FUNDAMENTOS DA COMPUTAÇÃO

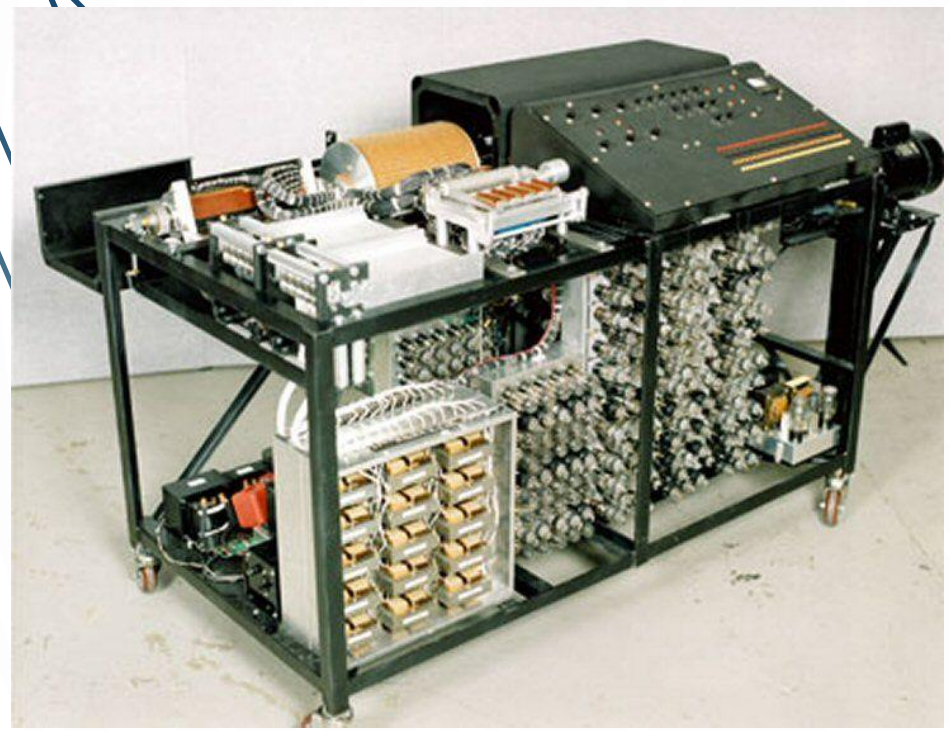
PROF. JOSENALDE OLIVEIRA

josenalde.oliveira@ufrn.br

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - UFRN

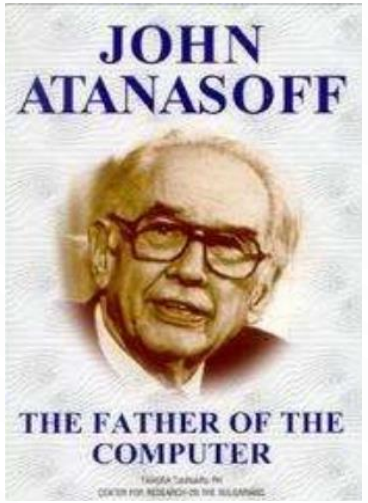
- Historicamente, as instruções (programa) era externo à máquina, sendo introduzido por exemplo por cartões ou fita perfurada, como no Mark I de Harvard (1937-1944, Howard Aiken) e a saída em cartões. Dimensões: 2,5 m x 16m, 4.500 kg de peso, 850 km de fios, 3.500 relés. Funcionou até 1959.
- Automatizar computação para que as operações aritméticas ocorressem em sequência controlada (Calculadora Automática de Sequência Controlada da IBM). Um grupo de botões permitia introduzir dados como números DEC. Tinha 72 "acumuladores", que eram contrarrodas eletromagnéticas; quando enviado ao acumulador o número era somado ao que já estivesse lá. Os acumuladores também faziam subtrações com o método dos complementos como nas máquinas mecânicas. Haviam unidades para multiplicar, dividir, contar e, mais importante, controlar a sequência das operações.

Mark I – parte do ASCC
Parceria Harvard University & IBM



Atanasoff 1903-1995
Berry Computer (ABC)
Considerado primeiro computador digital?

Clifford Berry (1918-1963) aluno de Atanasoff



- John V. Atanasoff, Univ. Estadual de Iowa: máquina compacta (tamanho de uma escrivaninha grande), a base de válvulas para a lógica, memória em forma de tambor, e armazenamento com capacitores, funcionava bem para equações simultâneas com muitas variáveis
- John Mauchly trabalhava no ENIAC e viu na máquina de Atanasoff o potencial da aritmética binária em conjunto com a eletrônica, isto gerou disputas jurídicas sobre a origem da ideia.
- Insights de Atanasoff: *eletricidade e eletrônica – não as peças mecânicas móveis, formam a base lógica da máquina; o "vai um" não usaria um mecanismo de engrenagens e catracas, mas válvulas.*
- *Números binários e não decimais e cálculos por lógica digital, não medições como num aparelho analógico*
- *Condensadores (capacitores) para o armazenamento elétrico de dados (a memória do computador) e "refrescamento" automático da memória para regenerar a carga que se reduz no capacitor*

- Processos sobre quem inventou o computador e tinha direito aos royalties se estenderam pelo século XX. Várias patentes referentes ao ENIAC no fim da segunda guerra pertenciam à Sperry Rand Corporation. Esta processou IBM e Berry. Quando as patentes do ENIAC saíram em 1964, a Sperry ainda quis processar GE, Honeywell e NCR. A Honeywell depois processou a Sperry informando (decisão saiu em 1973 segundo wikiPedia) que as patentes ENIAC estavam erradas.
- Em 1971 o juiz informou que as partes patenteadas, como a tecnologia digital eletrônica e a memória com auto refresh não tinham sido inventadas pela Sperry Rand nem por John Mauchly, mas por Atanasoff, e Mauchly partira delas. Talvez por isso a Sperry Rand não seja mais conhecida

- O ENIAC (Integrador e Computador Numérico Eletrônico): a inovação dessa máquina era a capacidade de fazer, numérica e eletronicamente, o que antes era feito por calculistas humanos usando analisadores diferenciais. Nele, dados e programas era armazenados em locais diferentes. Os programas eram armazenados diretamente nos circuitos eletrônicos e a sua troca realizada através de reconfiguração desses circuitos.
- Era maior, muito maior do que a máquina eletrônica de computação de Atanasoff. O segredo de sua velocidade (*333 multiplicações por segundo*) era um mecanismo de relógio/contador, e a capacidade de programação lhe dava uma versatilidade sem precedentes. Programar o ENIAC envolvia um emaranhado de cabos e conexões cruzadas, e os números eram inseridos por meio de discos. Cada problema exigia dias e não horas de configuração. A arquitetura do ENIAC e sua capacidade limitada de memória mostraram que o avanço era por outro caminho.

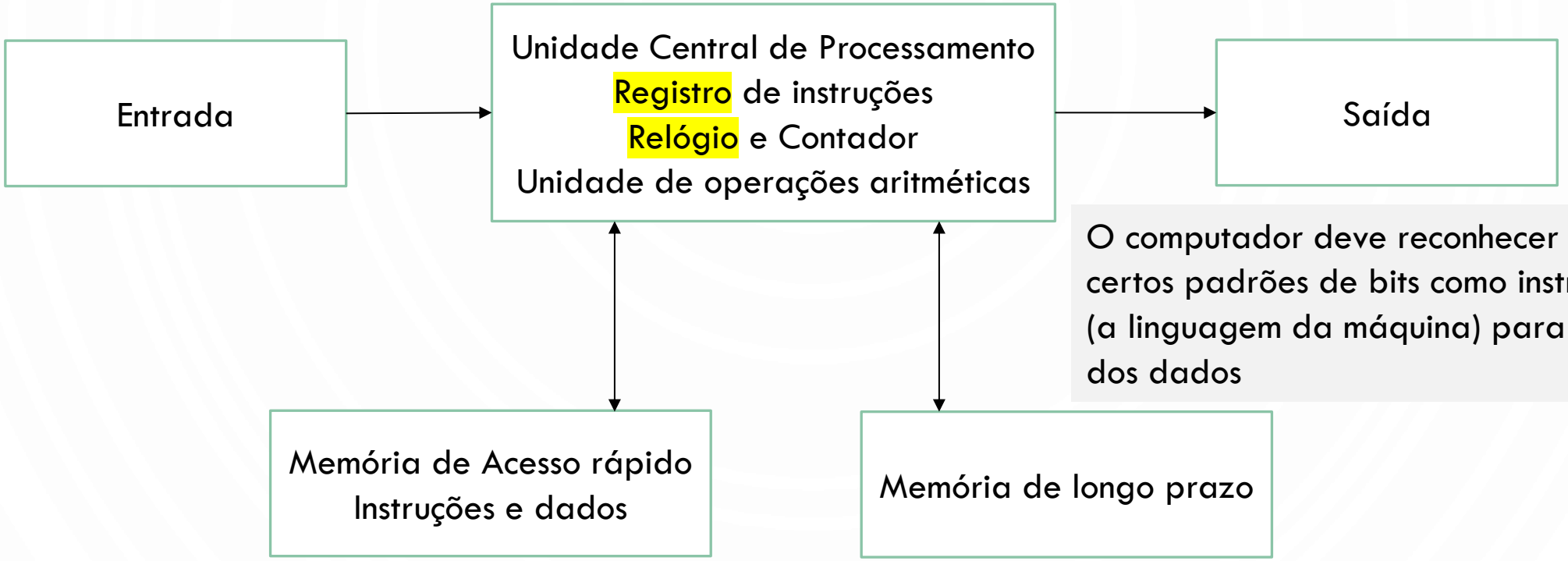
- John von Neumann encontra J. Eckert engenheiro chefe do ENIAC
- Trabalhou com Alan Turing na década de 1930, viu que a solução era o programa armazenado. Tudo aquilo de conectar e desconectar poderia ser codificado e inserido na máquina, tal como os dados
- Em 1945 discutem a máquina EDVAC, totalmente eletrônica, lidava com **variáveis discretas** (não contínuas). Fazia o cálculo do início ao fim, **descobrindo por conta própria o que fazer a seguir**
- Ele escreveu relatório sobre o EDVAC de 100 páginas (um esboço) e tinha o nome dele apenas na capa, embora trabalho colaborativo (talvez não por culpa dele!)
- Lançava ideias inovadoras sobre organização de um computador com programa armazenado, mas Mauchly e Eckert se preocupavam com royalties, licenciamento, PI. Eles pediram a patente em 1947, mas as revelações em 1945 invalidaram.

