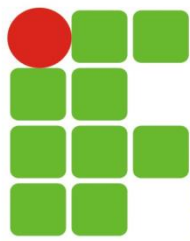


**INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**  
RIO GRANDE DO NORTE  
Campus Santa Cruz

# Organização de computadores

Aula 05



# Níveis de um computador

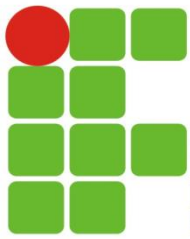
Computadores possuem uma série de níveis

Cada um construído sobre seus antecessores

Cada um representa uma abstração distinta com diferentes objetos e operações

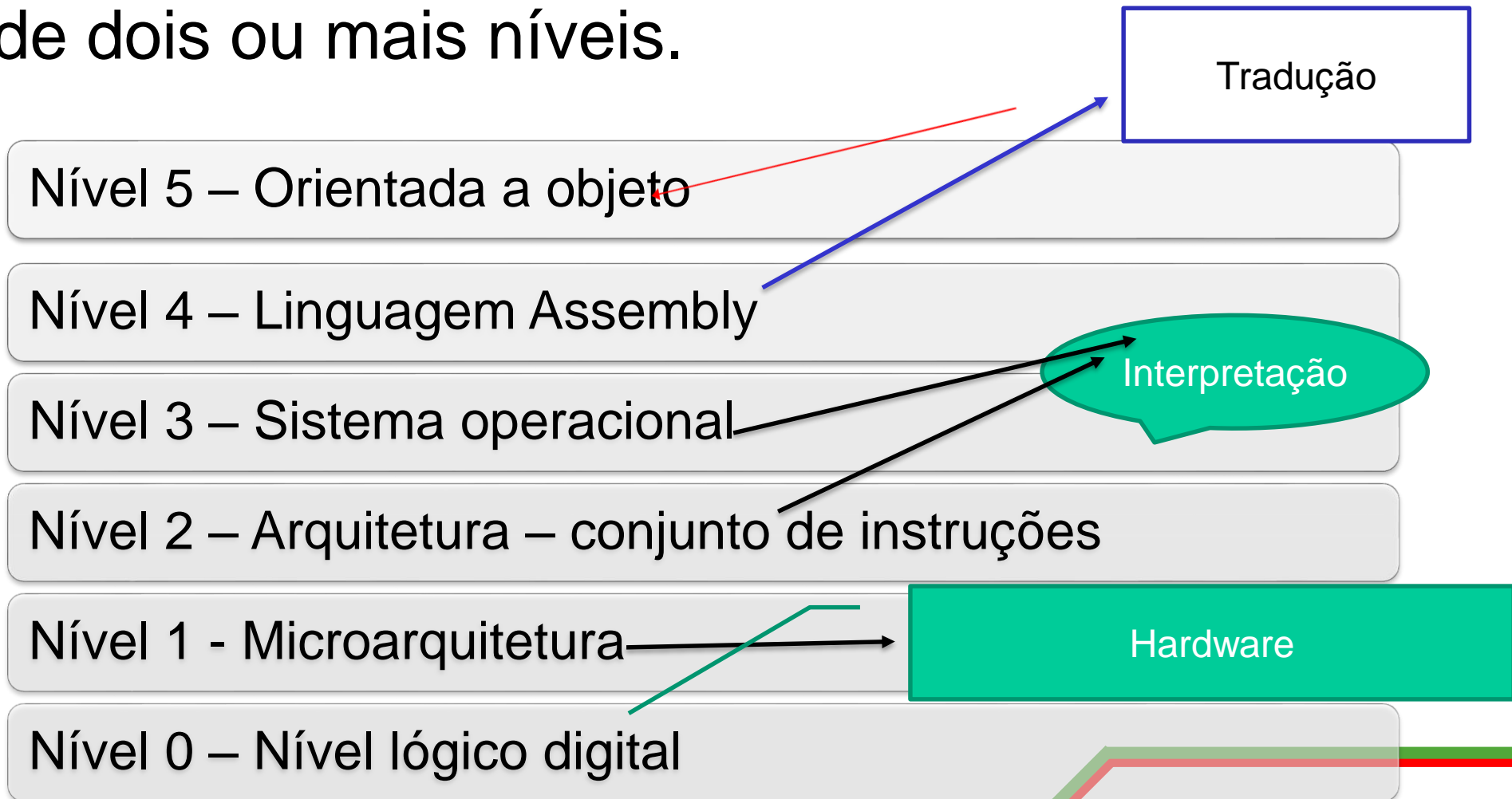
O conjunto de tipos de dados, operações e características de cada nível é denominado arquitetura.

-



# Níveis

A maioria dos computadores modernos consiste de dois ou mais níveis.





# Nível 0

## Nível lógico digital

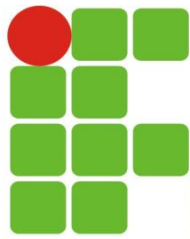
Portas lógicas

Transistores – Analógicos  $\Rightarrow$  Portas digitais

Portas digitais  $\Rightarrow$  Memória

Memórias podem ser combinadas em grupos

Registradores armazenam



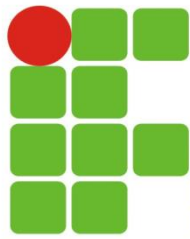
# Nível 1

Conjunto de 8 a 32 Registradores formando uma memória local e um conjunto ALU ou ULA

ULA – executa operações aritméticas simples

Registradores formam um caminho de dados transitando.

A ULA efetua algo com os registradores



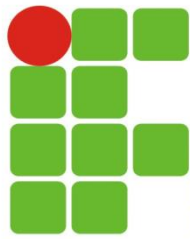
# Nível 2

*Busca, examina e executa instruções*

*Nível de Arquitetura do Conjunto de  
Instruções – ISA*

*Conjunto de instruções da máquina*

*Por interpretação de microprograma ou  
pelos circuitos de execução de hardware.*



# Nível 3

## Nível híbrido

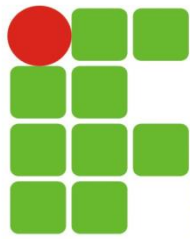
Parte das instruções estão em nível ISA

Conjunto de novas instruções

Organização diferente da memória

Capacidade de execução de dois ou mais programas ao mesmo tempo

Interpretador que roda no nível 2 - SO



# Nível 4

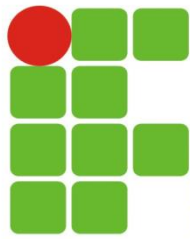
Sai os programadores de sistemas e entram os programadores aplicação

Sai interpretação e entra tradução

A linguagem já é mais próxima da linguagem humana

O programa que realiza a tradução é chamado de Assembler





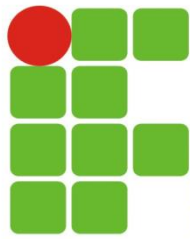
# Nível 5

Linguagem de programadores de  
aplicação

Alto nível

C++; Java; LISP; Prolog

Em geral são traduzidos para os níveis 3  
e 4 por compiladores



# A microprogramação

Os primeiros computadores tinham só dois níveis

ISA

Escrita a programação

Lógico digital

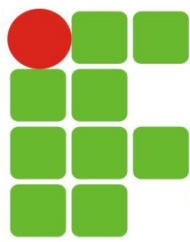
Execução dos programas

Surgiram computadores com 3 níveis

Simplificar o hardware

Interpretador embutido

Executar programas de nível ISA



# Arquitetura ou Organização

A arquitetura trata dos aspectos que são visíveis ao usuário do nível;

Características que o programador vê

Quantidade de memória, por exemplo

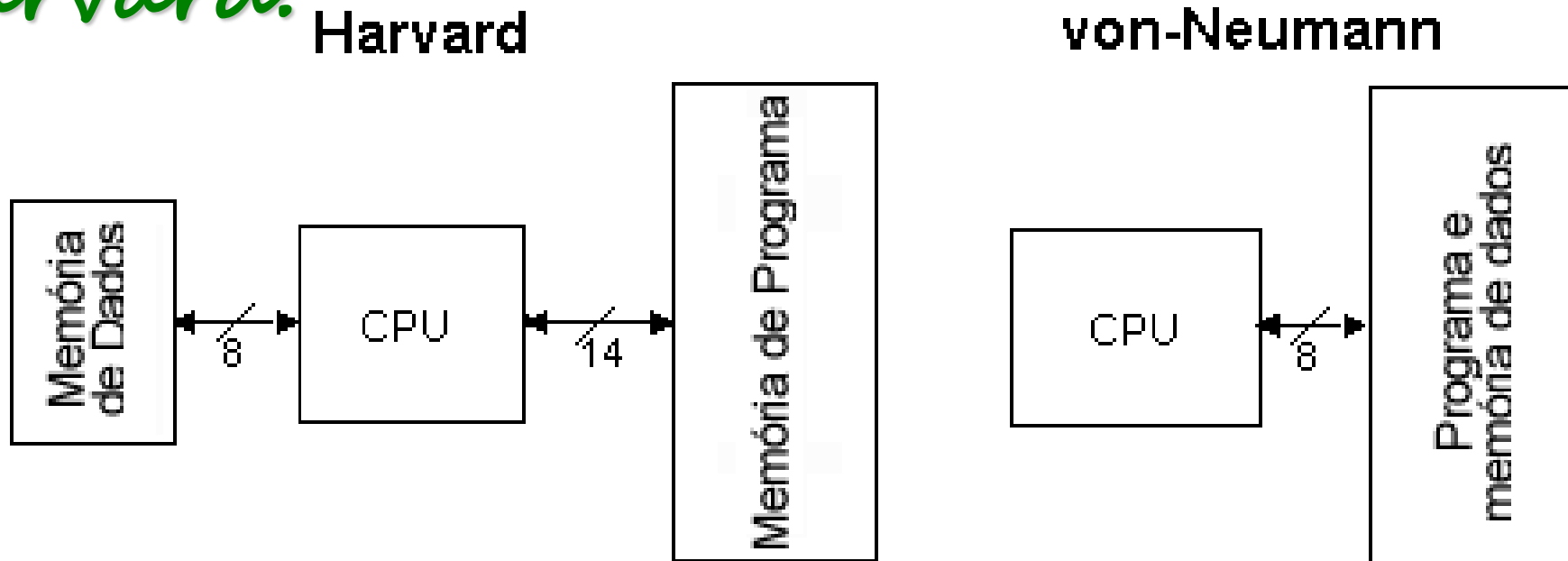
Arquitetura de computadores pode ser denominado como sendo o projeto das partes de uma sistema de computador visível ao programador

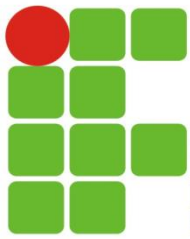


# ARQUITETURAS

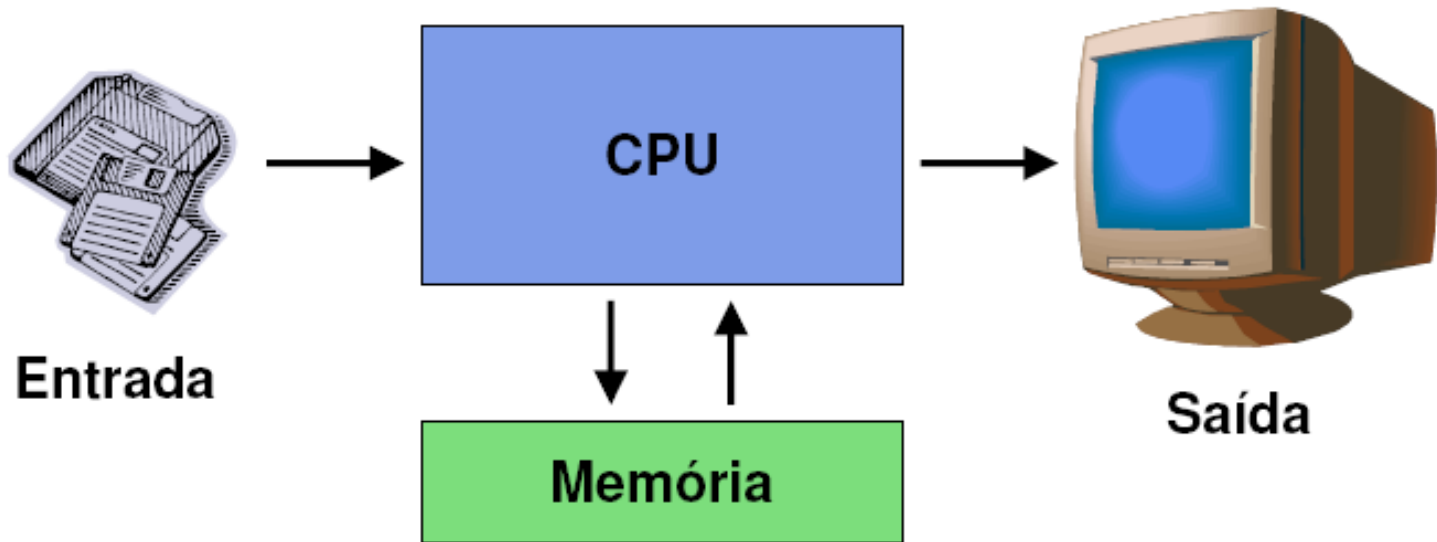
As principais arquiteturas de processadores são:

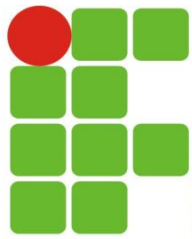
- Von Neumann;
- Harvard.



