



---

**DEPEL - Departamento  
de Engenharia Elétrica**

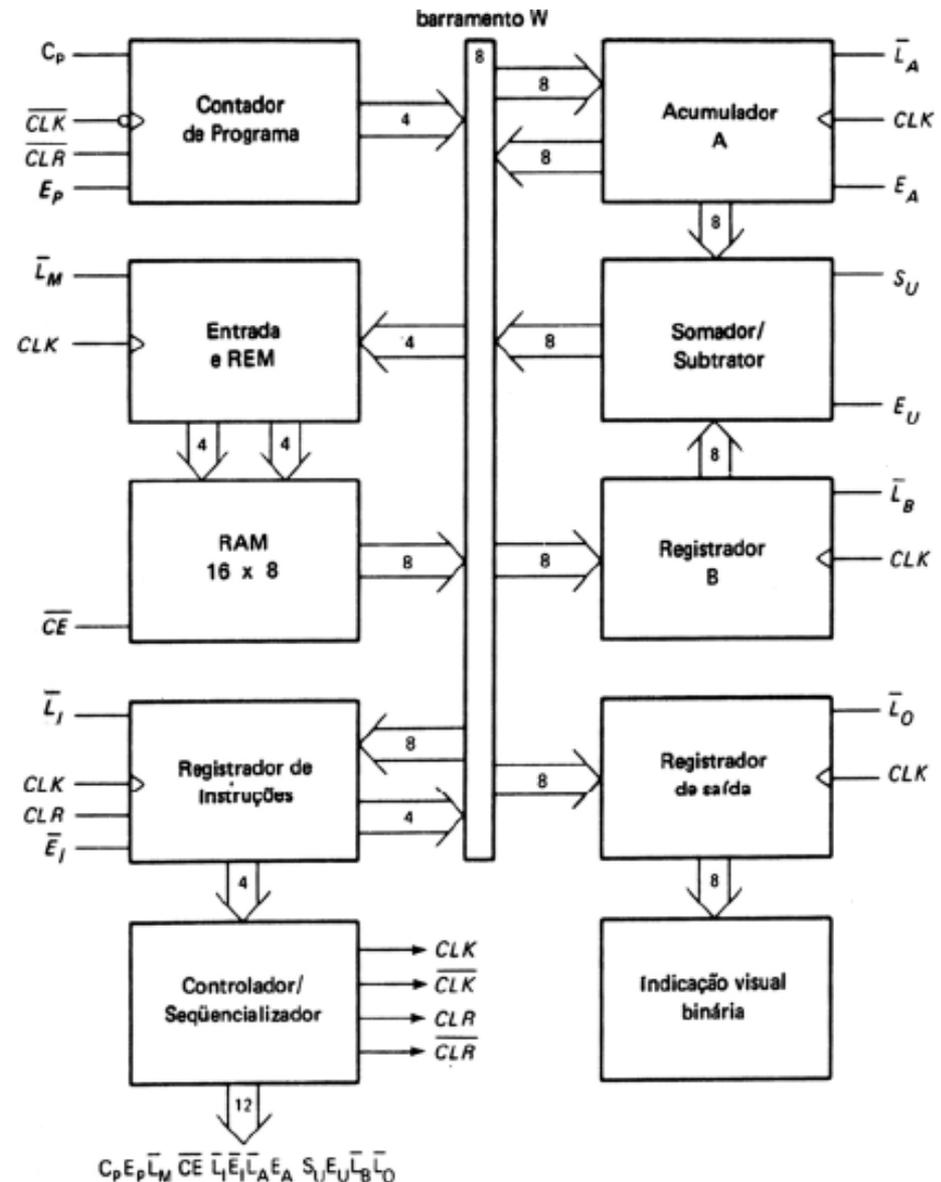
**- Arquitetura de sistemas digitais-**

**Cap 3 – Arquitetura e projeto de  
microprocessadores**

Prof. Alan Petrônio - [www.ufsj.edu.br/alan](http://www.ufsj.edu.br/alan)

# Projeto de um microprocessador

- Nome : uP1
- Arquitetura
- Documentação:
  - [www.ufsj.edu.br/alan](http://www.ufsj.edu.br/alan)



