

Unidade 1: Introdução à Arquitetura de Computadores

Prof. Daniel Caetano

Objetivo: Apresentar a evolução histórica dos computadores, os níveis de máquina e a importância da compreensão de diversas bases de contagem.

Bibliografia:

- MONTEIRO, M.A. **Introdução à Organização de Computadores**. 5ª. Ed. Rio de Janeiro: LTC, 2008.

- MURDOCCA, M. J; HEURING, V.P. **Introdução à Arquitetura de Computadores**. S.I.: Ed. Campus, 2000.

- TANENBAUM, A. S. **Sistemas Operacionais Modernos**. 2ª.Ed. São Paulo: Prentice Hall, 2003.

INTRODUÇÃO

- * O que é e como evoluíram os computadores?
- * O que existe entre nosso programa e os sinais elétricos?
- * Por que os computadores trabalham com números binários?

Atualmente é bastante incomum encontrar pessoas que jamais tenham visto ou operado um computador. Entretanto, para a maioria das pessoas o computador não passa de uma "caixa preta", isto é, um aparato "mágico" que produz resultados interessantes. O objetivo do curso de Arquitetura e Organização de Computadores é desvendar essa "caixa preta", apresentando os fundamentos que tornam possível seu funcionamento.

A compreensão dos computadores modernos sem conhecer a sua origem, entretanto, é muito mais árdua. Assim, esta primeira aula aborda a origem dos computadores e sua evolução, além de dar uma visão geral de toda a transição que ocorre entre aquilo que escrevemos na forma de um programa - o código - e aquilo que o computador realmente entende - sinais elétricos.

Finalmente, é apresentada uma breve introdução aos sistemas de numeração mais utilizados na informática, desvendando as razões pelas quais eles são utilizados.

1. EVOLUÇÃO DOS COMPUTADORES

- * Equipamentos Mecânicos e Eletro-Mecânicos
- * Equipamentos Eletrônicos e de Estado Sólido
 - Divisão em 6 Fases (Gerações)

A evolução dos equipamentos conhecidos hoje como "computadores" pode ser dividida duas grandes etapas: uma, inicial, envolve os equipamentos mecânicos e eletro-mecânicos. A segunda, mais recente, envolve os equipamentos eletrônicos e de estado sólido. Esta segunda etapa apresentou tantas transformações que acabou sendo dividida em diversas fases ou gerações.

1.1. Equipamentos Mecânicos e Eletro-Mecânicos

- * 500 a.C. - Ábaco
- * 1642 - Pascalene
- * Fim do Século XIX - Hermann Hollerith cria máquina de cartões
 - Fundação da IBM

Esse universo eletrônico que hoje conhecemos como "mundo da informática" teve seu início em épocas bem mais precárias, há muitos séculos.

Tudo começou com os babilônios, por volta de 500a.C., com a invenção do ábaco, que era uma ferramenta de cálculo manual. A primeira evolução do ábaco só veio em 1642, com a invenção da Pascalene, pelo físico e matemático Blaise Pascal. A Pascalene era um equipamento mecânico capaz apenas de realizar somas e subtrações. A evolução destes dispositivos foi muito lenta e eles eram pouco usados, devido ao uso limitado e desajeitado.

Foi apenas com a invenção do motor elétrico, já no fim do século XIX, foi possível construir máquinas mecânicas muito mais complexas e "rápidas". Uma das primeiras máquinas deste tipo foi usada com o propósito de realizar a contabilização do censo dos Estados Unidos da América. Esta máquina foi projetada por Hermann Hollerith, fundador da IBM e também criador da máquina que realizava o cálculo do pagamento dos funcionários, produzindo um pequeno resumo de contabilidade que recebeu o seu nome, sendo chamado até hoje de "Olerite".

1.2. Equipamentos Eletrônicos

- * Exigências militares: computadores humanos
- * Segunda guerra: Enigma
 - Computadores Humanos: Insuficientes

Como os equipamentos existentes até o fim do século XIX não eram adequados para resolver grandes problemas, todos os problemas mais complexos precisavam ser solucionados por seres humanos.