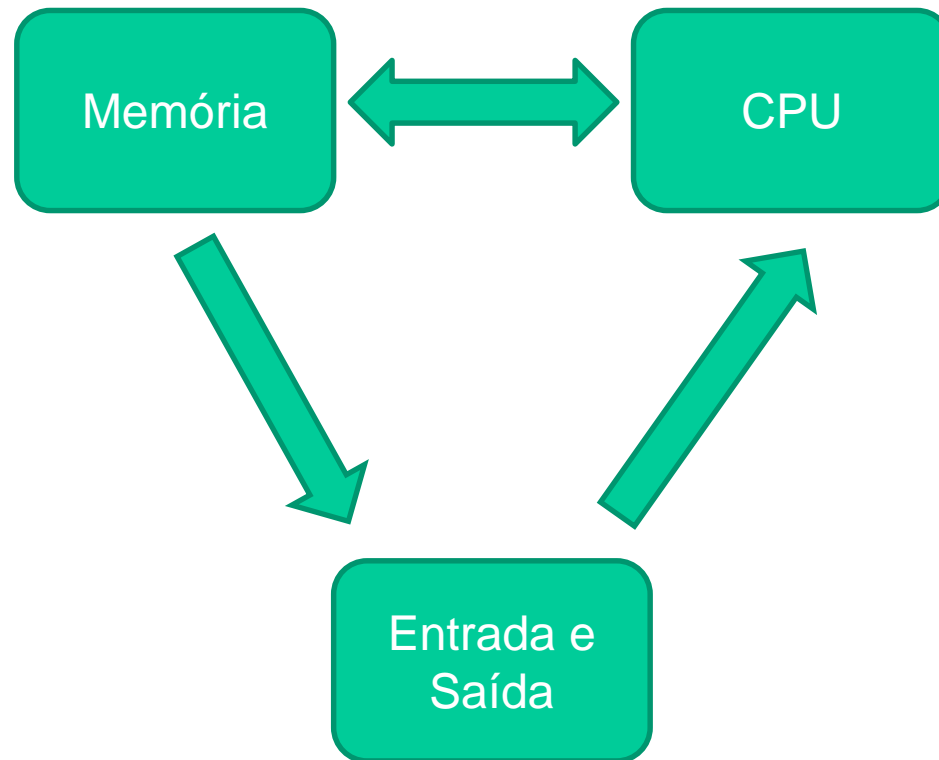


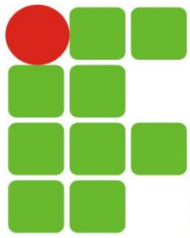
# Arquitetura Vonn Neumann;



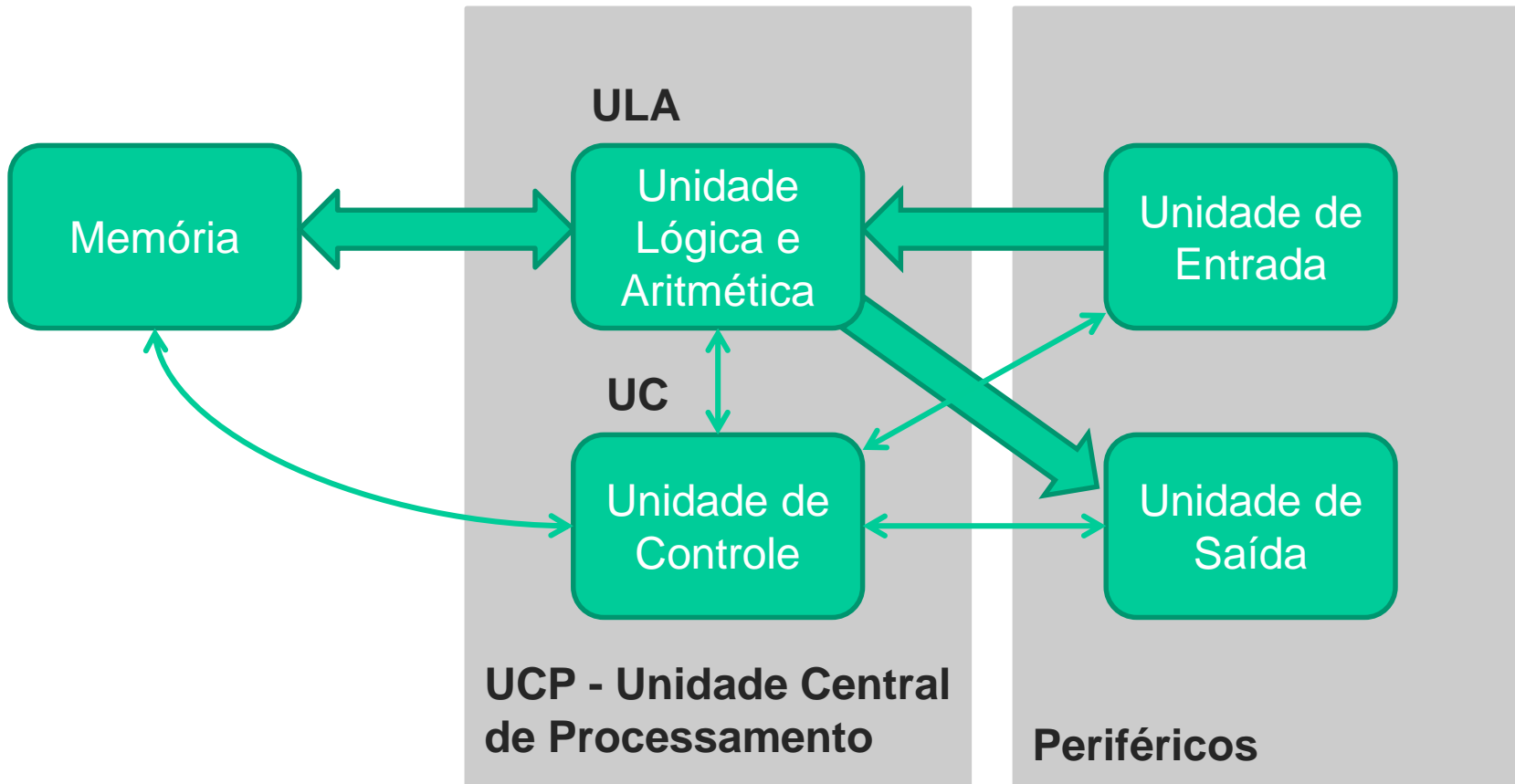


# O Modelo von Neumann





# O Modelo von Neumann



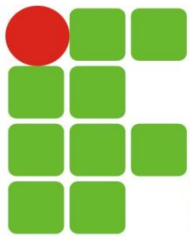


# MODELOS DE COMPUTAÇÃO

CISC - (Complex Instruction Set Computer, ou "computador com um conjunto complexo de instruções")

RISC (Reduced Instruction Set Computer, ou "computador com um conjunto reduzido de instruções")





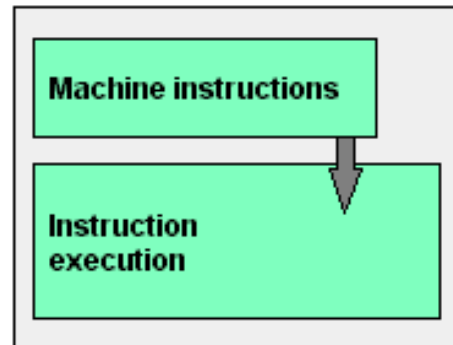
# MODELOS DE COMPUTAÇÃO

Os dois modelos de computação de processadores são:

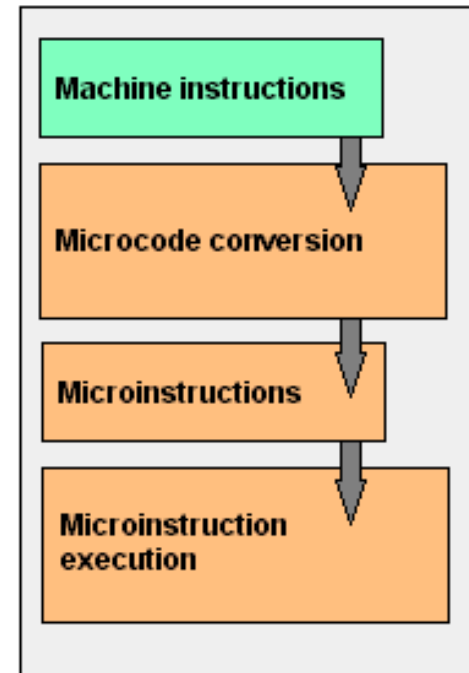
- CISC;
- RISC;

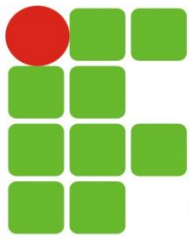
From Computer Desktop Encyclopedia  
© 1998 The Computer Language Co. Inc.

