

Arquitetura De Computadores

O termo arquitetura é principalmente utilizado na construção e decoração de edificações. Ele diz respeito à forma e a estrutura de uma construção. O termo refere-se à arte ou a técnica de projetar e edificar o ambiente habitado pelo ser humano. Na computação o termo foi adaptado para denominar a técnica (talvez até a arte também) de projetar e construir computadores. Nesse livro você não vai aprender a construir seu próprio computador.

Para isso eu recomendo outros autores, John I. Hennessy, David A. Patterson e Andrew S. Tanenbaum. Esses autores produzem livros para engenheiros de computadores e acompanhá-los antes de se tornar um pode ser uma tarefa bastante árdua. Aqui você vai conhecer o computador por dentro e saber como ele funciona.

Você não será capaz de construir um computador, mas saberá o suficiente para entender como os programas funcionam e até porque o computador para de funcionar às vezes, ou funciona lentamente, e que nessas situações, pressionar teclas do teclado rapidamente, ao mesmo tempo que move o mouse aleatoriamente, não faz o computador voltar a trabalhar novamente.

Por que estudar arquitetura de computadores?

É essencial que todos profissionais da computação tenham pelo menos conhecimentos básicos de arquitetura de computadores. Saber como o computador funciona nos permitirá entender sua capacidade (e incapacidade) de resolver problemas, sobre como programá-los da melhor forma possível, como deixar o computador e os dados contidos neles mais seguros, como ganhar desempenho e o que faz ele ficar tão lento às vezes a ponto de quisermos destruí-lo.

Então, estudar arquitetura de computadores é tão importante para um profissional de computação, como estudar anatomia é importante para um médico.

Antes de iniciar qualquer estudo na medicina, um médico precisa saber em detalhes o funcionamento do corpo humano. Quais são seus órgãos, como eles trabalham individualmente e como se relacionam para formar um sistema (digestivo, respiratório, motor etc.). Com a arquitetura de computadores é semelhante. Vamos aprender quais são os componentes de um computador, como eles funcionam e como eles trabalham em conjunto formando um sistema. Sem dúvidas o ser humano é a máquina mais perfeita já criada, mas vamos ver que o computador é uma das máquinas mais incríveis que o homem já criou.

Um aluno de licenciatura em computação precisa estudar arquitetura de computadores?

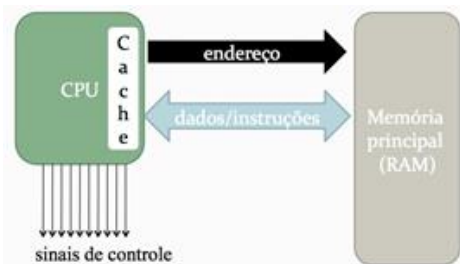
Ao longo de minha experiência como professor de arquitetura de computadores para alunos de licenciatura em computação, eu sempre ouvi essa pergunta. "por que precisamos estudar arquitetura de computadores?". Espero que isso já esteja claro para você.

Mas se ainda não estiver, aqui vai uma outra razão. Você será um licenciado e vai trabalhar no ensino e aprendizagem da ciência da computação. Como ensinar alguém sobre essa ciência se você não souber em detalhes como um computador funciona? Isso seria como um professor de farmácia que não conhece bem a química, ou um professor de matemática que não conhece os números.

Arquitetura Geral

Hoje em dia há muitos tipos de computadores e diversas arquiteturas. Elas são frutos de muitos estudos, pesquisas e avanços tecnológicos.

Mas todos computadores compartilham uma arquitetura comum. Essa arquitetura é o que separa um computador de uma calculadora de bolso, de um aparelho de televisão ou um relógio de pulso. Essa arquitetura é apresentada na.



Todo computador possui uma unidade central de processamento, ou, do inglês, central processing unit (cpu) e uma memória principal. Todos os dados a serem processados pela cpu, para operações lógicas e aritméticas, precisam estar na memória. Da memória os dados são transferidos para a cpu através de fios paralelos de comunicação, chamados de barramento de dados. Entretanto, a cpu não toma decisões por si própria.

Ela não sabe que dados deve trazer da memória, muito menos que operação executar com eles. Para isso, ela precisa que instruções, também armazenadas na memória, sejam trazidas para a cpu através do barramento de endereço. Cada instrução informa para a cpu que operação ela deve executar, com quais dados e o que ela deve fazer com o resultado da operação.

Para poder se localizar, a memória é organizada em endereços. Todos os dados e as instruções são localizadas através desses endereços. Cada instrução indica para a cpu que dados devem ser transferidos e processados através dos endereços desses dados.

Esse endereço é transferido para a memória pela cpu através do barramento de endereço. A memória localiza o tal dado e o transfere para a cpu via barramento de dados. As instruções são desenvolvidas pelo programador, através de linguagens de programação.

As ferramentas de compilação transformam os programas escritos em linguagens de alto nível, como c, java e python, em instruções de máquina, que são finalmente copiadas para a memória no momento em que precisam ser executadas.

Cada instrução é armazenada em um endereço diferente da memória. Na execução normal, a cpu passa para a memória, via barramento de endereço, o endereço da primeira instrução do programa, a memória transfere a instrução pelo barramento de instrução, a cpu a executa e, em seguida, solicita a instrução do endereço seguinte.

Assim, os programas são executados sempre de forma sequencial, a não ser que uma instrução especial solicite que ela salte para uma instrução que não seja a consecutiva.

Isso é o caso quando há instruções condicionais (como o "se" ou "if"), instruções de repetição (como "while" e o "for"), ou chamadas a sub-programas, ou mesmo, por ordem do sistema operacional, para que o programa pare de executar para que um outro tome seu lugar.

As memórias são, quase sempre, muito mais lentas do que as cpus. Isso exigiu, ao longo dos anos, que as cpus possuíssem também uma porção interna de memória muito rápida, chamada memória cache. A tecnologia que permite essas memórias serem mais rápidas, tornam-as também muito caras. Por isso que sua capacidade geralmente é muito limitada.

Para acelerar ainda mais, elas são instaladas dentro das cpus. Todos os dados e instruções transferidos da memória principal para a cpu são salvos também na cache. Como a cache não é capaz de guardar todos os dados da memória principal, apenas os dados mais recentes transferidos para a cpu permanecem na cache.

Técnicas muito avançadas são aplicadas para que se consiga, no máximo possível, manter os dados mais importantes daquele instante na memória cache.

A cpu também é responsável por enviar sinais de controle aos outros dispositivos do computador, como periféricos, dispositivos de entrada e saída, e memórias externas. Esses sinais são enviados quando uma instrução dá ordem para tal.

Por exemplo, quando uma instrução pede que uma mensagem sem impressa na tela, a cpu, ao receber e executar essa instrução, envia para o controle do monitor que imprima na tela a mensagem contida o endereço que também foi passada pela instrução.

É esse comportamento que diferencia um computador de outros dispositivos eletrônicos mais simples. A essência da cpu não é muito diferente de uma calculadora de bolso. Ela executa operações lógicas e aritméticas. Entretanto, no projeto do computador, o papel do homem foi substituído pela programação.

Todas instruções das tarefas que a cpu precisa executar são armazenadas na memória e, a partir de então, a cpu pode trabalhar sem qualquer interferência externa. Com a programação, a cpu pode também executar tarefas diversas, desde simulações, jogos, tocar músicas e vídeos etc. Simplificando, o computador é uma máquina programável e de propósito geral.

Operações Básicas

Todos computadores executam três operações básicas: * movimentação de dados * processamentos de dados * armazenamento de dados

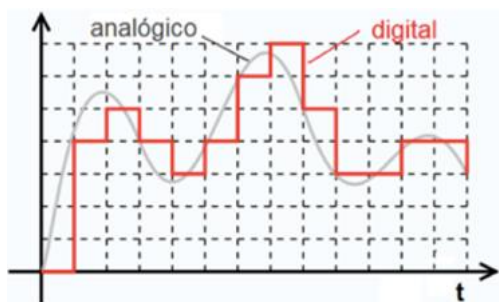
A movimentação de dados é a transferência de um dado de um ponto para outro do computador. Pode ser de um endereço de memória para outro, de um dispositivo de entrada para a memória, ou da memória para um dispositivo de saída.

O processamento de dados ocorre quando a cpu recebe um determinado dado e executa uma operação que o modifica de alguma forma. Já as operações de armazenamento ocorrem quando a cpu precisa registrar um dado em algum local específico, como salvar um dado no disco rígido, ou num pendrive, ou mesmo na memória principal.

Sistemas Analógicos X Sistemas Digitais

Para sabermos a importância de um computador e sua forma de funcionamento, precisamos conhecer suas potencialidades e suas limitações. O computador é um dispositivo eletrônico digital. Isso significa que ele armazena, processa e gera dados na forma digital. Por outro lado, o computador não é capaz de processar dados analógicos.

Eles antes precisam ser convertidos para digital para poderem ser utilizados por computadores. Mas o que venha a ser um dado analógico? Qualquer informação presente na natureza, como uma imagem, um som ou um cheiro, pode ser analisada em no mínimo duas componentes. Uma seria a sua intensidade e outra o tempo. A a seguir apresenta essa representação, onde o sinal em forma de onda cinza seria a representação de um sinal analógico.



Um som, por exemplo, é formado por vibrações no ar de diferentes intensidades (amplitudes) ao longo do tempo. Cada amplitude vai soar para nossos ouvidos como um tom diferente e alguns são até imperceptíveis aos nossos ouvidos.

Por outro lado, como o computador é um dispositivo baseado em números, para que ele armazene um som em sua memória e possa fazer qualquer processamento sobre ele (gravar, transmitir, mixar), ele deve antes representá-lo na forma de números. Ai que está a dificuldade. As intensidades possíveis de um som são tantas que se aproximariam do infinito.

Para tornar essa grandeza mais clara, imagine que pudéssemos emitir a intensidade do som emitido por um pássaro. Se em determinado momento dissermos que essa intensidade tem valor 5. Logo em seguida um outro som é emitido, medidos e constatamos que sua intensidade é 4.

Até aí tudo bem! Mas o pássaro poderá em seguida emitir diversos sons que estariam entre 4 e 5, como 4,23, ou 4,88938, ou até uma dízima periódica, como 4,6666... um ser humano, mesmo que não consiga medir a intensidade do canto do pássaro, consegue ouvi-lo, apreciá-lo e até repeti-lo com uma certa proximidade com alguns assobios. Mas o computador não trabalha assim!

Antes de tudo, um computador teria que discretizar esses valores medidos, ou seja, passá-los do domínio dos números reais para o domínio dos inteiros.

Assim, o que era 4 permanece 4, o que era 5, continua como 5, mas o que foi medido como 4,23 é convertido para 4, e o que era 4,88938 e 4,666 são convertidos para 5. Dessa forma, o computador passa a tratar com números reais e finitos. Um canto de um pássaro (ou até de uma orquestra sinfônica) pode ser armazenado e processado pelo computador. Na imagem apresentada, a onda quadrada representa um sinal digital.