Contratadas em 1945 como calculistas
pelo exército US e transferidas para o ENIAC.
Jean Jennings, Marlyn Wescoff, Ruth Lichterman,
Betty Snyder, Frances Bilas, Kay McNulty

Receberam plantas e diagramas de fiação e
Aprenderam sozinhas a programar.

Problema: simulação Monte Carlo do decaimento
De Neutrons durante a fissão nuclear

- Esta arquitetura ficou conhecida como von Neumann (Princeton)



O computador deve reconhecer certos padrões de bits como instruções (a linguagem da máquina) para diferenciar dos dados

DESAFIOS: blocos com funções diferentes, interligados

Necessário relógio para manter as coisas sincronizadas e regular o fluxo de dados,

De modo que as instruções buscadas na memória de curto prazo não se confundam com os dados sendo processados.

- Alan Turing leu o esboço de 1945 do EDVAC, entrou para o NPL (national physics lab) e escreveu uma resposta britânica ao EDVAC, chamada ACE (Automatic Computing Engine) – recorria ao software para cálculos aritméticos
- Em 1946 foi construída a primeira ACE, com 1 MHz de clock, 6000 válvulas, 4 tambores magnéticos, linhas de retardo de mercúrio para 768 números (memória) e dispositivos E/S com cartões perfurados – máquina lenta devido ao retardo!!. Outra máquina só em 1958, com Turing morto
- Maurice Wilkes lidera o projeto EDSAC em Cambridge. Tubos de raios catódicos (CRT) e fitas magnéticas resolveram o problema do retardo, tempos depois, mas não permitem acesso aleatório, típico das RAMs. Ele cria o conceito de **microcódigo**, ou seja, instruções de máquina poderiam ser executadas em hardware simples por este microcódigo

Em 1951, a J. Lyons & Co (empresa de chá) com equipe ENIAC e depois Maurice Wilkes, Lança o LEO 1, para adoção em indústrias como a Ford, meteorologia etc.

Em 1946 Mauchly e Eckert lançam sua própria Empresa e o computador UNIVAC, e em 1950 Se funde a Remington Rand de máquinas de Escrever.
O LEO então acabou na Fujitsu, e a Remington, Na Unisys.

SINCRONISMO

- Todos os circuitos numa arquitetura digital estão sincronizados por um RELÓGIO, ou melhor, o CLOCK, o qual usualmente é um sinal elétrico quadrado (trem de pulso) com tensão CC (**amplitude** TTL 5V ou CMOS 12V), e **frequência F**. A partir de uma ou mais frequências base geradas por cristais osciladores (quartzo), as demais frequências são divididas. Por exemplo, a partir de uma Frequência base = 50 MHz, obter 10 MHz, 1 MHz, 1 KHz, 1 Hz etc. para os demais circuitos, placas, barramentos etc.

SINCRONISMO

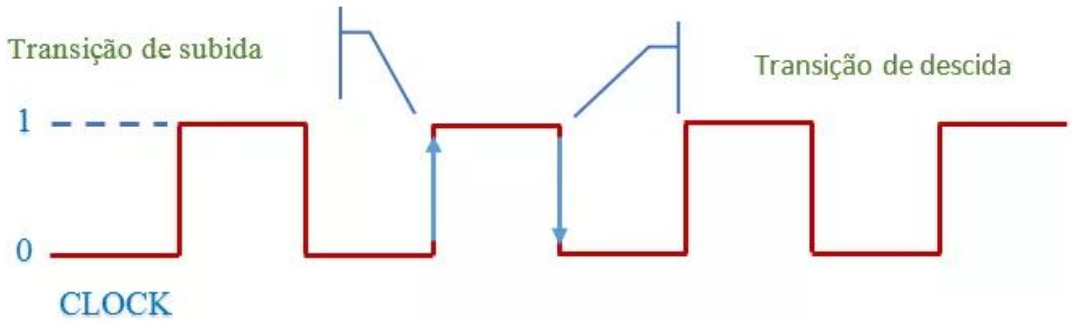
A frequência é definida em Hertz (Hz) ou radianos/segundo

F = 1/T, T é o período do sinal

Intervalo de tempo em que o fenômeno (sinal,

Chaveamento do transistor) ocorre

Em computação: KHz, MHz, GHz (comuns)



Fonte: http://eletronworld.com.br/eletronica/flip-flop-e-clock/

Outro termo: **FLOPS**

(Operações de ponto flutuante/Segundo)

Exemplo: Sunway TaihuLight (124 petaflops), em construção (2020), Exascale (1000 petaflops)
https://www.top500.org/system/178764/

VELOCIDADE DE PROCESSAMENTO - SUPER

Na UFRN, http://npad.ufrn.br/index.php (criado em 2016 o NPAD)

SUPERCOMPUTADOR: 72 nós de processamento, cada nó com 32 núcleos, num total de mais de 2.400 núcleos de processamento, 16 GPUs e 9 Terabytes de RAM.

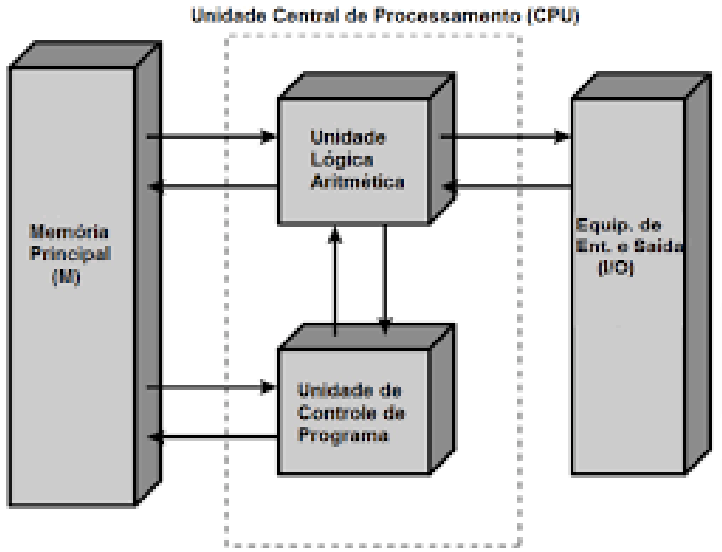
Adquirido por R$ 1.893.960,00, R$ 800.000,00 investimentos em refrigeração, expansão em 2019.

Aplicações: simulação em bioinformática, telecomunicações, previsões do tempo, otimização, projeto de circuitos, elementos finitos (térmica etc.)

https://www.youtube.com/watch?v=ULrtvX2Sunc

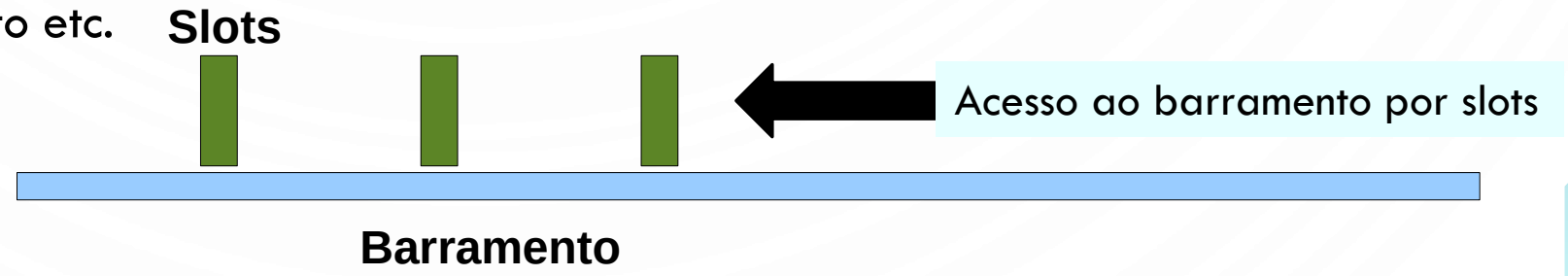
https://www.ufrn.br/imprensa/materias-especiais/40033/catalisador-da-pesquisa-cientifica-na-ufrn
Matéria de 17.09.2020

- Unidade Central de Processamento (CPU)
  - Manipulação direta ou indireta dos dados, executando instruções internas gravadas pelo fabricante (microcódigo) de acordo com as instruções externas (programas). A CPU é formada por vários componentes