RISC

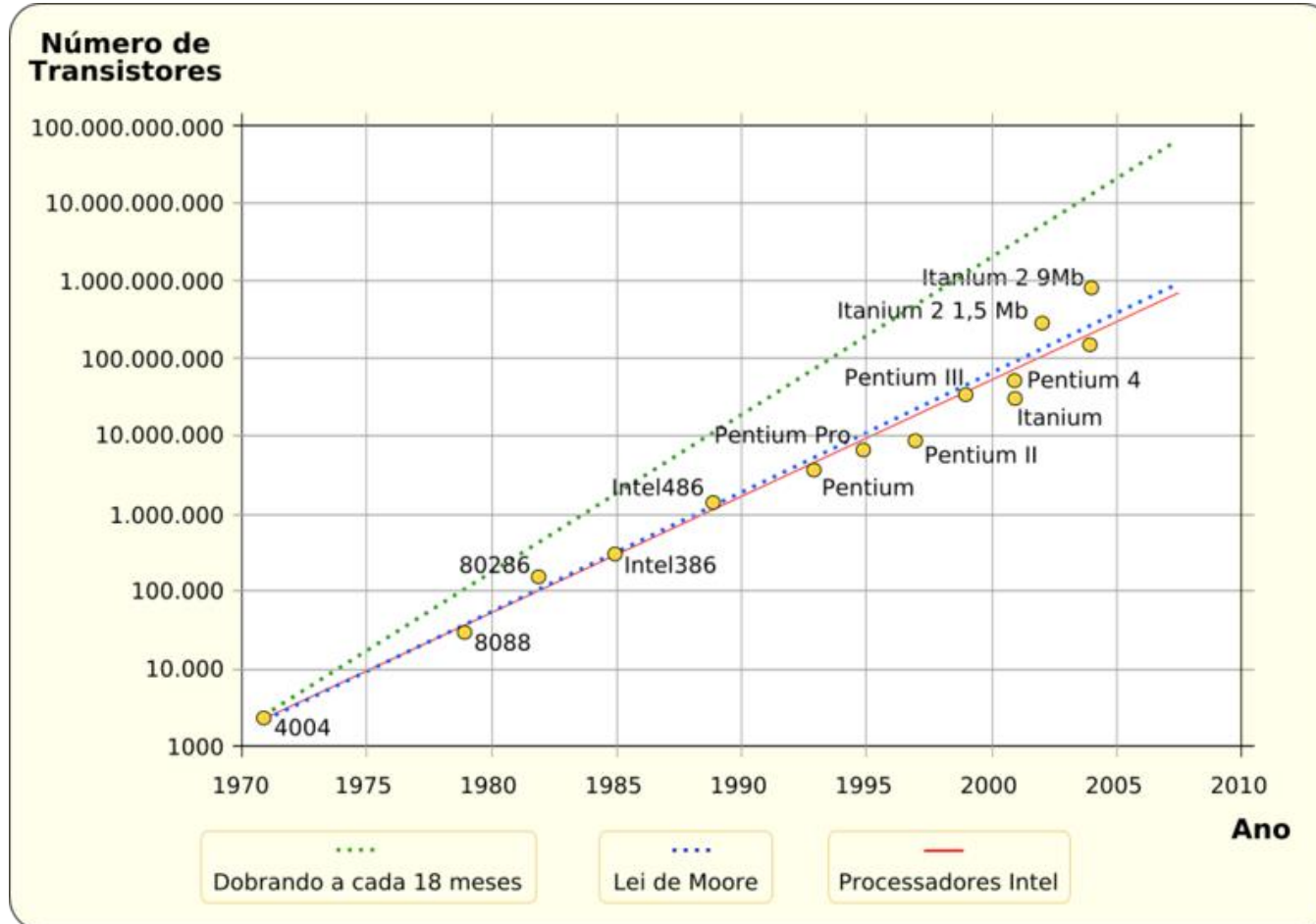


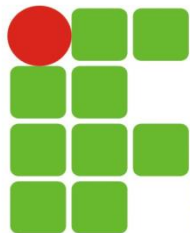
CISC





# LEI DE MOORE

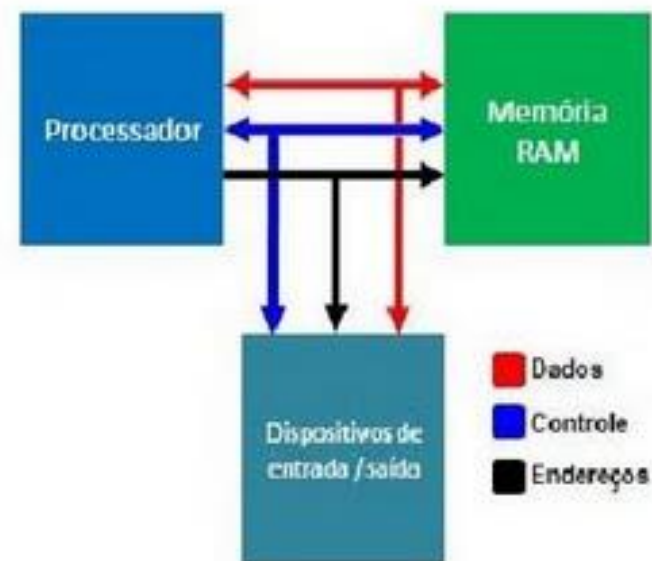


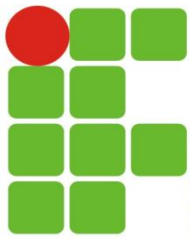


# BARRAMENTOS

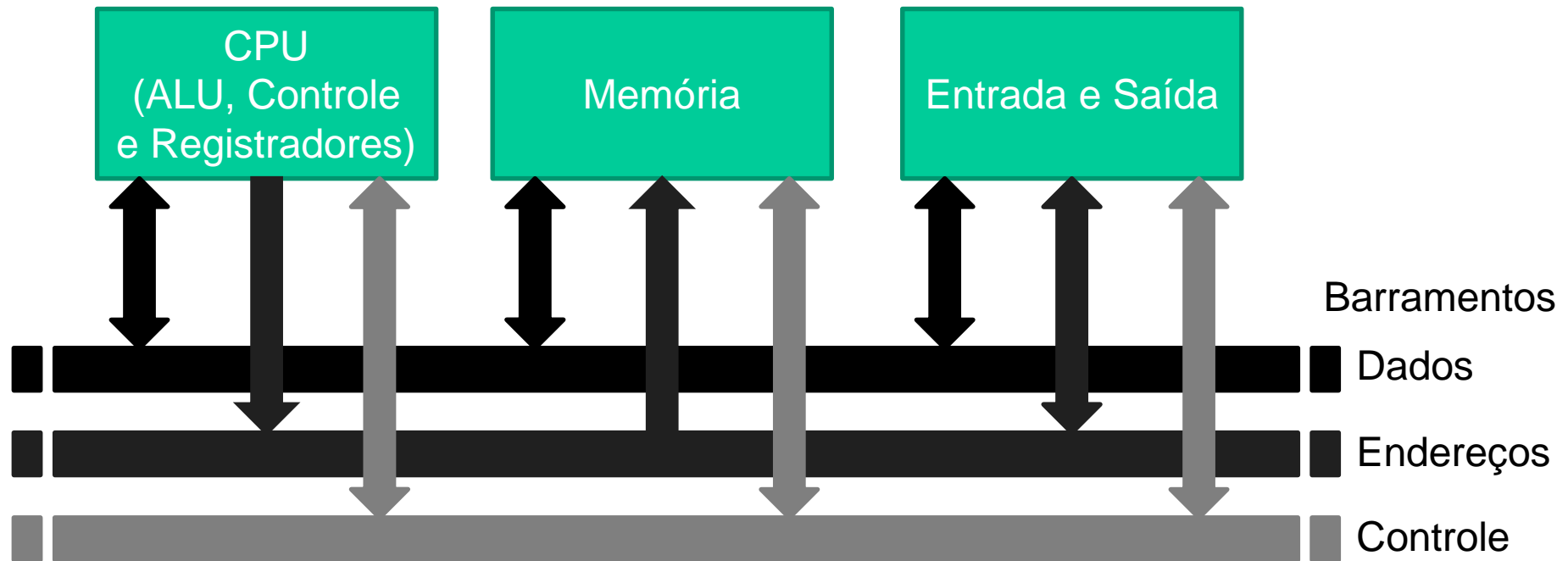
*Os principais barramentos são:*

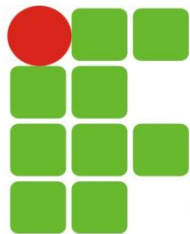
- *Barramento de Dados;*
- *Barramento de Endereço;*
- *Barramento de Controle.*



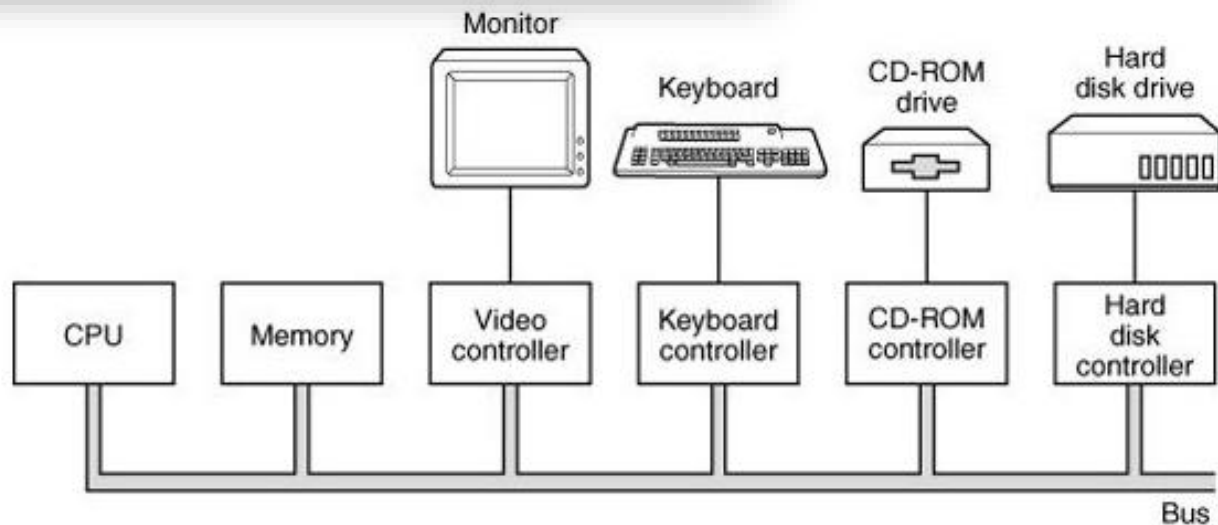
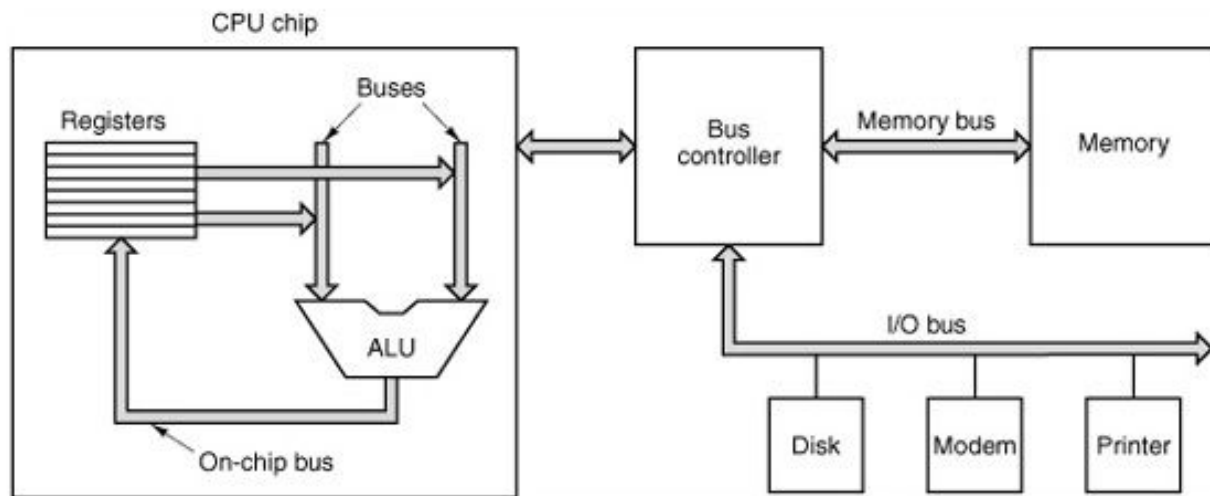


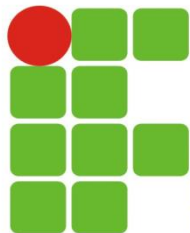
# Modelo de Barramento





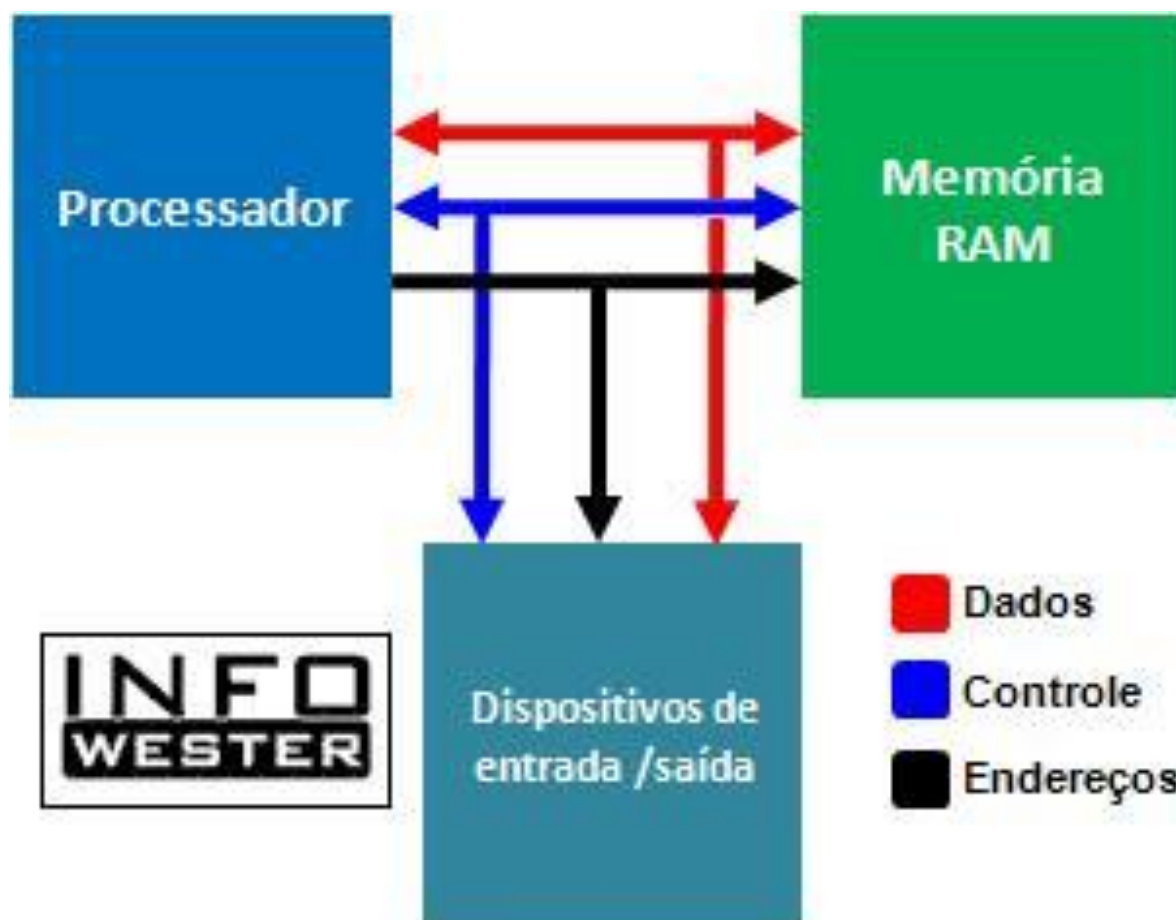
# Modelo de Barramento

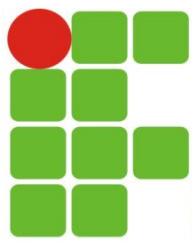




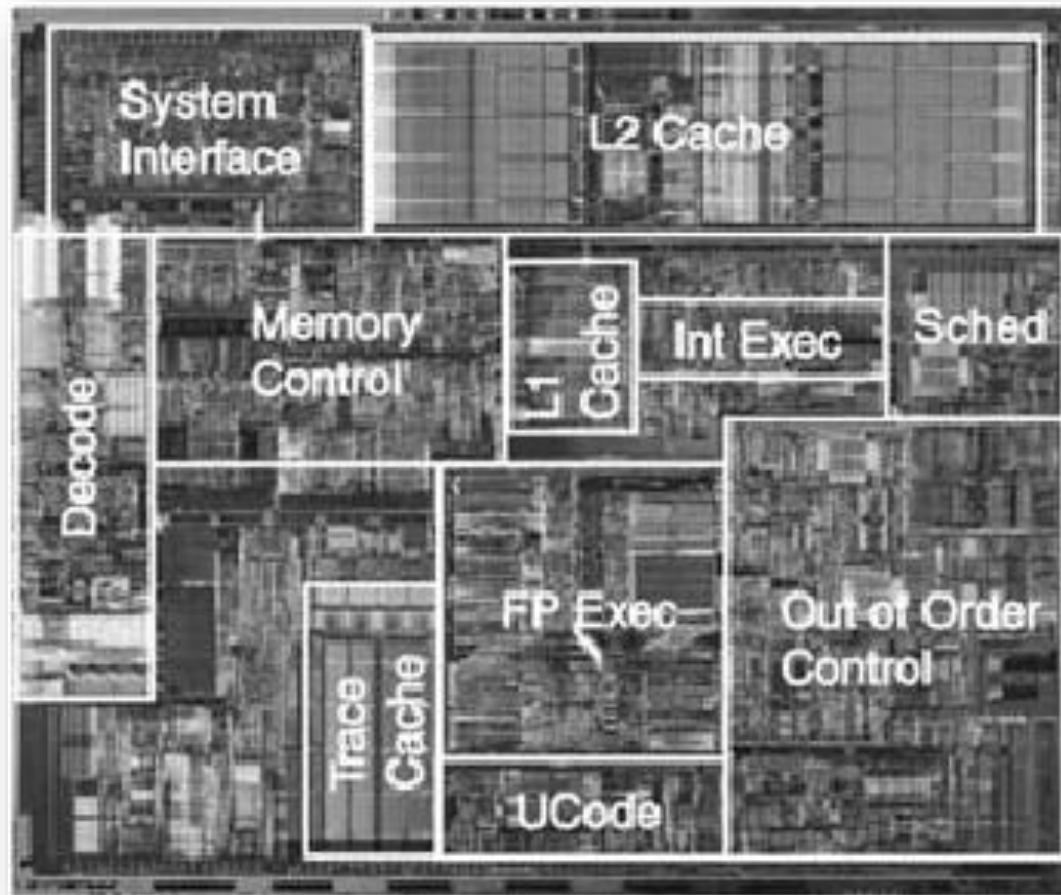
# PROCESSADOR

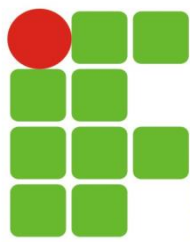
## Barramentos





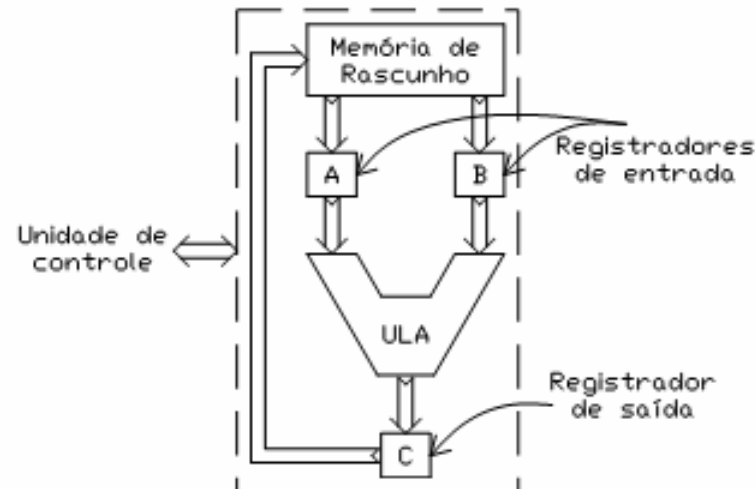
# Abstração do Processador



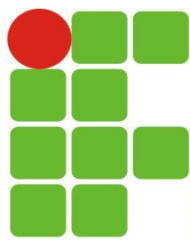


# UNIDADE LÓGICA E ARITMÉTICA

- RESPONSÁVEL POR EXECUTAR AS INSTRUÇÕES DOS PROGRAMAS;
- FUNCIONA COMO UMA GRANDE CALCULADORA ELETRÔNICA;
- EXECUTA AS OPERAÇÕES MATEMÁTICAS E LÓGICAS;
- SUBDIVIDE-SE EM: UNIDADE ARITMÉTICA, UNIDADE LÓGICA E UNIDADE DE DESLOCAMENTO.