- Unidades funcionais:
  - Contador de programa (ponteiro; PC)
    - Início lógico: ... 000h
  - Entrada e REM
    - Inserir dados e endereço (4bits endereço; 8bit dados a RAM)
    - REM = registrador de endereço de memória (recebe PC)
  - RAM
    - Programação por chaves (dip switch)
    - 4 bits de endereço (1byte de palavra)
  - Registrador de instruções
    - Coloca instrução no barramento W
  - Controlador-sequencializador
    - Obs.: registradores reagem à borda de subida!

$$\text{CON} = C_P E_P \bar{L}_M \bar{C}E \quad \bar{L}_I \bar{E}_I \bar{L}_A E_A \quad S_U E_U \bar{L}_B \bar{L}_O$$

- Acumulador
- Somador e subtrator
  - Complemento de 2
- Registrador B
- Registrador de saída
- Indicador visual binário
  
- Possui todos elementos:
  - UNIDADE DE CONTROLE
  - ULA
  - Memória
  - E/S

- Conjunto de instruções
  - Mnemônicos e Assembly

Mnemônicos	Operação
LDA	Carregue os dados da RAM no acumulador.
ADD	Some os dados da RAM com o acumulador.
SUB	Subtraia os dados da RAM do acumulador.
OUT	Carregue os dados do acumulador no registrador de saída.
HLT	Pare o processamento.

Mnemônico	Código op
LDA	0000
ADD	0001
SUB	0010
OUT	1110
HLT	1111

- Exemplo:

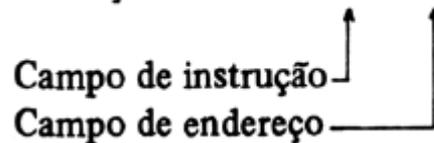
Endereço	Mnemônicos
0H	LDA 9H
1H	ADD AH
2H	ADD BH
3H	SUB CH
4H	OUT
5H	HLT
6H	FFH
7H	FFH
8H	FFH
9H	01H
AH	02H
BH	03H
CH	04H
DH	FFH
EH	FFH
FH	FFH