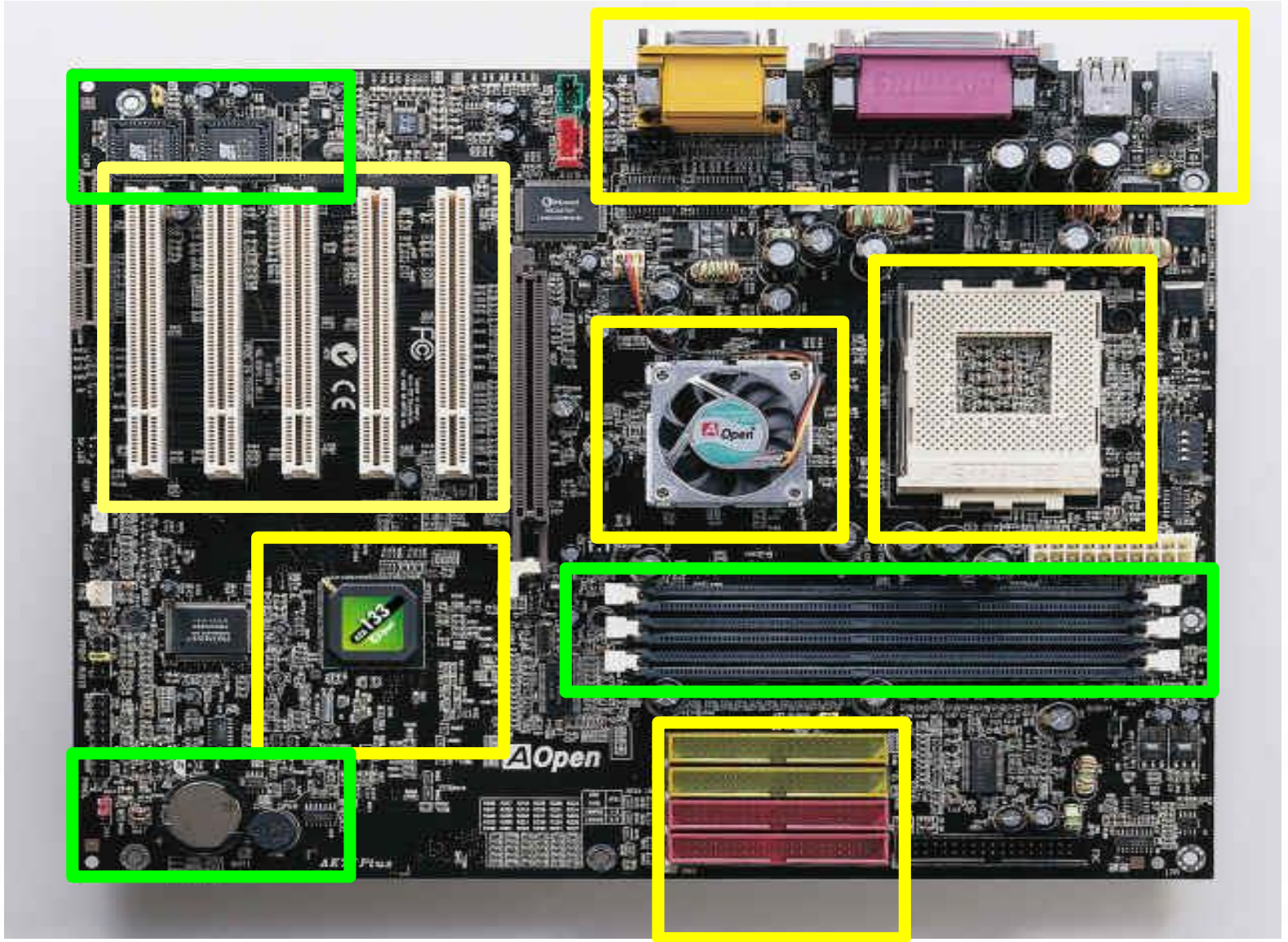


  - Conceito de UNIDADES (Controle, Lógica e Aritmética) e BARRAMENTOS (controle, instruções, dados, endereço). Um barramento pode ser visto como um conjunto físico de fios, uma espécie de "cabo" onde trafegam uma determinada quantidade de bits (byte, word, dword ou qword) a uma determinada velocidade (frequência), gerando uma taxa de transmissão de bytes/unidade de tempo. Este tamanho define a "largura" do barramento.

- Barramentos (bus)
  - Percurso principal entre dois ou mais componentes. Se estabelece apenas uma comunicação entre 02 componentes é chamado porta (**port**), sendo esta última mais associada à via de conexão entre o computador e os dispositivos externos (porta USB, serial, COM...)
  - Do PROCESSADOR – entrada e saída de dados do processador
  - De CACHE – pode haver barramento dedicado para acesso processador-cache (backside bus)
  - De MEMÓRIA – (ou frontal) conecta a memória ao processador. Em alguns casos, o de memória e processador são os mesmos
  - I/O padrão – usado por componentes mais lentos ou antigos
  - I/O local – alta velocidade, performance crítica, como controladores de vídeo, dispositivos de armazenamento etc.



- Barramentos (bus) e slots

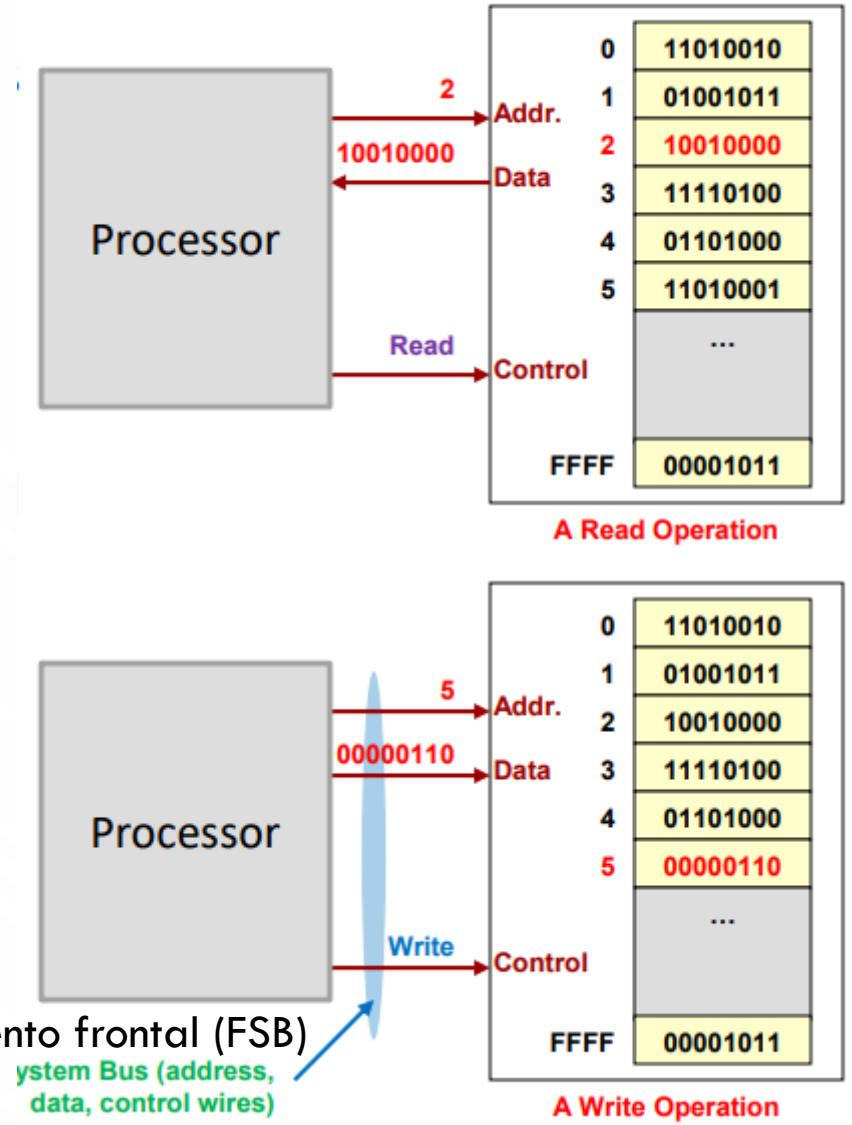


- Barramentos (bus) – todo barramento tem três partes:

Dados (data bus)	Carrega os dados através do sistema e o seu tamanho em bits determina a quantidade de dados que pode ser transferida ao mesmo tempo pelo processador, para a memória (palavra da memória). Isso influencia a velocidade total do sistema.
Endereços (address bus)	Carrega informação sobre a localização dos dados na memória principal. Cada combinação de bits que ele carrega indica uma posição de memória. Logo, seu tamanho indica o tamanho (posições) de memória que o processador pode endereçar. Ex: tamanho 16 bits, 2^16 = 64 kB; 32 bits, 2^32 = 4 GB; 64 bits, 2^64 = 16 PB
Controle (control bus)	Indica à memória quando a operação é de leitura ou escrita

- Os bits trafegam no barramento à determinada frequência (Hz). A largura de banda ou throughput é a quantidade teórica de dados transmitidos (bytes por segundo) – taxa de transmissão

