

Aula 2

• Conceitos fundamentais em Arquitectura de Computadores

• O Computador como sistema digital

• Os elementos básicos de um computador

• O ciclo básico de execução de uma instrução

• Arquitectura de Computadores

• *Instruction Set Architecture*

• Organização

• Níveis de Representação

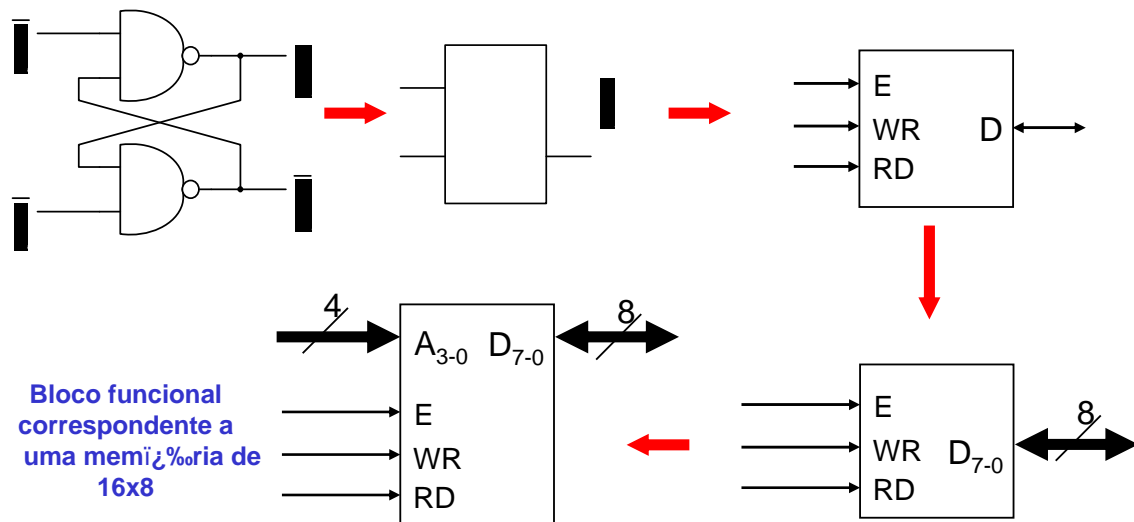
Bernardo Cunha, José Luís Azevedo, Arnaldo Oliveira e Silva

Há alguma relação entre Sistemas Digitais e Arquitectura de Computadores?

• Arquitectura de Computadores é uma das áreas de aplicação directa dos conceitos, técnicas e conhecimentos aprendidos em Sistemas Digitais.

• Em Arquitectura de Computadores, contudo, trabalha-se num nível de abstracção diferente, recorrendo na maior parte das vezes a blocos funcionais complexos com cuja síntese, normalmente, não temos que nos preocupar.

Por exemplo, uma "Memória" (um dispositivo capaz para armazenar informação digital binária) pode ser feita de blocos básicos bem conhecidos dos sistemas digitais: **flip-flops**



Computador: sistema digital complexo

As unidades fundamentais que constituem um computador

são:

CPU é o processamento da informação através da execução de uma sequência de instruções (programa) armazenadas na memória

Memória é o armazenamento de:

- Programas
- Dados para processamento
- Resultados

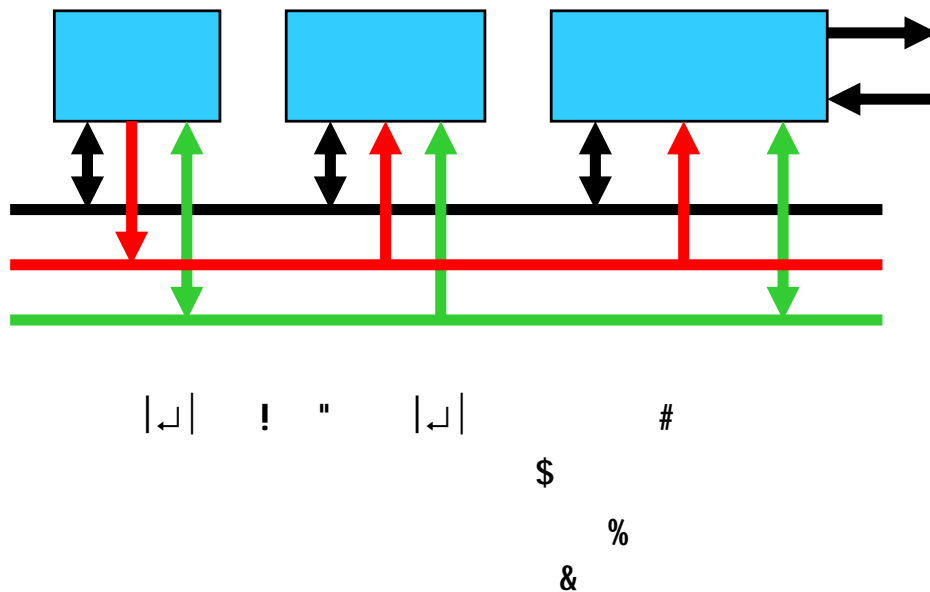
não esquecer que cada um destes blocos é um sistema digital !!!.

