

CURSO DE ENGENHARIA DE SOFTWARE

Disciplina: Unidade Central de Processamento

UNIDADE CENTRAL DE PROCESSAMENTO - CPU

Prof. Alexandre Tannus

- ▶ Detalhar o funcionamento da Unidade Central de Processamento (CPU - *Central Processing Unit*)
- ▶ Entender a operação da Unidade Lógica Aritmética e como ela realiza cálculos básicos (soma e subtração)
- ▶ Compreender a função dos diversos registradores presentes na CPU
- ▶ Explicar as atribuições da unidade de controle
- ▶ Identificar os tipos de barramentos
- ▶ Investigar o conjunto de instruções
- ▶ Comparar os modelos de paralelismo que podem ser utilizados

Introdução

Unidade Lógica Aritmética

Registradores

Unidade de Controle

Barramentos

Instruções

Paralelismo

Exercícios

Introdução

Unidade Lógica Aritmética

Registradores

Unidade de Controle

Barramentos

Instruções

Paralelismo

Exercícios

- ▶ *Central Processing Unit (CPU)*
- ▶ Responsável pelos cálculos e controle da operação do computador



- ▶ Unidade lógica aritmética (ULA)
- ▶ Unidade de controle (UC)
- ▶ Registradores
- ▶ Barramentos



Introdução

Unidade Lógica Aritmética

Registradores

Unidade de Controle

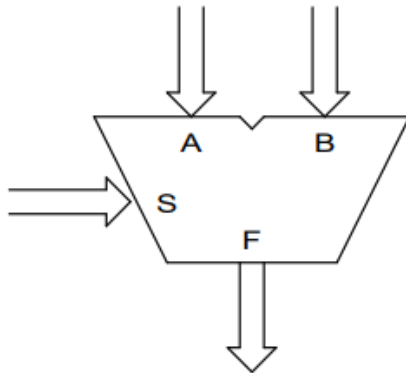
Barramentos

Instruções

Paralelismo

Exercícios

- ▶ Realiza as operações lógicas e aritméticas
 - ▶ NOT, OR, AND
 - ▶ Adição, Subtração
 - ▶ Comparação
 - ▶ Deslocamento



- ▶ Inteiros sinalizados
 - ▶ Sinal-magnitude
 - ▶ Complemento de 2
- ▶ Ponto Flutuante
 - ▶ Padrão ANSI/IEEE

- ▶ Bit mais significativo (MSB - *Most Significant Bit*) indica o sinal
 - ▶ 1: número negativo
 - ▶ 0: número positivo

- ▶ Representação de números negativos mais comum em hardware
- ▶ Complemento de 1 \rightarrow inversão bit a bit do número (complemento)
- ▶ Complemento de 2 \rightarrow adição de 1 ao complemento de 1

Exemplos (representação em 8 bits)

▶ $9 + 4$

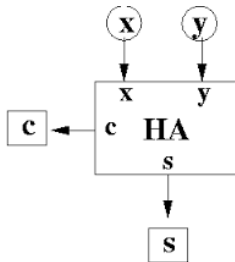
▶ $-9 + 4$

▶ $9 - 4$

▶ $-9 - 4$

Característica	Único/Curto	Duplo/Longo
Largura da palavra	32	64
Bits mantissa	23	52
Intervalo mantissa	$[1, 2 - 2^{-23}]$	$[1, 2 - 2^{-52}]$
Bits de expoente	8	11
Excesso do expoente	127	1023
Mínimo	$2^{-126} \approx 1,2 \times 10^{-38}$	$2^{-1022} \approx 2,2 \times 10^{-308}$
Máximo	$2^{128} \approx 3,4 \times 10^{-38}$	$2^{1024} \approx 1,8 \times 10^{308}$

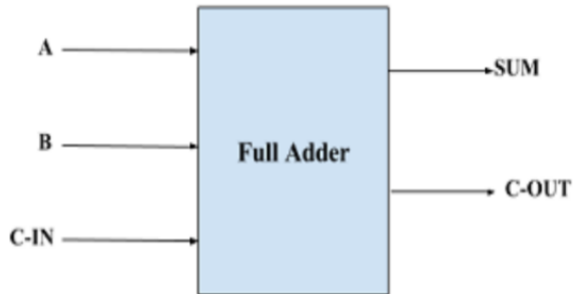
x	y	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Exemplos