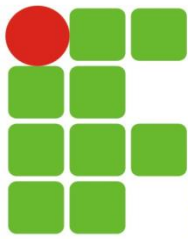
# UNIDADE DE PONTO FLUTUANTE

- É MAIS COMPLEXAS QUE A ULA;
- TRABALHA COM CÁLCULOS DE NÚMEROS REAIS;
- TRABALHAM COM OPERANDOS MAIORES:  
GERALMENTE ENTRE 64 E 128 BITS.

É um formato de representação digital de números reais, que é usada nos computadores.

Parte Inteira	Ponto ou Vírgula	Parte Fracionária
---------------	---------------------	----------------------

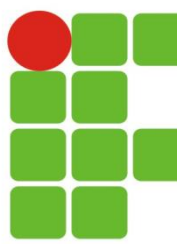




# UNIDADE DE CONTROLE

- GERA TODOS OS SINAIS QUE CONTROLAM AS OPERAÇÕES NO EXTERIOR DA CPU;
- FORNECE AS INSTRUÇÕES PARA O CORRETO FUNCIONAMENTO INTERNO DA CPU;
- É O DISPOSITIVO MAIS COMPLEXO DO PROCESSADOR;
- BUSCA INSTRUÇÕES NA MEMÓRIA PRINCIPAL E DETERMINA SEUS TIPOS;
- CONTROLA AS AÇÕES DA ULA;
- MOVIMENTA OS DADOS E INSTRUÇÕES PARA A CPU.





# UNIDADE DE GERENCIAMENTO DE MEMÓRIA

- TRANSFORMA ENDEREÇOS VIRTUAIS EM ENDEREÇOS FÍSICOS;
- ADMINISTRA A MEMÓRIA PRINCIPAL DO COMPUTADOR;





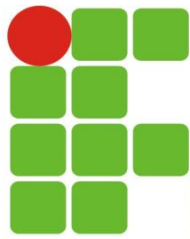
# Registradores

São unidades de memória capazes de armazenar  $n$  bits. Os registradores estão no topo da hierarquia de memória, sendo assim, são o meio mais rápido e caro de se armazenar um dado.

Lembrando que os registradores são tipicamente usados como um dispositivo de armazenamento temporário.

REGISTRADORES DE USO GERAL, DE SEGMENTO, DE PONTEIRO E DE ESTADO;

OS MAIS IMPORTANTES SÃO: CONTADOR DE PROGRAMA (PC) E REGISTRADOR DE INSTRUÇÃO (IR).



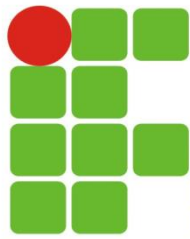
# Registradores

Registrador acumulador (EAX): registrador acumulador expandido de arquiteturas 8086

Registradores de base (EBX): registrador de base estendido de arquiteturas 8086

Registrador contador (ECX): registrador de laços de repetição em arquiteturas 8086

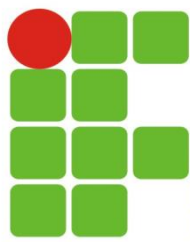
Registradores de dados (EDX): registrador estendido de "contas" com palavras de arquiteturas 8086



# Registrador acumulador

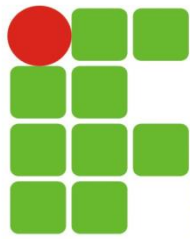
Realiza operações aritméticas, acessa portas de entrada e saída, faz chamadas de interrupções

# Registradores de base



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE  
Campus Santa Cruz

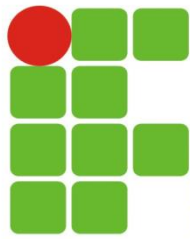
Registradores de base são registradores que recebem o endereço-base de um dado objeto. Este tipo de registrador oferece aos programadores um subterfúgio para a criação de "ponteiros" (variáveis, contendo o caminho para um endereço no software).



# Registrador contador

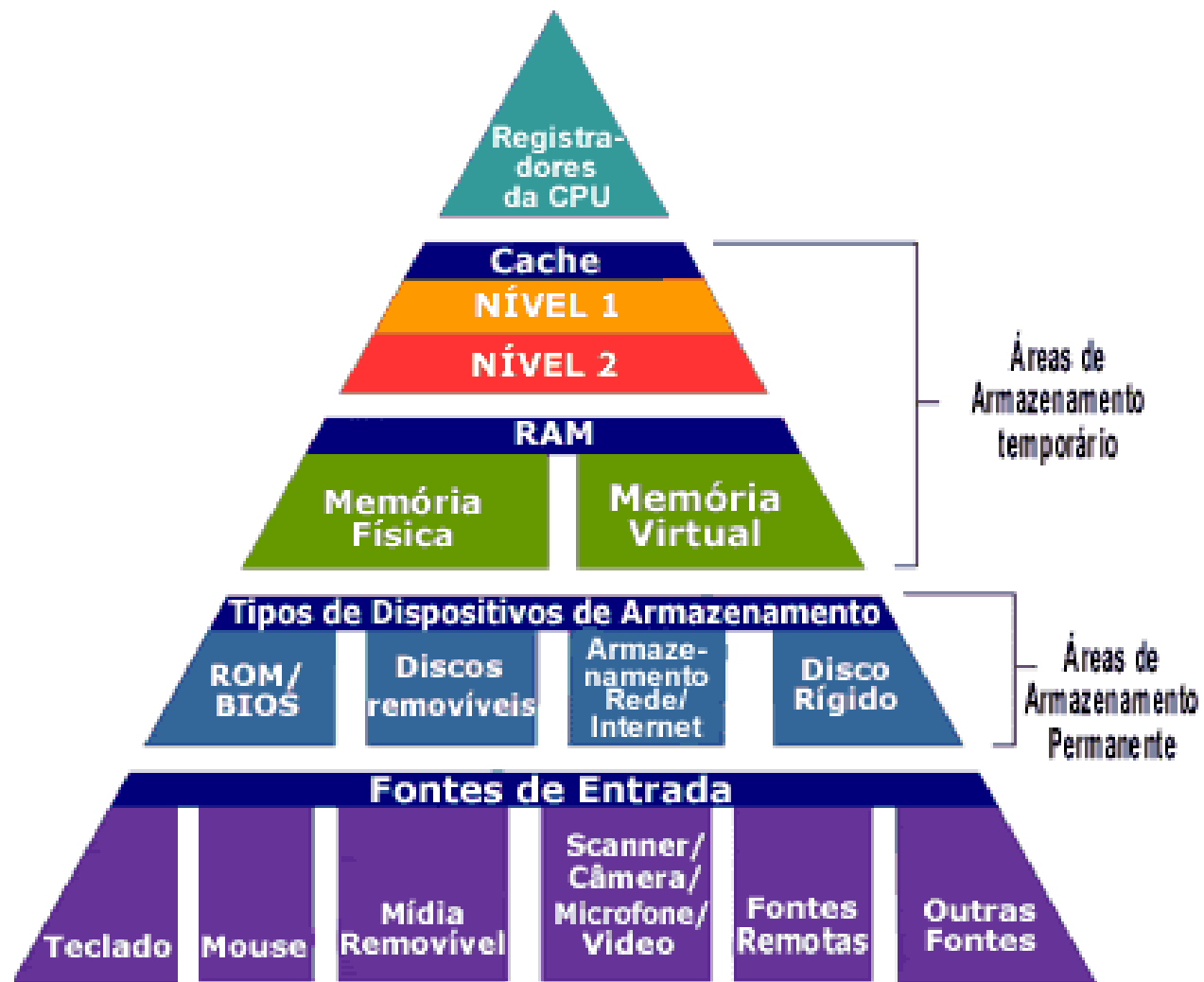
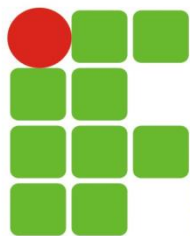
Conta todos as repetições e deslocamentos, podendo ser atribuído um valor de interrupção.

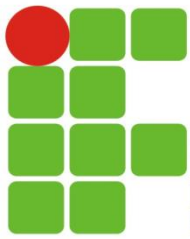




# Registradores de dados

Em algumas CPUs antigas e mais baratas, é um registrador de dados especial, conhecido como acumulador, e é utilizado implicitamente em muitas operações. O acumulador funciona como um recipiente onde são colocados e somados valores de cálculos e comparações.

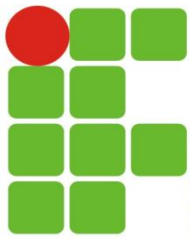




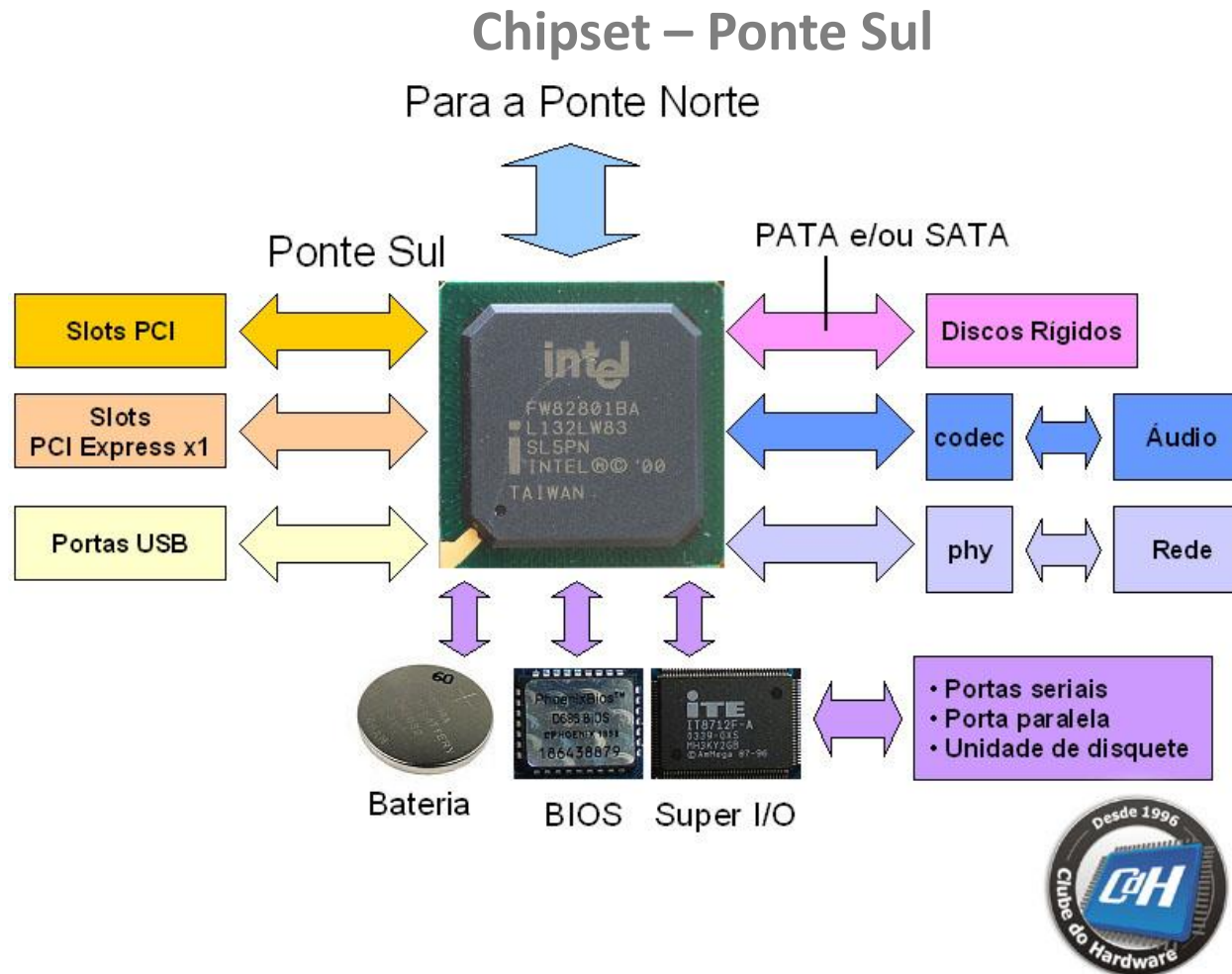
# Chipset

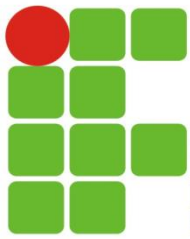
*Chipset é o nome dado ao conjunto de chips (set significa “conjunto”, daí o seu nome) usado na placa-mãe.*