Unidades de I/O é a comunicação com o exterior

Unidades de entrada permitem a recepção de informação do exterior (dados, programas) e que é armazenada na memória

Unidades de saída permitem o envio de resultados para o exterior

Arquitectura básica de um sistema computacional (modelo de von Neumann)



Endereço (address) é um número (único) que identifica cada registo de memória. Os endereços são contados sequencialmente, começando em 0

Exemplo: o conteúdo da posição de memória 0x2000 é 0x32 (0x2000 é o endereço, 0x32 o valor armazenado)

Espaço de endereço (address space) é a gama total de endereços que o CPU consegue referenciar (depende da dimensão do barramento de endereços).

Exemplo: um CPU com um barramento de endereços de 16 bits pode gerar endereços na gama: 0x0000 a 0xFFFF (i.e., 0 a $2^{16}-1$)

Qual o espaço de endereçamento de um processador com um barramento de endereços de 32 bits?

O CPU consiste, fundamentalmente, em duas secções:

Secção de dados (data path) registos, elementos operativos:

Registos internos

Unidade Aritmética e Lógica (ALU) Add, Sub, And,

Unidade de controlo responsável pela coordenação dos elementos do *data path*, durante a execução de um programa

Máquina de estados síncrona (estado seguinte é função do estado actual e das entradas)

As entradas correspondem a informação recebida de uma das instruções lidas da memória

Ciclo-base de execução de uma instrução (execute cycle)

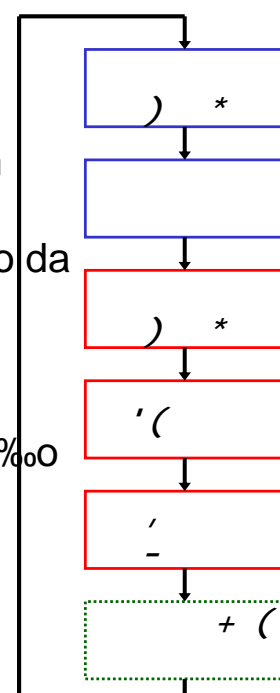
Instruction fetch: leitura do código máquina da instrução (reside em memória)

Instruction decode: descodificação da instrução pela unidade de controlo

Operand fetch: leitura do(s) operando(s)

Execute: execução da operação especificada pela instrução

Store result: armazenamento do resultado da operação no destino especificado na instrução



Processos de Representação

Programa em
Linguagem de Alto
Nível (e.g. C)

$a = b + c + d;$

Compilador

Programa em
Assembly

add \$10, \$11, \$12
add \$10, \$10, \$13

Assembler

Programa em
Linguagem Máquina

00000001011011000101000000100000 (32 bits)
00000001010011010101000000100000

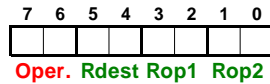
Descodificação e execução pela
máquina

Especificação dos
sinais de controlo

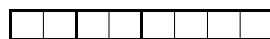
ALUOp ← "10"
ALUIn1 ← Reg[Inst(25:21)]
ALUIn2 ← Reg[Inst(20:16)]

Exemplo - CPU hipotético

Formato de codificação das instruções (8 bits)



Formato 1



Formato 2

Oper. Reg. End. Memória

Registos Internos do CPU:

00 Reg. 0
01 Reg. 1
10 Reg. 2
11 Reg. 3

Operações possíveis:

- 00 Somar o conteúdo de dois registos
- 01 Ler da memória para um registo interno do CPU (LD)
- 10 Escrever o conteúdo de um reg. interno na memória (STORE)
- 11 Não definida (N.D.)

Exemplo de programa em código máquina para este processador hipotético

Qual é a expressão aritmética implementada neste programa?

Hex	Binário	
0x58	01011000	Ler o conteúdo da posição de memória 8 para o registo interno 0
0x79	01111001	Ler o conteúdo da posição de memória 9 para o registo interno 1
0x15	00010101	Somar o conteúdo do reg. 1 c/ o reg. 1 e depositar o result. no reg. 1
0x07	00000111	Somar o conteúdo do reg. 1 c/ o reg. 3 e depositar o result. no reg. 0
0x8A	10001010	Escrever o conteúdo do reg. 0 na posição de memória 10

Arquitectura de Computadores =
**Arquitectura do Conjunto de Instruções (ISA) +
 Organização da Máquina**

Conjunto (Set) de Instruções: a coleção de todas as operações que o processador pode executar

ISA Instruction Set Architecture

• **Arquitectura do Conjunto de Instruções** (também designada por "modelo de programação"):

- Uma importante abstracção que representa **interface entre o h/w e o nível mais básico de s/w**
- Descreve tudo o que o programador necessita de saber para programar correctamente, em linguagem máquina, um determinado processador
- Descreve a funcionalidade, independentemente do h/w que a implementa. Pode assim falar-se de "**arquitectura**" e "**implementação de uma arquitectura**"
- Exemplo em que a mesma arquitectura do conjunto de instruções tem 2 implementações distintas:
 - Processadores AMD compatíveis com Intel x86

• Alguns exemplos de ISAs:

- MIPS
- ARM (Nintendo DS, iPod, Canon PowerShot ...)
- Intel x86 (PCs, MACs)
- PowerPC
- ...

