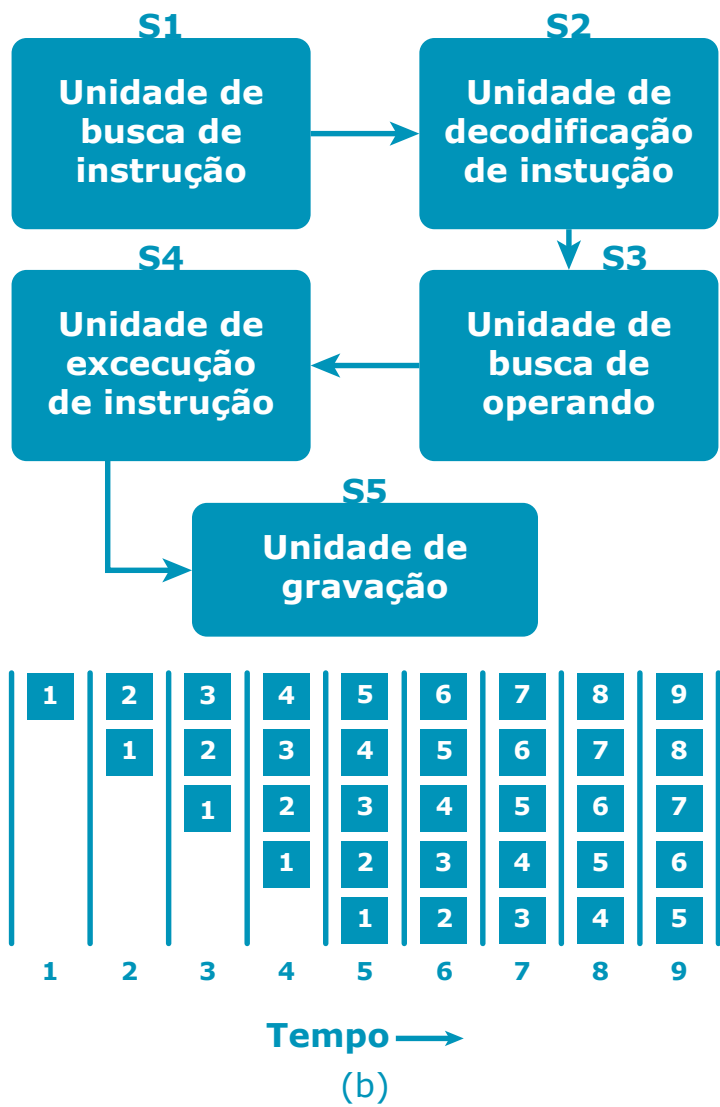


Figura 7: Exemplo de Pipeline. **Fonte:** Tanenbaum e Austin (2013, p. 51).

Nessa figura, observamos o funcionamento de um pipeline. Tratemos agora do que acontece a cada estágio de acordo com Tanenbaum e Austin (2013, p. 50):

O estágio 1 busca a instrução na memória e coloca-a em um buffer até que ela seja necessária. O estágio 2 decodifica a instrução, determina seu tipo e de quais operando ela necessita. O estágio 3 localiza e busca os operandos, tanto nos registradores como na memória. O estágio 4 realiza o trabalho de executar a instrução [...]. Por fim o estágio 5 escreve o resultado de volta no registrador adequado.

A figura ainda mostra como o pipeline trabalha em função do tempo durante cada ciclo de *clock*:

Durante o ciclo de clock 1, o estágio S1, está trabalhando na instrução 1, buscando-a na memória. Durante o ciclo 2, o estágio S2 decodifica a instrução, enquanto o estágio S1 busca a instrução 2. Durante o ciclo 3, o estágio S3 busca os operandos para a instrução 1, o estágio 2 decodifica a instrução 2 e o estágio 1 busca a terceira instrução. Durante o ciclo 4, o estágio 4 executa a instrução 1, S3 busca os operandos para instrução 2, S2 decodifica a instrução 3 e S1 busca a instrução 4. Por fim, durante o ciclo 5, S5 escreve (grava) o resultado da instrução 1 de volta ao registrador, enquanto os outros estágios trabalham nas instruções seguintes (TANENBAUM; AUSTIN, 2013, p. 51).

Dessa forma, o tempo total para efetivar uma execução em um pipeline é ligeiramente maior do que o tempo para realizar a mesma operação sem pipeline, já que um sistema de pipeline demanda mais recursos do que um que executa uma instrução de cada vez, pois seus estágios não podem reutilizar os

recursos de um estágio anterior. Contudo, as técnicas modernas de pipeline cuja finalidade é aumentar o desempenho de computadores, ao construir novas alternativas arquiteturais.