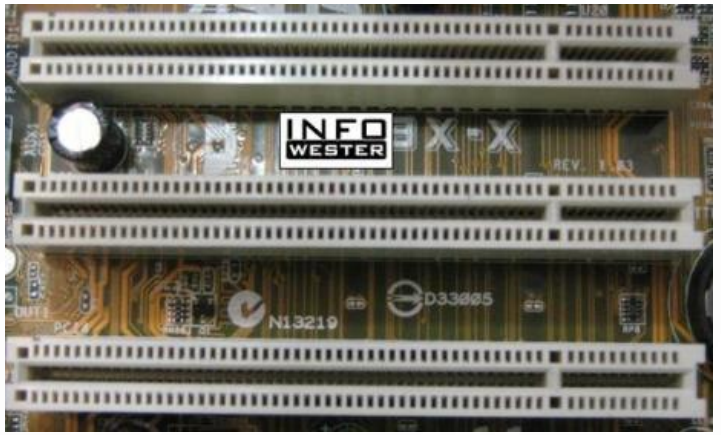
- Barramento (bus) – três partes:

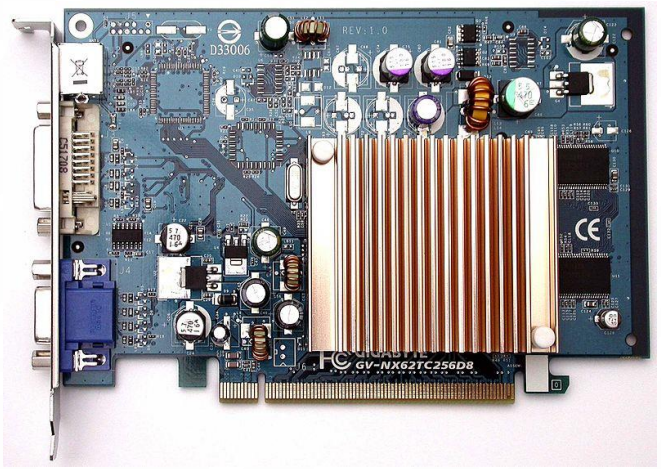


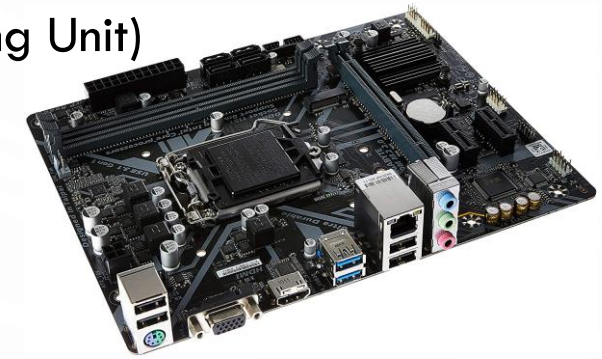
Barramento de sistema; Barramento frontal (FSB)

- Exemplo: (Peripheral Component Interconnect (PCI, 1993, Plug and Play)
  - PCI (32 bits) – 33 MHz = 133,33 MB/s
  - PCI-X QDR (64 bits) – 533 MHz = 4,266 GB/s
- PCI Express (PCIe, 2004)
  - Ligação ponto a ponto entre o dispositivo

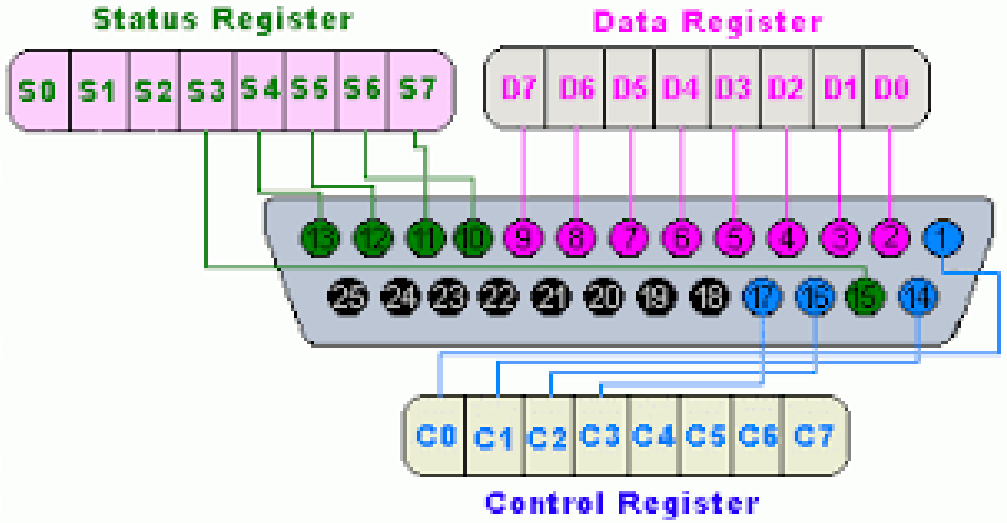
    e o barramento de controle
  - PCIe V1.x (4 GB/s) – v4.0 (31,51 GB/s)
  - Placa de vídeo offboard PCIe x16 (3.0, 4.0)
  - GPU (Graphics Processing Unit)





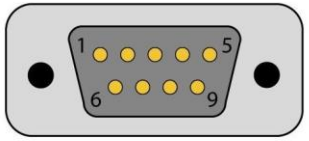


- Porta paralela – 1 byte por vez
  - Fios paralelos levam sinais elétricos
  - Interferência entre eles proporcional ao comprimento do fio – limita cabos a 3m
  - Conector DB25 (centronics)
  - EPP – 1 MB/s
  - ECP

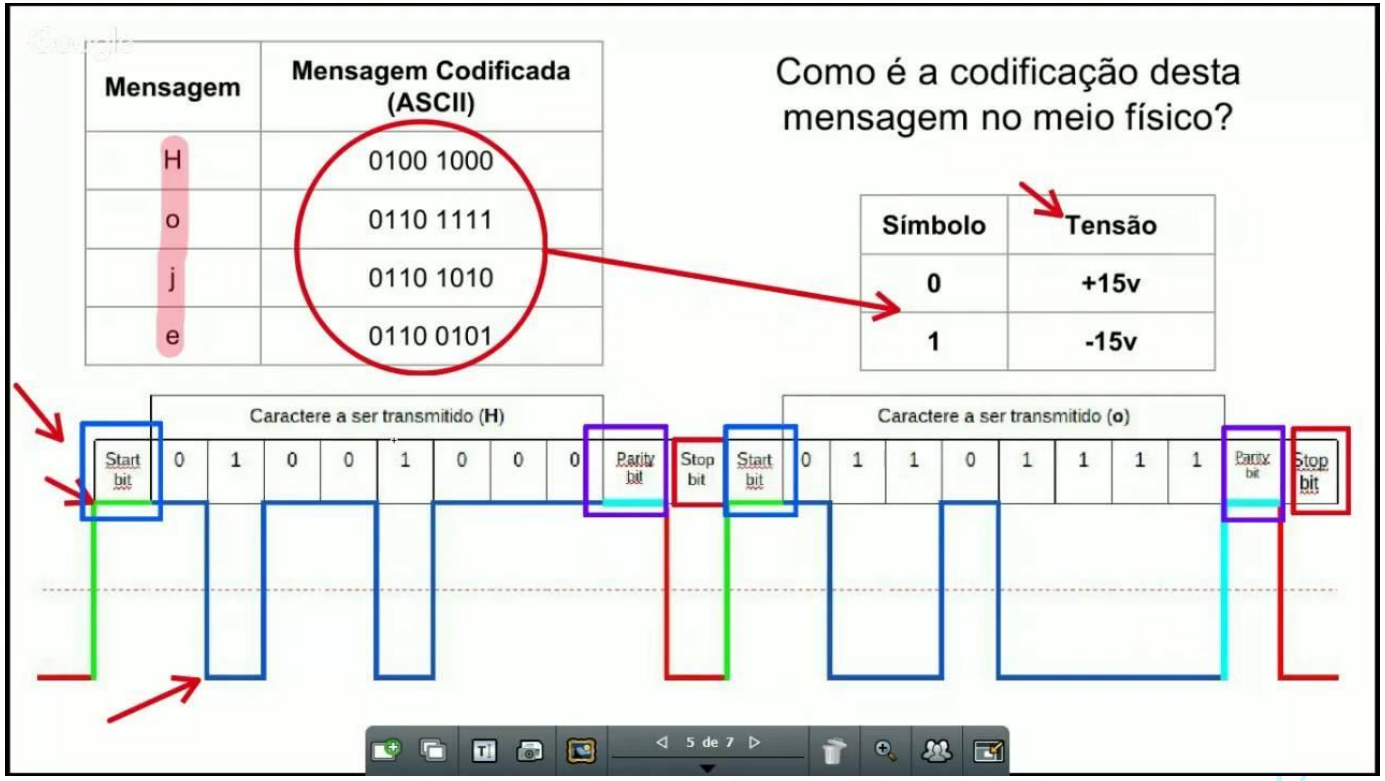


- Porta serial
  - Bits enviados em série
  - Mais veloz que paralela
  - Técnicas de detecção de erro
  - Cabos até 15m
  - Conector mais comum DB-9
  - Padrões EIA-232C, 422-C, 485
  - (RS-232, RS-422, RS-485)