Os militares, por exemplo, que frequentemente precisavam de soluções sistematizadas para problemas complexos - fosse para distribuir produtos ou para construir edificações -, usavam profissionais específicos da área de cálculo e lógica, que compunham uma espécie de linha de produção de soluções de problemas. Haviam os profissionais que, em tempos de guerra, estabeleciam a lógica de solução para os problemas de distribuição de armas e suprimentos; a atividade exercida por eles ficou conhecida como "Logística". Entretanto, os "logísticos" só descreviam os procedimentos de solução, que precisavam ter seus resultados computados para que pudessem ser postos em prática. Os profissionais que realizavam os cálculos eram chamados de "Computadores".

Esta organização era suficiente para todas as necessidades até que, na segunda guerra mundial, os engenheiros alemães criaram máquinas complexas de criptografia de dados, chamadas "Enigma". As forças militares aliadas, incluindo o exército norte-americano, eram capazes de captar as mensagens transmitidas pelos alemães mas, como estas estavam codificadas, não era possível compreendê-las. Percebendo que era fundamental decifrar tais mensagens para a vitória e, verificando que a decodificação pelos computadores humanos era impossível, engenheiros foram chamados para que fossem propostas máquinas capazes de decifrar as mensagens codificadas pelo Enigma.

Das pesquisas nesta área, deu-se início aos primeiros equipamentos eletrônicos de computação, substitutos dos computadores humanos.

1.2.1. Primeira Fase

- * 1940 a 1955
- * ENIAC: 5000 adições por segundo
 - O que isso representa hoje?
- * Colossus
- * Linguagem de Máquina: Wire-up
- * Não existia conceito de Sistema Operacional
- * Falta de Confiabilidade: executar várias vezes!
- * EDVAC, IAS, UNIVAC I...

A primeira fase (ou primeira geração) dos computadores ocorreu aproximadamente durante o período entre 1940 e 1955, quando surgiram os primeiros computadores digitais usados durante a segunda guerra mundial.

O ENIAC (*Electronic Numerical Integrator and Computer*) foi o primeiro computador de propósito geral, desenvolvido para cálculo de Balística. Era gigantesco e tinha uma capacidade de processamento em torno de 5000 adições por segundo, valor este muito inferior ao de uma calculadora atual da Hewlett-Packard. Outro computador desenvolvido à mesma época foi o Colossus, este de propósito específico: decodificar as mensagens criadas pela máquina Enigma (e posteriormente Enigma 2) dos Alemães.

Estes primeiros computadores eram baseados em válvulas e programados com o método chamado *wire-up*, isto é, conectando fisicamente, com fios, diversos polos, fechando bits de forma a construir um programa. A programação era, assim, feita diretamente em linguagem de máquina e o equipamento não oferecia qualquer recurso para facilitar este trabalho de programação.

Outros grandes computadores construídos na época incluem o EDVAC, o IAS e o UNIVAC I, construído pela IBM para a computação do censo americano de 1950.

1.3. Equipamentos de Estado Sólido

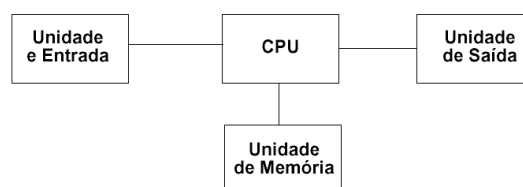
- * Problemas dos equipamentos eletrônicos
- * Transístores
- * Modelo de Von Neumann

Os equipamentos eletrônicos produzidos na primeira fase possuíam diversas limitações. Para começar, consumiam uma quantidade monstruosa de energia. Adicionalmente, eram enormes, esquentavam demasiadamente, demoravam horas para poderem ser usados a partir do momento em que eram ligados e, para finalizar, alguns cálculos tinham de ser repetidos diversas vezes, pois a confiabilidade dos resultados não era exatamente alta.