FIQUE ATENTO

Ciclo de clock (ciclos de relógio) é o tempo que a CPU gasta para executar uma instrução, em vez de informar o tempo de execução em segundos, geralmente é usado ciclos.

Buffer (um tipo de armazenamento) é uma pequena área de memória temporária empregada para melhorar a velocidade de acesso a um determinado dispositivo.

Acesse o Podcast 1 em Módulos

INTRODUÇÃO A SISTEMAS OPERACIONAIS

Grande parte dos usuários de sistemas computacionais já teve alguma experiência com um Sistema Operacional (SO), que é extremamente importante, pois torna possível a utilização de um computador. Um sistema operacional “é um programa que controla a execução de programas, aplicativos e age como uma interface entre o usuário e o hardware do computador” (STALLINGS, 2010, p. 241). Mesmo assim, é difícil reconhecer de modo absoluto o que é um SO. Esse problema ocorre porque os SO's exercem basicamente dois papéis não relacionados: estender a máquina e gerenciar recursos, que podemos denominar como conveniência e eficiência.

FIQUE ATENTO

Um sistema computacional (ou baseado em computador) abarca um conjunto de dispositivos eletrônicos (hardware) capazes de processar informações de acordo com um programa (software); ele automatiza ou apoia a execução de tarefas humanas por meio do processamento de informações.

O SO é como uma máquina estendida que trabalha com a abstração, uma vez que a arquitetura (conjunto de instruções, disposições de memória, E/S e composição de barramentos na maioria dos computadores em nível de linguagem de máquina) é primitiva e evolve programação complexa. Então, o SO controla a inicialização, sinalização, reinicialização dessas unidades, de modo mais simples e abstrato, escondendo os detalhes do hardware, provendo uma interface mais conveniente para a utilização do sistema (STALLINGS, 2010). Como gerenciador de recursos, ele controla o processamento, o armazenamento e a transferência de dados, realizando o compartilhamento de duas maneiras: no tempo e no espaço.

Quando se compartilha o recurso no tempo, diferentes usuários esperam por sua vez para usar um recurso. Por exemplo, com apenas uma CPU e múltiplos programas, o SO faz a alocação da CPU em um programa e depois, se ele foi suficientemente executado, outro programa consegue seu uso, em seguida, outro e, por fim, o primeiro programa novamente.

No compartilhamento do espaço, os usuários não aguardam pela sua vez, cada um toma sua parte no recurso requerido. Um exemplo disso é o compartilhamento da memória principal que, geralmente, se divide entre os múltiplos programas em execução. Desse modo, cada um pode operar ao mesmo tempo na memória (FARIAS; MEDEIROS, 2013).

Os sistemas operacionais mais conhecidos são Windows, MacOS e Linux. Mas algumas características os diferenciam devido à arquitetura do hardware em que irão executar, em suma, um SO é um sistema de computação **interativo** ou um sistema de processamento em **lotes** (*batch*) (STALLINGS, 2010).

Em um sistema interativo, o usuário/programador interage diretamente com o computador por meio de periféricos (monitor ou teclado) para requisitar a execução de tarefas ou realização de transações; assim, dependendo da natureza da aplicação, é possível a comunicação durante a execução da tarefa. De forma oposta, o processamento em lotes é realizado pelo agrupamento de um programa a outros programas de usuários distintos para serem submetidos à execução por um operador de computador. O usuário não tem acesso às informações durante o processamento. Quando termina a execução, os resultados são impressos ao usuário, e essa especificidade os torna mais raros hoje em dia.

Em um cenário independente, especifica-se se o sistema emprega multiprogramação ou não. A **multi-programação** tem por objetivo manter o processador o mais ocupado possível, o que o faz trabalhar em mais de um programa de cada vez, já que vários programas são carregados na memória e o processador reveza rapidamente entre eles. Uma alternativa a esse sistema é o sistema de **uniprogramação**, que atua somente com um programa de cada vez.

REFLITA

Imagine a seguinte situação: você chega em casa e, ao ligar seu notebook, constata que há um problema na inicialização do sistema operacional. A questão é que você não tem tempo para leva-lo a uma assistência técnica, pois precisa imprimir um documento importante que está armazenado no seu Hard Drive (HD).

Quais caminhos você acha que deveria seguir para resolver o problema e quais conceitos estão envolvidos?