- Exemplo 1:
  - Nibble 1: instrução
  - Nibble 2: endereço

Endereço	Instrução
0H	LDA FH
1H	ADD EH
2H	HLT

LDA FH = 0000 1111  
 ADD EH = 0001 1110  
 HLT = 1111 XXXX

Instrução = XXXX XXXX

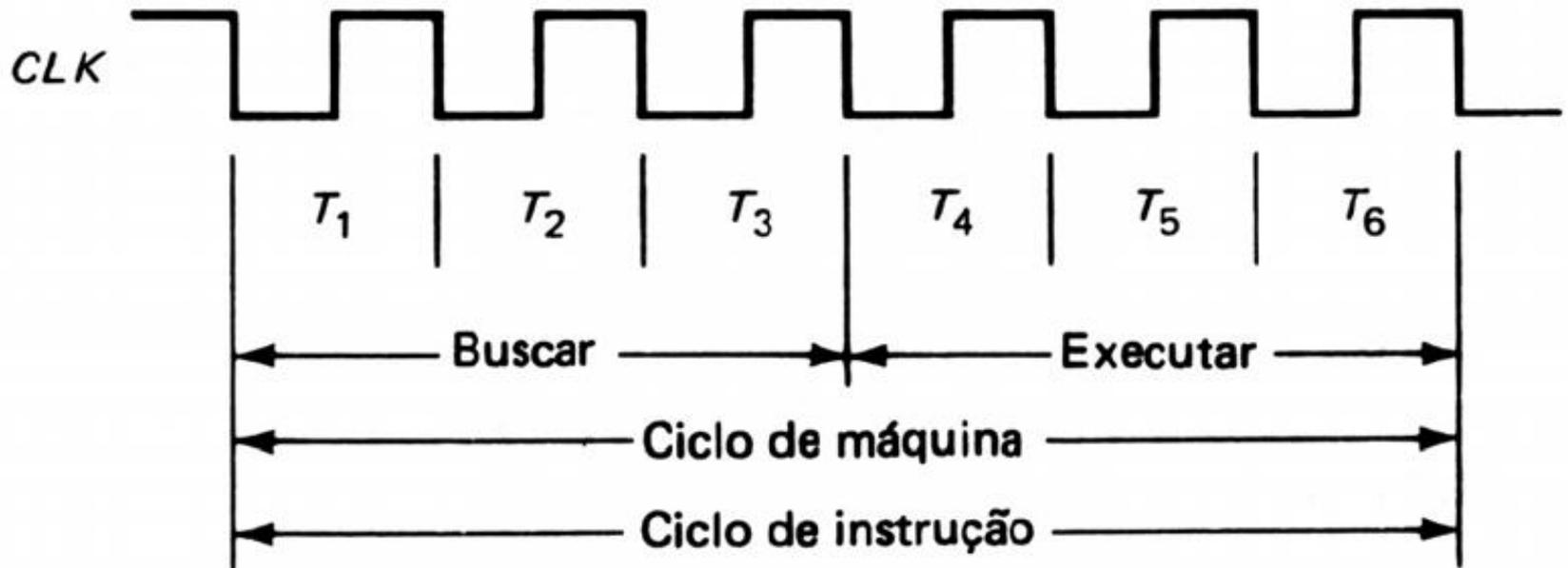


- Exemplo 2: fazer um programa SAP1 para resolver a expressão  $16+20-32$

- Solução em diferentes níveis:

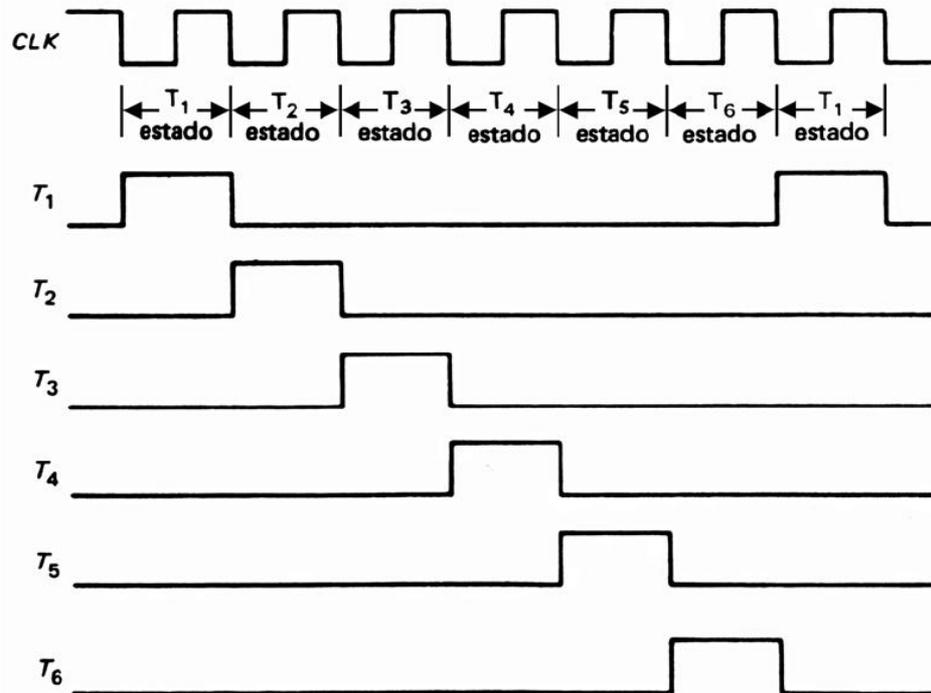
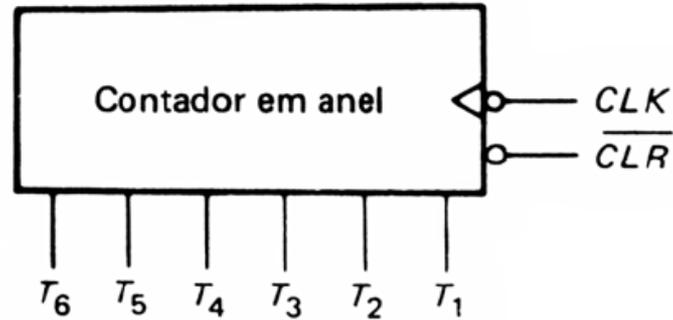
Em linguagem assembly (mneumônicos) temos:		Em linguagem de máquina (binário) temos:		Em linguagem de máquina (hexadecimal) temos:	
Endereço	Conteúdo	Endereço	Conteúdo	Endereço	Conteúdo
0H	LDA 9H	0000	0000 1001	0H	09H
1H	ADD AH	0001	0001 1010	1H	1AH
2H	ADD BH	0010	0001 1011	2H	1BH
3H	SUB CH	0011	0010 1100	3H	2CH
4H	OUT	0100	1110 XXXX	4H	EXH
5H	HLT	0101	1111 XXXX	5H	FXH
6H	XX	0110	XXXX XXXX	6H	XXH
7H	XX	0111	XXXX XXXX	7H	XXH
8H	XX	1000	XXXX XXXX	8H	XXH
9H	10H	1001	0001 0000	9H	10H
AH	14H	1010	0001 0100	AH	14H
BH	18H	1011	0001 1000	BH	18H
CH	20H	1100	0010 0000	CH	20H