Uma mudança radical neste cenário só foi possível com a criação dos transístores, que permitiram a eliminação das válvulas e, com elas, a maior parte dos transtornos mencionados acima. Entretanto, o transistor, por si só, não eliminava uma das principais limitações destes equipamentos: a programação física através de fios. Como a memória dos computadores eletrônicos era muito pequena, apenas os dados do processamento eram armazenados nela.

Com a possibilidade de memórias maiores - seja pelo uso de transístores ou pelas novas "fitas magnéticas", este problema foi resolvido com o conceito de *software*, isto é, um programa armazenado em memória, conforme descrito por John Von Neumann.

Na proposta de Von Neumann, todo equipamento de computação deveria possuir quatro componentes: memória, unidade de processamento, dispositivos de entrada e dispositivos de saída, conforme apresentado na figura a seguir. Adicionalmente, um certo conjunto de dados armazenado na memória deve ser interpretado pela CPU como *instruções*, eliminando a necessidade de fios para a programação do equipamento.



1.3.1. Segunda Fase

- * 1955 a 1965
- * Transístores => confiabilidade
- * Memórias magnéticas
 - Maior velocidade e maior capacidade
- * Programas armazenados em memória (Modelo de Von Neumann)
 - Funções de E/S => embrião dos Sistemas Operacionais
 - + Eliminação de bugs
 - + Interface padronizada para dispositivos de E/S
 - + Escrita com Independência de Dispositivos
- * Automação de Processo Sequenciais
 - Processamento Batch => vários programas em sequência
 - + Cartão x Batch: qual a vantagem?
 - Sem intervenção
- * Escrita direta entre dispositivos (DMA)

A segunda fase ocorreu aproximadamente entre 1955 e 1965, e a grande inovação era o uso de transístores, o que permitiu uma grande redução no tamanho dos equipamentos e aumento de sua velocidade, além do aumento da **confiabilidade** do processamento. Também é desta época o surgimento das memórias magnéticas, que permitiram um aumento na capacidade e velocidade do armazenamento.

Nesta fase surgiram as primeiras linguagens e compiladores e, pela primeira vez, surgiu o conceito de sistema operacional como um software para automatizar todas as tarefas repetitivas utilizadas por diversos softwares e que, até então, eram realizadas manualmente (processamento *batch*). Originalmente, sempre que se desejasse executar um programa, o programador deveria inserir este programa no equipamento (através de um cartão perfurado), este programa seria executado e finalmente o resultado seria impresso. Entretanto, em geral este processamento durava horas e era comum que se passassem horas até alguém perceber que o processamento havia finalizado. Isso fazia com que o equipamento ficasse ocioso mesmo que muitas tarefas ainda estivessem por ser executadas, já que o equipamento dependia que um ser humano fosse até ele e o alimentasse com um novo programa.

Nesta geração, então, passou a ser possível introduzir diversos programas e o "sistema operacional" existente era capaz de executá-los em seqüência, ou seja, assim que um terminava, ele iniciava o seguinte e assim por diante, eliminando o problema da ociosidade. Ainda no final desta fase a IBM criou o conceito de "canal", hoje comumente chamado de "DMA", que permitia a escrita direta entre dispositivos sem a necessidade de intervenção da CPU.

1.3.2. Terceira Fase

- * 1965 a 1980
- * Circuitos Integrados => redução de tamanho e custos
- * IBM Série 360 / DEC PDP-8
- * ...Sistemas Operacionais... Ex.: OS/360
 - Multiprogramação: multitarefa cooperativa
 - + Processamento de um programa durante espera de outro
- * Terminais de Impressão e de Vídeo
 - Interação Online
- * PDP-7 (POSIX/UNIX e Linguagem C), Computadores Apple, CP/M

Esta fase ocorreu mais ou menos no período de 1965 a 1980, e foi marcada pela utilização dos circuitos integrados, que reduziram tamanho e custos e ampliaram enormemente a capacidade de armazenamento, processamento e confiabilidade dos computadores. Nesta época surgiu o conceito de família de processadores (IBM Série 360), em que vários equipamentos, com dispositivos diferentes conectados, eram compatíveis entre si. Também surgiram computadores de custo menor, como o PDP-8 da DEC.