- ▶ $(11001)_2 + (1011)_2$
- ▶ $(101101)_2 + (11100111)_2$
- ▶ $(100111)_2 + (1110)_2 + (1011)_2$

A	B	C-IN	C-OUT	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Introdução

Unidade Lógica Aritmética

Registradores

Unidade de Controle

Barramentos

Instruções

Paralelismo

Exercícios

- ▶ Armazenam dados e instruções
- ▶ Baixa capacidade de armazenamento
- ▶ Alta velocidade de acesso
- ▶ Podem ser
 - ▶ Propósito geral - operações lógicas e aritméticas
 - ▶ Especiais - Acumuladores, *Program Counter*, registrador de *flags*, etc.

- ▶ Contador de programa (*Program Counter* - PC)
 - ▶ Armazena o endereço de memória onde será lida a próxima instrução que será executada
 - ▶ Atualizado após a busca da instrução
- ▶ Registrador de Instrução (*Instruction Register* - IR)
 - ▶ Armazena a instrução em execução

- ▶ Registrador de endereços de memória (*Memory Address Registers* - MAR)
- ▶ Registrador *buffer* de memória - (*Memory Buffer Register* - MBR)

Registradores responsáveis pela troca de informações (dados e instruções) entre memória e CPU

- ▶ Palavra de estado do programa (*Program State Word* - PSW)
- ▶ Informações de estado
 - ▶ sinal
 - ▶ zero
 - ▶ *carry*: *carry out* bit de uma operação
 - ▶ *overflow*
 - ▶ *interrupt enable/disable*: habilita ou desabilita interrupções

- ▶ AX - Acumuladores
- ▶ BX - Base
- ▶ CX - Contador
- ▶ DX - Dados
- ▶ BP - Ponteiro de base
- ▶ SI e DI - usado em operações que envolvem *strings*

Introdução

Unidade Lógica Aritmética

Registradores

Unidade de Controle

Barramentos

Instruções

Paralelismo

Exercícios

- ▶ Controla toda a operação do microprocessador
- ▶ Constituída por
 - ▶ Circuito de temporização
 - ▶ Controle e decodificação
 - ▶ Decodificador de instruções

Introdução

Unidade Lógica Aritmética

Registradores

Unidade de Controle

Barramentos

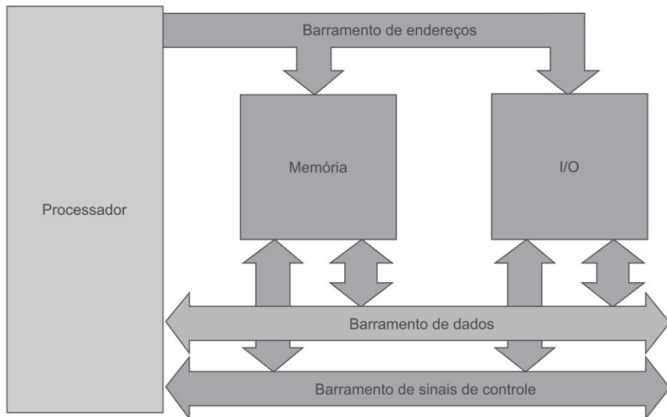
Instruções

Paralelismo

Exercícios

- ▶ Vias que interligam os dispositivos (CPU, memória e periféricos), permitindo a comunicação entre os mesmos.

- ▶ Três tipos
 - ▶ Dados
 - ▶ Endereços
 - ▶ Controle



- ▶ Trafega dados, informações ou instruções
- ▶ Composto por vias
 - ▶ Cada via trafega um bit
 - ▶ Quantidade de vias define largura do barramento
- ▶ **ATENÇÃO:** Largura do barramento de dados pode ser diferente da quantidade de bits que o processador utiliza

Processador	Processamento interno	Barramento externo
i8080 (1974)	8	8
8088 (1979)	16	8
80286 (1982)	8	16
80386DX (1985)	32	32
80486DX (1989)	32	32
Pentium (1993)	32	64
Athlon 64 (2003)	64	128