Mas perceba que o som emitido pelo pássaro teve que ser modificado. Ele antes era complexo, rico e cheio de detalhes. Agora se tornou algo mais simples e reduzido. Houve uma perda de informação ao passarmos o dado do analógico para o digital. Processo semelhante ocorre quando outras informações da natureza são passadas para o computador, como uma imagem através de uma foto, ou uma cena através de um vídeo. Parte da informação deve ser ignorada para que possa ser armazenada em computadores.

Você deve estar se perguntando então, quer dizer que imagens e sons analógicos possuem mais qualidade do que digitais? A resposta rigorosa para essa pergunta é, sim! Mas uma resposta mais consciente seria, as vezes! Isso porque a perda causada pela digitalização pode ser reduzida até níveis altíssimos que modo que nem o ouvido, nem a visão humana serão capazes de perceber.

Como exemplo de dados analógicos podemos citar tudo o que vem da natureza, som, imagem, tato, cheiro, enquanto que digitais são todos aqueles armazenados por dispositivos eletrônicos digitais, como computadores, celulares e tvs (exceto as antigas analógicas).

Se uma foto digital, por exemplo, possui menos qualidade do que uma analógica, por que todos procuram apenas máquinas fotográficas digitais, transformando as analógicas quase em peças de museu? A resposta está na praticidade. Os computadores só entendem informações digitais.

Uma máquina fotográfica, mesmo com qualidade inferior, vai nos permitir passar as fotos para o computador, compartilhar com os amigos, aplicar edições e melhorias, ampliar e copiar quantas vezes quisermos.

Tarefas que antes eram impossíveis com máquinas analógicas. O mesmo pode ser refletido para músicas, documentos e livros. O mundo hoje é analógico!

O Transistor

O transistor é um componente eletrônico criado na década de 1950. Ele é o responsável pela revolução da eletrônica na década de 1960. Através dele foi possível desenvolver sistemas digitais extremamente pequenos. Todas funcionalidades de um computador são internamente executadas pela composição de milhões de transistores. Desde operações lógicas e aritméticas, até o armazenamento de dados em memórias (a exceção do disco rígido, cd, dvd e fitas magnéticas), tudo é feito pelos transistores.

Os primeiros eram fabricados na escala de micrômetros 10^{-6} metros). Daí surgiram os termos microeletrônica e micro-tecnologia. Depois disso deu-se início a uma corrida tecnológica para se desenvolver transistores cada vez mais rápidos, menores e mais baratos.

Essa revolução dura até hoje, mas foi mais forte nas décadas de 1980 e 1990. Foi emocionante acompanhar a disputa entre as empresas norte-americanas intel e amd para dominar o mercado de computadores pessoais.

A cada 6 meses um novo processador era lançado por uma delas, tomando da concorrente a posição de processador mais rápido do mercado. Poucos eram aqueles consumidores que conseguiam se manter a atualizados com tantos lançamentos.

O princípio básico é utilizar a eletrônica (corrente elétrica, resistência e tensão) para representar dados e depois poder executar operações com eles. A forma mais fácil de fazer isso foi primeiramente limitar os dados a apenas dois tipos. Zero e um.

O sistema de numeração binário é muito mais fácil de representar com dispositivos eletrônicos do que o decimal, por exemplo. O transistor possui dois estados. Ou ele está carregado, ou está descarregado, assim como uma pilha. Isso facilmente pode ser mapeado para o bit 1 (carregado) e o bit (0).

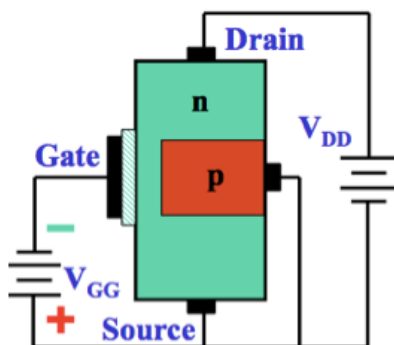
O revolucionário, diferente de uma pilha, foi possibilitar que esse estado pudesse ser mudado eletronicamente a qualquer momento e de forma muito rápida. Assim, com 8 transistores em paralelo, eu posso representar, por exemplo um número de 8 bits. Posso mudar seus valores mudando suas cargas, e posso ler seus valores chegando se cada um possui, ou não carga. Esse é o princípio básico de construção de uma memória.

De forma semelhante, é possível integrar transistores para que os mesmos executem operações lógicas e aritméticas. As portas lógicas estudadas por você em introdução à computação são todas fabricadas utilizando transistores. Quanto menores são os transistores, mais dados podem ser armazenados por área.

Ao mesmo tempo, transistores menores guardam menos carga. Isso torna mais rápido o processo de carregamento e descarregamento, que, por consequência, torna o processamento e armazenamento de dados muito mais rápidos também.

Com a evolução da nanoeletrônica, os transistores são tão pequenos que possibilitou a construção de memórias de 1gb (um giga byte) do tamanho da unha da mão de um adulto.

Para ser ter uma ideia, 1 giga é a abreviação de 10^9 , ou seja, um bilhão. Um byte são 8 bits. Então, uma memória de 1gb possui, pelo menos, 8 bilhões de transistores. Os processadores também se tornaram bastante velozes com a miniaturização dos transistores. Os processadores atuais trabalham na frequência de ghz (giga hertz), ou seja, na casa de bilhões de ciclos por segundo (diferente de operações por segundo).



Na imagem anterior é apresentada a estrutura de um transistor mosfet. Esse transistor é o mais utilizado para se construir sistemas eletrônicos digitais, como os computadores. O nome vem da abreviação de "metal-oxide semiconductor field-effect transistor".

Vamos ver o que significa cada palavra dessas, e isso nos ajudará a conhecer um pouco mais o mosfet e sua relevância. O termo mos ("metal-oxide semiconductor") vem dos materiais utilizados para compor um mosfet, que são principalmente, óxido metálico e semiconductor.

Semicondutores são materiais que possuem propriedades que nem os permitem classificar como condutor, nem como isolante. Em algumas condições ele age como um isolante, e em outras, como um condutor.

O semicondutor mais utilizado em transistores é o silício (símbolo si na tabela periódica). Em condições ambientes, o silício age como um isolante, mas se misturado a outros materiais, ele pode se tornar um condutor até a intensidade desejada.

No transistor da imagem a cor verde representa um cristal de silício que foi dopado com cargas negativas. Já vermelha, representa a parte que foi dopada com cargas positivas. Na situação normal uma corrente elétrica aplicada no dreno (drain) consegue produzir percorrer o estreito canal negativo e seguir até a fonte (source). Nessa condição dizemos que o transistor está ativo.

Porém, for aplicada uma tensão negativa na porta (gate), as cargas positivas da região p (em vermelho) serão atraídas para mais próximo da porta, e isso irá fechar o canal por onde passava a corrente elétrica. Nesse caso, dizemos que o transistor está inativo.

Por que isso tudo nos interessa? Quando o transistor está ativo, ele pode ser visto com o valor 1, e quando inativo, ele pode ser visto com o valor 0. Assim, temos a menor memória possível de ser construída.

Quando quisermos que ela guarde o valor 1, basta desligar a tensão da porta e aplicar uma corrente no dreno. Já quando quisermos que ele armazene o valor 0, precisamos aplicar uma corrente na porta e fechar o canal. Então, a memória de 8 bilhões de bits, pode ser elaborada com 8 bilhões de transistores como esses.

Agora conhecemos o primeiro aspecto que faz dos transistores essenciais para o entendimento do computador. Eles são usados para a construção de memórias. Memórias feitas a base de transistores são chamadas também de memórias de estado sólido. Mas há outras, não tão eficientes e miniaturizadas, como memórias ópticas e magnéticas. O importante percebermos é que quanto menores pudermos construir esses transistores, melhor.

O processo de abertura e fechamento do canal não é instantâneo. Ele leva um curtíssimo tempo, mas quando somados os tempos de todos os bilhões de transistores, ele passa a se tornar relevante.

Quanto menor ele for, mais estreito é o canal e, portanto, mais rápido ele liga e desliga, da mesma forma, menor será a distância entre o dreno e a fonte, levando também menos tempo para os elétrons deixarem o dreno em direção à fonte. Isso tudo fará a memória mais rápida.

Transistores pequenos também possibilitam que mais dados sejam armazenados por área. É por isso que hoje enormes capacidades de armazenamento são disponíveis em dispositivos tão reduzidos, como são os exemplos de pen-drives e cartões de memória.

Os transistores também são usados para executar operações lógicas e aritméticas. A carga retirada de um transistor pode servir para alimentar um outro e que, se combinados de forma correta, podem executar as operações lógicas básicas, e, ou, não e as aritméticas, adição, subtração, divisão e multiplicação.

Com isso, os transistores não apenas podem ser utilizados para armazenar dados, mas como executar operações lógicas e aritméticas sobre esses dados.

Isso é fantástico e vem revolucionando todo o mundo. Não só na ciência da computação, mas como também em todas áreas do conhecimento. O que seria da humanidade hoje sem o computador? Sem o telefone celular? Sem os satélites?

A Lei de Moore

Durante os anos de 1950 e 1965, as indústrias do vale do silício disputavam pelo domínio do recém-surgido mercado da computação e eletrônica.

Naquela época ainda não havia surgido o termo tic (tecnologia da informação e comunicação), mas ele seria mais apropriado para definir o nicho de clientes e serviços que eles disputavam. Eles dominavam a produção de circuitos eletrônicos digitais, dominados pela intel e amd, a produção de computadores e equipamentos de comunicação, como a dell, apple, ibm, hp e cisco, além da indústria e software e serviços, como a apple, microsoft e, mais tarde, a google. A disputa era grande e nem sempre leal.

Entretanto, não se sabia naquela época onde essa disputa ia parar, nem quem seriam os vencedores, nem mesmo, se haveria sequer vencedores.

Até um dos sócios e presidente da intel, gordon moore, lançou um trabalho minucioso onde ele destacava a experiência que ele adquiriu ao longo de alguns anos trabalhando na indústria de fabricação de processadores e circuitos para computadores.

Ele percebeu que, sempre a indústria avançava em sua tecnologia e conseguia reduzir o tamanho de cada transistor de um circuito integrado, os computadores tornavam-se também muito mais velozes do que antes.

Porém, essa redução no tamanho dos transistores requer uma total atualização nos equipamentos da indústria, tornando os equipamentos anteriores obsoletos.

Assim, só seria viável a evolução para transistores menores se o lucro da empresa fosse o suficiente para pagar todas essas despesas.

Por outro lado, ele também percebeu que os computadores e equipamentos mais obsoletos ainda possuíam mercado aberto em países menos desenvolvidos economicamente.

Ele concluiu então que a indústria seria sim capaz de continuar evoluindo na redução do tamanho dos transistores porque os novos computadores, sendo tornando mais velozes, seriam tão mais eficientes e atrativos, que todos os clientes, principalmente as empresas, fariam de tudo para trocar seus computadores antigos por novos, afim de se tornarem cada vez mais produtivos.