- Ciclo de máquina ou instrução
  - Clock = 1KHz
  - Unidade de controle
    - palavras de controle : buscam e executam cada instrução
  - “ESTADOS DE TEMPORIZAÇÃO” (T)



# • Ciclo de busca (fetch)

– Contador em anel



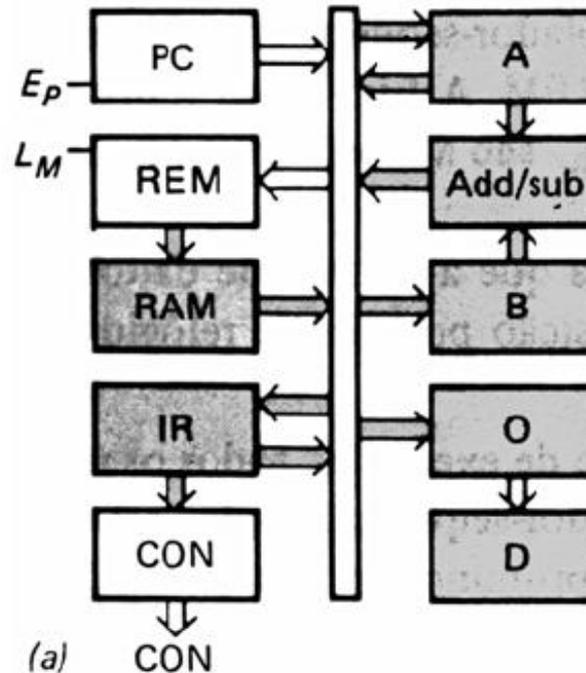
- $T_1$  = Estado de endereço

- Endereço do PC é transferido para registrador REM

- Palavra de controle:  $C_P E_P \bar{L}_M \bar{C}E \quad \bar{L}_1 \bar{E}_1 \bar{L}_A E_A \quad S_U E_U \bar{L}_B \bar{L}_O$   
 $0 \ 1 \ 0 \ 1 \quad 1 \ 1 \ 1 \ 0 \quad 0 \ 0 \ 1 \ 1$