- ▶ Endereçamento dos periféricos do sistema
 - ▶ Memórias
 - ▶ Controlador de vídeo
 - ▶ Disco
 - ▶ Rede
- ▶ Quantidade de vias define quantidade máxima de endereços possíveis

Processador	Largura do barramento de endereços	Quantidade máxima de endereços
i8088	20 bits	1 Mb
i80286	24 bits	16 Mb
i386	32 bits	4 Gb
Pentium	32 bits	4 Gb
Core i5	35 bits	32 Gb

- ▶ Recebimento/envio de sinais de controle para os dispositivos do sistema
 - ▶ RESET
 - ▶ Interrupções
 - ▶ HALT
 - ▶ HOLD
 - ▶ Seleção
- ▶ Sinais de controle são específicos de cada arquitetura

Introdução

Unidade Lógica Aritmética

Registradores

Unidade de Controle

Barramentos

Instruções

Paralelismo

Exercícios

- ▶ Interface entre o programador e a máquina
- ▶ Cada instrução realiza uma tarefa simples
- ▶ Operações complexas podem ser construídas a partir de operações simples

- ▶ Execução sequencial
- ▶ Exceções
 - ▶ Instruções de salto
 - ▶ Instruções de desvio

Tipo	Formato (bits)					
R	opcode (6)	rs(5)	rt(5)	rd(5)	shamt(5)	function(6)
I	opcode (6)	rs(5)	rt(5)	imediato(16)		
J	opcode(6)	endereço(26)				

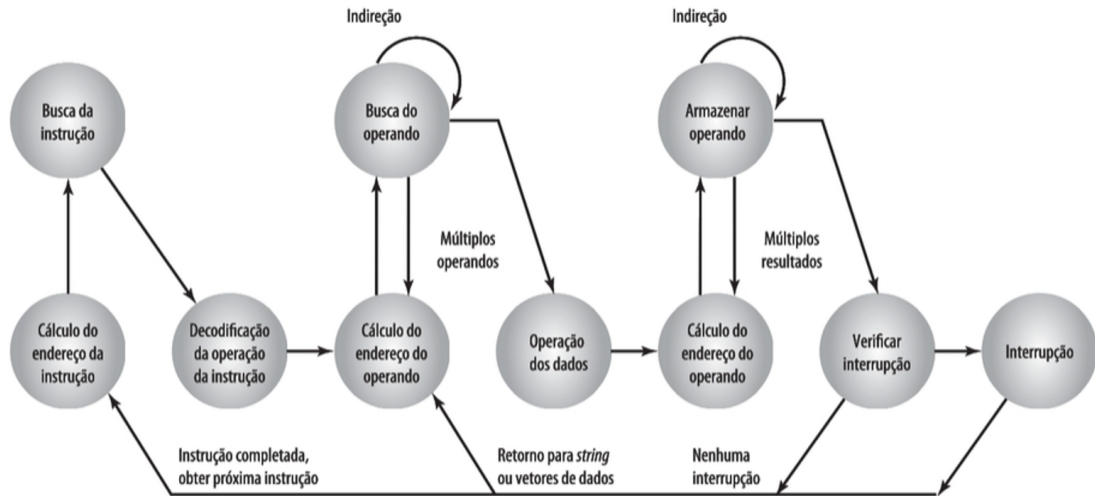
Operação	Comando	Sintaxe	Função
Adição	<i>add</i>	<i>add</i> \$t0, \$s0, \$s1	32
Subtração	<i>sub</i>	<i>sub</i> \$t0, \$s0, \$s1	34
Lógica AND	<i>and</i>	<i>and</i> \$t0, \$s0, \$s1	36
Lógica OR	<i>or</i>	<i>or</i> \$t0, \$s0, \$s1	37
Lógica XOR	<i>xor</i>	<i>xor</i> \$t0, \$s0, \$s1	38
Lógica NOR	<i>nor</i>	<i>nor</i> \$t0, \$s0, \$s1	39
Adição imediata	<i>addi</i>	<i>addi</i> \$t0, \$s0, <i>constante</i>	8
Lógica AND imediata	<i>andi</i>	<i>andi</i> \$t0, \$s0, <i>constante</i>	12
Lógica OR imediata	<i>ori</i>	<i>ori</i> \$t0, \$s0, <i>constante</i>	13
Lógica XOR imediata	<i>xori</i>	<i>xori</i> \$t0, \$s0, <i>constante</i>	14