Além dessa análise de mercado, ele analisou como o processo industrial era concebido e como os novos computadores se beneficiariam da redução do tamanho dos transistores.

Ao final, ele afirmou que “a cada ano a quantidade de transistores por chip irá dobrar de tamanho, sem alteração em seu preço”. Essa frase pode ser interpretada também pelas consequências da quantidade de transistores por chip.

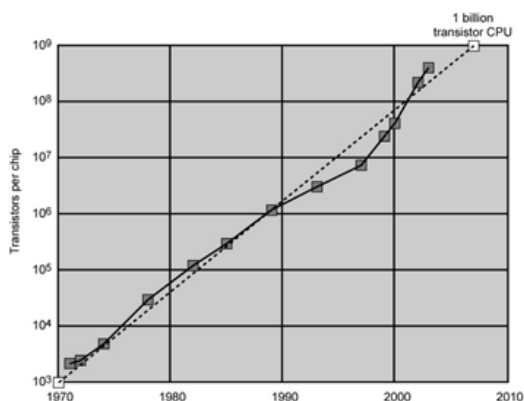
Ou seja, a cada ano, com o dobro dos transistores, os chips se tornarão duas vezes mais rápidos. Um exemplo mais comum de chip são os processadores dos computadores.

Então, por consequência, os computadores irão dobrar sua velocidade de processamento a cada ano, e ainda vão permanecer com o mesmo preço.

Naquela época essa era uma afirmação muito forte e ambiciosa. Muitos receberam esse estudo com cautela.

Mas não demorou muito para todos perceberem que as previsões de moore estavam se realizando. Foi tanto, e o trabalho dele foi depois chamado de “lei de moore” e ela ainda é válida até os dias de hoje.

Na figura a seguir é possível perceber como a quantidade de transistores por processadores cresceu dos anos 1970 até por volta de 2003 (linha contínua). É possível ver que ela não se afastou muito das previsões de moore (linha tracejada).



A lei de moore se tornou tão importante que ela não é usada apenas como uma meta a ser buscada e batida a cada ano, mas também como um meio para se verificar se a indústria está evoluindo na velocidade esperada. Apesar de moore está muito correto em suas previsões, todos sabem, inclusive ele próprio, que esse crescimento não vai durar para sempre. Os transistores hoje estão na escala de 25 nanômetros.

Essa é a mesma escala de alguns vírus e bactérias. Reduzir mais do que isso está se tornando cada vez mais difícil. Pesquisadores e cientistas buscam outras formas de fazer com que os computadores continuem evoluindo em sua velocidade e reduzindo seu tamanho.

Alguns pensam na substituição de transistores de silício por outros materiais, como grafeno. Outros até são mais radicais e defendem que a forma de computação deve mudar, talvez através de computadores quânticos ou de bio-computadores.

Quanto menores forem os transistores, mais rapidamente eles podem ser carregados e descarregados. Isso possibilita que o sistema trabalhe cada vez mais veloz. Mas há ainda outra limitação para a redução do tamanho dos transistores é a dissipação de calor. Quanto menores os transistores, mais deles são adicionados num mesmo circuito.

O funcionamento dos transistores, como dito anteriormente, é feito através da passagem de corrente elétrica (elétrons em movimento). Como toda máquina elétrica, nem toda corrente é aproveitada. Muito dela é desperdiçada através da dissipação de calor.

Então, uma vez que há milhões desses transistores trabalhando juntos, a dissipação de calor é ainda maior.

É muito importante para toda a humanidade que os computadores continuem evoluindo. A redução do tamanho dos computadores, aliada ao aumento de desempenho e sem o crescimento dos preços, permitiu que as ciências evoluíssem ao mesmo tempo, com a mesma velocidade.

A meteorologia, a medicina, as engenharias e até as ciências humanas avançaram sempre em conjunto com o avanço da computação. Para se ter um exemplo, foi a evolução dos transistores que permitiu que computadores se comunicassem numa velocidade tão grande que permitiu a formação da rede mundial de computadores, a internet.

Qualquer pessoa hoje consegue em poucos milissegundos fazer uma pesquisa por informações que estão do outro lado do planeta.

Algo que antes só era possível viajando até bibliotecas distantes e cheirando bastante mofo e poeira. Hoje, ter em casa bilhões de bytes (giga bytes) armazenados num minúsculo cartão de memória, é algo corriqueiro.

A informação está hoje disponível numa escala tão grande e numa velocidade tão intensa que parece que mais nada é impossível para a humanidade. Após a revolução industrial do século xviii que substituiu os trabalhadores braçais por máquinas, o século xx, puxado pela evolução dos transistores, passou pelo o que muitos consideram a revolução da informação e o século xxi, já é considerado a "era do conhecimento".

A Evolução dos Computadores

O Eniac

O primeiro computador criado foi o eniac ('electronic numerical integrator and computer'), desenvolvido por eckert e mauchly na universidade da pennsylvania, estados unidos. O projeto iniciou em 1943 financiado pelo governo americano.

O período era da segunda guerra mundial e o objetivo era poder calcular de forma mais ágil as melhores trajetórias para transporte de armas e mantimentos em meio aos exércitos inimigos.

Esse é o tipo de cálculo que pequenos aparelhos celulares fazem hoje para encontrar rotas nas cidades através de gps ('global positioning system') e análise de mapa. O projeto só foi concluído em 1946, tarde demais para ser utilizado para a segunda guerra, mas foi bastante utilizado até 1955.

O eniac ocupava uma área de 4500 metros quadrados, pesava 30 toneladas e consumia cerca de 140kw. Ele era capaz calcular 5000 somas por segundo. A programação era feita manualmente através da manipulação de chaves, ou seja, não havia linguagem de programação, nem compiladores ou interpretadores de comandos.

O sistema operacional só surgiu bem depois e tomou o emprego de muitos funcionários chamados na época de operadores de computadores. Profissão hoje extinta! O eniac ainda não utilizava transistores, mas válvulas que, dependendo de seu nível de carga, representavam um número.

Cada válvula precisava estar devidamente aquecida para funcionar corretamente, então o processo de ligar o eniac era trabalhoso e levava bastante tempo. Ele trabalhava com o sistema de numeração decimal, o que parecia óbvio naquela época, mas que depois dos transistores, se tornaram complexo demais e foi adotado o sistema binário.

Após a segunda guerra iniciou-se o período chamado de guerra fria, quando a espionagem, sabotagem e muito especulação reinava entre os países liderados pela união soviética e estados unidos. Prato cheio para os computadores. Possuir um computador que fosse capaz de decifrar mensagens codificadas dos inimigos era o sonho de consumo de todo general daquela época.

A Arquitetura de Von Neumann

Muitas empresas e governos corriam para construir seu próprio computador que fosse mais avançado do que os anteriores. Muitos projetos surgiram depois do eniac. Mas todos eles eram barrados por algumas dificuldades e limitações. Como por exemplo, o fato de não serem programados e trabalharem com números decimais.

O problema de trabalhar com decimais é que cada algarismo armazenado possui 10 estados possíveis, representando os números de 0 a 9.

Dentro de um sistema eletrônico, isso é complicado por que a carga de cada dispositivo, seja transistor, seja válvula, deveria ser medida para verificar se que número ela estava representando. Os erros eram muito frequentes. Bastava que uma válvula estivesse fora da temperatura ideal para que os resultados das operações comessem a sair errado.

Von neumann recomendou em sua arquitetura que os dados e instruções fossem agora armazenados em binário, facilitando a análise dos mesmos e reduzindo a quantidade de erros.

Em 1952, o professor John von Neumann, da universidade de Princeton, Estados Unidos, apresentou um projeto inusitado para a arquitetura de um computador.

Ele sugeriu que o computador fosse organizado em componentes, cada um executando apenas uma única tarefa e de forma muito mais organizada. Ele propôs que o computador fosse composto por:

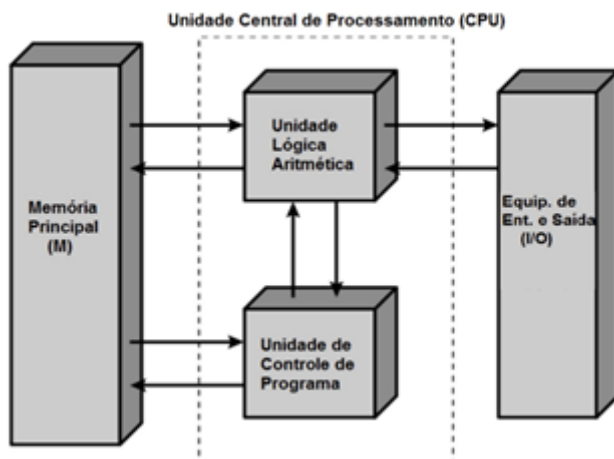
Memória principal: responsável por armazenar os programas a serem executados, assim como os dados a serem processados

Unidade lógica e aritmética (ULA): para realização das operações lógicas e aritméticas

Unidade de controle: que, baseado nas instruções lidas da memória, enviaria sinais de controle para a ULA para que a mesma executasse as operações devidas

Unidade central de processamento (CPU): que agruparia a ULA e a unidade de controle

Unidade de entrada e saída: responsável pela comunicação com os periféricos do computador (teclado, monitor, memória externa etc.)



A arquitetura de von Neumann deu tão certo que todos os fabricantes começaram a segui-la. Os computadores utilizados até hoje em dia seguem os preceitos básicos propostos por ele. Muitos avanços surgiram, melhorias foram feitas, mas até hoje os computadores são formados por unidades de controle, CPU, ULA, memória e unidades de entrada e saída.

John von Neumann deixou um legado para toda a humanidade.

A IBM

A International Business Machines, ou apenas IBM, foi fundada em 1911 com o nome de Computing Tabulating Recording (CTR) e iniciou produzindo e comercializando calculadoras para empresas e empresários.

Só em 1924 é que ela muda de nome para International Business Machines ou apenas IBM. Ela é uma das poucas empresas que sobreviveram a todos os avanços da computação e continua sendo uma potência mundial.

Apenas em 1953 a IBM entra no mercado de computadores com o IBM 701, tendo sempre as grandes organizações como alvos. Só muitos anos depois é que os computadores pessoais foram surgir.

O IBM 701 trabalhava com cartões perfurados, ou seja, toda programação dele era feita através de uma perfuradora que marca buracos para representar o bit 1, e deixava íntegra uma área para representar o 0.

O mesmo acontecia depois que os programas eram lidos e processados.

Uma folha de papel era perfurada pelo computador para representar os resultados das operações executadas.

Não preciso nem dizer o que isso era trabalhoso!

Em 1955 a IBM lança o IBM 702 que agora não fazia apenas cálculos científicos, mas também aplicações comerciais, visando deixar de ser um equipamento apenas para cientistas, mas também para empresários. Depois desses vários outros computadores foram lançados nas séries 700. Essas máquinas ainda utilizavam válvulas para armazenar os dados.