- Transição + do clock ocorre à meia distância de T

- carrega o REM com conteúdo do PC

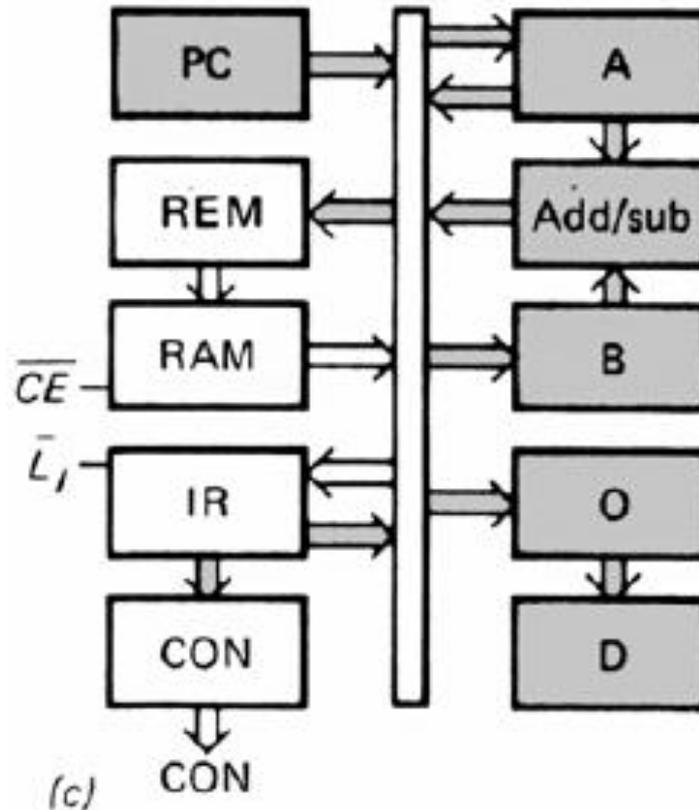




- $T_3$  = Estado de memória

- Instrução da RAM é enviada ao registrado de instrução

- Palavra de controle:  $C_p E_p \bar{L}_M \bar{C}\bar{E}$     $\bar{L}_I \bar{E}_I \bar{L}_A E_A$     $S_U E_U \bar{L}_B \bar{L}_O$   
0 0 1 0   0 1 1 0   0 0 1 1



# CICLO DE EXECUÇÃO

- $T_4$ ,  $T_5$ ,  $T_6$  ... dependem de cada instrução

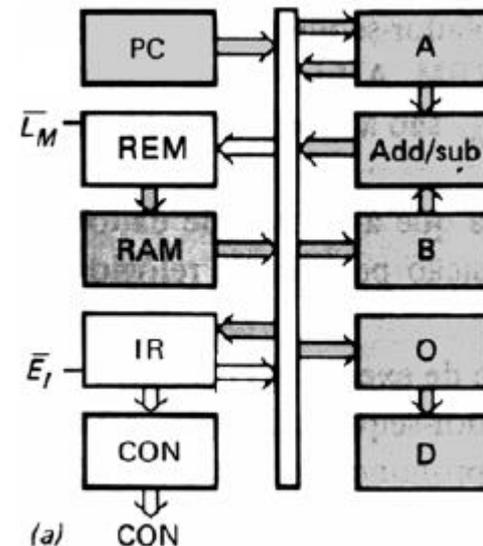
- Instrução **LDA**

- Exemplo: LDA 9H;