- ▶ Transferência de 32 bits entre memória e registradores
- ▶ Instrução tipo I
- ▶ Registrador *rt* - destino (*load*) ou origem (*store*)
- ▶ Endereço de memória - Constante de deslocamento + valor do registrador *rs*

- ▶ Alteram o fluxo de execução sequencial
- ▶ Salto
 - ▶ Instrução *j* ou *jr*
- ▶ Desvio
 - ▶ *bne* - diferente de
 - ▶ *bltz* - menor que
 - ▶ *beq* - igual a

- ▶ Implícito
- ▶ Imediato
- ▶ Por registrador
- ▶ Base
- ▶ Relativo ao PC
- ▶ Pseudodireto

- ▶ Busca
 - ▶ Leitura da próxima instrução da memória
- ▶ Execução
 - ▶ Interpretação e efetuação da operação indicada
- ▶ Interrupção
 - ▶ Verificação de ocorrência de interrupção



- ▶ Anos 80 – computadores mais complexos
- ▶ CISC – *Complex Instruction Set Computer*
- ▶ Conjuntos de instruções cada vez mais complexas e maiores
 - ▶ Pode afetar o desempenho
 - ▶ Maior dificuldade de implementação de outras funções

- ▶ RISC – *Reduced Instruction Set Computer*
- ▶ Diminuição do número de instruções disponíveis
- ▶ Padronização do tamanho das instruções
- ▶ Introdução do pipeline

- ▶ Controle via *hardware*
- ▶ Maior número de registradores
- ▶ Modos de endereçamento limitados

Introdução

Unidade Lógica Aritmética

Registradores

Unidade de Controle

Barramentos

Instruções

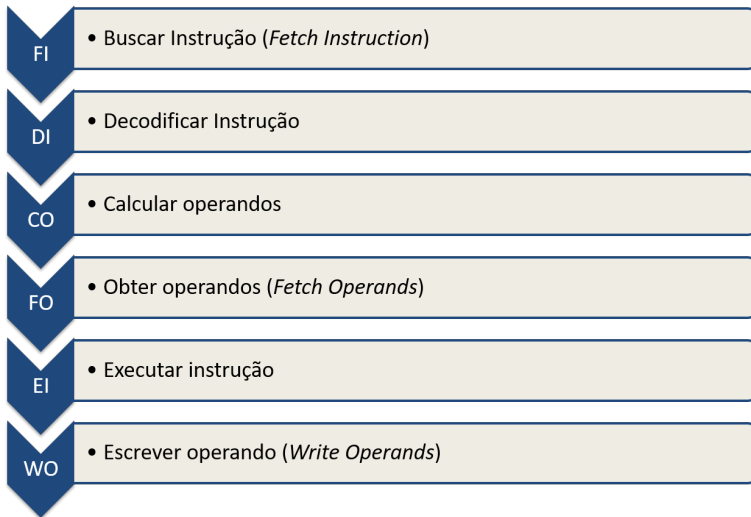
Paralelismo

Exercícios

- ▶ Realizar várias coisas ao mesmo tempo
- ▶ Dois níveis
 - ▶ Instrução
 - ▶ Processador

- ▶ Executar mais instruções em um determinado tempo
- ▶ Dois modelos
 - ▶ *Pipelines*
 - ▶ Superescalares

- ▶ Dividir uma instrução em vários estágios
- ▶ Dedicar uma parte do *hardware* para cada estágio
- ▶ Executar paralelamente vários estágios