Só em 1957 é que surge a segunda geração de computadores, com a utilização de transistores. Isso tornou os computadores mais leves, baratos, rápidos e mais energeticamente eficientes. Os primeiros computadores dessa geração foram o IBM 7000 e o PDP-1, da DEC, empresa que não existe mais.

A IBM lança em 1964 o IBM série 360, substituindo os antigos computadores da série 7000. O IBM 360 inicia a primeira família de planejada de computadores. Isso significava que todos computadores seguintes da série 360 seriam compatíveis com os anteriores.

Todos programas desenvolvidos ou adquiridos pelas empresas poderiam ser usados mesmo que a empresa substituísse os computadores pela geração mais nova. Isso tornou a ibm uma das empresas mais poderosas do mundo na época, com filiais e representantes em as continentes do planeta.

As Gerações Dos Computadores

As gerações de computadores surgiram com a miniaturização dos transistores e sua integração em chips em escalas cada vez maiores. Podemos então ver as gerações dos computadores como:

1946 a 1957: computadores baseados em tubos de vácuo

1958 a 1964: surgimento dos transistores

1965: indústrias atingiram a integração de até 100 transistores num único chip

1971: chamada de integração em média escala, com até 3000 transistores por chip

1971 a 1977: integração em larga escala, com até 100.000 transistores por chip

1978 a 1991: integração em escala muito grande (vlsi), com até 100 milhões de transistores por chip

1991 até a atualidade: integração ultra-vlsi, com mais de 100 milhões de transistores por chip

Memórias de Semicondutores

Em 1970, uma empresa chamada fairchild desenvolveu pela primeira vez uma memória utilizando a mesma tecnologia utilizada para fabricar os processadores, os transistores. Isso possibilitou que memórias muito menores, mais rápidas e mais baratas fossem desenvolvidas.

E melhor, elas poderiam ser inseridas muito próximas, e até dentro dos processadores, acompanhando sua miniaturização. E foi o que aconteceu.

A medida que a tecnologia foi avançando e produzindo transistores cada vez menores, as memórias também foram encolhendo.

Os processadores tornaram-se cada vez menores e mais velozes, mas infelizmente o avanço não ocorreu também com a velocidade das memórias, mas apenas com o seu tamanho. Isso até hoje é um problema.

Armazenamentos rápidos são muito complexos de fabricar e, por consequência, caros. Isso vem limitando o avanço da velocidade dos computadores, mas sempre os cientistas vêm encontrando alternativas para manter gordon moore e todos nós muito orgulhosos.

A Intel

A intel corporation, ou simplesmente intel, surgiu nos estados unidos em 1968, como uma empresa focada no projeto e fabricação de circuitos integrados. Ela foi fundada por gordon moore (o mesmo da lei de moore) e robert noyce.

Ela era inicialmente uma concorrente da ibm, mas logo se tornaram parceiras. A intel fabricava os processadores e memória, e a ibm fazia a composição deles com outros componentes para montar os computadores.

Em 1971 a intel lança seu primeiro processador, o 4004, que trabalhava com operações e dados de 4 bits. Foi uma revolução, pois todos componentes da cpu estavam num único chip. No ano seguinte eles lançam o 8008, já de 8 bits. Em 1974 é lançado o 8080, primeiro processador de propósito geral. Ou seja, com ela tanto era possível executar aplicações científicas, financeiras, gráficas e jogos. O mesmo princípio dos processadores atuais.

Ele foi substituído pelo 8086 de 16 bit. O próximo foi o 80286 que já era capaz de trabalhar com uma memória de 16mbytes. O 80386 trabalhava com 32 bits e tinha suporte a multi-tarefas, ou seja, era finalmente possível executar mais de uma aplicação simultaneamente.

Depois veio o 80486 com muito mais memória e bem mais rápido, além de um co-processador específico para aplicações matemáticas. A partir do 80286 as pessoas omitiam o 80 ao falar do processador, chamando-o apenas de 286, 386 e 486.

Em seguida veio a geração pentium, focando cada vez mais na execução de tarefas paralelas, adicionando várias unidades de processamento e armazenamento de dados dentro do processador.

Agora os processadores não teriam apenas uma unidade ou uma memória dentro do processador, mas várias delas. Hoje estamos na geração dos processadores multi-núcleos, ou multi-cores, que nada mais são do que vários processadores replicados dentro de um mesmo chip e coordenados por uma unidade única.

A Apple e a Microsoft

Em 1974 Steve Jobs e Steve Wozniak trabalhavam noites a fio para tentar, pela primeira vez, criar um computador que fosse voltado não apenas para empresas, mas também para pessoas. Seria a ideia de um computador pessoal.

Eles compraram todos os componentes necessários para montar um computador, fizeram várias improvisações e inovações, acoplaram uma TV e um teclado. Wozniak, um gênio da eletrônica e programação, desenvolveu o software para controlar o computador e ainda alguns aplicativos, como uma planilha de cálculos e alguns jogos.

Assim que o protótipo ficou pronto, Steve Jobs, exímio negociador e vendedor, colocou o computador na mala de seu carro e foi visitar várias empresas para conseguir algum apoio financeiro para poder fabricar o produto em escalas maiores. Foi até na IBM, mas ouviu deles que o mercado de computadores pessoais não era promissor e o negócio deles era a produção de grandes computadores para empresas.

Assim que conseguiram o primeiro cliente, em 1976, Jobs e Wozniak fundaram a Apple e lançaram o Apple I. Um produto mais maduro e melhor acabado. Jobs sempre gostava de produtos de design diferenciado, que fossem não apenas eficientes, mas bonitos e, principalmente, fáceis de usar. Suas apresentações anuais de lançamento de novos produtos eram sempre aguardadas com grande expectativa e especulações.

A IBM inicialmente também desenvolvia o sistema operacional e os programas que iriam ser executados por suas máquinas. Logo ela percebeu que poderia fazer parcerias com outras empresas e agregar ainda mais valor aos seus produtos. Foi aí que surgiu a Microsoft, liderada pelo seu fundador, Bill Gates, com o seu sistema operacional MS-DOS.

Não demorou muito para que todos os computadores lançados pela IBM trouxessem também o MS-DOS integrado a eles. Depois surgiram as evoluções do MS-DOS, o Windows e suas várias gerações.

A Microsoft se beneficiou bastante dessa parceria, já que todos a grande maioria dos computadores do mundo executavam seu sistema, as pessoas teriam que aprender e se familiarizar com seu sistema operacional.

As empresas de desenvolvimento de aplicativos e jogos tinham que fazê-los compatíveis com o MS-DOS e Windows e foi aí que a Microsoft se tornou uma das líderes do mercado e, por muitos anos, a mais rica empresa do mundo.

Steve Jobs sempre acusou o Bill Gates de ter copiado dele o código principal para o funcionamento do MS-DOS. Gates nunca negou. Eles sempre trocavam acusações e isso gerou muito assunto para a imprensa e fanáticos por tecnologia.

A verdade é que a Microsoft cresceu bastante e a Apple passou por vários apertos. Só no ano 2000, quando Jobs retornou à Apple depois de ter sido expulso da própria empresa que ele fundou, foi que as coisas melhoraram para a Apple.

Eles lançaram produtos em outras linhas que não fossem computadores pessoais, como o iPod para ouvir música e o telefone celular iPhone. A Apple passou então a dominar o mercado de música online com sua loja de músicas, iTunes e o iPhone é o smartphone mais vendido do mundo.

Steve jobs seguia a filosofia não de fazer clientes, mas de criar fãs. E deu certo. Hoje há vários “apple maníacos” que compram seus produtos antes mesmo deles serem apresentados ao público. Nos dias em que esse livro está sendo escrito, a apple ultrapassou a ibm e a microsoft em valor, e é a empresa mais valiosa do mundo.

Recapitulando

Ao final desse capítulo vimos o que é a arquitetura de um computador e porque é tão importante estudá-la. Vimos que o transistor é o dispositivo básico para todo o funcionamento de um computador. Estudar seu funcionamento e sua evolução, é estudar a própria ciência da computação e a eletrônica digital.

Depois de seu surgimento, os computadores foram possíveis e avançaram a medida que eles encolhiam de tamanho e aumentavam de velocidade, consumindo menos energia. Com a evolução dos computadores, cada vez mais rápidos, menores e mais baratos, toda a humanidade avançou na mesma velocidade. No próximo capítulo vamos estudar mais a fundo como os processadores funcionam. Como os programas são executados e o que é feito nos dias de hoje para que eles sejam cada vez mais eficientes.

Atividades

Quais as quatro funções básicas que todos os computadores executam? Dê um exemplo de operação de cada uma delas.

Quais os elementos básicos de um computador e quais as funcionalidades de cada um deles?

Quais as diferenças entre um sinal analógico e um digital? Apresente os pontos fortes e fracos de cada um deles. Na sua opinião, qual dos dois sinais apresentam maior qualidade?

Caracterize o que é uma máquina de von neumann

O que são transistores? Quais as vantagens na concepção de computadores com o surgimento dos transistores?

Por que quanto menores os transistores, mais velozes os computadores? Há desvantagens nessa miniaturização das máquinas? Quais?

O que diz a lei de moore? Em sua opinião, há um limite para esse crescimento? Onde vamos chegar?

Que outras técnicas podem ser utilizadas para aumento do desempenho dos processadores que não pela redução do tamanho dos transistores? Explique cada uma delas.

Unidade central de processamento (cpu)

Objetivos do capítulo

Ao final deste capítulo você deverá ser capaz de:

Identificar os principais componentes de uma cpu

Conhecer o funcionamento de uma cpu

Saber como um programa é executado em ciclos

Definir e explicar o que é uma interrupção e suas consequências

Argumentar sobre aspectos que influenciam no desempenho de uma cpu

Nesse capítulo vamos estudar a parte mais importante de um computador, que é a unidade central de processamento, ou ucp, ou, do inglês, cpu. A cpu é responsável não apenas por executar os programas contidos na memória, mas também de controlar todos os dispositivos de entrada e saída. Seu avanço ao longo dos anos tem permitido que programas fossem executados cada vez mais rapidamente. Hoje temos processadores de vários núcleos capazes de executar várias atividades ao mesmo tempo. São esses processadores e funcionalidades que iremos estudar nesse capítulo.

O que é um programa?

Nesse momento, você não apenas deve saber o que é um programa, como já deve ter até já escrito seus próprios programas e entendido um pouco como o computador funciona. Os programas são sequências finitas de passos que foram definidas por um programador para alcançar um objetivo específico.

Cada passo desse programa é chamado de instrução. Não necessariamente, uma instrução escrita em uma linguagem de alto nível, como c, java, python, por exemplo, é diretamente transformada em uma instrução de máquina e armazenada em memória para execução da cpu. Na verdade, geralmente, uma instrução de uma linguagem de alto nível embute vários comandos e ações a serem executadas pela cpu. Essa é a principal razão da criação dessas linguagens de alto nível.

O programador deve ter o menor trabalho possível ao escrever um programa. Ele deve se preocupar com o problema que está tentando solucionar, e em memorizar dezenas de comandos de uma linguagem de máquina extensa e repleta de detalhes.

