



UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

## Microprocessadores

Regente: Eng.º Albino B Cuinhane  
Assistente: Eng. Edson Camilo Fortes  
Monitor: Erick Mahanjane

(P/C) A.B. Cuinhane UEM - Microprocessadores

## AULA 2 SUMÁRIO

- ▶ 2.1. Origem e Formato da Instrução
- ▶ 2.2. Quantidade de Operandos e OpCodes

ABC UEM - Digital

## Capítulo 2

## Codificação e Manipulação de Instruções

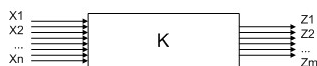
ABC UEM - Digital

## 2.1. Origem e Formato da Instrução

ABC UEM - Digital

### 2.1. Origem e Formato da Instrução

A. O PROBLEMA: Dado um conjunto de entradas pretendemos que combinadas produzam um conjunto de saídas com determinado comportamento



As saídas Z são funções de X e do tempo.

Por vezes há realimentação

B. SOLUÇÃO: A solução passa por encontrar K.

Vários métodos para resolver:

B1- equações matemáticas que descrevem o comportamento do sistema

B2. Algoritmos (conjunto finito de passos): FSM ou HDL

(P/C) A.B. Cuinhane UEM - Digital II

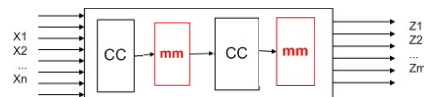
### 2.1. Origem e Formato da Instrução

C. TIPO DE SOLUÇÃO:

C1. teoria de sinais analógicos

C2. teoria de sinais discretos (nosso caso)

Entrando no sistema vemos que começamos por instalar Circuitos Combinatórios (CC) para resolução de tarefas específicas



Para que o sistema possa reter informação inserimos a memória

Desta forma fizemos o CC um pouco mais flexível e formamos a ARQUITECTURA

(P/C) A.B. Cuinhane UEM - Digital II

### 2.1. Origem e Formato da Instrução

Do ponto de vista do projecto de microprocessadores, a arquitectura representa um conjunto de circuitos combinatórios associados a elementos de memória com o objectivo de resolver um determinado problema.

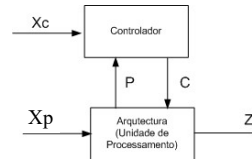
#### D. CIRCUITO FINAL – FINITE STATE MACHINE

A arquitectura tem em falta uma capacidade importante: CONTROLO

- Para comandar esta arquitectura adiciona-se um controlador. Assim, a arquitectura informa em que estado está para que o controlador responda com sinais de controlo.
- Com esta estrutura pode-se resolver problemas complexos através da quebra dos mesmos em problemas mais simples que serão resolvidos pelos circuitos combinatórios.

(P/C) A.B. Coimbra UEM - Digital II

### 2.1. Origem e Formato da Instrução



Na verdade o CONTROLADOR é também uma arquitectura, porém se destaca pelo facto dele gerar sinais de controlo no momento exacto para gerir a arquitectura que processa as equações do problema

O sistema construído, FSM, é capaz de resolver problemas complexos através de subdivisão em etapas simples que são feitas no CC (operações aritméticas, lógicas e manipulação de números)

(P/C) A.B. Coimbra UEM - Digital II

### 2.1. Origem e Formato da Instrução

•Mas, sendo um circuito físico, as operações são feitas em sequência fixa de acordo com a disposição dos componentes. É preciso encontrar uma forma de alterar a sequência de utilização de recurso conforme o problema actual.

•Então acrescentamos um outro nível de controlo que age sobre o controlador que, por sua vez, age sobre a arquitectura e decide a sequência.

•A partir daqui nomeamos as sequências predefinidas de utilização de recursos – INSTRUÇÃO. Por exemplo ADD usará certos recursos e SUB outros.

•Finalmente, a solução dum problema é expresso por símbolos combinados de acordo com a sequência desejada de sequências predefinidas - PROGRAMA

(P/C) A.B. Coimbra UEM - Digital II

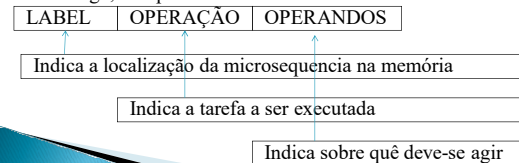
### 2.1. Origem e Formato da Instrução

•Um programa começa com a sequência predefinida GO, termina com a END. Pelo caminho usa outras como ADD, SUB, LOAD, REPEAT, etc.