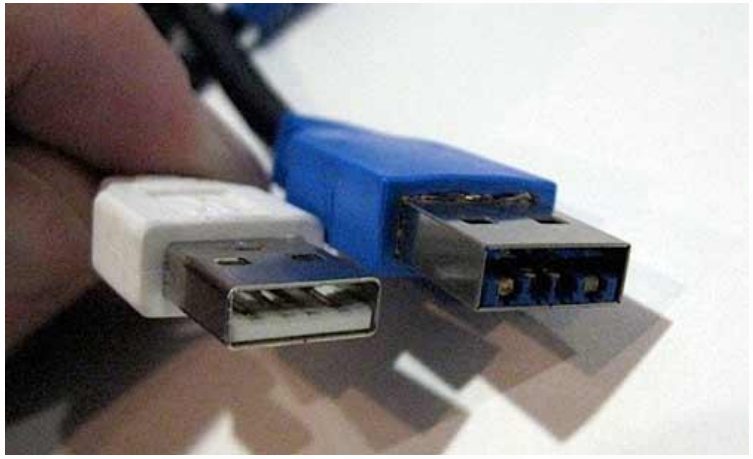
DB9M Connector

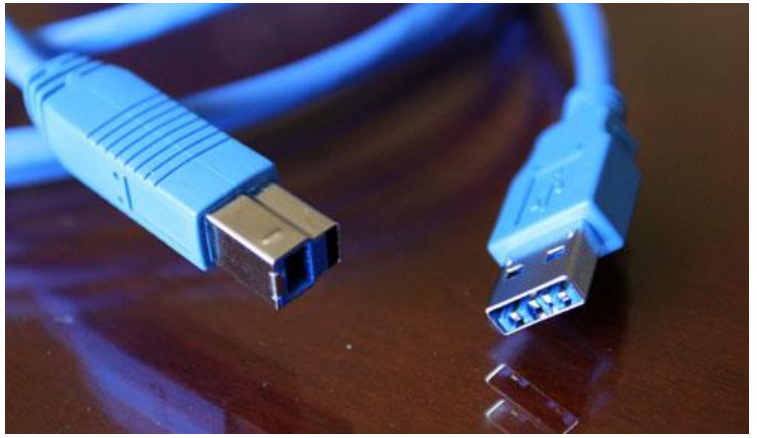


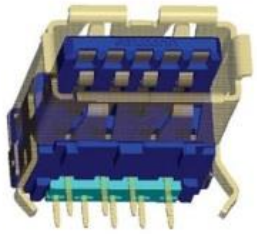
Pino	Sinal	Descrição do sinal
1	DCD	Data Carrier Detect
2	RXD	Receive Data
3	TXD	Transmit Data
4	DTR	Data Terminal Ready
5	GND	Signal Ground (Terra)
6	DSR	Data Set Ready
7	RTS	Request do Send
8	CTS	Clear to Send
9	RI	Ring Indicator

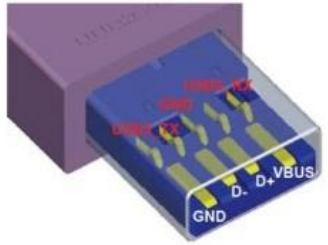


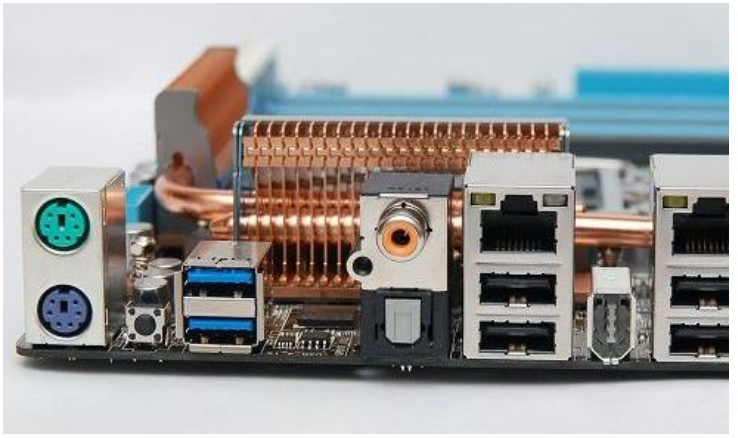
- USB – Universal Serial Bus
  - Até 127 dispositivos por meio de HUB
  - Cabo de 5m
  - De USB 1.0 (1,5 MB/s) a USB 3.2 SuperSpeed+ (2,5 GB/s), USB 4.0 (type-C)



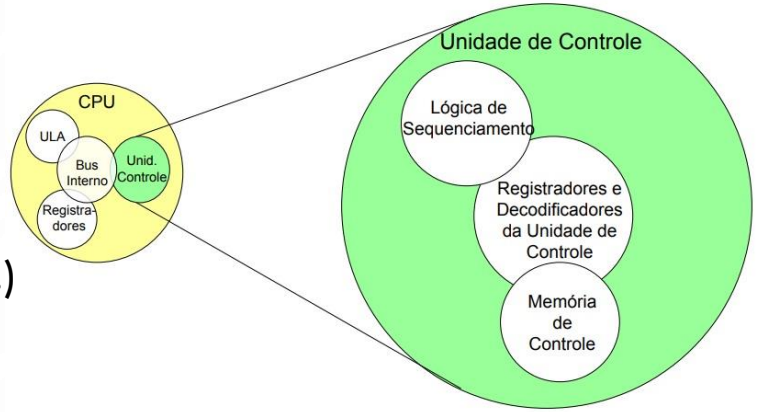






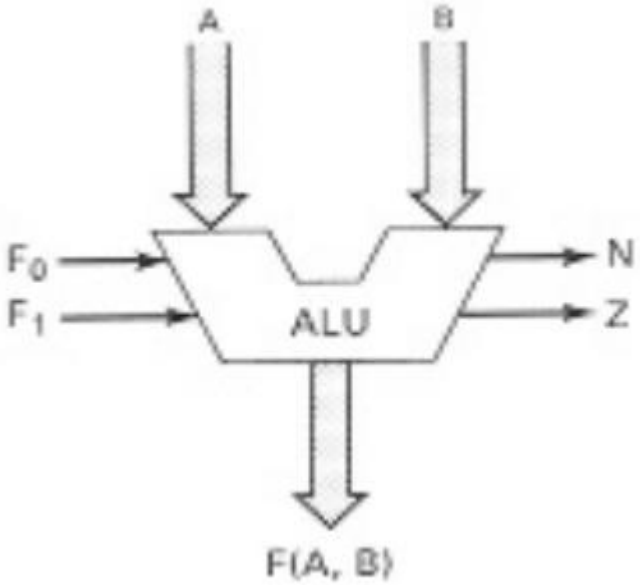


- Unidade de Controle (UC)
  - Coordena as atividades realizadas pela CPU, fornecendo sinais de controle que sincronizam e ordenam as micro-operações (realizadas pela UC)
  - Sincronismo necessário; a UC gera série de pulsos elétricos de sincronização transmitidos aos demais componentes do sistema, que utilizam tais pulsos para coordenar suas operações. O sincronismo ocorre na borda de subida ou na borda de descida do sinal.
  - Algumas instruções gastam um ciclo, outras vários.
  - A UC contém instruções gravadas no seu hardware, o chamado microcódigo, um conjunto de instruções básicas, que indicam quais operações a CPU é capaz de realizar e como vai realiza-las. Operações
    - Ariméticas (soma, subtração, multiplicação e divisão)
    - Lógicas (xor, or, and, etc.)
    - Movimentação de dados (memória-CPU, registrador-memória etc.)
    - Desvio
    - Entrada e saída (troca de dados com as memórias secundárias ou dispositivos ES)



- Unidade de Controle (UC)
  - Sequência de execução de programa
    - A CPU recebe sequência de instruções e procura realiza-las na sequência de recebimento
    - A UC compara a instrução recebida com o microcódigo. Se estiver na lista, a UC pode executar
    - As CPUS podem ter conjunto de instruções diferentes, mas a **compatibilidade ascendente** é preservada.
- Unidade Lógico Aritmética (ULA)
  - Quanto UC encontra instrução com operação aritmética ou lógica, passa o controle para a ULA. A ULA possui conjunto de circuitos eletrônicos para realizar operações simples. Uma operação complexa pode ser reduzida a vários passos simples. Exemplo: 3^2 = 3*3 = 3+3+3 (soma, básica).
  - A ULA possui códigos de condição, que indicam (sinalizam) eventos que podem ocorrer
    - Overflow (não pode ser representado, "ultrapassou" capacidade
    - Sinal (se o resultado é positivo ou negativo)
    - Carry ("vai-um" na soma (carry-out))
    - Zero (resultado da operação) é nulo
    - Existem ULAS para inteiros, para ponto flutuante, e na ULA não é armazenado nenhum dado

- Exemplo de ULA, supor 4 operações possíveis
  - Se F0F1 = 00, F = A+B
  - Se F0F1 = 01, F = A and B
  - Se F0F1 = 10, F = A
  - Se F0F1 = 11, F = not(A)
  - Se o resultado for negativo N = 1
  - Se o resultado for positivo Z = 0

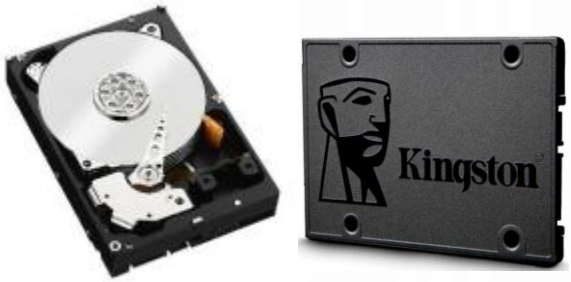


- IMPLEMENTADA POR MULTIPLEXADOR (MUX), SELETOR DE ENTRADAS

- Vamos observar o seguinte: onde reside cada item abaixo?