Após compilado, a programada de linguagem de alto nível é transformado em um programa apenas com instruções de máquina. Cada instrução de máquina contém apenas uma única operação a ser realizada pela cpu. Para ser executado, esse programa deve ser transferido para a memória principal. No princípio, um operador de máquina, copiada todas instruções para a memória de maneira quase que manual.

Hoje em dia essa operação é realizada pelo sistema operacional (windows, linux etc.). Assim que um usuário clica com o mouse, ou pressiona a tecla enter do teclado solicitando que um determinado programa execute, o sistema operacional copia o programa para a memória e solicita que a cpu o execute.

Não podemos esquecer que a memória do computador apenas armazena números binários. Então, podemos dizer que um programa em linguagem de máquina é formado por instruções em binário. A cada instrução trazida da memória, a cpu lê seu código binário de operação para saber do que se trata, e inicia o processo de execução.

Dependendo da operação, que pode ser de movimentação de dados, uma operação lógica, ou aritmética, ou uma operação de armazenamento de dados, a cpu envia ordens para que os outros dispositivos do computador atuem de forma a completar a operação. Essas ordens são enviadas através de pulsos elétricos passados por fios dentro do computador.

Esses fios são chamados de barramento de controle.

Software X Hardware

O computador é composto por dois elementos, o software e o hardware. Tanto o hardware quando os softwares foram escritos por um programador, ou engenheiro, para se resolver um determinado problema.

O início é sempre o mesmo. O profissional se depara com um problema e projeta uma solução algorítmica para ele. A diferença está na concepção. O hardware é concebido em chip, utilizando transistores interconectados.

Uma vez elaborado, o hardware não pode mais ser modificado. Ele é uma solução rígida (do inglês, hard) para o problema.

Já o software é elaborado para ser armazenado numa memória e ser executado com um processador de propósito geral. Ele é uma solução flexível (do inglês, soft) para o problema, já que o programador pode, a cada momento, modificar seu programa afim de torná-lo cada vez melhor.

Soluções em software são sempre mais lentas do que soluções equivalentes em hardware. Isso porque para executar um programa, cada instrução deve antes ser armazenada em memória, transferidas para a cpu (lembre-se que memórias são muito mais lentas do que cpus) e, só então, ser executada pela cpu. Já as soluções em hardware não utilizam instruções, elas executam as operações diretamente.

Por outro lado, as soluções em software ganham em flexibilidade, já que os programas podem ser facilmente modificados. Já as soluções em hardware, não. Uma vez concebido, um hardware não pode mais ser modificado, ele deve ser descartado para dar lugar a uma versão mais nova. Isso torna projetos em hardware muito mais caros.

Para entender melhor, podemos citar alguns exemplos de implementações em hardware comumente utilizadas. Todas são escolhidas devido ao seu caráter de pouca necessidade de modificação, mas muito demanda por alto desempenho.

Por exemplo, chips de criptografia para celulares (geralmente smartphones), processadores aritméticos para acelerar os cálculos, aceleradores gráficos para gerar gráficos mais rápidos, alguns chips para fazer edições rápidas em fotos, geralmente acoplados às câmeras digitais.

As implementações são feitas em software quando a demanda por desempenho não é tanta, ao mesmo tempo em que as atualizações são frequentes, como os sistemas operacionais, os jogos e aplicativos em geral.

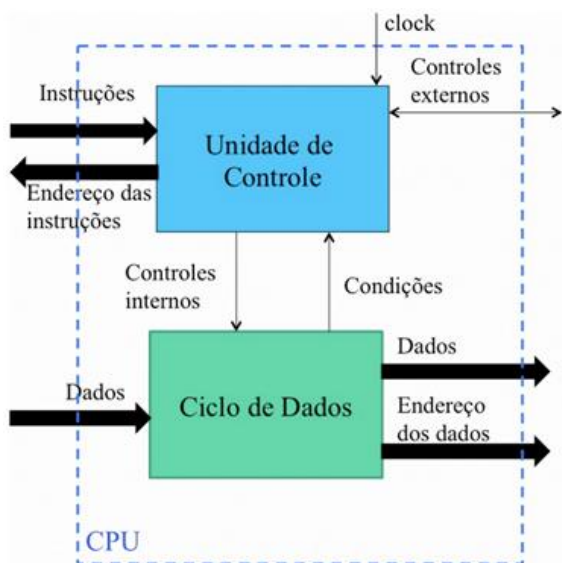
Apesar de não ser tão rápida quanto gostaríamos, a cpu é uma solução muito boa por permitir a execução de, praticamente, qualquer tipo de programa, se tornando uma máquina de propósito geral.

Estrutura de uma cpu

Toda cpu é formada por duas unidades, como podem ser vistas na [fig_cpu]:

Unidade de controle (uc)

Unidade de ciclo de dados (ucd)



A unidade de controle é responsável por receber instruções pelo barramento de instruções. As instruções vêm da memória de acordo com o endereço enviado pela uc para a memória através do barramento de endereço das instruções (à esquerda da uc na figura da cpu).

Já unidade de ciclo de dados, como o próprio nome deixa entender, é responsável por tratar os dados propriamente ditos. A unidade de controle não executa as instruções.

Ela as lê, decodifica e passa os comandos para a ucd determinando como as instruções devem ser executadas e com quais dados. Baseada nesses comandos, a ucd pode ir buscar os dados necessários na memória, executa as devidas operações e envia o resultado de volta para a memória para ser armazenado.

Tudo controlado de acordo com os comandos internos enviados pela unidade de controle, que por sua vez se baseia na instrução decodificada.

Os dados lidos, ou enviados para a memória, são transmitidos através do barramento de dados. Os endereços são enviados para a memória através do barramento de endereço.

Tudo isso é controlado por um sinal síncrono de relógio (clock, do inglês). A cada batida do relógio a unidade sabe que deve executar um passo, passar os dados para quem deve, e se preparar para o próximo passo.

Quanto mais rápido é o relógio mais operações por segundo o processador consegue executar e mais rápido pode se tornar. A velocidade do relógio é medida em frequência, utilizando a unidade herz (abreviatura é hz).

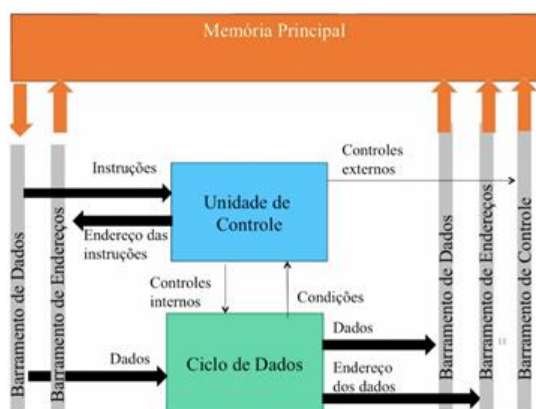
Um herz significa um passo por segundo. Os processadores atuais trabalham na faixa dos poucos ghz (leia-se giga herz), entre 1 ghz e 5 ghz.

Um giga herz significa um bilhão de passos por segundo. É por isso que os computadores são tão incríveis. Eles não executam operações extraordinárias.

Pelo contrário. Executam operações extremamente simples, como somas, subtrações e multiplicações, mas fazem isso numa velocidade incrível.

Os Papéis dos Barramentos e da Memória

Saindo um pouco de dentro da cpu, podemos enxergar os barramentos e a memória principal, como é apresentado na figura de cpu com barramento. Para facilitar a visualização, os barramentos de dados e de endereço são apresentados replicados, tanto do lado esquerdo, quanto do direito da figura.



A comunicação da unidade de controle e da unidade de ciclo de dados é feita sempre com a memória principal através dos barramentos.

Os endereços são transmitidos sempre via barramento de endereços para a memória, sempre de forma unidirecional da cpu para a memória. Quando as instruções são transmitidas da memória para a unidade de controle, elas utilizam o barramento de dados.

Isso porque as instruções são tratadas pela memória como um conteúdo como um outro qualquer. Ela não faz distinção entre dados e instruções.

O mesmo barramento de dados é utilizado pela unidade de ciclo de dados para receber os operandos das operações a serem realizadas e para enviar os resultados de volta para a memória.

Fica claro então a importância da memória principal. Todo e qualquer programa só poderá ser executado a partir dela. Quando você, por exemplo, deseja executar um programa de um pendrive conectado pela usb do computador, ele antes precisa ser copiado para a memória principal. Só então ele será executado.

A memória precisa ser grande o bastante para armazenar a maior quantidade possível de programas, e também precisa ser rápida o suficiente para buscar os dados e enviá-los o mais rapidamente possível à cpu, e também salvá-los no menor tempo possível.

A velocidade das memórias é determinada essencialmente pela tecnologia de transistores utilizada. Essa tecnologia é relacionada ao preço. Quanto mais rápidas, mais caras elas são.

Os Registradores

Os registradores são memórias elaboradas com o mínimo de transistores possível, utilizando o que há de mais moderno em tecnologia de armazenamento.

Elas são as memórias mais rápidas que podem ser construídas e por isso são também as mais caras. Por essa razão, elas aparecem numa quantidade muito pequena em um computador, na casa de alguns kilo bytes. Eles podem ser divididos em dois grupos.

Os registradores de propósito geral, e os de propósito específico. Como o próprio nome diz, os primeiros podem ser utilizados pelos programas para quaisquer objetivos, já os segundos são específicos para algumas tarefas.

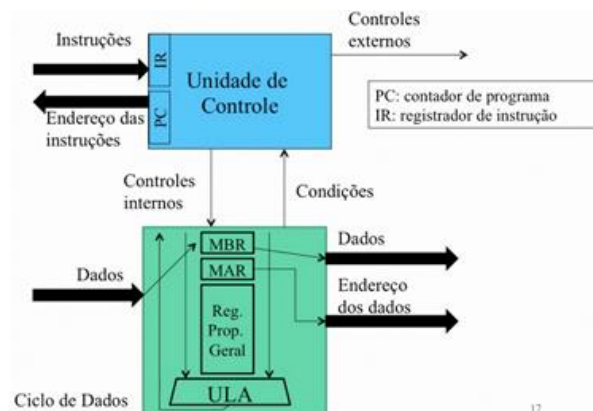
Por exemplo, há um registrador na cpu para controlar se o processador deve continuar em execução, ou entrar em modo de espera por nova ordem. Se esse registrador receber um valor diferente de zero, o processador entrará em modo de espera, até que receba a ordem de modificar esse valor. Na figura os registradores de propósito específico apresentados são:

Program counter (pc): contador de programas

Instruction register (ir): registrador de instrução

Memory address register (mar): registrador de endereço

Memory buffer register (mbr): registrador de dados



O pc contém o endereço de memória que será utilizado para buscar a próxima instrução a ser executada pela cpu. Antes de executar qualquer instrução, a cpu envia o conteúdo de pc para a memória através do barramento de endereço, a memória envia o conteúdo da memória nesse endereço através do barramento de dados.