- IR: 0000 1001

- $T_4$

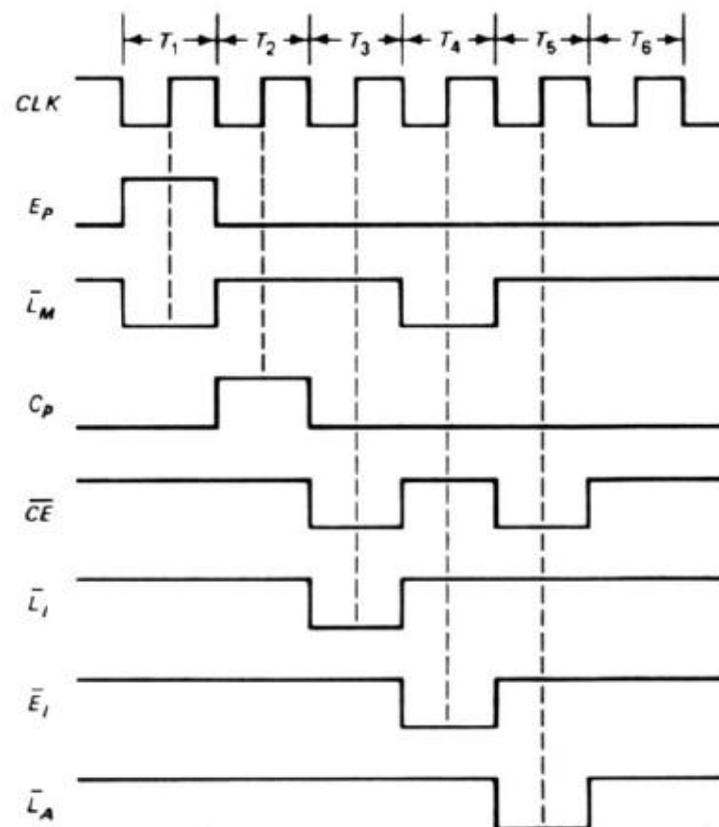
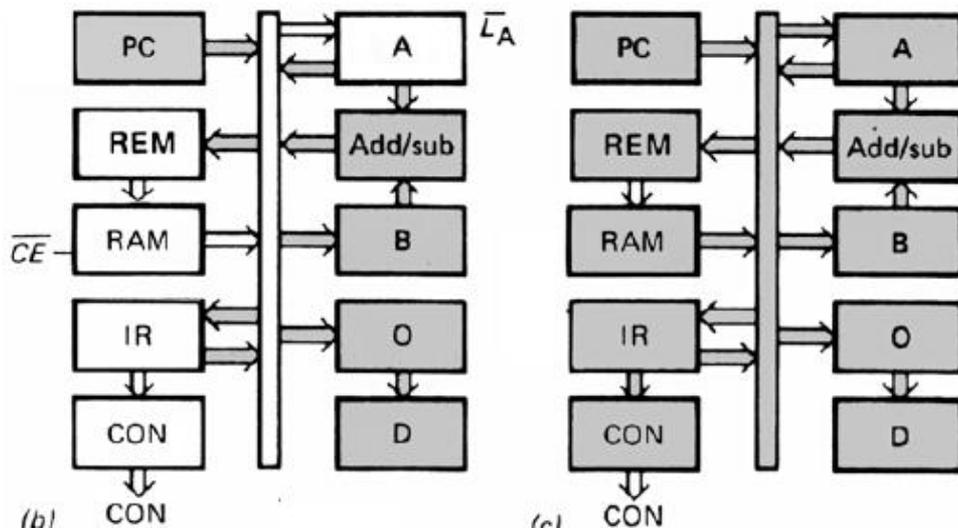
- 0000 vai para o controlador-sequencializador onde é decodificado
    - 1001 é carregado no REM



– T<sub>5</sub>

- Palavra de dados endereçada na RAM será carregada no acumulador
- Transição positiva de clock

– T<sub>6</sub> = sem operação (nop)

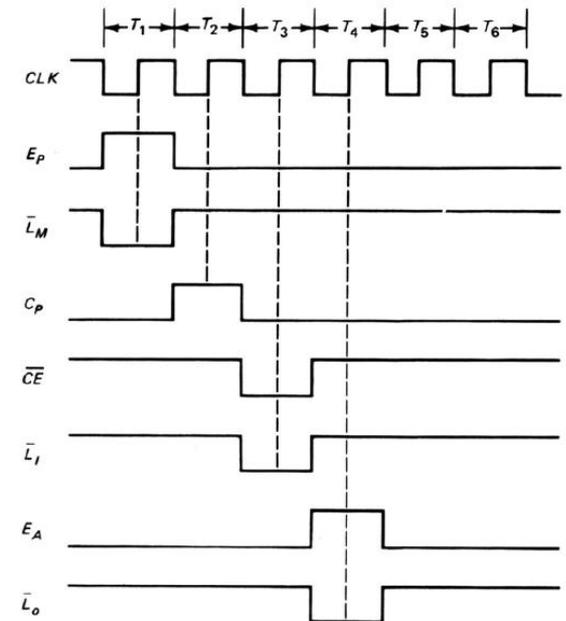
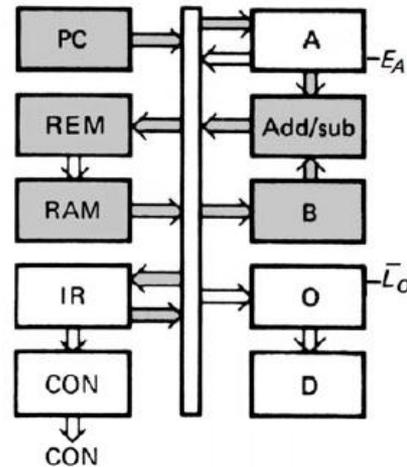




- Rotina **OUT**

- T4

- A instrução vai ao controlador-sequencializador para decodificação que emite a palavra para carregar o conteúdo do acumulador na saída



- Rotina **HALT**

- T4

- Avisa ao controlador-sequencializador para interromper CLOCK