Com o grande aumento de recursos destes equipamentos, os novos sistemas operacionais (como o OS/360) traziam novas possibilidades de gerência de processamento, permitindo que enquanto quando um programa esperava pela entrada de dados do usuário, outro fosse processado. Esta tecnologia ficou conhecida como *multiprogramação* e é uma técnica básica envolvida na *multitarefa cooperativa*. Nesta geração também passou a existir uma "interação online"; foram criados os primeiros terminais de vídeo e teclados para comunicação com o software *durante sua execução*.

Ainda nesta geração, a multiprogramação evoluiu de maneira a melhorar os tempos de resposta na interação com os usuários, desenvolvendo o conceito de *time-sharing*, isto é, cada processo compartilha a CPU por um intervalo de tempo. Este conceito é a base da *multitarefa preemptiva*.

Surgiu nesta fase, ainda, o sistema operacional UNIX, concebido inicialmente para o computador PDP-7, desenvolvido em Linguagem C, e tornou-se bastante conhecido por sua portabilidade. Outras grandes novidades desta época foram os computadores de 8 bits da Apple e o sistema operacional CP/M (Control Program Monitor).

Vale ressaltar que, nesta geração, houve a criação do padrão POSIX (Portable Operating System IX), que definiu uma interface mínima que sistemas UNIX devem suportar.

1.3.3. Quarta Fase

- * 1980 a 1990
- * Integração em Larga Escala (LSI e VLSI)
- * Computadores Pessoais (no Brasil, do MSX ao IBM-PC)
 - Recursos limitados: DOS era suficiente (sem *time-sharing* etc)
- * Computadores de Grande Porte
 - VMS: multitarefa monousuário
- * Computadores Multiprocessados
- * LANs, WANs, TCP/IP... Sistemas Operacionais de Rede

Nesta fase, que durou toda a década de 1980, a integração em larga escala (LSI e VLSI) permitiram uma redução substancial no tamanho e no preço dos equipamentos. Com isso houve o surgimento de diversos computadores menores mas muito potentes (variando desde os mais simples como o MSX até os mais poderosos IBM-PCs), ambiente no qual surgiu o DOS (Disk Operating System), base dos "computadores pessoais" do período. Estes equipamentos tinham processamento relativamente limitado e, portanto, o DOS não suportava muitas das características de multiprogramação, *time-sharing* e outros.

No campo dos computadores de grande porte, surgiu o sistema VMS (Virtual Machine System) que, implementando todos os recursos concebidos até então, criou oficialmente o conceito de *multitarefa* em um sistema monousuário.

Nesta fase surgiram os computadores capazes de *multiprocessamento*, com várias CPUs em paralelo, e os primeiros sistemas capazes de lidar com este tipo de característica também surgiram. Nesta fase houve proliferação das LANs e WANs, com o surgimento de diversos protocolos de comunicação e uma grande aceitação do protocolo TCP/IP.

Alguns autores (como Tanenbaum, 2003) não consideram a quinta e sextas fases, colocando os avanços posteriores ao da quarta fase dentro da própria quarta fase. Por questões didáticas, neste trabalho foi feita a opção pela separação.

1.3.4. Quinta Fase

- * 1990 a 2000
- * Enorme aumento da capacidade de processamento e armazenamento
- * Multitarefa nos computadores pessoais

A quinta fase compreendeu basicamente a década de 1990, sendo a tônica principal o aumento da capacidade de processamento e armazenamento em proporções não previstas anteriormente, possibilitando aplicação de inteligência artificial, bancos de dados e mídiã em praticamente qualquer tipo de aplicação, tornando-as muito mais complexas.