Esse conteúdo é então armazenado no ir. Já o ir, que recebeu a instrução que veio da memória, tem o objetivo de guardar a instrução e passá-la para a unidade de controle, que é quem vai lê-la e tomar as decisões necessárias para que ela seja executada pela unidade de ciclo de dados.

Por se tratarem do processo de busca de instruções, o pc e o ir ficam instalados na unidade de controle. O pc possui conexão direta com o barramento de endereços, e o ir com o barramento de instruções.

Com relação ao mar e ao mbr, eles possuem funções análogas ao pc e ir, respectivamente, mas referentes a dados e não a instruções. Quando uma operação precisa ser realizada com algum dado que está na memória (e não em um registrador), o endereço desse dado é passado para o mar.

A cpu então passa o conteúdo de mar para a memória através do barramento de endereço, que retornará o conteúdo da memória nesse endereço através do barramento de dados. O conteúdo trazido pela

memória será armazenado em mbr. Só então o dado poderá ser utilizado para o processamento inicialmente planejado.

O mbr e mar possuem, respectivamente, conexões diretas com os barramentos de dados e de endereços. Ambos são situados na unidade de ciclo de dados, por serem utilizados nas fases de processamento das instruções.

O tamanho e quantidade dos registradores de uma cpu é uma das principais decisões de projeto. Se forem grandes demais, ou em quantidade maior do que a necessária, podem resultar em desperdício e aumento desnecessário no preço do processador.

Já se forem pequenos, ou em pouca quantidade, com certeza vão tornar o computador muito mais lento do que o desejado. Encontrar o tamanho e quantidade ideais é trabalhoso e geralmente é feito através de simuladores e de muito testes e anos de experiência.

Os registradores de propósito geral são utilizados para guardar as variáveis dos programas. Como eles estão presentes em quantidades muito pequenas, são poucas as variáveis que ficam armazenadas em registradores. As demais ficam na memória principal.

Quando uma operação precisa ser realizada e seus dados estão nos registradores de propósito geral, a cpu não precisa buscá-los na memória e o processamento torna-se muito mais rápido.

A unidade lógica e aritmética, ou ula, se assemelha muito com uma calculadora convencional. Ela executa operações lógicas e aritméticas. As ulas modernas executam operações tanto com inteiros, como com números reais.

A ula recebe como entrada dois diferentes dados que são trazidos para ela dos registradores (de propósito geral, ou específicos). Quem decide que registradores passarão seus dados para a ula é a unidade de controle baseada instrução que está sendo executada. A unidade de controle também envia para a ula qual operação será realizada (soma, multiplicação, divisão, and, or etc.).

Assim que isso é feito, a ula executa a operação e gera um resultado na sua saída. Esse resultado também é passado para um registrador escolhido pela unidade de controle, baseando-se na instrução em execução.

Unidade de Controle (Uc)

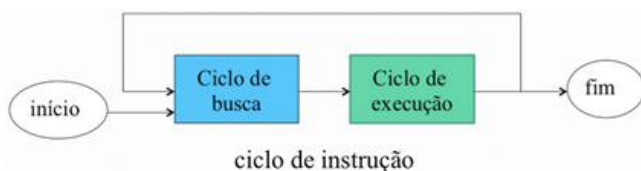
A unidade de controle, ao receber a instrução que está armazenada em ir, a decodifica e envia os sinais de controle para onde for necessário. Decodificar nada mais é do que ler um código em binário e interpretar a operação relativa a esse código.

Dependendo da operação, os sinais de controle podem ser internos, por exemplo, para a ula executar uma soma, ou para o conteúdo de um registrador ser transferido para a ula. Ou pode ser externo, para um dispositivo de entrada e saída, por exemplo, ou mesmo para a memória principal. Tudo isso depende da instrução a ser executada.

Na próxima seção será apresentada a execução de instruções em mais detalhes, o que facilitará o entendimento do funcionamento das cpus.

Ciclo de Instrução

Toda cpu trabalha em dois ciclos principais, o ciclo de busca e o ciclo de execução, como pode ser visto na figura, assim que o computador é iniciado, a cpu entra no ciclo de busca, em seguida passa para o ciclo de execução e depois volta para o ciclo de busca. Ela continua nesse processo até que precise ser desligada, saindo do ciclo de execução para o estado final.



Durante o ciclo de busca, é a unidade de controle que atua. Uma nova instrução é busca da memória para que possa ser decodificada. Nessa fase os registradores pc e ir são utilizados, como apresentados na seção anterior.

O pc é logo lido para se saber que instrução será executada, essa instrução é trazida para o ir e, finalmente, é decodificada pela unidade de controle.

Assim que esse processo termina, caso a instrução não diga respeito à um laço, ou à uma repetição, o conteúdo de pc é incrementado. Ou seja, pc recebe $pc + 1$.

Assim, no próximo ciclo de busca a instrução do endereço seguinte será carregada da memória e executada. Esse comportamento garante a característica de execução sequencial dos programas.

No passo seguinte a cpu entra em ciclo de execução. Nessa etapa atua a unidade de ciclo de dados. Agora a unidade de controle já sabe exatamente que operação será executada, com quais dados e o que fazer com o resultado.

Essa informação é passada para a ula e os registradores envolvidos. Durante o ciclo de execução há cinco possíveis tipos de operação que podem ser executadas:

Processador e Memória

Trata simplesmente da transferência de dados entre cpu e memória principal;

Processador e entrada e saída

Diz respeito à transferência de dados entre a cpu e um dispositivo de entrada e saída, como teclado, mouse, monitor, rede, impressora etc.;

Processamento de Dados

São operações simplesmente de processamento dos dados, como operação aritmética ou lógica sobre os registradores da cpu;

Controle

São instruções que servem para controlar os dispositivos do computador, como para ligar um periférico, iniciar uma operação do disco rígido, ou transferir um dado que acabou de chegar pela internet para a memória principal;

Operações compostas

São operações que combinam uma ou mais intrusões das outras em uma mesma operação.

Busca de Dados

Em operações entre processador e memória, é necessário que dados sejam trazidos da memória para servirem de entrada para a ula, e/ou o resultado seja levado para armazenamento na memória no final da execução.

Para isso acontecer, é executada uma busca de dados. Isso é determinado durante a decodificação da instrução, no ciclo de busca de instrução. Isso acontece quando um dos parâmetros de uma operação aritmética é um endereço de memória, e não um valor diretamente, nem um registrador.

Para isso, parte do conteúdo de ir é transferido para o mar. Essa parte é justamente o endereço do parâmetro da instrução. Em seguida a unidade de controle requisita à memória uma leitura.

Assim, o endereço, agora em mar, é passado para a memória e o conteúdo lido da memória é passado para o mbr. Agora o conteúdo é transferido para a ula para que a operação seja executada (lógica ou aritmética).

Se a instrução tiver dois ou mais parâmetros de memória, serão necessárias outras buscas de dados. Como a memória é sempre mais lenta do que a cpu, instruções que necessitam buscas de dados são muito mais lentas do que instruções de processamento de dados.

Perceba que cada instrução pode exigir mais tempo de execução do que outras. Isso depende de quantos acessos à memória ela exigirá.

Quanto mais acessos à memória, mais lenta a instrução. O ideal é sempre usar registradores. Mas nem sempre é possível utilizar registradores. Eles estão sempre em poucas quantidades e em menores tamanhos.

Principalmente por serem caros. O que os computadores sempre tentam fazer é passar os dados da memória para os registradores assim que puderem, para que as próximas instruções sejam aceleradas.

Interrupções

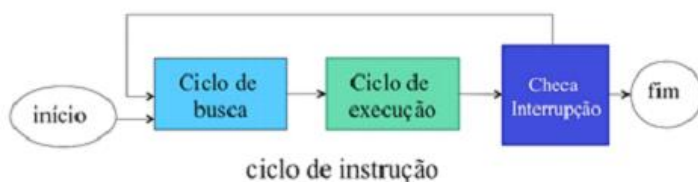
Além do ciclo básico de instrução apresentado anteriormente, a cpu pode ainda executar outro tipo de tarefa. Ela diz respeito ao processamento de pedidos oriundos dos dispositivos de entrada e saída.

Como o ciclo de instrução da cpu que vimos até o momento é fechado, ou seja, a cpu sempre fica em estado de repetição até que seja desligada, ela não pode atender a nenhum evento externo que não seja a execução de um programa.

Por exemplo, quando um usuário pressiona uma tecla do teclado, ou faz um movimento com o mouse, ou mesmo, quando uma mensagem chega pela internet através da placa de rede.

O que a cpu deve fazer? Se ela estiver em um ciclo de instrução fechado como mostrado anteriormente, nada. Ela precisa parar o que está fazendo para atender ao evento ocorrido e, só então, voltar ao ciclo de instruções. Esse processo de parar o ciclo de instrução para atender a um evento externo é chamado de interrupção.

O ciclo de instrução pode agora ser visto modificado para atender às interrupções. Todas interrupções são recebidas e armazenadas internamente por um dispositivo chamado gerenciador de interrupções. Esse dispositivo é um chip, semelhante à uma cpu, mas bem mais simples.



Na maioria dos computadores eles vêm soldados na placa-mãe, mas podem também vir dentro do chip da cpu. Toda interrupção possui um código de identificação.

Sempre que uma nova interrupção chega nesse gerenciador, ele armazena esse código em sua memória e manda um sinal para cpu através do barramento e controle.

Durante seu ciclo de instrução, sempre que uma instrução é executada, antes de voltar para o ciclo de busca, a cpu chega de algum sinal de interrupção foi enviado pelo gerenciador de interrupção.

Quando não há uma interrupção, a execução volta ao ciclo de busca e o programa em execução continua a ser executado. Mas se houver uma interrupção, a cpu agora vai parar a execução do programa atual para atender a interrupção. Por exemplo, vamos supor que o usuário pressionou uma tecla do teclado. O código armazenado pelo gerenciador de interrupção indica que a interrupção veio do teclado.

A cpu para sua execução do programa anterior e vai iniciar a execução de um programa especial, o tratador de interrupção. O código do dispositivo (aqui seria o teclado) serve para a cpu saber o endereço do tratador de interrupção ela vai buscar da memória. Então, ao sair do checkagem de interrupção, a cpu muda o endereço do pc para o endereço do tratador de instrução.

Assim, no ciclo de busca a próxima instrução a ser trazida da memória e posteriormente executada será a do tratador do teclado.

Cada tipo de interrupção precisa de um tratamento específico a ser feito. No caso do teclado, o tratador vai checar que tecla foi pressionada.

Isso é feito através de uma leitura à memória do teclado (sim, todos dispositivos possuem uma pequena memória) para saber que tecla foi pressionada. Dependendo da tecla, uma operação diferente será executada. Geralmente, a cpu adiciona o código da tecla pressionada num endereço específico de memória. Cada programa, lendo essa informação, tomará sua própria decisão sobre o que deve ser feito.

O que acontece é que apenas o programa ativo no momento, vai ler esse conteúdo, executar a ação da tecla e limpar essa área de memória. Se o programa for um editor de texto, por exemplo, o código pode representar escrever a letra pressionada na posição atual do cursor dentro do texto.