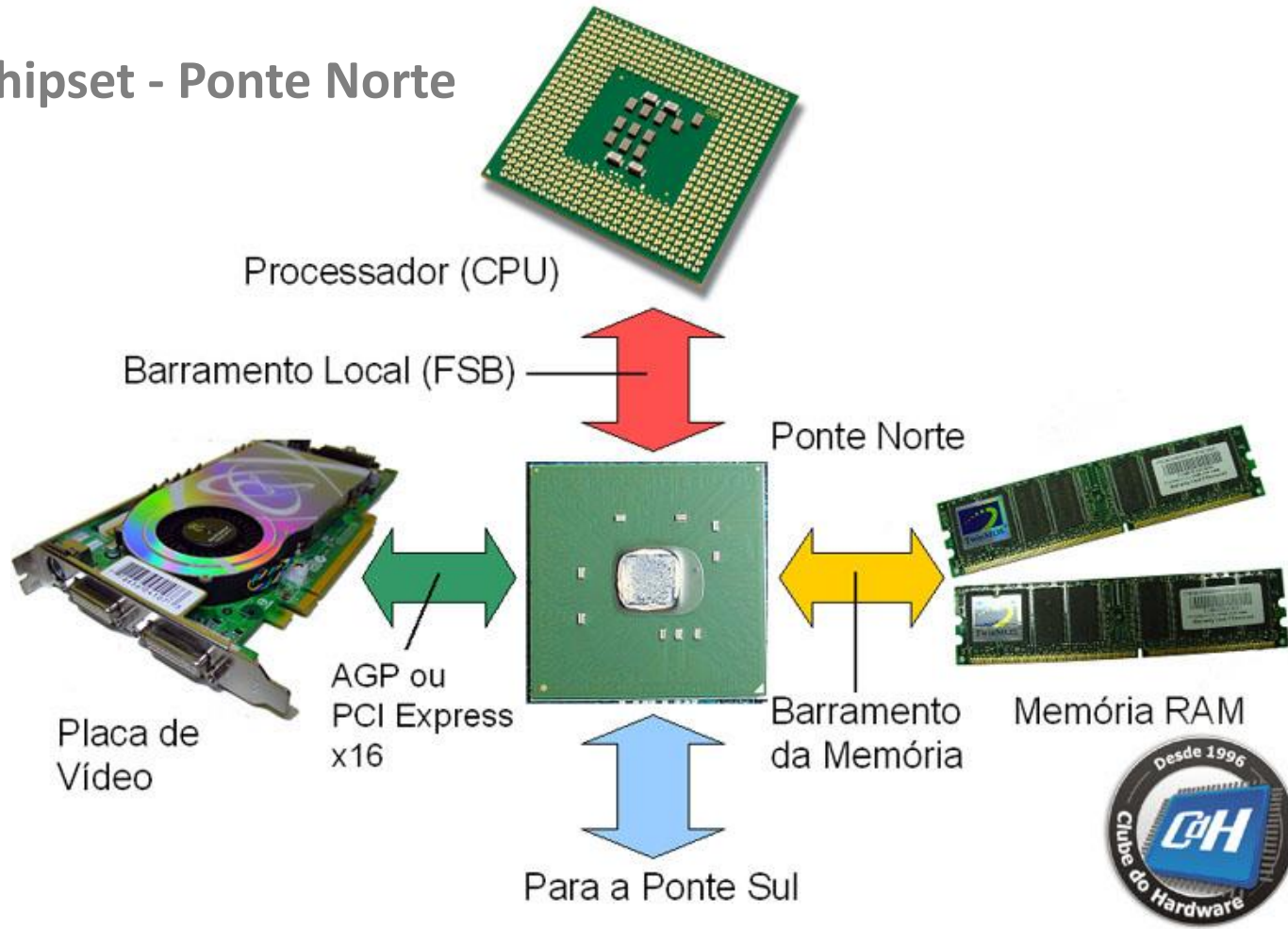
# Ponte Sul

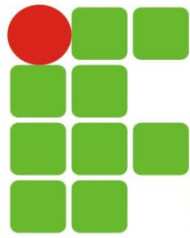




# Ponte Norte

## Chipset - Ponte Norte

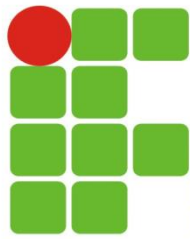




# Pipeline

Método de processamento usado em um computador, que permite o rápido processamento paralelo dos dados.

O processamento em pipeline é feito através da sobreposição de operações através de um pipe, ou uma parte da memória que transfere informações de um processo para outro.



# Pipeline

Os Pipelines, levam muitas estruturas diferentes:

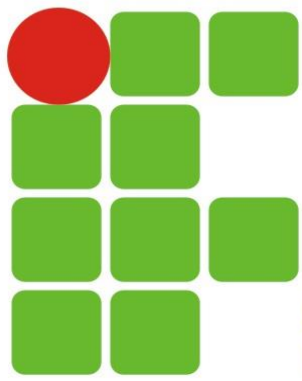
- Superescalar,
- Dinâmico,
- Out-of-Order, entre outros,

Minimizar o uso de bolhas nos processadores, assim maximizando o uso do processador.

processadores ficam mais sofisticados

diminuído de tamanho através do uso de tecnologias de fabricação mais precisas.

Problema de dissipação de calor.

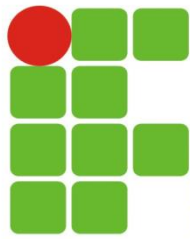


**INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**  
RIO GRANDE DO NORTE  
Campus Santa Cruz

# Organização de computadores

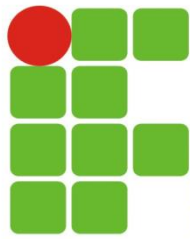
Aula 06





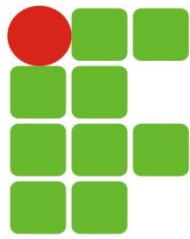
# MIPS

- ✓ Microprocessor without interlocked pipeline stages (microprocessador sem estágios intertravados de pipeline)
- ✓ é uma arquitetura de microprocessadores RISC desenvolvida pela MIPS Computer Systems.
- ✓ é uma arquitetura baseada registrador, ou seja, a CPU usa apenas registradores para realizar as suas operações aritméticas e lógicas.



# MIPS

- ✓ Existem outros tipos de processadores, tais como processadores baseados em pilha e processadores baseados em acumuladores.
- ✓ Processadores baseados no conjunto de instruções do MIPS estão em produção desde 1988.
- ✓ Revisões que foram introduzidas são MIPS I, MIPS II, MIPS III, MIPS IV e MIPS V.
- ✓ Cada revisão é um super conjunto de seus antecessores.
- ✓ definição alterada para definir um conjunto de instruções MIPS32 de 32 bits e um MIPS64 de 64 bits.



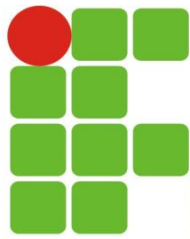
# MIPS

- ✓ **Resumo:**
- ✓ A arquitetura MIPS é do tipo load-store, ou seja, as operações lógicas e aritméticas são executadas exclusivamente entre registradores da arquitetura ou entre constantes imediatas e registradores.
- ✓ As operações de acesso à memória só executam ou uma leitura da memória (load) ou uma escrita na memória (store).



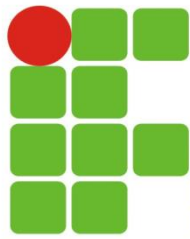
# MIPS

- ✓ **Resumo:**
  - ✓ Devido à característica load-store, o processador disponibiliza um conjunto relativamente grande de registradores, para reduzir o número de acessos à memória externa, pois estes últimos representam perda de desempenho em relação a operações entre registradores internos ao processador.



# Outras Arquiteturas

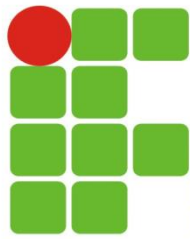
- ✓ Arquiteturas baseadas em acumulador
- ✓ Mantêm todos os dados em memória
- ✓ Realiza operações aritméticas entre um conteúdo que está em memória e um, ou poucos registradores de dados, os denominados acumuladores.
- ✓ Exemplo: Um `for (i=0; i < 1000; i++)`
  - ✓ caso `i` esteja armazenado em memória, tem-se 2000 acessos à memória,
  - ✓ tempo de acesso a um registrador é (10 a 100 vezes) menor que o da memória



# MIPS

## Outras características

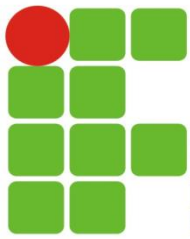
- ✓ Dados e endereços são de 32 bits, outros: Itanium da Intel (64 bits).
- ✓ O endereçamento de memória orientado a byte
- ✓ cada endereço de memória é um identificador de uma posição e guarda apenas 1 byte (8 bits).
- ✓ palavra do processador armazenada em memória ocupa 4 posições, tipicamente em quatro endereços consecutivos.
- ✓ O banco de registradores da arquitetura possui 32 registradores de uso geral



# MIPS

## Outras características

- ✓ formato regular para as instruções. Todas possuem exatamente o mesmo tamanho,
- ✓ ocupam 1 palavra em memória (32 bits), ou seja, o equivalente a 4 endereços consecutivos em memória.



# Ponto Fixo

- É a representação de números inteiros em binário;
- É direta e não se preocupa com sinal, nem com formatação dos bits.

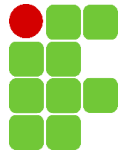
## EXEMPLO:

6.75 com 4 bits para inteiros e 4 bits para a fração

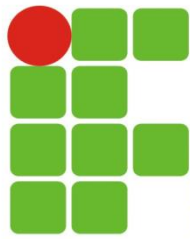
6 0.75  
0110.1100

01101100 ←

$2^2 + 2^1 + 2^{-1} + 2^{-2} = 6.75$

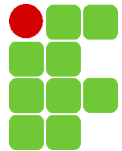


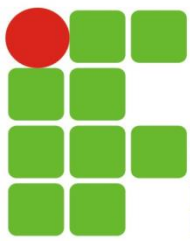




# OVERFLOW

- Situação anormal que ocorre quando o resultado de uma operação não pode ser representado com um dada quantidade de bits;
- Tratamento de overflow depende do compilador e do sistema operacional;





# Ponto Fixo → Operações

## Adição:

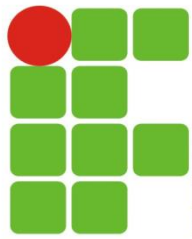
- Dígitos são somados bit a bit, da direita para a esquerda;
- Carries (vai-um) são passados para o próximo **dígito à esquerda**.

$$\begin{array}{r} 11 \leftarrow \text{carries} \\ 1011 \\ + 0011 \\ \hline 1110 \end{array}$$

## Subtração:

- Nega-se o subtraendo e soma-se um (complemento de 2);
- Soma-se o resultado anterior com o diminuendo.

$$\begin{array}{r} (+5) \\ - (+2) \\ \hline (+3) \end{array} \quad \begin{array}{r} 0101 \\ - 0010 \\ \hline \end{array} \Rightarrow \begin{array}{r} 0101 \\ + 1110 \\ \hline 10011 \\ \uparrow \\ \text{ignore} \end{array}$$

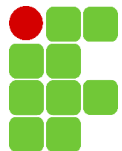


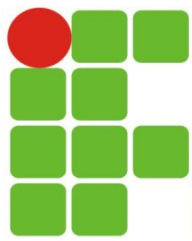
# Ponto Fixo → Operações

## Multiplicação:

1. Se bit da posição  $i$  do multiplicador for **1**: desloca-se o multiplicando em  $i$  posições;
2. Se bit do multiplicador for **0**, coloca-se **0**;
3. Somar cumulativamente os produtos parciais.

0010	(multiplicando)
× 0111	(multiplicador)
<hr/>	
0010	
0010	
0010	
+ 0000	
<hr/>	
0001110	(produto)





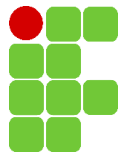
# Ponto Fixo → Operações

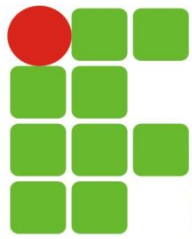
## Divisão:

1. Registradores do Resto e Divisor terão 8 bits;
2. Registrador do Quociente terá 4 bits;
3. No início a parte menos significativa do Registrador Resto conterá o dividendo e os 4 bits mais significativos do registrador Divisor, o valor inicial do divisor.

```
Dividendo → 10010011' | 1011 ← Divisor
                -1011
                001110
                -1011
                001111
                -1011
                100 ← Resto

00001101 ← Quociente
```





# Ponto Fixo → Par de registradores

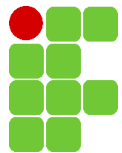
Registradores especiais para guardar resultado das multiplicações e divisões, Lo e Hi.

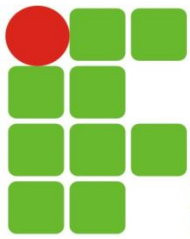
## Multiplicação:

- Hi – armazena a parte mais significativa;
- Lo – armazena a parte menos significativa.