•Desta maneira foi criada a chamada ISP (Instruction Set Processor), ou seja um Processador de Conjunto de Instruções.

•De forma geral, segundo Deepti Mehrotra (2013) “uma instrução é um comando dado a um computador para desempenhar uma operação específica em torno de um conjunto de dados”.

•Ou seja a INSTRUÇÃO é uma ordem para se fazer alguma acção sobre algo, e as partes básicas dela são:



(P/C) A.B. Coimbra UEM - Digital II

### 2.2. # de Operandos e OpCodes

Tipo de operações	Nome das operações
Operações de transferência de dados	Move, store, load, exchange, clear, set, push e pop.
Operações aritméticas e lógicas	ADD, SUB, AND, OR, NOT, Exclusive OR, Test, Compare, Set, Shift, Rotate e jump.
Operações de transferência de controlo	Call subroutine, return, Execute, Skip, Halt, Wait e no operation.
Operações de Entrada e Saída	Input, Output, Start I/O e Test I/O
Operações de Conversão	Translate e Convert.

(P/C) A.B. Coimbra UEM - Digital II

### 2.2. # de Operandos e OpCodes

ABC UEM - Digital

## 2.2. # de Operandos e OpCodes

### •# DE OPERANDOS

- A escolha da quantidade de operandos é difícil e controversa.
- Fazer instruções com poucos operandos poupa memória por instrução mas limita a quantidade de tarefas por instrução e alonga os programas são por vezes designados por máquinas de m-operandos
- Muitos operandos por instrução implica circuitos decodificadores mais complexos e maior consumo de memória/instrução
- A instrução deve trazer normalmente a origem e destino dos dados. Há casos de várias origens e 1 destino como  $Z=A+B$ . Assim os processadores são por vezes designados por máquinas de m-operandos

(P/C) A.B. Coimbra UEM - Digital II

## 2.2. # de Operandos e OpCodes

### •OPCODE

- Em muitos processadores o OpCode é um conjunto de 8 a 16 bits destinados a codificar as operações a serem realizadas. Para o caso de 16 bits uma parte do OpCode pode ser usado para identificar registos internos
- Esta parte dos OpCodes é muito diversificada, dependendo do fabricante do processador
- Em alguns computadores, as instruções podem ser todas elas do mesmo tamanho, noutros estas podem apresentar tamanhos diferentes.
- Apresentar todas as instruções do mesmo tamanho torna a decodificação mais fácil, mas em certos casos o espaço não é optimizado pois todas as instruções devem apresentar o mesmo tamanho que a instrução mais longa.

(P/C) A.B. Coimbra UEM - Digital II

## 2.2. # de Operandos e OpCodes

- ▶ Uma instrução é composta por um opcode e operandos. Pode-se ter instruções com zero, um ou mais operandos.

OPCODE

OPCODE OPERANDO

OPCODE OPERANDO 1 OPERANDO 2

OPCODE OPERANDO 1 OPERANDO 2 OPERANDO 3

(P/C) A.B. Coimbra UEM - Digital II

## 2.2. # de Operandos e OpCodes

### ▶ Execução de instruções:

- A sequência de operações realizadas pela CPU para executar uma instrução é conhecida como ciclo de instrução.
- Para executar uma instrução são necessários os seguintes passos:
  - Busca: a instrução é buscada na memória;
  - Decodificação: a instrução é decodificada; e
  - Execução: as operações especificadas pela instrução são executadas.

### ▶ Tipos de Operandos:

- Endereços de memória;
- Números;
- Caracteres;
- Dados lógicos.

### ▶ Fonte de operandos:

- Memória principal;
- Registos internos;
- Dispositivos de entrada/saída.



(P/C) A.B. Coimbra UEM - Digital II

## Referências

- ▶ [1] - AKHTAR, Shamim, KESARI, Vivek, et al, COMPUTER ORGANIZATION AND ARCHITECTURE, 2013, VIKAS publishing house pvt LTD, New Delhi;
- ▶ [2] - STALLINGS, William, COMPUTER ORGANIZATION AND ARCHITECTURE, Eighth Edition, 2006, Pearson Education LTD;
- ▶ [3] - PATTERSON, David, HENNESSY, John, COMPUTER ORGANIZATION AND DESIGN, Fifth Edition, 2014, USA.
- ▶ [4] - Apontamentos do regente.

17