Quando esse processo encerra, o tratamento é encerrado, e a cpu deve voltar a execução do programa que havia sido interrompido. Isso só é possível porque, antes de passar a execução do tratador de interrupção, a cpu salva os conteúdos de todos os registradores da cpu (inclusive o pc e o ir).

Então, antes de devolver a execução para o programa, cpu restaura todos os valores dos registradores antes salvos. Dessa forma, o programa retoma exatamente do ponto em que parou.

As interrupções também ocorrem se o próprio programa em execução executar uma operação ilegal. Isso é feito para evitar que a cpu entre em erro. Por exemplo, se um programa tentar acessar uma área da memória que é proibida para ele, como a área de outro programa ou do sistema operacional.

Nesse caso, o programa é interrompido e não volta mais a executar, ele é finalizado e a execução é devolvida ao sistema operacional.

Algo semelhante ocorre em caso de defeitos em alguns dispositivos. Por exemplo, se um programa estiver lendo um arquivo que está em um pendrive, e esse pendrive é removido subitamente, uma interrupção é lançada e o programa é encerrado, já que ele não faz mais sentido estar em execução.

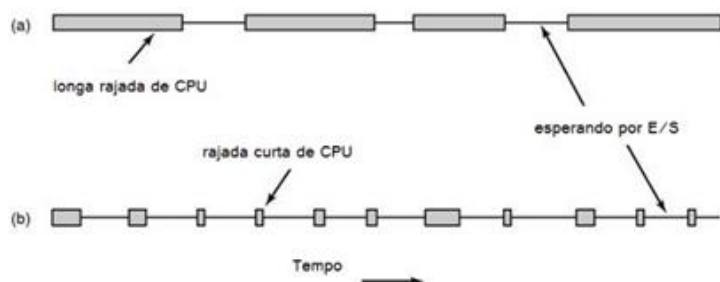
Sobre o Desempenho

É possível agora perceber que o desempenho das cpus depende de muitos outros fatores além da velocidade do seu clock.

O computador precisa ter memórias rápidas para reduzir o tempo dos ciclos de busca, precisam de mais registradores para usar menos a memória e também que poucas interrupções ocorram.

Cada vez que uma interrupção ocorre, o programa deve ser interrompido e a chamada deve ser atendida. Isso vai atrasar demais o tempo de execução dos programas, dando a impressão de baixo desempenho.

Basicamente, há dois tipos programas, os orientados à cpu e os orientados a entrada e saída. Na figura de entrada e saída o comportamento dos primeiros é mostrado na parte a) e o dos segundos na parte b).



Quando um programa é orientado à cpu, há momentos longos de processamento de cpu e curtos momentos de espera por um evento de entrada e saída. É o exemplo de programas que fazem muitos

cálculos matemáticos, como ferramentas de simulação, projetos de engenharia, computação gráfica e planilhas de cálculos.

Inicialmente os dados de entrada são passados por um dispositivo de entrada, há longos momentos de cálculos e depois os resultados são passados para um dispositivo de entrada e saída.

Já nos programas orientados à entrada e saída (b), são aqueles chamados também de interativos. Há muitos momentos de interação e uso de dispositivos de entrada e saída, e poucos momentos de uso de cpu. Como é o caso de programas que usam muito de mouse e teclado, como os jogos e a própria navegação na internet.

O que temos que ter em mente é que o desempenho de um computador está muito ligado ao perfil de cada usuário. Os sistemas operacionais são os responsáveis por escolher que tarefa colocar para executar a cada momento e por quanto tempo ela deve executar até que uma nova tarefa entre em execução. Assim, o papel do sistema operacional também é fundamental e determinante no desempenho do sistema. O que ele tenta fazer no máximo que pode, é tentar ocupar os tempos de espera de um programa com a execução de outro.

Introdução de Microcomputadores

A indústria dos computadores pessoais começou em 1971 com a introdução do primeiro microprocessador, o intel 4004.

Mas a indústria decolou de verdade logo após a edição de janeiro de 1975 da revista popular electronics, da ziff-davis, que anunciava o "sucesso do projeto" altair 8800, da mits, citado pela revista como "o primeiro kit para minicomputador do mundo a concorrer com os modelos comerciais".

Pelos padrões atuais, este kit inicial desenvolvido por ed roberts, que liderava a mits, uma pequena companhia de componentes eletrônicos de albuquerque, novo méxico, era bastante limitado. Ele se baseava no microprocessador 8080 da intel e tinha apenas 256 bytes de memória. Com um preço bem acessível, us\$7, o altair foi o primeiro computador pessoal disponível em grande escala para o público em geral. Ele atraiu centenas de pedidos por parte de entusiastas da eletrônica.

Um dos que notaram este evento embrionário foi um jovem programador da honeywell chamado paul allen, que mostrou o artigo da popular electronics a um velho amigo, um calouro da universidade de harvard chamado bill gates. A dupla rapidamente uniu suas forças para elaborar uma versão do basic para o altair.

Em pouco tempo, allen foi trabalhar para a mits como diretor de software e, logo em seguida, gates deixou harvard para juntar-se a allen em albuquerque e dar início a uma empresa que, mais tarde, seria conhecida como microsoft. (outro ex-funcionário da mits, david bunnell, publicaria, mais tarde, uma variedade de revistas especializadas em computação, dentre elas a pc magazine.)

Com a introdução do altair, a indústria de computadores pessoais decolou. O ano de 1977 assistiu a uma explosão de interesse pelos computadores pessoais e à introdução de uma longa sucessão de máquinas - commodore pet, radio shack trs-80 e - a mais importante de todas - a apple ii, de steve wozniak e steve jobs.

O apple ii desenvolveu rapidamente seu próprio padrão, com o inestimável auxílio do projeto de wozniak, em 1978, de uma econômica unidade de disco flexível e - o mais importante - do visicalc, a primeira planilha eletrônica, elaborado por dan bricklin e bob frankston. Com a introdução do visicalc, os homens de negócios encontraram, repentinamente, uma razão para utilizar os computadores pessoais. Este já não era mais um mundo de lazer.

O restante da década viu passar vários projetos diferentes, enquanto uma companhia após outra tentava definir uma combinação exclusiva de potência, preço, desempenho e recursos.

As máquinas introduzidas neste período abrangiam desde as ofertas para usuários domésticos e aficionados - como o vic-20 e o 64, da commodore, a série 400, da atari, e o ti-99, da texas instruments - até os dispositivos mais orientados à área comercial, como uma série de máquinas da tandym/radio shack e diversos projetos que executavam o sistema operacional cp/m, da digital research, elaborado pelo pioneiro da computação pessoal, gary kidall.

Devido ao rápido crescimento do mercado e ao fato de que a compatibilidade descendente não significava muito no início, o período foi marcado por uma criatividade em hardware jamais vista. Obviamente, o ramo do software também começou a crescer, com a rápida aparição de uma variedade de linguagens de programação, jogos e até mesmo de aplicativos comerciais, como o popular processador de textos wordstar.

Em pouco tempo, ninguém mais via os computadores pessoais como brinquedos ou hobby, mas sim como dispositivos de produtividade pessoal com visíveis aplicações comerciais. A era do computador pessoal estava estabelecida, de uma vez por todas.

E a ibm, que há muito tempo dominava a área dos computadores de grande porte, queria a sua fatia deste bolo.

A ibm de 1980, muito mais do que a ibm atual, não era uma companhia acostumada com os mercados de mudanças rápidas e com as vendas ao consumidor final.

Ela vendia máquinas comerciais - principalmente computadores e máquinas de escrever - a grandes empresas, utilizando sua própria tecnologia e apoiando-se excessivamente num bem estruturado sistema de vendas e prestação de serviços às grandes contas.

O ramo dos pcs precisava de algo diferente. Este novo mercado estava mudando com grande velocidade e um recém-chegado teria que se movimentar depressa.

Além disto, teria que se dividir entre os usuários individuais e as empresas, mesmo que a meta principal fosse continuar a vender computadores comerciais. Isto foi o que disse william c. Lowe, diretor de laboratório da unidade de sistemas de nível de entrada da ibm em boca raton, flórida, ao comitê de gerenciamento corporativo da ibm, o que incluía o presidente da ibm, john open, em julho de 1980.

Lowe disse ao comitê que a ibm precisava construir um computador pessoal e que havia um espaço no mercado ainda não canalizado pela apple e outras empresas.

Contudo, disse ele ao comitê, ele não poderia ser construído dentro da cultura padrão da ibm daquela época. Sendo assim, eles lhe deram a liberdade de recrutar 12 engenheiros para formar uma força-tarefa, chamada projeto chess, e construir um protótipo de computador.

No mês seguinte, a força-tarefa de lowe já tinha várias reuniões com outros representantes da jovem indústria e tomou algumas decisões importantes que, posteriormente, viriam a afetar a arena do pc pelos próximos anos.

Uma destas decisões foi a de comercializar o computador pessoal da ibm através de lojas de varejo, além de oferecê-lo por meio da equipe de vendas comissionada da própria ibm. Mas talvez a decisão mais importante da companhia tenha sido a de utilizar uma "arquitetura aberta": selecionar os componentes básicos e o sistema operacional de fontes externas à ibm.

Esta era uma grande mudança para a ibm que, até este ponto, projetava todos os principais componentes de suas máquinas.

Em agosto, lowe e mais dois engenheiros, bill sydnes e lew eggebrecht, fizeram a demonstração de um protótipo ao comitê de gerenciamento corporativo, que aprovou o plano básico e deu ao projeto chess o ok para a criação de um computador pessoal chamado acorn.

Para liderar o grupo que o construiria, lowe procurou philip d. "don" estridge, outro antigo funcionário da ibm que trabalhava no laboratório de boca raton. Estridge recrutou uma equipe que incluía sydnes, líder da engenharia, dan wilkie, responsável pela manufatura, e h. L. "sparky" sparks, para a liderança das vendas.

Uma das primeiras decisões a serem tomadas era a escolha do processador que alimentaria o pc. A força-tarefa havia decidido que queria um computador de 16 bits, já que ele seria mais potente e mais fácil de programar do que as máquinas de oito bits existentes.

A intel havia anunciado recentemente o 8086 de 16 bits, mas sydnes disse, mais tarde, que a ibm teve receio de que o 8086 fosse potente demais e concorresse demais com outros itens da ibm.

Assim, eles optaram pelo 8088, uma versão do chip com barramento de oito bits e estrutura interna de 16 bits. Esta tecnologia de oito bits oferecia o benefício adicional de trabalhar com as placas de expansão oito bits existentes e com dispositivos oito bits relativamente baratos, como os chips controladores, que poderiam, assim, ser incorporados de maneira simples e barata à nova máquina.

Outra decisão importante era o software. Em julho, membros da força-tarefa fizeram uma visita à digital research para solicitar à empresa que ela portasse seu sistema operacional cp/m para a arquitetura 8086.