REDES DE COMPUTADORES: FUNDAMENTOS

A interconexão de computadores é um axioma comum em espaços corporativos e, mais atualmente, em ambientes domésticos. Uma rede de computadores é composta por um conjunto de módulos processadores preparados para trocar informações e compartilhar soluções, que estão conectados por um sistema de comunicação por meio de cabos, fibras óticas ou links sem fio (CÓRDOVA JUNIOR; SANTOS; KISLANSKY, 2018). E um sistema computacional pode ser composto de uma rede de computadores, dependendo da conjuntura e necessidade. Sob essa ótica, uma rede precisa ser capaz de atender a três critérios básicos:

1. Performance: o desempenho de uma rede de computadores pode ser aferido por meio de alguns indicadores:

I. Tempo de tráfego: tempo necessário para que uma mensagem de rede trafegue de um dispositivo para outro da rede.

II. Tempo de resposta: tempo decorrido entre o envio da informação e a sua resposta.

III. Eficiência de software: métrica utilizada para verificar o tráfego de rede com softwares específicos.

IV. Número de usuários: a quantidade de usuários conectados a uma rede influencia diretamente na performance da rede.

V. Capacidade de hardware: capacidade de processamento de dados dos dispositivos físicos que compõem a rede também influencia na sua performance (CÓRDOVA JUNIOR; SANTOS; KISLANSKY, 2018, p. 184).

2. Confiabilidade: em redes de computadores, a confiabilidade diz respeito à preservação de dados de qualquer usuário ou a acesso não permitido. Ao navegar pela rede, os dados trafegam por múltiplas camadas de rede e podem ser rastreados, caso necessário. Por esse motivo, a proteção dos dados é um atributo indispensável em redes de computadores. Uma rede pode ter a sua confiabilidade avaliada pela taxa de pacotes de rede extraviados no processo de comunicação. Apesar disso, os protocolos de comunicação de rede apresentam tolerância a falhas, o que assegura a retransmissão de pacotes de rede que foram perdidos.

3. Escalabilidade: em uma rede de computadores, a escalabilidade indica sua habilidade de manipular uma porção crescente de trabalho de forma estável, ou estar organizada para crescer, sem que seja perdida a qualidade do serviço de rede.

Assim, em uma rede de computadores, todas as atividades que envolvam duas entidades comunicantes remotas são administradas por um protocolo. Computadores, placas de redes, fios de cobre e roteadores são instrumentos físicos que compõem

uma rede. Entretanto, análogo ao corpo humano, tais instrumentos carecem de uma “consciência” para torná-los úteis. Podemos chamá-las de protocolos de rede ou protocolos de internet.

“Basicamente, um protocolo é um acordo entre as partes que se comunicam, estabelecendo como se dará a comunicação” (TANENBAUM, 2003, p. 23). Logo, “um protocolo define o formato e a ordem das mensagens trocadas entre duas entidades ou mais entidades comunicantes, bem como as ações realizadas na transmissão e/ou no recebimento de uma mensagem no outro evento” (KUROSE; ROSS, 2009, p. 7).

Nas comunicações, existem protocolos que atuam em diferentes camadas e com funções distintas. Para a difusão de uma sequência de bits através de um fio de cobre, entre duas placas de rede contidas em dois computadores que se comunicam, é imperativo a implementação de protocolos de comunicação em seus hardwares, admitindo assim a codificação/decodificação da informação durante a transmissão no meio físico. O mais conhecido é **TCP/IP**, abreviatura de dois protocolos combinados: Transmission Control Protocol (TCP - Protocolo de Controle de Transmissão) e Internet Protocol (IP - Protocolo de Internet); juntos, entre todos os protocolos de rede, compõem a base de envio e recebimento de informações por toda a internet.

Outro ponto importante é a classificação de redes de computadores. A técnica mais comum de classificação de redes de computadores está relacionada

com seu alcance. Nesse sentido, temos as principais categorias:

- **LAN** (*Local Area Network*): também conhecida como “rede local”, a LAN interliga computadores localizados dentro de um mesmo ambiente físico. Pode abranger uma empresa ou redes relativamente pequenas (dentro de uma sala), sendo possível a troca de dados e recursos entre os dispositivos participantes.
- **MAN** (*Metropolitan Area Network*): uma rede de área metropolitana é análoga a uma LAN, porém, sua extensão abrange uma cidade ou um *campus* inteiro e é formada pela conexão de várias redes locais.
- **WAN** (*Wide Area Network*): ou “rede de longa distância”, é uma rede de comunicação que abarca uma ampla área geográfica, como cidades, países ou continentes. Pode ser privada ou pública. Um exemplo prático de WAN é a internet, a maior WAN do mundo. Seu funcionamento se dá pelo emprego de provedores de acesso, conectando diversas LANs ou MANs.
- **WLAN** (*Wireless Local Area Network*): rede de área local sem fio capaz de conectar dispositivos eletrônicos próximos e transmitir dados e informações sem o uso de cabeamento.
- **VPN** (*Virtual Private Network*): rede privada virtual é uma tecnologia que cria uma conexão criptografada e encapsulada em uma rede menos segura, como a internet. Como a internet é uma rede pública, existe a necessidade de estabelecer alguns mecanismos de segurança para que as informações trocadas entre os computadores de uma VPN não sejam lidas por outros.

SAIBA MAIS

Saiba mais sobre esse conteúdo, assistindo a **Redes de Computadores**: Introdução às Redes de Computadores (Aula 1), que se encontra disponível em: <https://www.youtube.com/watch?v=1csTmCZj-io>. Acesso em: 10 jul. 2019.

De acordo com Stallings (2010), à medida que a internet cresce, crescem também as necessidades computacionais nessa área, por conta do emaranhado de interações dadas cotidianamente nesse espaço. Assim, a implantação de sistemas computacionais modernos é exigida pela alta conectividade das redes de computadores devido ao grande volume de dados que nelas trafegam. Portanto, é importante saber como que uma rede de computadores, sob a ótica de sistemas computacionais, funciona, além de entender que quando essa estrutura é bem definida e administrada, a chance de sucesso no bom funcionamento aumenta significativamente.

Acesse o Podcast 2 em Módulos

CONSIDERAÇÕES FINAIS

Em sistemas computacionais, é muito importante que o aluno tenha uma visão holística do funcionamento dos computadores. Com a tecnologia avançando a cada dia, ela traz consigo uma crescente complexidade de processos. Notadamente, uma enorme quantidade de informações circula hoje em computadores no mundo todo. Para acompanhar esses avanços e o enorme volume de atualizações diárias, os profissionais da tecnologia da informação devem estar sempre atentos não só às mudanças, mas também à definição de alguns conceitos necessários para o entendimento da arquitetura e organização de computadores.

Inicialmente, expomos as portas lógicas, uma vez que executam operações sobre operadores binários (0 e 1), e podem ser aplicadas em diversas finalidades. As principais portas são AND, OR, NOT e XOR.

Em seguida, falamos da hierarquia de memórias, da diferença entre RAM e ROM, a importância da memória principal para um computador e como sua performance está intrinsecamente ligada ao custo. Depois, apresentamos as arquiteturas RISC e CISC. A RISC emprega um número pequeno de instruções, ao passo que a CISC é melhor com processamentos complexos, por trabalhar com um conjunto bem maior de instruções. Uma breve discussão sobre a utilização das duas arquiteturas em computadores modernos foi abordada.

Você aprendeu o conceito de Pipeline e como o processamento dos estágios de um pipeline são executados paralelamente, de forma que todos terminem ao mesmo tempo. Saímos do cenário dos processadores e fomos direto para os Sistemas Operacionais, mostrando as principais funções e sua interação com o hardware. O objetivo de aprendizagem desse tópico foi perceber que ele é responsável pela gerência e coordenação das atividades de compartilhamento dos recursos do computador.

Concluimos o módulo estudando as redes de computadores. O intuito foi fazer com que você seja capaz de verificar a utilidade das redes de computadores, compreender o conceito de protocolos e perceber a importância da internet. O entendimento básico do funcionamento de uma rede de computadores é essencial para facilitar a utilização dos recursos disponíveis em sistemas computacionais modernos. O assunto é extenso, e aqui foi explorado o conhecimento básico em relação à arquitetura e organização de computadores, mas você pode aprender muito mais com as referências indicadas no texto. É muito importante um constante aprofundamento no conteúdo para tornar-se possível o trabalho com os sistemas computacionais atuais, de complexidade e calibre sempre crescentes.



SISTEMAS COMPUTACIONAIS

Este módulo deu continuidade aos temas relacionados à arquitetura e organização de computadores. Primeiro, foi abordado o tópico **Portas Lógicas**, mostrando que elas são utilizadas para realizar operações aritméticas, a partir de portas lógicas básicas, uma vez que trabalham com sinais lógicos de entrada e saída e que podem ser expressos por tabela-verdade; e isso demonstra o poder da matemática no auxílio dos procedimentos computacionais.