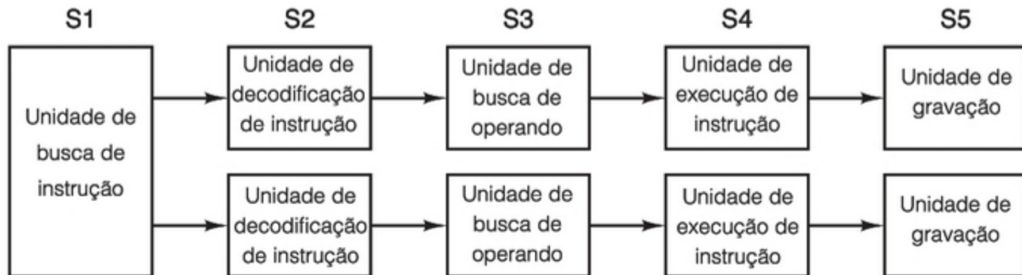
- ▶ Buscar Instrução (Fetch Instruction)
 - ▶ Leitura da próxima instrução
- ▶ Decodificar Instrução
- ▶ Calcular operandos
 - ▶ Cálculo do endereço efetivo de cada operando

- ▶ Obter operandos (*Fetch Operands*)
 - ▶ Obtenção dos operandos da memória
- ▶ Executar Instrução
 - ▶ Realização da operação e armazenamento do resultado no registrador
- ▶ Escrever operandos (*Write Operands*)
 - ▶ Cálculo do endereço efetivo de cada operando

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Instrução 1	FI	BI	CO	FO	EI	W0												
Instrução 2							FI	BI	CO	FO	EI	WO						
Instrução 3													FI	BI	CO	FO	EI	WO

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Instrução 1	FI	BI	CO	FO	EI	WO												
Instrução 2		FI	BI	CO	FO	EI	WO											
Instrução 3			FI	BI	CO	FO	EI	WO										

- Execução de vários *pipelines* simultaneamente



- ▶ Projeto de computadores com várias CPUs
- ▶ Modelos
 - ▶ Computadores paralelos
 - ▶ Multiprocessadores
 - ▶ Multicomputadores

Introdução

Unidade Lógica Aritmética

Registradores

Unidade de Controle

Barramentos

Instruções

Paralelismo

Exercícios

Sobre Processadores, analise as assertivas e assinale a alternativa que aponta a(s) correta(s).

I. A CPU é o 'cérebro' do computador, sua função é executar programas armazenados na memória principal, buscando suas instruções, examinando-as e então executando-as uma após a outra.

II. Barramentos podem ser externos à CPU, conectando-a à memória e aos dispositivos E/S, mas também podem ser internos à CPU.

III. A CPU é composta por várias partes distintas. A unidade de controle é responsável por buscar instruções na memória principal e determinar seu tipo.

IV. A unidade de aritmética e lógica efetua operações como adição AND (E) booleano para executar as instruções.

Em relação à arquitetura, a CPU é representada pelo microprocessador, sendo responsável pela principal função dos microcomputadores, que é o processamento dos dados. Conceitualmente, a CPU é constituída de

- a Registradores / Memória Cache / Coprocessador Aritmético e Lógico.
- b Registradores / Unidade de Controle / Unidade Lógica e Aritmética.
- c Buffers / Memória Cache / Coprocessador Aritmético e Lógico.
- d Buffers / Unidade de Controle / Unidade Lógica e Aritmética.

A CPU gera endereços que são colocados no barramento ...I.....e a memória os recebe através deste barramento. O caminho inverso desta operação não é possível (isso pode ser observado na figura). Durante a execução de um programa, cada instrução é levada até a ALU a partir da memória, uma instrução de cada vez, junto com qualquer dado que seja necessário para executá-la, cujo valor é transmitido através do barramento...II.... . A saída do programa é colocada em um dispositivo como um monitor de vídeo ou disco. A comunicação entre os componentes do sistema é sincronizada pelo barramento...III.. .

As lacunas I, II e III são correta e, respectivamente, preenchidas por:

- a De controle, de endereços, de dados
- b De endereços, de dados, de sincronização
- c De dados, de endereços, de controle
- d De endereços, de dados, de controle

A função do registro de instrução é armazenar o identificador da próxima instrução a ser executada pelo processador.

Se, para reduzir custos de fabricação, for criado um computador em que o tamanho do registrador PC (program counter) seja a metade do REM, então, embora ocorra a redução do custo, essa máquina não irá funcionar, pois o PC deve ser projetado, no mínimo, com o mesmo tamanho do REM.



William Stallings.

Arquitetura e Organização de Computadores.

Pearson, São Paulo, 8 edition, 2010.



Andrew S. Tanenbaum.

Organização estruturada de computadores.

2007.