(1) Processador

(2) Memória



(3) Armazenamento

* Fonte: adaptado de https://ee.usc.edu/~redekopp/cs356/slides/CS356Unit4_x86_ISA.pdf

- Código fonte
- Código de programa em execução
- Variáveis globais
- Executável compilado (antes da execução)
- Instruções em execução
- Variáveis locais

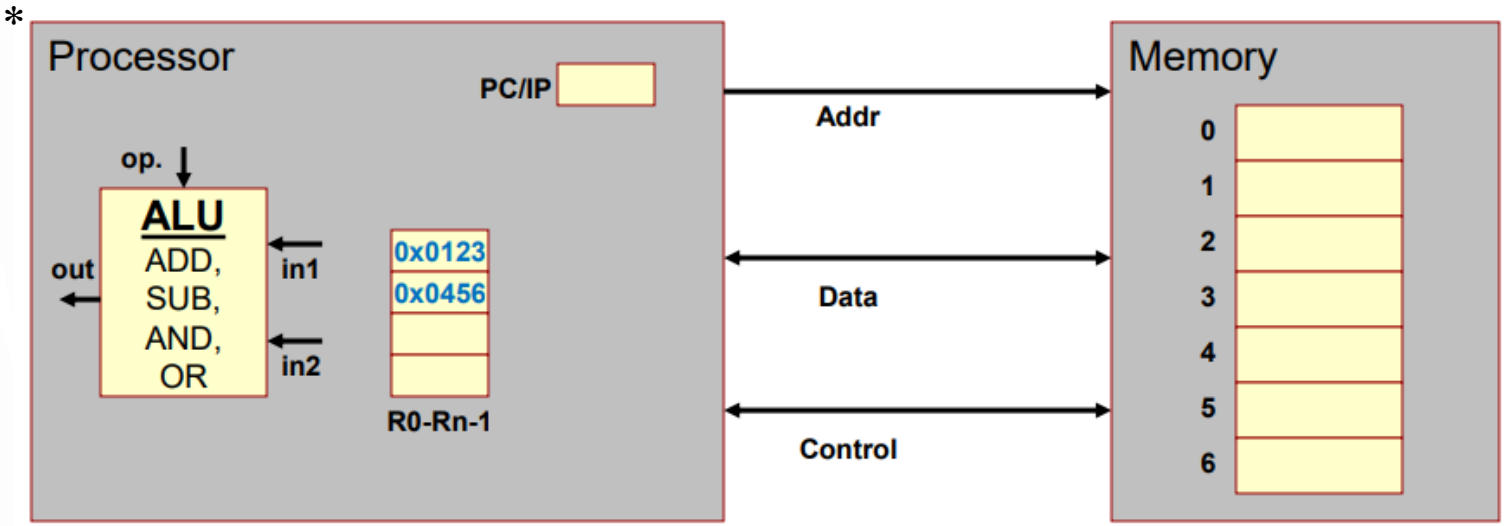
- Registradores

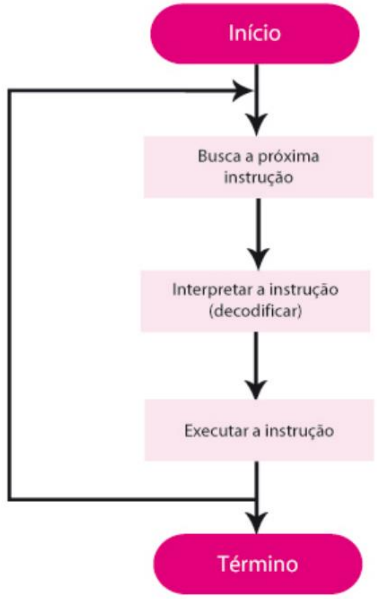
https://cs.brown.edu/courses/cs033/docs/guides/x64_cheatsheet.pdf

  - Auxiliam UC e ULA no armazenamento temporário (e rápido) de dados os quais estas unidades manipulam. Pertencem à CPU. Podem ser **especiais** ou de **uso geral**
  - Suponha que a CPU precisa SOMAR 1 ao que está armazenado em algum lugar da memória principal: A = A + 1
    - CPU copia o conteúdo dessa posição de memória onde o Sistema Operacional alocou A num registrador R0
    - A ULA recebe esse valor do registrador R0 e soma 1, armazenando o resultado no registrador R0
    - Então este valor atualizado é copiado do registrador R0 para a posição original da memória principal onde foi alocado A
    - OBS: O tamanho em bits do registrador R0 determina a quantidade de dados que ele pode processar ao mesmo tempo, o limite de valores que ela pode trabalhar e até a velocidade com que ela consegue realizar operações – PALAVRA DA CPU
  - Registradores especiais
    - CI, CP ou PC (Contador de programa ou Program Counter) – mantém atualizado o endereço de memória da próxima instrução a ser realizada. (x86_64: RIP (register instruction pointer)
    - ACC (acumulador) – armazena dados E/S da ULA. Quando recebe o sinal de carga, copia o conteúdo da ULA e elimina o conteúdo anterior

- Ideia Geral

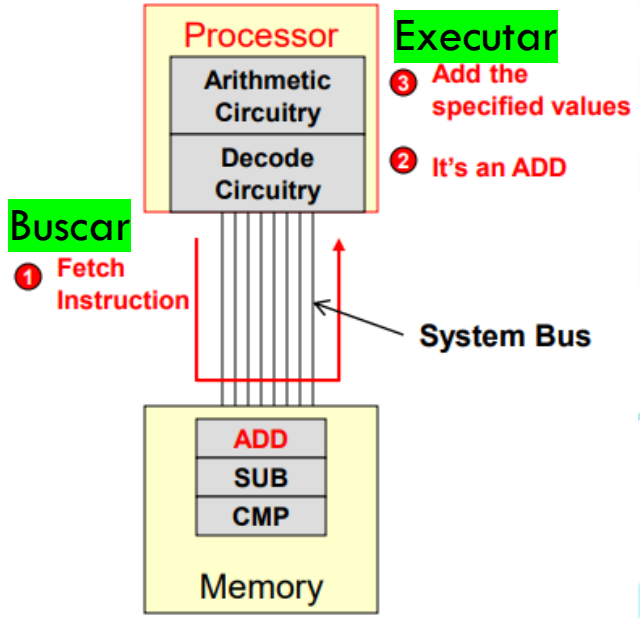
*



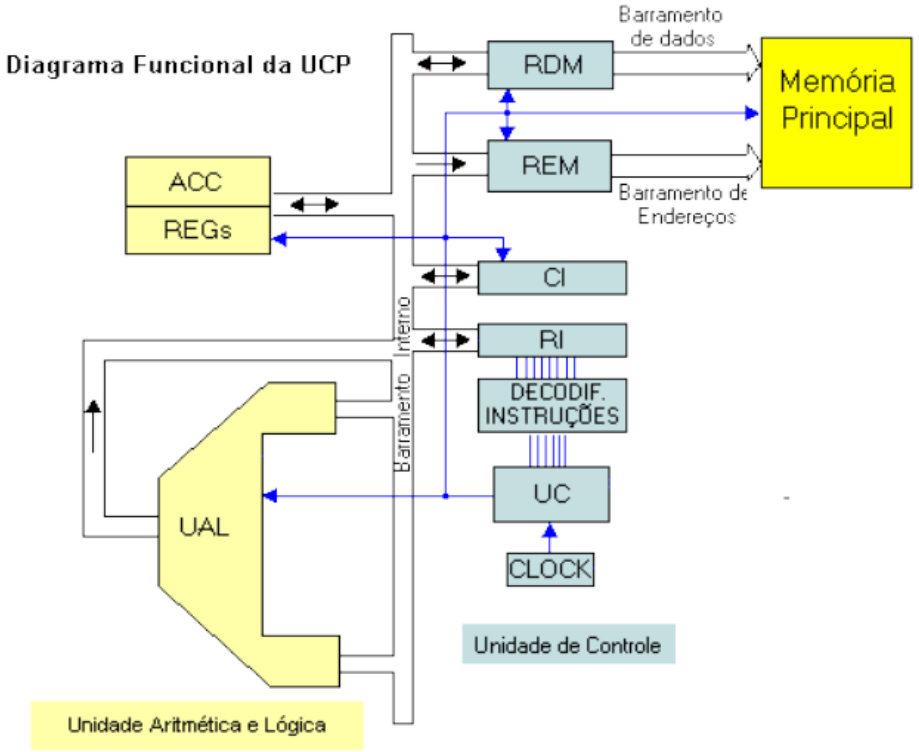


Ciclo de máquina (ou de instrução) – controlado pela UC
BUSCA (fetch)
DECODIFICAÇÃO
EXECUÇÃO
- cálculo do endereço de um operando
- busca de operando na memória
- seleção de uma operação na ULA
- carga de um registrador
- escrita de operando na memória
- atualização do PC para desvios

ARMAZENAMENTO (write-back)



- Registradores
  - Resistrador de Instruções (RI) – armazena a instrução que está sendo executada
  - Registrador de Dados da Memória (RDM ou MBF – memory buffer register) – armazena temporariamente os dados transferidos da memória principal para a CPU ou transferidos da CPU para a memória principal. Envia esses dados posteriormente (BIDIRECIONAL)
  - Registrador de Endereços de Memória (REM ou MAR – memory address register) – armazena temporariamente o endereço da mem. Principal que será acessado pela CPU para uma operação de escrita (write, w) ou leitura (read, r) no referido endereço



E se não tivéssemos registradores?
Ex.: F = (A + B) – (A * B)
- 3 operações: ADD, SUB, MULT
a) Carregue A e B da memória e guarde o resultado na memória
b) Carregue A e B da memória e guarde o resultado na memória
c) Carregue os resultados de a) e b), subtraia e guarde na memória
9 ACESSOS À MEMÓRIA! – Quantos seriam usando REGS?