Em seguida, apresentou-se a **Hierarquia de memórias**, muito importante para o entendimento do armazenamento e processamento de informações em um sistema baseado em computador, expondo então o conceito de **memória principal**, **cache** e **registradores**.

Foi discutida ainda a arquitetura de **processadores RISC/CISC** e suas diferenças. E depois foi apresentado o **Pipeline** e como se dá sua execução. No tópico seguinte, abordaram-se os conceitos de **Sistemas Operacionais**, com a intenção de que você entenda que como se faz o intermédio entre o hardware e o usuário, exercendo basicamente dois papéis não relacionados, a saber, máquina estendida e gerenciador de recursos, que podem ser denominados como conveniência e eficiência.

Por fim, estudamos as **Redes de Computadores**, o conceito básico de funcionamento de uma rede de computadores, os tipos e as classificações, e o objetivo desse tópico é devido à sua importância para os agentes computacionais ligados à internet.

Referências Bibliográficas & Consultadas

CÓRDOVA JUNIOR, R. S.; SANTOS, S. C. B.;
KISLANSKY, P. **Fundamentos computacionais**.
Porto Alegre: SAGAH, 2018.

CORRÊA, A. G. D. **Arquitetura e Organização de computadores**. São Paulo: Pearson Education do Brasil, 2016. [Biblioteca Virtual]

CRISTO, E. F.; PREUSS, E.; FRANCISCATTO, F.
Arquitetura de computadores. Universidade
Federal de Santa Maria, Colégio Agrícola de
Frederico Westphalen, 2013.

DELGADO, J.; RIBEIRO, C. **Arquitetura de computadores**. 5. ed. Rio de Janeiro: LTC, 2017. [Minha Biblioteca]

FARIAS, G.; MEDEIROS, E. S. **Introdução à Computação**, 2013. Disponível em: <http://producao.virtual.ufpb.br/books/gilbertofarias/introducao-a-computacao-livro/livro/livro.pdf>. Acesso em: 28 ago. 2019.

FÁVERO, E. M. B. **Organização e arquitetura de computadores**. Pato Branco: Universidade Tecnológica Federal do Paraná, 2011.

FERNANDEZ, M. P. **Informática: Arquitetura de Computadores**. 3. ed. Fortaleza: Editora UECE, 2015. Disponível em: http://www.uece.br/computacaoead/index.php/downloads/doc_download/2123-arquiteturadecomputadores. Acesso em: 08 jul. 2019.

GRAIG, J. J. **Robótica**. 3. ed. São Paulo: Person Education do Brasil, 2012. [Biblioteca Virtual].

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet: Uma abordagem top-down**. 3. ed. São Paulo: Person Education, 2009.

LESKOVEC, J.; RAJARAMAN, A.; ULLMAN, J. D. **Mining of Massive Datasets**. 2. ed. Cambridge University Press, 2014.

PATTERSON, D. A.; HENNESSY, J. L. **Organização e projeto de computadores: a interface hardware/software**. 2. ed. Rio de Janeiro: LTC, 2000.

REBONATTO, M. T. **Organização de Computadores: Notas de aula**. Passo Fundo, 2003. Disponível em: <http://usuarios.upf.br/~rebonatto/tsi120.htm>. Acesso em: 06 jul. 2019.

RICARTE, I. L. M. **Organização de Computadores**. FEEC/UNICAMP, 1999. Disponível em: <http://www.dca.fee.unicamp.br/courses/EA960/>. Acesso em: 08 jul. 2019.

ROGERSM D, L. **Transformação digital**: repensando o seu negócio para a era digital. São Paulo: Autêntica Business, 2017.

SEBASTIÃO, R. *et al.* **Fundamentos computacionais**. Porto Alegre: SAGAH, 2018. [Minha Biblioteca].

STALLINGS, W. **Arquitetura e organização de computadores**. 8. ed. Person Prattice Hall, 2010.

TANEMBAUM, A. S. **Redes de Computadores**. 4. ed. São Paulo: Campus, 2003.

TANENBAUM, A. S.; AUSTIN, T. **Organização estruturada de computadores**. 6. ed. São Paulo: Pearson Prentice Hall, 2013.

TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. **Sistemas digitais**: princípios e aplicações. 10. ed. São Paulo: Pearson Prentice Hall, 2007. [Biblioteca Virtual].

FaTM
ONLINE