Nesta fase os computadores se tornaram muito baratos e passaram a fazer parte da vida de praticamente todas as pessoas. O conceito de *processamento distribuído* passou a fazer parte das pesquisas e a *multitarefa* veio para os computadores pessoais.

1.3.5. Sexta Fase

- * 2000 até hoje
- * Rede sem fio ubíqua
- * Limite físico para processamento de uma CPU
 - Multiprocessamento nos computadores pessoais
- * Processamento Distribuído é comum
- * Computação móvel

A sexta fase teve início juntamente com o século XXI e ainda não foi finalizada. As inovações trazidas são conhecidas pela maioria das pessoas, como a ubiquidade do acesso à rede, com redes sem fio e internet, com um aparente limite físico estabelecido da capacidade de processamento de uma unidade central de processamento e o *multiprocessamento* chegando aos computadores pessoais a baixos preços.

A quantidade de memória e velocidade de comunicação das redes permitem que grandes massas de dados sejam processadas e transmitidas, possibilitando video-conferências a um baixo custo. O processamento distribuído tornou-se uma realidade comum, embora ainda explorada apenas por aplicações científicas. A computação móvel tornou-se uma realidade, com a proliferação dos laptops e palmtops, levando os recursos computacionais a qualquer lugar.

2. NÍVEIS DE MAQUINAS

Na seção anterior foi possível verificar como os equipamentos foram evoluindo ao longo do tempo. Entretanto, a partir das máquinas de estado sólido, estabeleceu-se um paradigma de organização que facilita o estudo destes equipamentos.

Cada um dos níveis desta organização é denominado "nível de máquina" e, dentro de certos limites, cada um destes níveis de abstração pode ser estudado independentemente. Nestes termos, podem ser definidos 7 níveis de uma máquina, do mais alto para o mais baixo:

1. Programas Aplicativos
2. Linguagens de Alto Nível
3. Linguagem Assembly / de Máquina
4. Controle Microprogramado
5. Unidades Funcionais
6. Portas Lógicas
7. Transistores e Fios

Esta "independência" é que permite, na prática, que o usuário de um software qualquer não precise conhecer programação e que um programador não precise entender de eletrônica e portas lógicas, ficando esta tarefa apenas para os Engenheiros Eletrônicos.

A compreensão destes níveis é importante para que sejam compreendidos os diferentes níveis de compatibilidade que podem existir entre dois equipamentos.

Programas Aplicativos: é o nível com que, obviamente, o usuário de computador está mais familiarizado. É neste nível que o usuário interage com o computador, usando um programa (como jogos, editores gráficos ou de texto). Neste nível, quase nada (ou nada mesmo) da arquitetura interna é visível. Neste nível, existe a compatibilidade de "usabilidade", do tipo que você espera ao executar um programa como Microsoft Office ou Firefox independente de estar executando em um PC ou Mac.

Linguagens de Alto Nível: é o nível com que lidam os programadores de linguagens como C/C++, Pascal, Java etc. O programador lida com todos os detalhes de instruções e tipos de dados da linguagem que não necessariamente têm a ver com as instruções e tipos de dados da linguagem de máquina. É interessante citar a exceção do C/C++, onde algumas vezes o programador é obrigado a lidar com características especiais da linguagem de máquina. Por esta razão, o C/C++ às vezes é chamado, informalmente, de "a única linguagem de médio nível". Neste nível temos a chamada "compatibilidade de código fonte", em que um código escrito da maneira correta pode ser *compilado* para "qualquer" processador (ou CPU) e funcionar normalmente.