Arquitectura do Conjunto de Instruções (ISA)

Os atributos de um sistema computacional tal como são vistos pelo programador, i.e. a estrutura conceptual e o comportamento funcional, de forma distinta e independente da organização do fluxo de informação e dos respectivos elementos de controlo, do desenho lógico e da implementação física.

Amdahl, Blaaw, and Brooks, 1964

Conjunto de Instruções

Organização da memória

Número e tipos de Registos

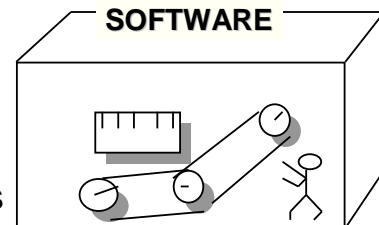
Tipos de dados e estruturas de dados

Codificação e representação

Modos de endereçamento (acesso a dados e instruções)

Formato das Instruções

Condições que podem desencadear "exceções"



Arquitectura do Conjunto de Instruções:

Formato e codificação das instruções
como são decodificadas?

Operandos das instruções e resultados

onde podem residir?

quantos operandos explícitos?

como referenciar?

quais podem residir na memória externa?

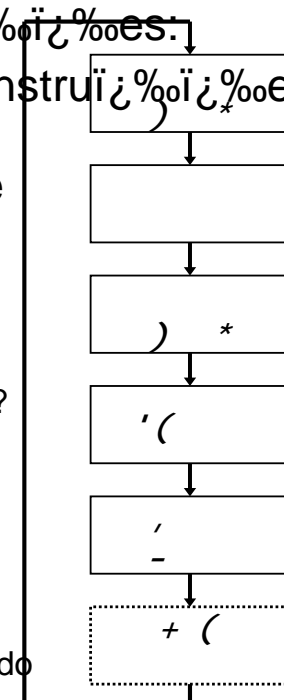
Tipo e dimensão dos dados

Operações

quais são suportadas?

Instruções auxiliares

jumps, conditions, branches - para controlo do fluxo de execução



Requisitos básicos da Arquitectura do Conjunto de Instruções:

- Implementação simples e eficiente
- Fácil de entender e programar
- Desenvolvimento de compiladores eficientes

Conjunto de Instruções: um Interface Crítico



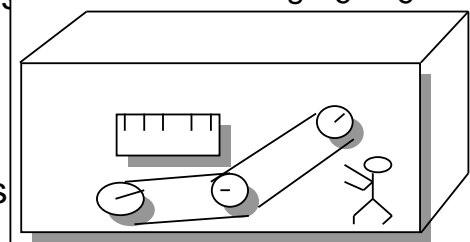
Organização da máquina

- Características operativas e de desempenho das principais unidades funcionais
 - (ALU, Registradores, Shifters, Unidades Lógicas, ...)
- De que modo esses componentes são interligados
- Fluxo de informação entre componentes
- Lógica e meios através dos quais esse fluxo é controlado
- Coreografia das Unidades Funcionais para implementar a ISA

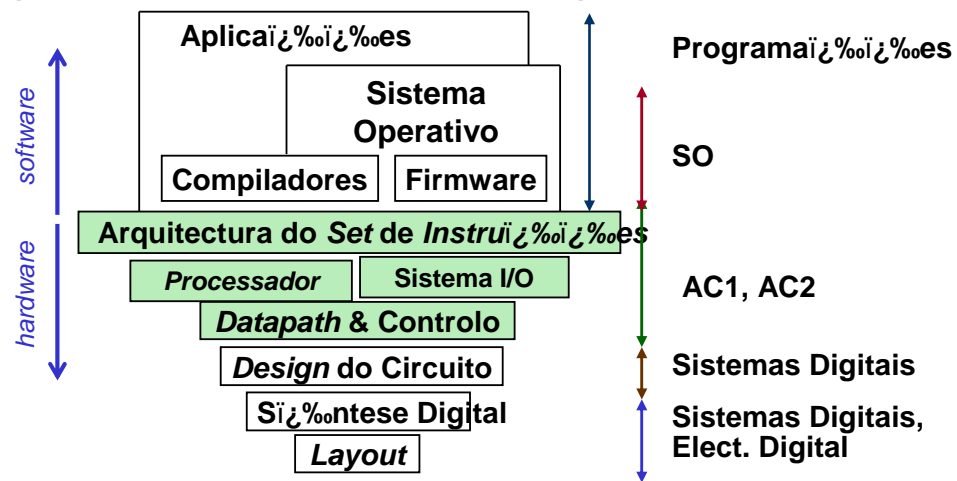
Perspectiva do Designer Lógico

Nível ISA

UFs & Interligações



Arquitectura dos Sistemas Computacionais:



- Coordenação de muitos níveis de abstracção...
- Sob o efeito de um conjunto de forças em permanente mudança
- Design, Medida e Avaliação

A evolução da arquitectura de computadores depende de múltiplos factores