## Divisão:

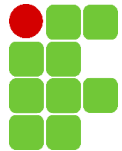
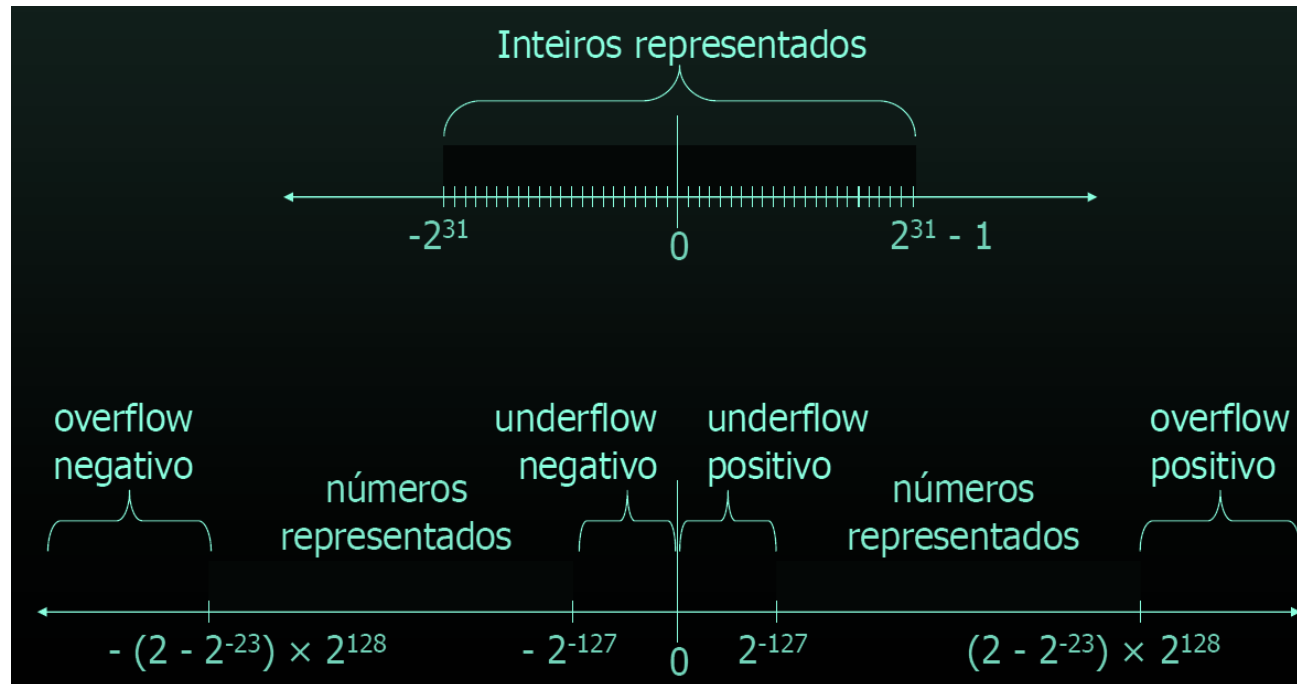
- Hi – armazena o resto;
- Lo – armazena o quociente.

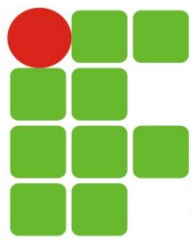




# Ponto Flutuante

Números representados em ponto flutuante não são igualmente espaçados, tal como na notação de ponto fixo.





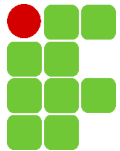
# Ponto Flutuante → Outros

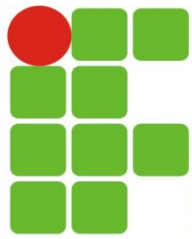
## Adição e subtração:

Ambos operandos precisam ter o mesmo expoente.

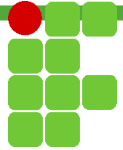
## Divisão e multiplicação:

São mais simples de serem calculadas.





# VIA DE DADOS

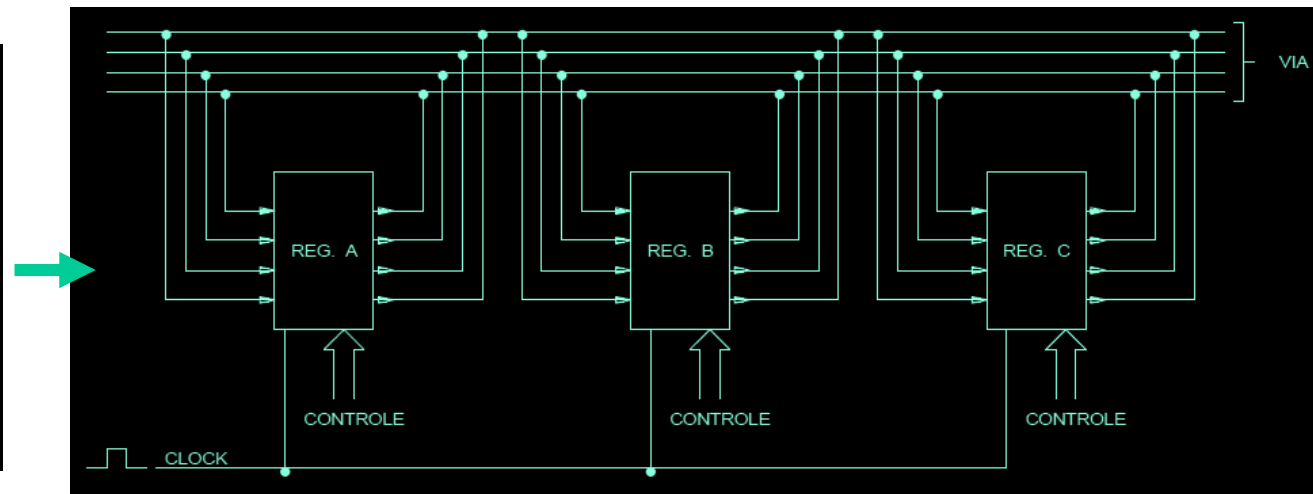
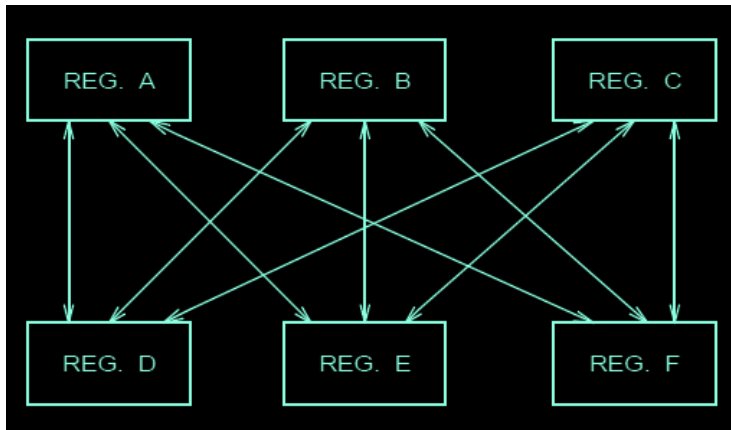


## Via de Dados (datapath):

- Parte do processador que contém o hardware necessário para execução de todas as operações requeridas pelo computador.

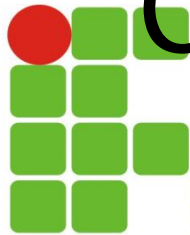
## Via de Controle:

- Parte do processador que comanda as ações da via de dados.

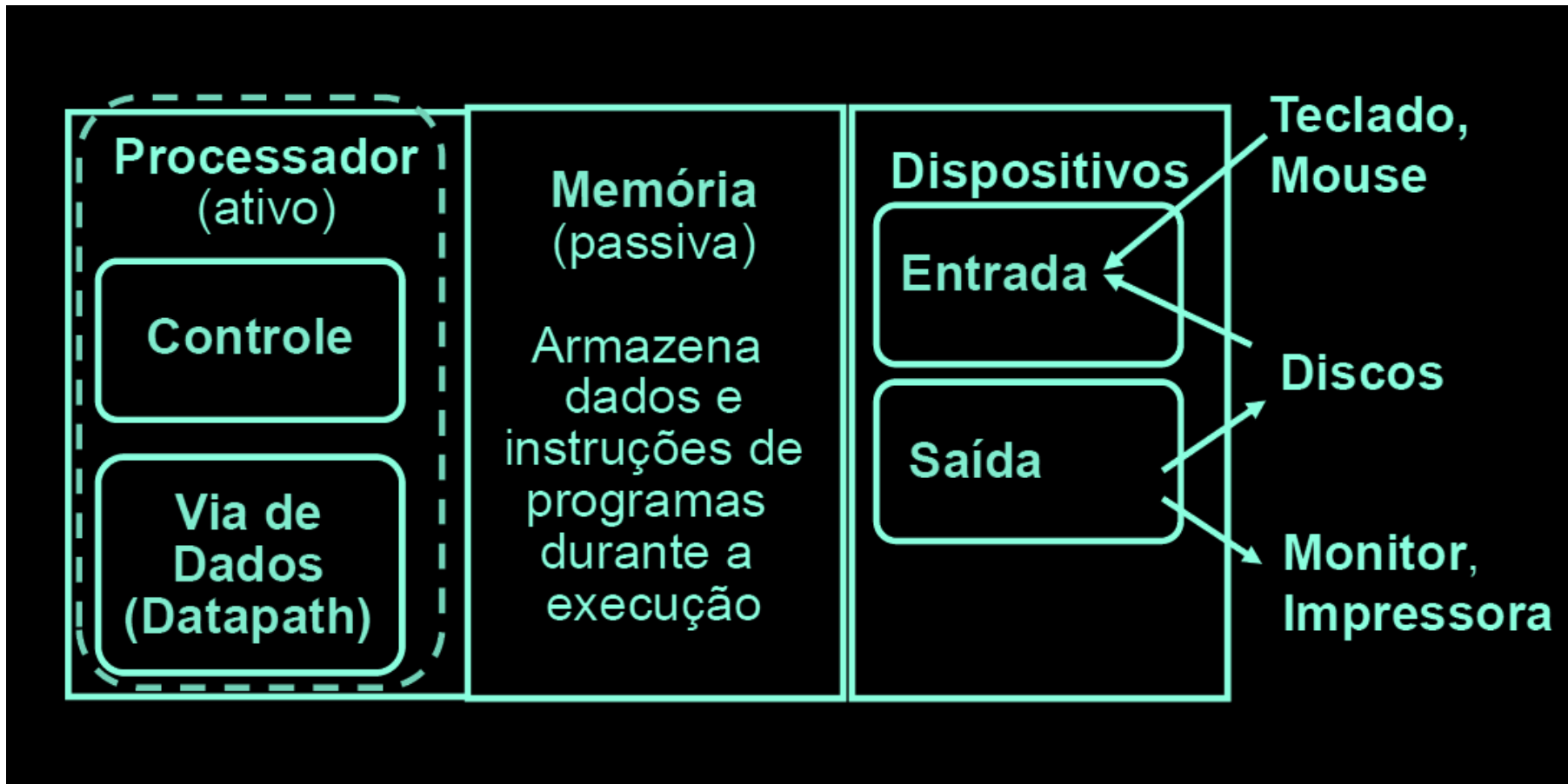


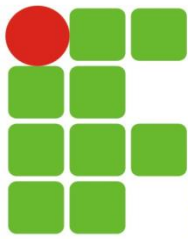


# Componentes Clássicos das vias de dados e controle



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE  
Campus Santa Cruz



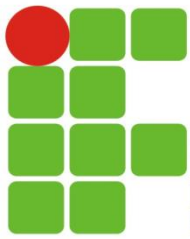


# Monociclo X Multiciclo

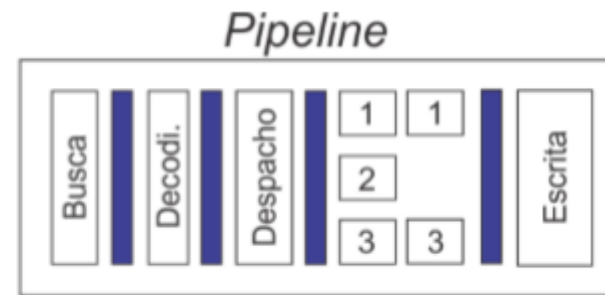
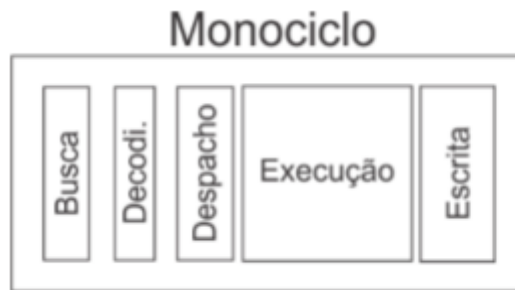
Arquitetura monociclo (todas as instruções demoram um ciclo),

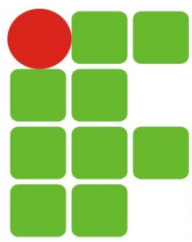
O período de relógio do processador é determinado pela instrução mais longa

Multiciclo- para uma arquitetura pipeline, que dividem as tarefas em estágios e unidades funcionais heterogêneas que podem executar instruções de forma paralela. Com isso, o tempo de um ciclo pode ser reduzido, aumentando a performance



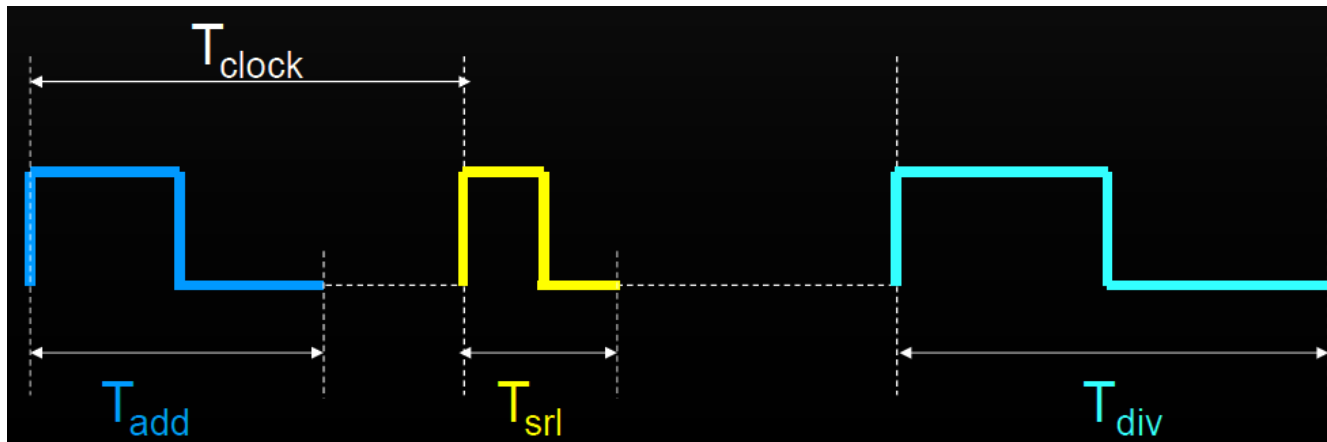
# Monociclo X Multiciclo

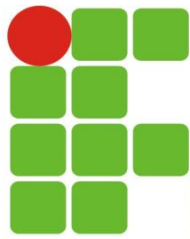




# Implementação de Instruções Ciclo Único

- Cada instrução é executada em um 1 ciclo de clock;
- Ciclo de clock deve ser longo o suficiente para executar a instrução mais longa;
- Desvantagem: velocidade global limitada à velocidade da instrução mais lenta.