Linguagem Assembly / de Máquina: enquanto uma linguagem considerada de alto nível tem pouco a ver (ou nada a ver) com as instruções e estruturas de dados típicas de uma dada CPU, a linguagem de máquina (de baixo nível) é exatamente a linguagem desta CPU, com instruções próprias e tipos de dados intimamente ligados à forma como a CPU funciona. Estas instruções de uma CPU são chamadas de **conjunto de instruções** da máquina. Para programar neste nível, o programador precisa conhecer muito bem toda a arquitetura da máquina e também seu conjunto de instruções. Quando máquinas são compatíveis neste nível - ainda que o circuito seja completamente diferente de uma para outra, é dito que elas têm **compatibilidade binária**, pois uma é capaz de executar códigos de máquina da outra. A compatibilidade entre os diversos processadores Intel x86 e "compatíveis" vem até este nível.

Nos computadores digitais, a linguagem de máquina é composta por instruções binárias (longas seqüências de zeros e uns), também chamado de **código de máquina binário**. Entretanto, nenhum programador com um mínimo de recursos disponíveis trabalha com tais códigos, por ser um trabalho extremamente tedioso e sujeito a erros de digitação. Ao trabalhar com programação de baixo nível é comum o uso de **montadores** (*assemblers*), que foram, certamente, um dos primeiros tipos de software escritos. Estes montadores permitem que usemos palavras chamadas **mnemônicos** para expressar instruções da CPU (LOAD, MOVE, JUMP etc.) e o trabalho destes montadores é justamente o de traduzir estes mnemônicos para códigos de máquina. O conjunto de mnemônicos cujas construções têm relação direta de um para um com a linguagem de máquina é chamada **linguagem de montagem** (linguagem *assembly*).

Controle Microprogramado: é o nível que faz a interface entre a linguagem de máquina (código de máquina) e os circuitos que realmente efetuam as operações, interpretando instrução por instrução, executando-as uma a uma. Nos processadores "compatíveis com x86", incluindo os da própria Intel, é nessa camada que é feita a "mágica" da compatibilidade. Este nível foi "criado" pela IBM com a série de computadores **IBM 360**, em meados da década de 1960. Existem duas formas de fazer a microprogramação: uma delas é através de circuitos lógicos (*hardwired*), o que é extremamente eficiente e rápido, mas de projeto bastante complexo. Uma outra solução é através do microprograma, que nada mais é que um pequeno programa escrito em uma linguagem de ainda mais baixo nível executado por um **microcontrolador**. Este microprograma é também chamado de **firmware**, sendo parte hardware e parte software.

Unidades Funcionais: a grande maioria das operações da Unidade de Controle (parte da CPU) são exatamente para mover dados para dentro e para fora das "unidades funcionais". Estas unidades têm esse nome porque executam alguma tarefa importante para o funcionamento da máquina e, dentre elas, temos os registradores da CPU (memórias internas da CPU que possuem um nome específico), a ULA (que realiza as contas de fato) e a memória principal.

Portas Lógicas, Transistores e Fios: este é o nível mais baixo que ainda remete ao funcionamento de mais alto nível. As unidades funcionais são compostas de **portas lógicas**, que por sua vez são compostas de **transistores** interconectados. Abaixo deste nível existem apenas detalhes de implementação de circuitos (como níveis de voltagem, atrasos de sinal etc).

3. BASES NUMÉRICAS

Como foi visto nos níveis de máquina, no nível mais baixo tudo ocorre através de portas lógicas, isto é, com informações lógicas do tipo "falso" ou "verdadeiro". Eletronicamente falando, é comum considerar "falso" o sinal 0, isto é, um fio cuja tensão vale 0V, e considerar "verdadeiro" o sinal 1, isto é, um fio cuja tensão vale algo diferente de 0V (usualmente 3.3V ou 5V).

Como os fios do computador só podem estar "sem tensão" ou "com tensão" - 0 ou 1 -, o sistema de numeração mais próximo do funcionamento básico do computador é o sistema binário, em que apenas os dígitos 0 e 1 são usados. Assim, a menor unidade de informação tratada por um computador é chamada "bit", e tem o valor 0 ou o valor 1.