- Registradores
  - Algoritmo - GERAL

O registrador CI (Contador de Instrução ou Program Counter PC) contém o endereço da próxima instrução a ser executada de um determinado processo. A UC (Unidade de Controle) solicita a obtenção da instrução na memória principal, colocando o endereço que está em CI no registrador REM (Registrador de Endereço de Memória), o qual comunica-se com a memória RAM (principal). Esta comunicação se dá pelo Barramento de Endereços. A instrução é entendida como um DADO e, portanto, retorna da memória principal pelo Barramento de Dados para o registrador RDM (Registrador de Dados de Memória) e então o mesmo passa a instrução recém obtida para o registrador RI (Registrador de Instrução), o qual armazena **efetivamente** a instrução. Perceba na Figura 1 que o Barramento de Endereços é unidirecional (CPU-RAM e não RAM-CPU). A UC então ativa o DECODIFICADOR DE INSTRUÇÕES que interpretará a instrução. Interpretar a instrução significa saber qual é a instrução e quais são os operandos envolvidos naquela instrução, por exemplo, em uma soma de dois números, quais são os números?

Lembre que no interior da CPU tudo é binário, portanto a instrução é um número binário de determinada quantidade de bits (depende da CPU) e consiste, basicamente, de um campo OPCODE, com o código da instrução e o campo OP, com o operando.

Podem ocorrer algumas situações em relação ao operando:

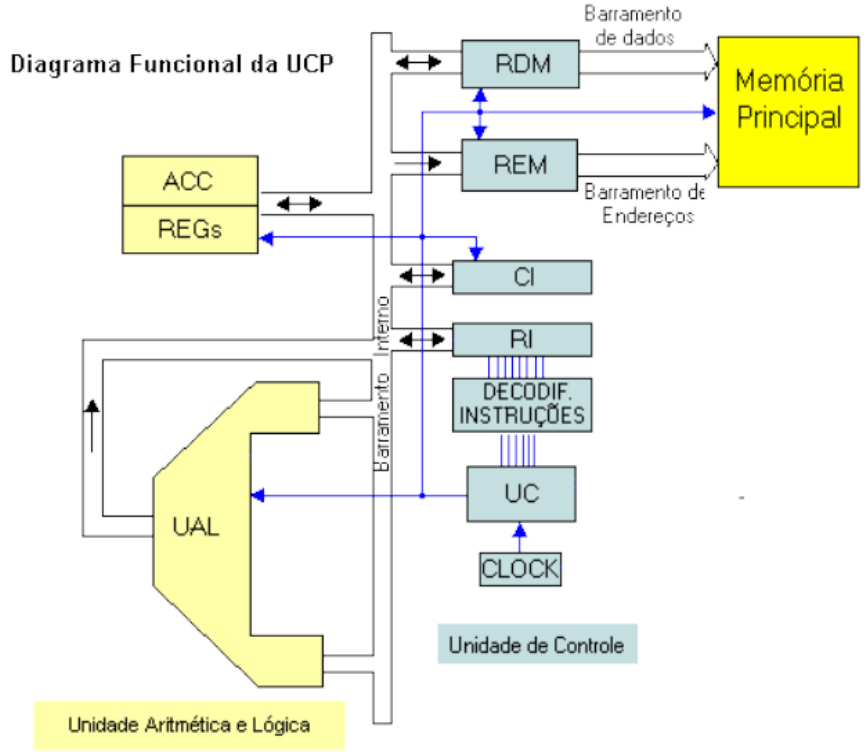
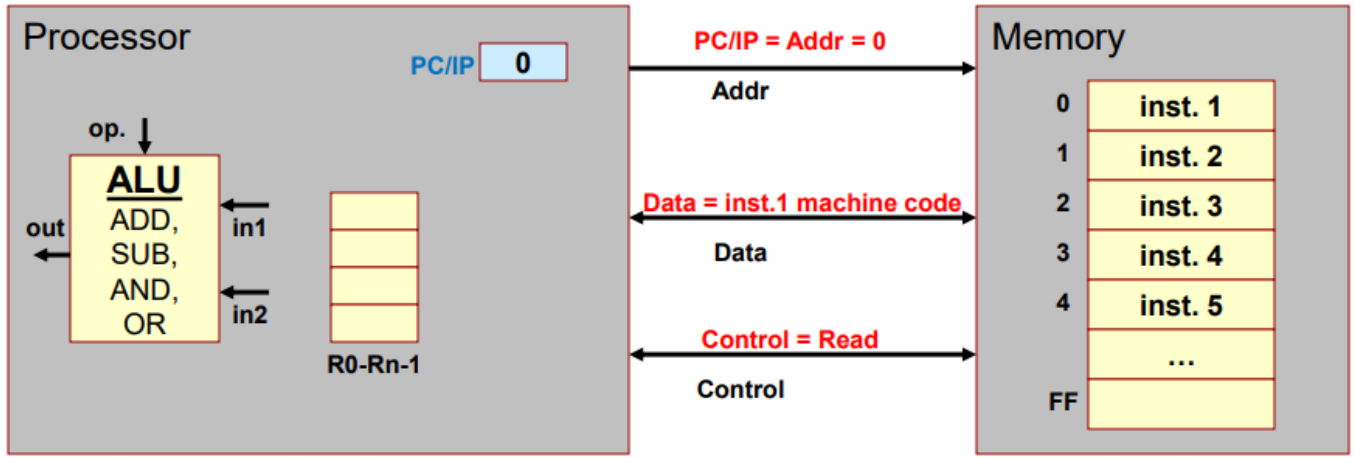


Figura 1

- Registradores
  - Algoritmo – FASE BUSCA

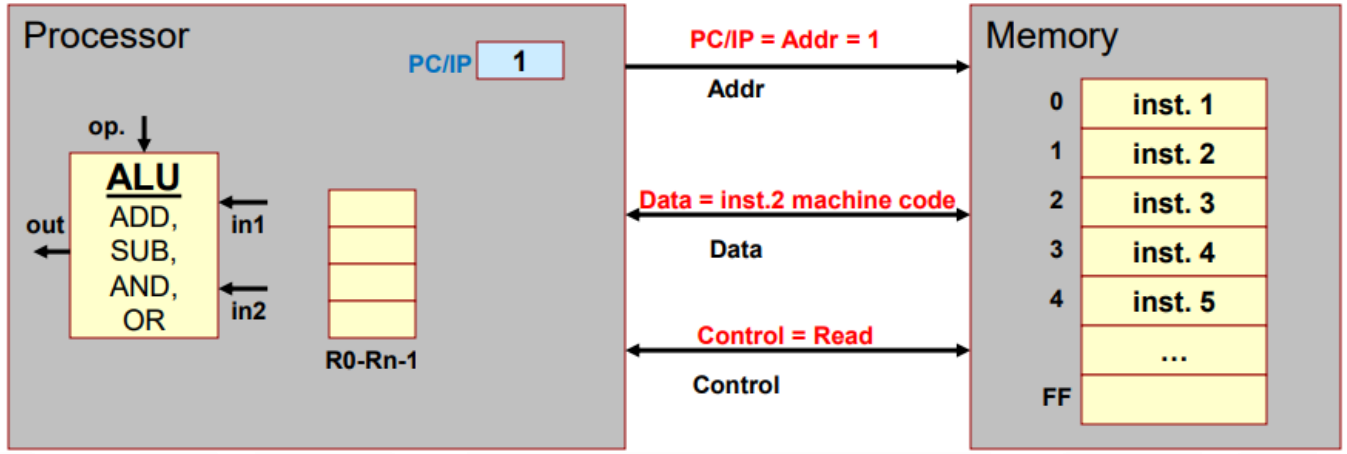
*

1)