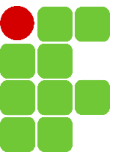


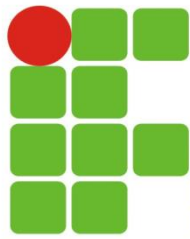
# Ciclo Único: Inclusão de Multiplexadores

Como várias unidades funcionais são compartilhadas para diferentes finalidades é preciso:

- Incluir multiplexadores;
- Expandir os multiplexadores existentes.

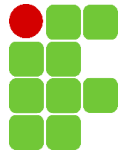
Os MUX e registradores são muito pequenos se comparados a uma unidade de memória ou ALU portanto, a economia na eliminação destes compensa os custos de adição daqueles.

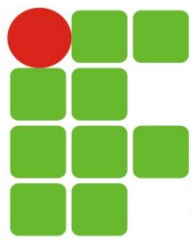




# Ciclo Único: Linhas de Controle

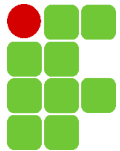
- As unidades de estado visíveis ao programador (PC, memória, banco de registradores) e IR precisarão de sinais de controle de escrita;
- Memória precisará de um sinal de leitura.

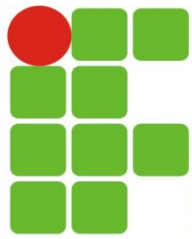




# Ciclo Único: Implementação de Controle

- Todos os elementos de estado (registradores, memória) têm o clock como uma entrada implícita;
- Unidade de controle pode definir todos os sinais de controle baseada no campo opcode da instrução, exceto Branch, que depende da saída zero da ALU no caso de instruções de desvio.

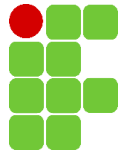




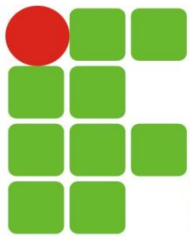
# Ciclo Único: Desempenho da arquitetura

- Ciclo de clock tem mesma duração para todas as instruções: ciclos de clocks por instrução (CPI) = 1;
- Ciclo de clock é longo suficiente para executar a instrução mais demorada, que neste caso é lw;
- Apesar do CPI ser 1, o período de clock ( $T_{clock}$ ) é elevado o que prejudica o desempenho do CPU.

$$T_{exec} = N_{instr} \times CPI \times T_{clock}$$

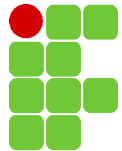
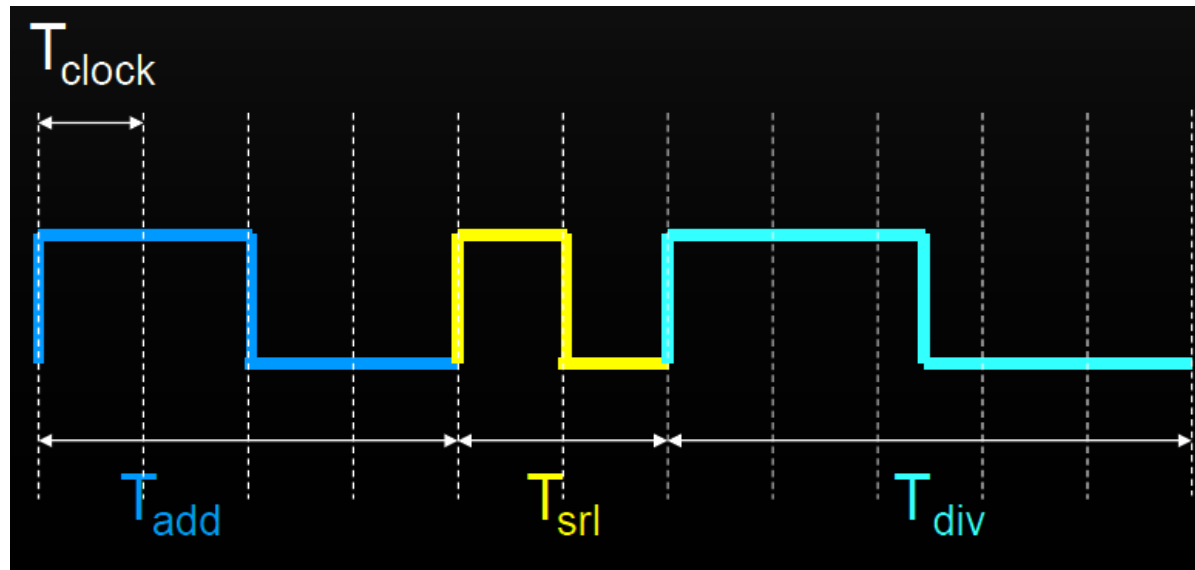


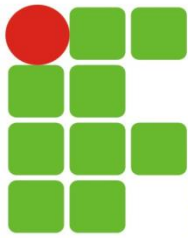




# Implementação Multi Ciclo

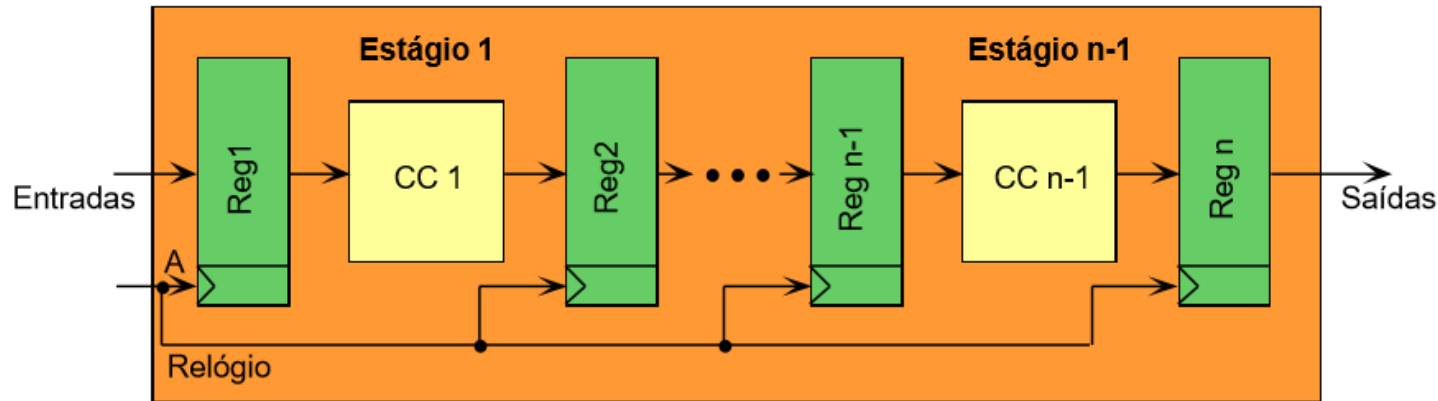
- Quebra o ciclo de execução em vários passos;
- Executa cada passo em um ciclo de clock;
- Vantagem: cada instrução usa apenas o número de ciclos que ela necessita.

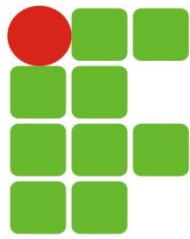




# Multi ciclo

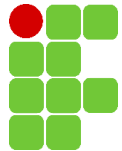
- ✓ Cada estágio realiza uma parte do trabalho
- ✓ Registradores usados para isolar os resultados – são as chamadas “barreiras temporais”





# Multi Ciclo: Via de dados

- Instruções MIPS levam o tempo de 3 a 5 etapas de execução;
- As duas primeiras etapas (IF e ID) são comuns a todas instruções;
- As células vazias na tabela indicam que a classe de instruções representada pela coluna leva menos ciclos;
- No hardware multi-ciclo, esses ciclos não ficam ociosos, pois uma instrução é executada logo após a anterior.

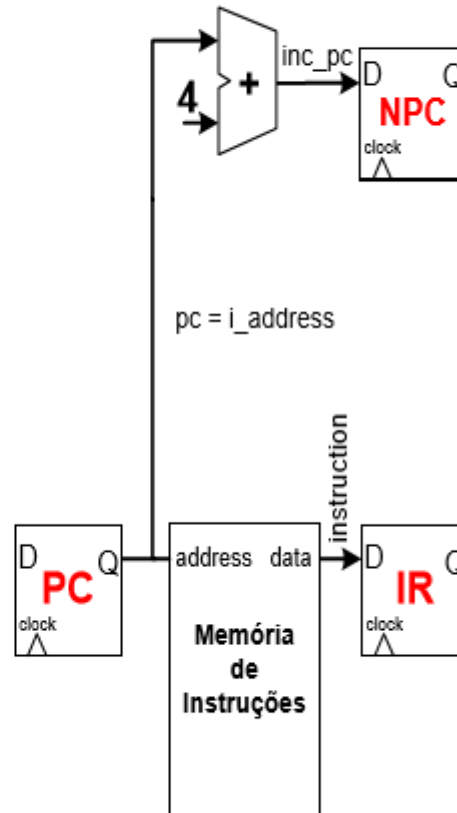


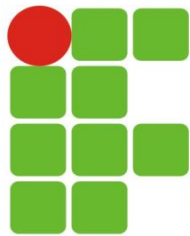


# Primeiro Estágio

## Primeiro estágio

- **Comum a todas as instruções**
  - Incrementa o PC
  - Armazena instrução no IR
  - **NPC** é novo registrador
    - Porquê existe?
      - Um motivo: não gerar transitórios em saltos
      - Ver especificação das instruções!

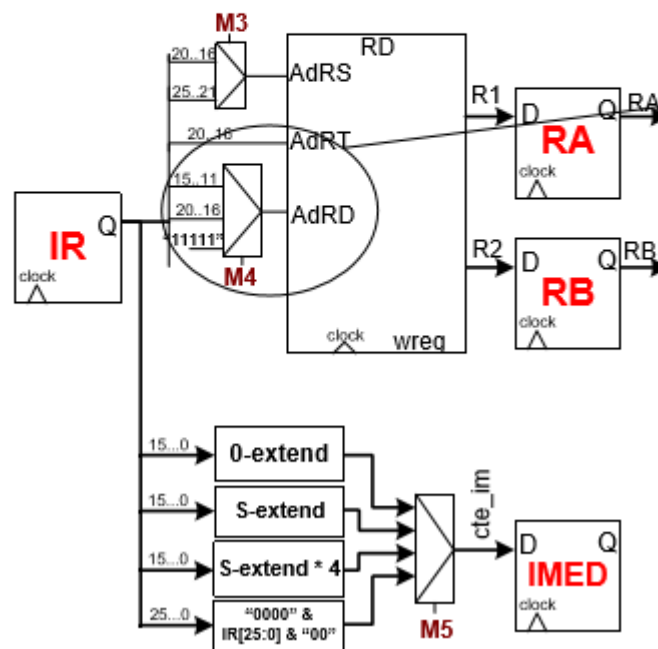




# Segundo Estágio

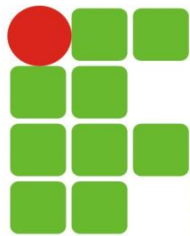
## Segundo estágio

- Comum para todas as instruções
- Leitura do Banco de Registradores
- Determinação da constante imediata



Controle de escrita no banco de registradores.

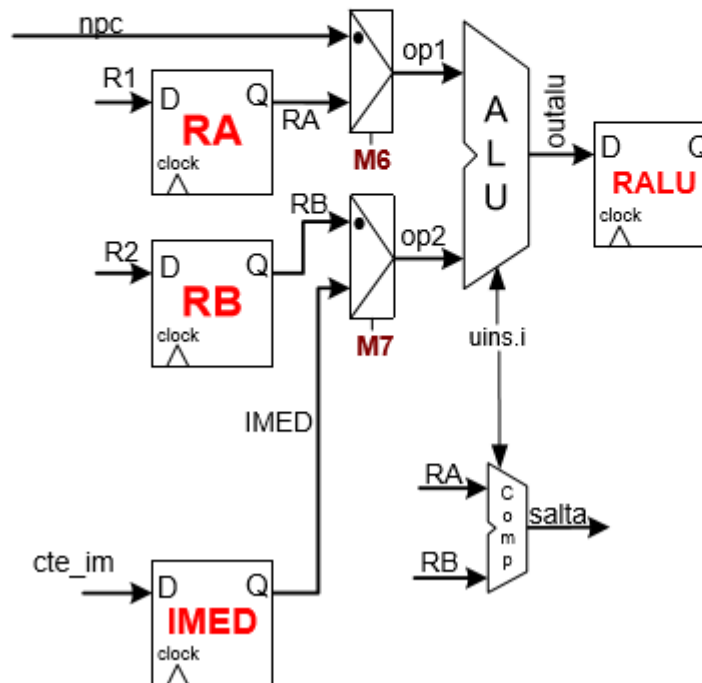
Discutido no último estágio

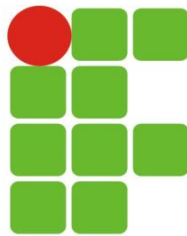


# Terceiro Estágio

## Terceiro estágio

- Comum a todas as instruções
- Realiza a operação com a ULA
- Determina um *flag* (salta) que indica se a condição de uma instrução de salto é verdadeira ou não