Por outro lado, para o programador, representar tudo através de números 0 e 1 pode ser bastante aborrecido (para não dizer chato, mesmo). Assim, nos níveis mais altos, é comum o uso de outras bases numéricas.

Nos primeiros computadores microprocessados, as instruções e informações eram organizadas em grupos de 4 bits; entretanto, apenas os 3 bits inferiores (os mais à direita)

eram "úteis"; como os computadores não eram exatamente confiáveis, o bit mais alto (o mais à esquerda) era usado como um dígito verificador (chamado "bit de paridade") e o programador tinha acesso a apenas 8 combinações numéricas. Assim, a notação OCTAL ganhou destaque, pois permitia exprimir cada sequência de 3 bits apenas com um dígito:

000	-	0	100	-	4
001	-	1	101	-	5
010	-	2	110	-	6
011	-	3	111	-	7

Por parecer pouca economia, mas observe o ganho no tempo de digitação e leitura. Cada tríade de bits era substituída por um único dígito.

000 010 111 100 001 000 100 101 010 <=> 0 2 7 4 1 0 4 5 2

Quando os computadores se tornaram mais confiáveis, as informações passaram a ser organizadas em grupos de 4 bits, denominado "*nibble*". Os "octais", de 0 a 7, não era mais suficientes, e o uso de decimais não era o mais adequado, como pode ser visto abaixo, devido à necessidade de dois dígitos para representar cada *nibble*.

0000	-	00	1000	-	08
0001	-	01	1001	-	09
0010	-	02	1010	-	10
0011	-	03	1011	-	11
0100	-	04	1100	-	12
0101	-	05	1101	-	13
0110	-	06	1110	-	14
0111	-	07	1111	-	15

Por essa razão, alguém teve a idéia de usar uma representação "hexadecimal", que os dígitos considerados são 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F, onde A representa o 10, o B representa o 11, o C representa o 12, o D representa o 13, o E representa o 14 e, finalmente, o F representa o 15. Assim, a tabela anterior pode ser escrita da seguinte forma:

0000	-	0	1000	-	8
0001	-	1	1001	-	9
0010	-	2	1010	-	A
0011	-	3	1011	-	B
0100	-	4	1100	-	C
0101	-	5	1101	-	D
0110	-	6	1110	-	E
0111	-	7	1111	-	F

E o conjunto de nibbles abaixo pode ser convertido sem qualquer dúvida:

0000 0001 1000 0100 1100 1001 1110 <=> 0 1 8 4 C 9 E

Quando os dados passaram a ser considerado em 8 bits, percebeu-se que não seria viável o uso de novas letras, uma vez que há 256 combinações possíveis (de 0 a 255) com 8 bits e, passou-se então, a especificar os números em hexadecimal, com um dígito por *nibble*:

00101101 \Leftrightarrow 2D

Os números de 16 bits passaram a ser representados como 4 dígitos hexadecimais e os números de 32 bits passaram a ser representados por 8 dígitos hexadecimais, separados por : quatro a quatro:

00101101 11110000 : 11001110 00010110 \Leftrightarrow 2DF0:CE16

Um ganho significativo! Reduz-se o tempo de digitação e a probabilidade de erro.

Entretanto, o uso de notação binária, octal e hexadecimal não é natural para nós, seres humanos, que estamos habituados ao sistema decimal. Assim, nas próximas aulas trabalharemos bastante as conversões ente bases numéricas, para que todos se familiarizem com as novas notações, que aparecerão com frequência tanto nos projetos de circuitos digitais quanto nos códigos de programação.

4. BIBLIOGRAFIA

MONTEIRO, M.A. **Introdução à Organização de Computadores**. 5ª. Ed. Rio de Janeiro: LTC, 2008.

MURDOCCA, M. J; HEURING, V.P. **Introdução à Arquitetura de Computadores**. S.I.: Ed. Campus, 2000.

TANENBAUM, A. S. **Sistemas Operacionais Modernos**. 2ª.Ed. São Paulo: Prentice Hall, 2003.