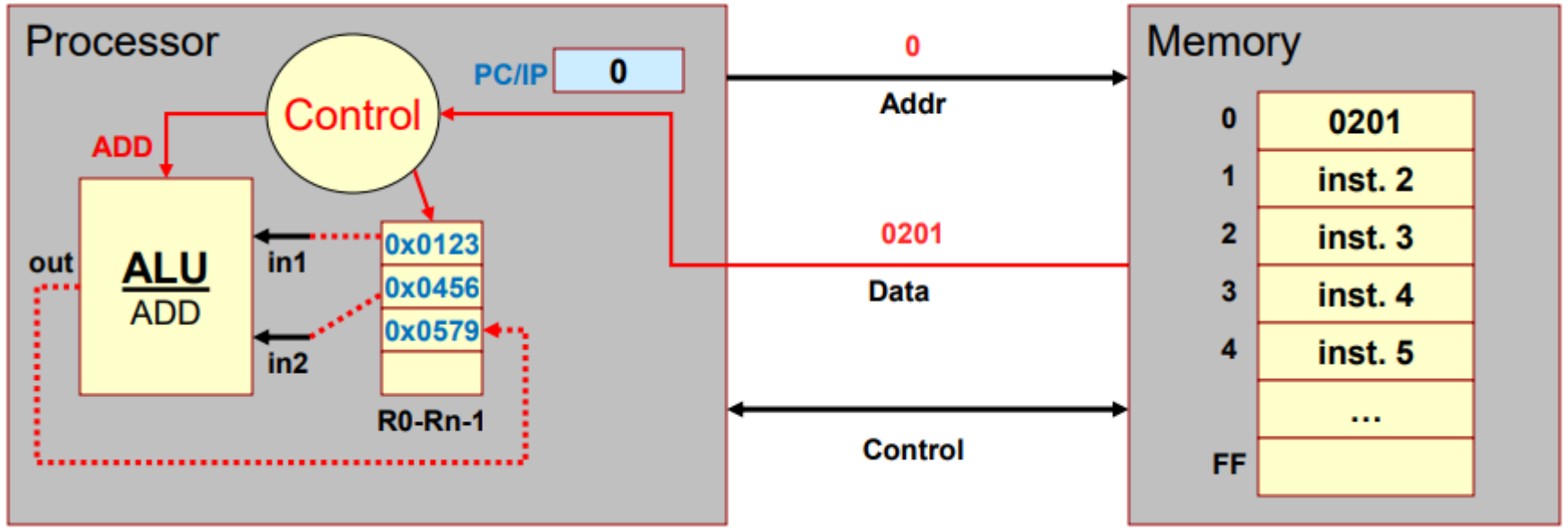
*

2)



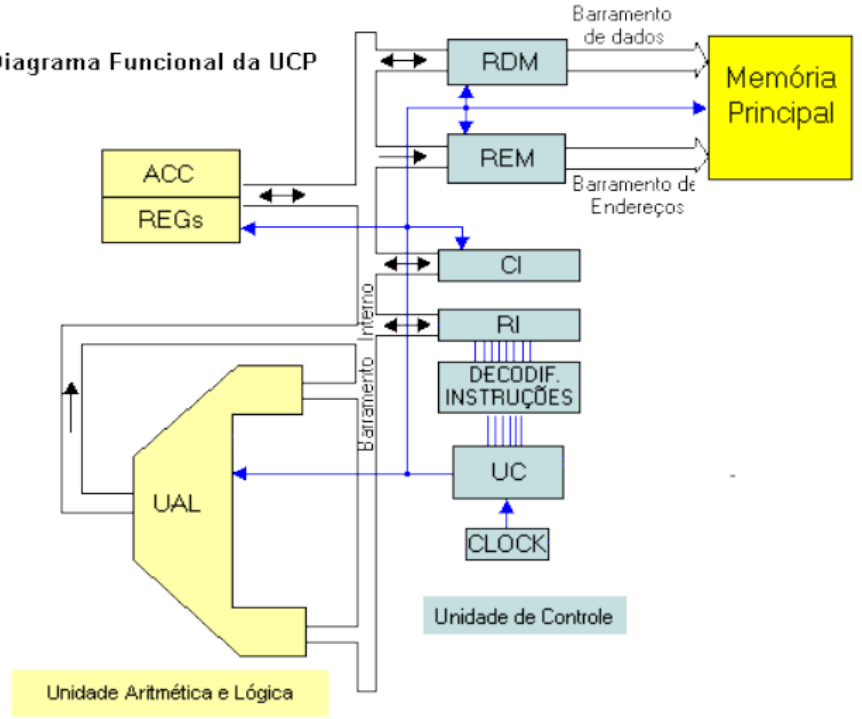
- Registradores
  - Algoritmo – FASE DECODIFICAÇÃO

*



- Registradores
  - Algoritmo (uma medida desse ciclo é **MIPS**, milhões de instruções por segundo)

1. Se o operando já for o próprio dado a ser manipulado pela ULA, este é direcionado ao registrador ACC (Acumulador) e a partir daí está pronto para ser acessado pela ULA através do barramento interno.
2. Na segunda situação, o campo operando da instrução contém o endereço para o dado na memória principal. Nesse caso, a UC coloca este endereço no registrador REM, realiza-se a busca na memória principal, o dado retorna pelo Barramento de Dados ficando em RDM. Do RDM segue para o ACC e novamente está pronto para uso pela ULA.
3. Pode acontecer também do operando ser um registrador onde o dado está.



• Ideia de instruções e modo de endereçamento

```
0000xxxxxxxxxxxx LODD ac:=m[x]
0001xxxxxxxxxxxx STOD m[x]:=ac
0010xxxxxxxxxxxx ADDD ac:=ac+m[x]
0011xxxxxxxxxxxx SUBD ac:=ac-m[x]
0100xxxxxxxxxxxx JPOS if ac >= 0 then pc:=x
0101xxxxxxxxxxxx JZZR if ac=0 tehn pc:=x
0110xxxxxxxxxxxx JUMP pc:=x
0111xxxxxxxxxxxx LOCO ac:x (0 <= x <= 4095)
1000xxxxxxxxxxxx LODL ac:=m[sp+x]
1001xxxxxxxxxxxx STOL m[x+sp]:=ac
1010xxxxxxxxxxxx ADDL ac:=ac+m[sp+x]
1011xxxxxxxxxxxx SUBL ac:=ac-m[sp+x]
1100xxxxxxxxxxxx JNEG if ac<0 then pc:=x
1101xxxxxxxxxxxx JNZE if ac not= 0 then pc:=x
1110xxxxxxxxxxxx CALL sp:=sp-1; m[sp]:=pc; pc:=x
1111000000000000 PSHI sp:=sp-1; m[sp]:=m[ac]
1111001000000000 POPI m[ac]:=m[sp]; sp:=sp+1
1111010000000000 PUSH sp:=sp-1; m[sp]:=ac
1111011000000000 POP ac:=m[sp]; sp:=sp+1
1111100000000000 RETN pc:=m[sp]; sp:=sp+1
1111101000000000 SWAP tmp:=ac; ac:sp; sp:tmp
11111100yyyyyyyy INSP sp:=sp+y (0 <= y <= 255)
```

Imaginemos máquina com barramento de endereços de 12 bits (largura do REM), ou seja, isto irá definir o tamanho da memória, 2^12 = **4096** endereços (espaço de endereçamento), onde em cada endereço pode estar um dado de 16 bits (palavra da memória)

Uma instrução tem o CÓDIGO DE OPERAÇÃO (OPCODE) e o operando (OP). Se o código tem 4 bits, esta máquina só teria 16 instruções

Endereçamento imediato – o dado encontra-se na própria instrução ou registrador especial (como o ACC)

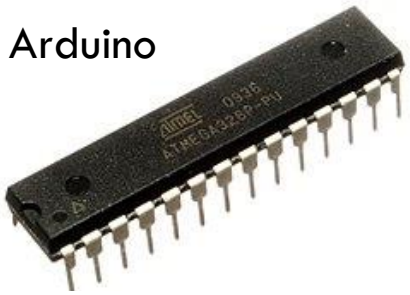
Endereçamento direto – instrução contém o endereço na memória do dado

Endereçamento indireto – a instrução contém um endereço da memória em que se encontra o endereço do dado

Endereçamento indexado – o endereço do dado é obtido somando o endereço na instrução com um valor fixo (k) contido num registrador especial

- Arquitetura CISC X RISC (o que se sabia...)
  - Complex Instruction Set Computer
    - Instruções realizam funções mais complexas
    - Facilita trabalho do programador, com menor microcódigo e consequente menor espaço na memória
    - Logo, CISC são mais lentos, pois instruções demandam mais ciclos de clock para execução
    - Os programas podem ser menores, fazendo uso mais eficiente da memória
    - Já o set de instruções é maior
    - Em 1960 IBM observou que 10% das instruções realizavam 90% das tarefas
  - Reduced Instruction Set Computer
    - Instruções mais simples, executadas em um ciclo de clock
    - Set de instruções menor, sem decodificação
    - Processadores menores que os CISC e mais baratos
    - Com o tempo passaram a ser usados em workstations
    - Permite execução em paralelo (pipelining)

ATMEGA328P
Plataforma UNO Arduino
RISC



Parâmetro	Valor
Tipo de CPU	8-bit AVR
Desempenho	20 MIPS às 20 MHz
Memória Flash	32 kB
SRAM	2 kB
EEPROM	1 kB
Contagem de pinos	28 pinos PDIP, MLF, 32 pinos TQFP, MLF
Frequência máxima	20 MHz
Número de canais de toque	16
Hardware de Aquisição QTouch	Nenhum
Máximo de pinos de I/O	23
Interrupções externas	2
Interface USB	Nenhum
Velocidade USB	–

- Arquitetura CISC X RISC: A = B + C
  - Complex Instruction Set Computer

```
add mem(B), mem(C), mem(A)
```

  - Reduced Instruction Set Computer

```
load reg(1), mem(B);
load reg(2), mem(C);
add reg(3), reg(1), reg(2);
store mem(A), reg(3)
```

A discussão CISC x RISC não é mais relevante à medida que a percepção transitou da ISA (Instruction Set Arch) para a micro arquitetura, a implementação de fato da ISA (x86, ARM...). A geração do microcódigo é otimizada! Fabricantes com Intel, Apple etc. tem adotado ARM para computadores de uso geral.