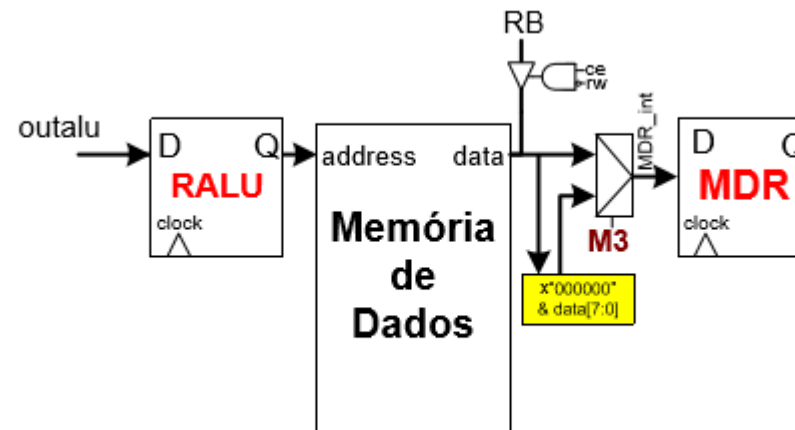


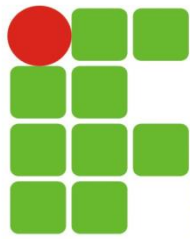


# Quarto estágio

## Quarto estágio

- Utilizado apenas pelas instruções LW/LB/SW/SB
- Leitura (LB/LW)
  - Valor endereçado pelo registrador RALU é lido da memória de dados e escrito no registrador MDR
  - Valor endereçado pelo registrador RALU recebe o conteúdo do registrador RB quando for instrução de escrita

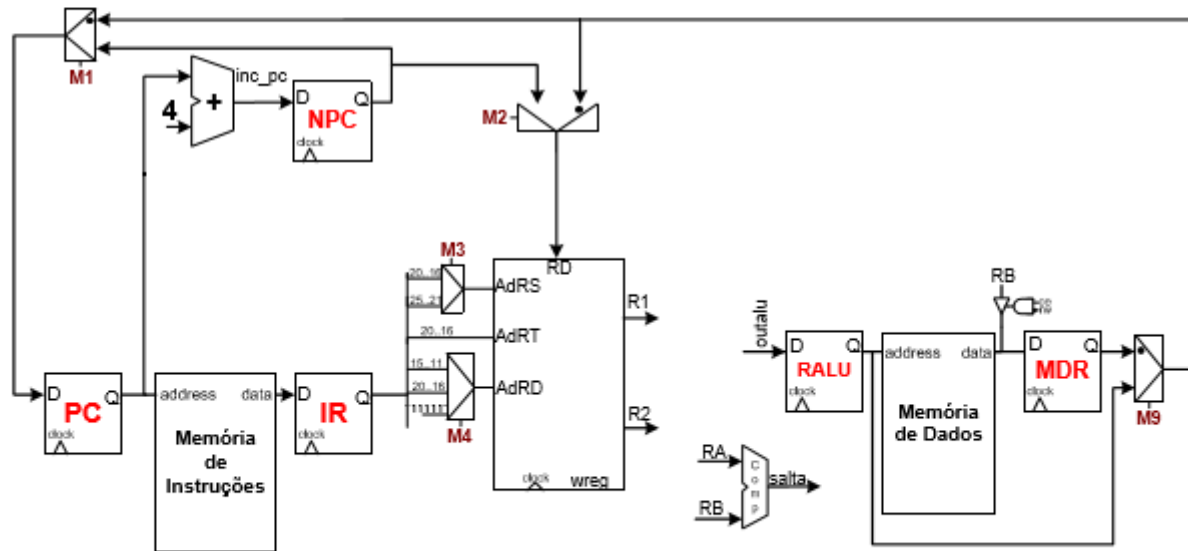




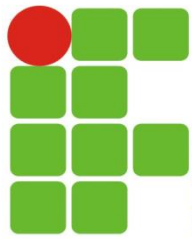
# Quinto estágio

## Quinto estágio

- Estágio que conclui a execução de uma determinada instrução
- Denomina-se em inglês *write-back*
- Depende da instrução corrente

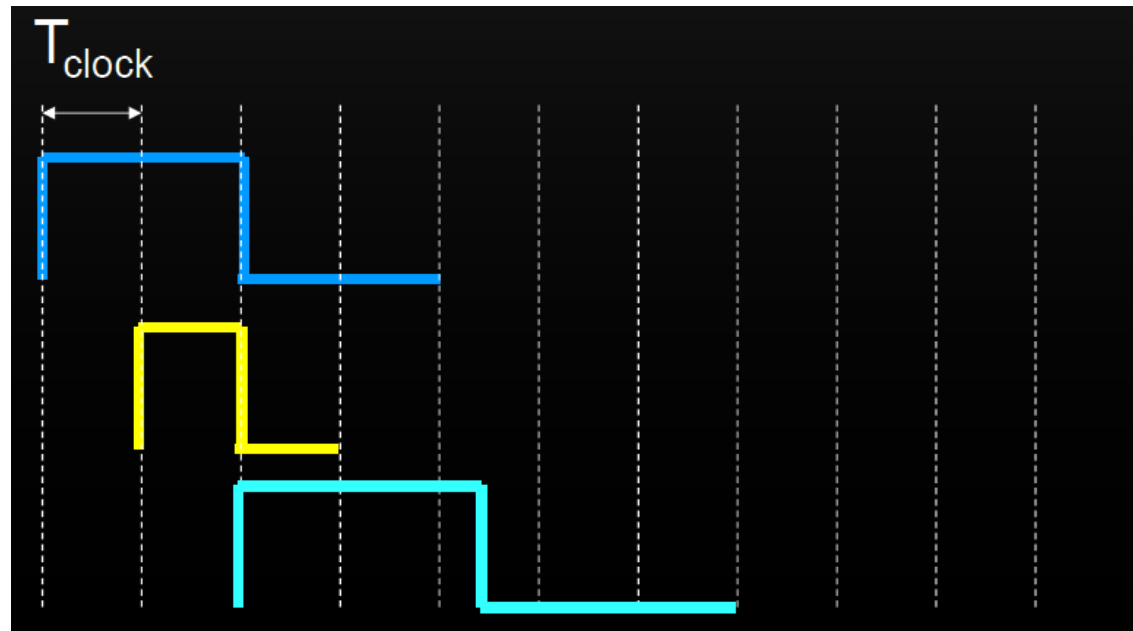


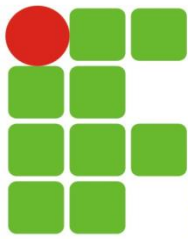




# Implementação Pipelined

- Cada instrução é executada em múltiplos ciclos;
- Executa uma etapa de cada instrução em cada ciclo;
- Processa múltiplas instruções em paralelo.

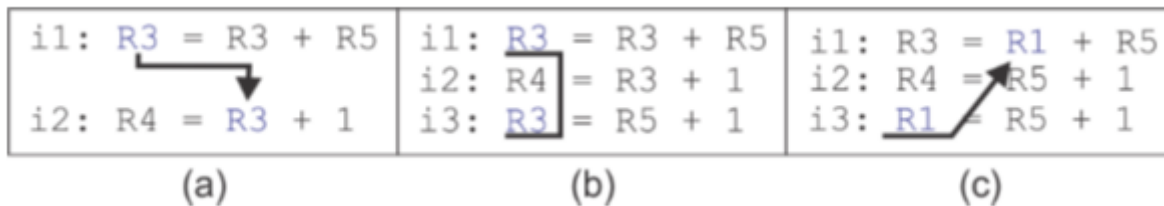


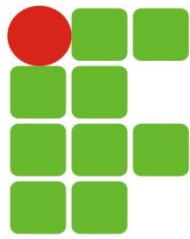


# Pipeline

A utilização de diversas unidades funcionais com pipeline trouxe uma nova gama de problemas inexistentes em arquiteturas monociclo, que entre elas são as dependências, que podem ser classificadas em:

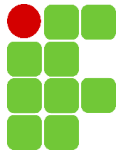
- Dependência verdadeira (Figura 2.a), que acontece quando um dos operandos de uma instrução dependem dos valores de uma instrução anterior;
- Dependência de saída (Figura 2.b), que acontece quando instruções escrevem em um mesmo registrador;
- Antidependência (Figura 2.c), que acontece quando uma instrução escreve em um registrador, e uma instrução anterior tem esse registrador como operando;

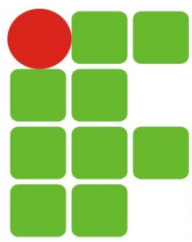




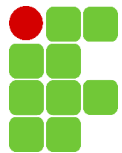
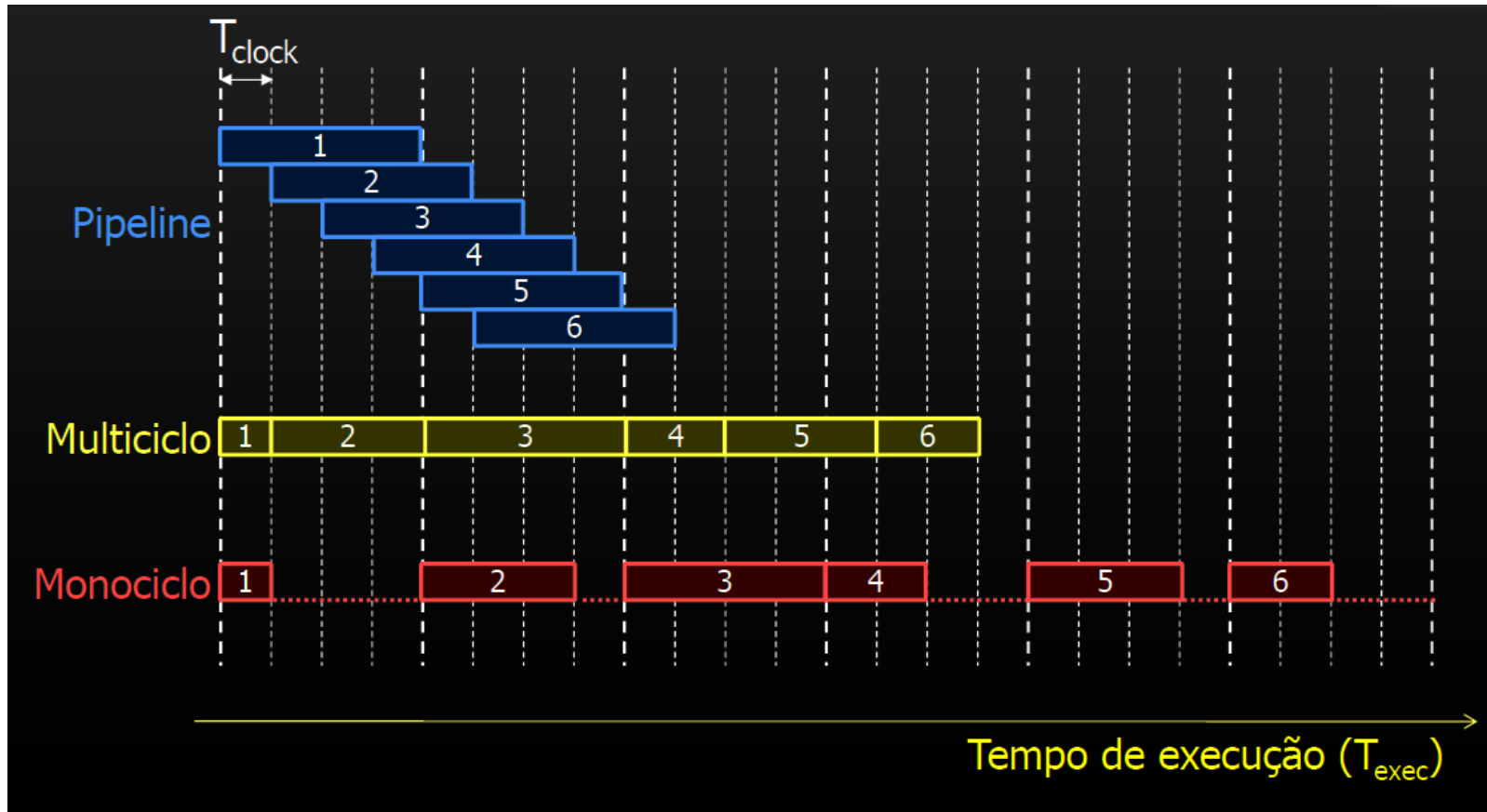
# Comparando Desempenhos

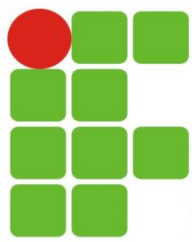
- Ciclo Único: Ciclo de clock tem mesma duração para todas instruções: ciclos de clocks por instrução (CPI) = 1;
- Multi Ciclo: Cada instrução usa apenas o número de ciclos que ela necessita;
- Pipelined: Pipeline explora o paralelismo entre as instruções em um fluxo de instruções sequenciais.





# Conjunto de Instruções Comparando Desempenhos

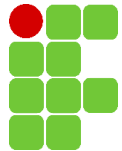


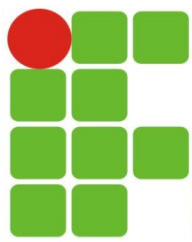


# Conjunto de Instruções

## Divisão de instrução em estágios

- Possíveis estágios:
  - Instruction Fetch (IF) – Carga de instrução e incremento do registrador PC;
  - Instruction Decode (ID) – Decodificação e carga de registrador(es) do banco;
  - Execução (EX) – operação da ALU, cálculo de endereço de memória ou finalização de desvios;
  - MEM – Acesso à memória ou finalização de instrução R;
  - Write Back (WB) – Finalização de leitura de memória.

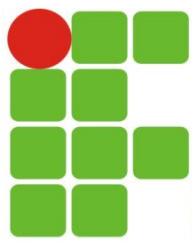




# Conjunto de Instruções

## Divisão de instrução em estágios

- Cada estágio toma um ciclo de clock;
- Nem todas as instruções usam todos os estágios;
- Em MIPS, as instruções tomam entre 3 – 5 ciclos (estágios);
- Cinco estágios de execução de cada instrução:
  - IF - Busca (fetch) da instrução;
  - ID - Decodificação da instrução;
  - EX - Execução ou cálculo do endereço efetivo;
  - MEM - Acesso à memória ou desvio;
  - WB - Atualização dos registradores (write-back).
- Cada fase de uma instrução é processada por um estágio, em um ciclo de clock;
- Uma nova instrução é iniciada a cada ciclo.



# Divisão de instrução em estágios

Instrução	Ciclo de clock								
	1	2	3	4	5	6	7	8	9
i	IF	ID	EX	MEM	WB				
i+1		IF	ID	EX	MEM	WB			
i+2			IF	ID	EX	MEM	WB		
i+3				IF	ID	EX	MEM	WB	
i+4					IF	ID	EX	MEM	WB