Diz a lenda que seu fundador, gary kildall, estava pilotando seu avião naquela ocasião. Seja qual for a razão, a esposa de kildall, dorothy, e os advogados da dr não assinaram o contrato de exclusividade apresentado pela ibm. Assim, a equipe da ibm foi embora, seguindo ao norte, até seattle, para reunir-se com a microsoft, de quem esperavam obter uma versão do basic.

Os executivos da microsott assinaram um contrato com a ibm para o fornecimento do basic e, logo, bill gates e a companhia estavam discutindo não só o basic como também um sistema operacional.

Imediatamente após, a microsoft adquiriu um sistema operacional 8086 que atendia por diversos nomes, incluindo "quirk and dirty dos", ou qdos, elaborado por tim patterson, de uma companhia chamada seattle computer products. A microsoft incrementou este sistema operacional, licenciando-o para a ibm, que o comercializava como pc-dos.

Seguiram-se, então, meses febris de união de hardware e software, até que, numa quarta-feira, 12 de agosto de 1981, quase um ano após o ok dado ao projeto chess, a ibm apresentou o ibm personal computer. Comercializado inicialmente pelas lojas computerland e nas centrais comerciais da sears, aquele primeiro pc - com uma cpu 8088, 64kb de ram e uma unidade de disco flexível de 160 kb de face simples - tinha um preço de tabela de us.880.

Arquitetura de John Von Neumann

Nos primórdios da computação não exista o conceito de programa armazenado, isto significa que os programas eram desenvolvidos, executados e apresentavam os resultados imediatamente.

O conceito de programa armazenado surgiu da necessidade de armazenar programas em um computador, pois, até então, ainda não havia formas de armazenar programas em um computador. John von neumann e outros pesquisadores descobriram que, utilizando dispositivos de memória em formas de linha de retardo de mercúrio, poderiam armazenar instruções de programas.

Pra nós hoje esse conceito de programa armazenado já está muito bem edificado. Nós codificamos, desenvolvemos software em diversas linguagens de programação, armazenamos nosso código fonte em um hd externo, ou no hd interno, ou na nuvem se assim desejarmos.

O código executável desses softwares compilado, que está armazenado em algum desses dispositivos, sempre passa pela memória principal, depois pela memória cache e então é executado na cpu.

O conceito de programa armazenado surgiu durante o projeto edvac e john von neumann acabou recebendo a maior parte dos créditos pela definição, mas a verdade é que várias pessoas participaram da construção.

John von neumann ganhou mais destaque que os outros por ter escrito um relatório sobre o edvac, com 101 páginas, descrevendo formalmente o conceito, criando assim uma espécie de guia para a construção de computadores.

A Arquitetura de John Von Neumann

A proposta inicial de um computador de programa armazenado, relatada pelo próprio von neumann, era dividida em cinco partes:

Central aritmética;

Central de controle;

Memória;

Gravação;

Entrada e saída.

Primeira parte específica - central arithmetic (ca): o dispositivo deve realizar as operações elementares da aritmética mais frequentemente, e por este motivo deve ter unidades especializadas apenas para essas operações.

Segunda parte específica - control center (cc): a sequenciação apropriada das operações pode ser executada, de forma mais eficiente, por um controle central.

Terceira parte específica:

Memória (m): qualquer dispositivo que tiver que executar longas e complicadas sequências de operações precisa ter uma memória considerável.

Recording (r): deve ser capaz de manter contato com a entrada e a saída, que são como os neurônios correspondentes sensoriais e motores do cérebro humano.

Quarta parte específica - input (i): é necessário ter unidades para transferência de informações de r para m.

Quinta parte específica - output (o): é necessário ter unidades para transferência de informações de m para r.

Características

A arquitetura é composta por três grandes sistemas de hardware: sistema central de processamento, sistema de memória e sistema de entrada/saída. A arquitetura tem a capacidade de executar instruções sequencialmente e possui apenas um caminho de dados entre a ucp e a memória principal, o que é chamado de gargalo de von neumann.

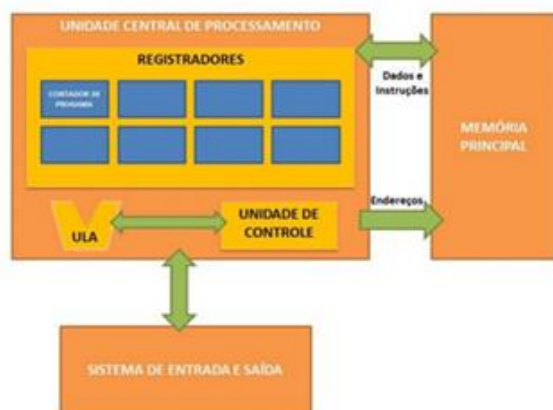
Atualmente, é claro, isso não existe mais e sofreu uma grande evolução.

Unidade central de processamento (ucp) ou central processing unit (cpu): a cpu deve ser composta por uma unidade de controle (uc), uma unidade lógica aritmética (ula), vários registradores (memórias internas da cpu) e um contador de programa (pc);

Sistema de memória principal;

Sistema de entrada/saída.

Vamos ver agora uma imagem que representa o conceito da arquitetura de john von neumann. Observe a figura 1:



A figura 1 apresenta a arquitetura de john von neumann representando os três grandes sistemas de hardware. A seguir descrevo as funções dos componentes ilustrados na figura 1.

Função da cpu: a unidade central de processamento tem como função executar programas que estão armazenados na memória principal, buscar as instruções desses programas, examinar essas instruções e executar as instruções uma após a outra (sequência).

Função da uc: a unidade de controle tem como função buscar instruções na memória principal e determinar o tipo dessas instruções.

Função da ula: a unidade lógica aritmética tem como função efetuar operações aritméticas e efetuar operações booleanas (e, ou, not, etc);

Função dos registradores: os registradores da unidade central de processamento são memórias de altíssima velocidade que armazenam resultados temporários. Alguns registradores têm uma função e um tamanho específico (em bits e/ou bytes) e são lidos/escritos em alta velocidade pois são internos à cpu.

A cpu não consegue manter todos os valores manipulados por um programa apenas em registradores, por isso necessita de uma memória para o armazenamento das informações.

Função do pc: a função do contador de programa, ou programm counter, é o de indicar a próxima instrução a ser buscada para execução pela cpu. Esse é um exemplo de registrador com função específica.

Função do ir: a função do registrador de instrução é o de conter (armazenar) a instrução que está sendo executada no momento. Este é um outro exemplo de registrador com função específica.

O Ciclo de Execução

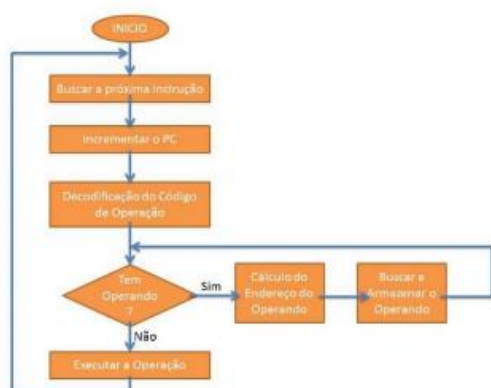
O ciclo de execução diz respeito à forma com que uma instrução é executada pela cpu, os caminhos que ela deve fazer e as unidades funcionais pelas quais ela passará, ou seja, descreve como o hardware trabalha com as instruções que serão executadas.

O ciclo de execução de von neumann é simples e normalmente é citado como busca-decodificação-execução, pois é exatamente isto o que ocorre: busca uma instrução, a decodifica e a executa.

Vamos ver um algoritmo passo a passo que ilustra a execução de uma instrução aritmética (que faz parte de um programa executável) em máquinas von neumann:

1. A unidade de controle busca a próxima instrução do programa na memória principal;
2. O contador de programa é usado pela unidade de controle para determinar onde a instrução está localizada;
3. A instrução é decodificada para uma linguagem que a unidade lógica aritmética possa entender;
4. Os operandos de dados requeridos para executar a instrução são carregados da memória e colocados em registradores;
5. A unidade lógica aritmética executa a instrução e coloca os resultados em registradores ou na memória.

Isso pode ser representado por um algoritmo que é ilustrado na figura 2:



O algoritmo começa buscando, na memória principal, a próxima instrução do programa que será executada. A unidade de controle é responsável por isso. O contador de programa deve ser incrementado para que a próxima instrução a ser executada seja encontrada.

Em seguida a instrução deve ser decodificada, identificando primeiramente o seu código. Se a instrução tiver operandos, como em uma operação aritmética, então, é necessário buscar esses operandos (valores).

Para isso é preciso calcular o endereço do operando, para saber onde ele se encontra na memória e, somente depois pegar esse valor e armazenar em registradores para manipulação.

Quando os operandos terminarem de ser buscados e armazenados, somente então a operação é executada pela unidade lógica aritmética. Se a instrução não tiver operandos, a instrução simplesmente é executada logo após a avaliação da condição.

A arquitetura de John von Neumann é referência nos estudos sobre microprocessadores e arquitetura. Por mais que esse tipo de sistema computacional não seja mais utilizado, foi ele o precursor de tudo o que conhecemos hoje.

Para podermos avançar e evoluir, é muito importante conhecer o passado, o que nos levou a estar no ponto onde estamos e, o que podemos fazer para melhorar ainda mais.

Nos primórdios da informática, nas décadas de 50, 60 e 70, vários fabricantes diferentes disputavam o mercado. Cada um desenvolvia seus próprios microcomputadores, que eram incompatíveis entre si, tanto em nível de hardware, quanto em nível de software.

Apesar de executarem as mesmas operações básicas, praticamente tudo era diferente: os componentes de um não serviam em outro, os programas eram incompatíveis, e até mesmo as linguagens de programação eram diferentes. Porém, com a popularização dos microcomputadores era inevitável uma padronização.

No início da década de 80 tínhamos basicamente apenas duas arquiteturas, ou “famílias” de computadores pessoais diferentes: o PC, desenvolvido pela IBM, e o Macintosh, desenvolvido pela Apple. Como era mais barato, o PC tornou-se popular, ficando o uso dos Macintosh restrito a nichos nos quais suas características peculiares o tornam mais atraente, como a edição de imagens ou sons e editoração eletrônica.

Como os PC possuem uma arquitetura aberta, ou seja, a possibilidade de vários fabricantes diferentes desenvolverem seus próprios componentes e padrões tem-se uma lista enorme de componentes compatíveis entre si. Pode-se escolher entre várias marcas e modelos, os componentes que melhor atendam as necessidades e montar a própria configuração.

Também é possível melhorar posteriormente o micro montado, por meio de upgrades, trocando alguns componentes para melhorar seu desempenho.

Mesmo micros de grife: IBM, Compaq, Dell, etc. Também são montados, já que quase todos os seus componentes são comprados de outros fabricantes.

Têm-se, por exemplo, um processador da Intel, um disco rígido da Quantum, uma placa-mãe da ASUS, memórias da Kingston, CD-ROM e drive de disquetes da Mitsumi, um monitor da LG, e por aí vai.
