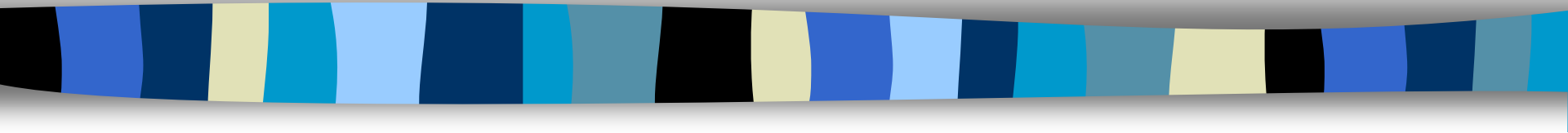


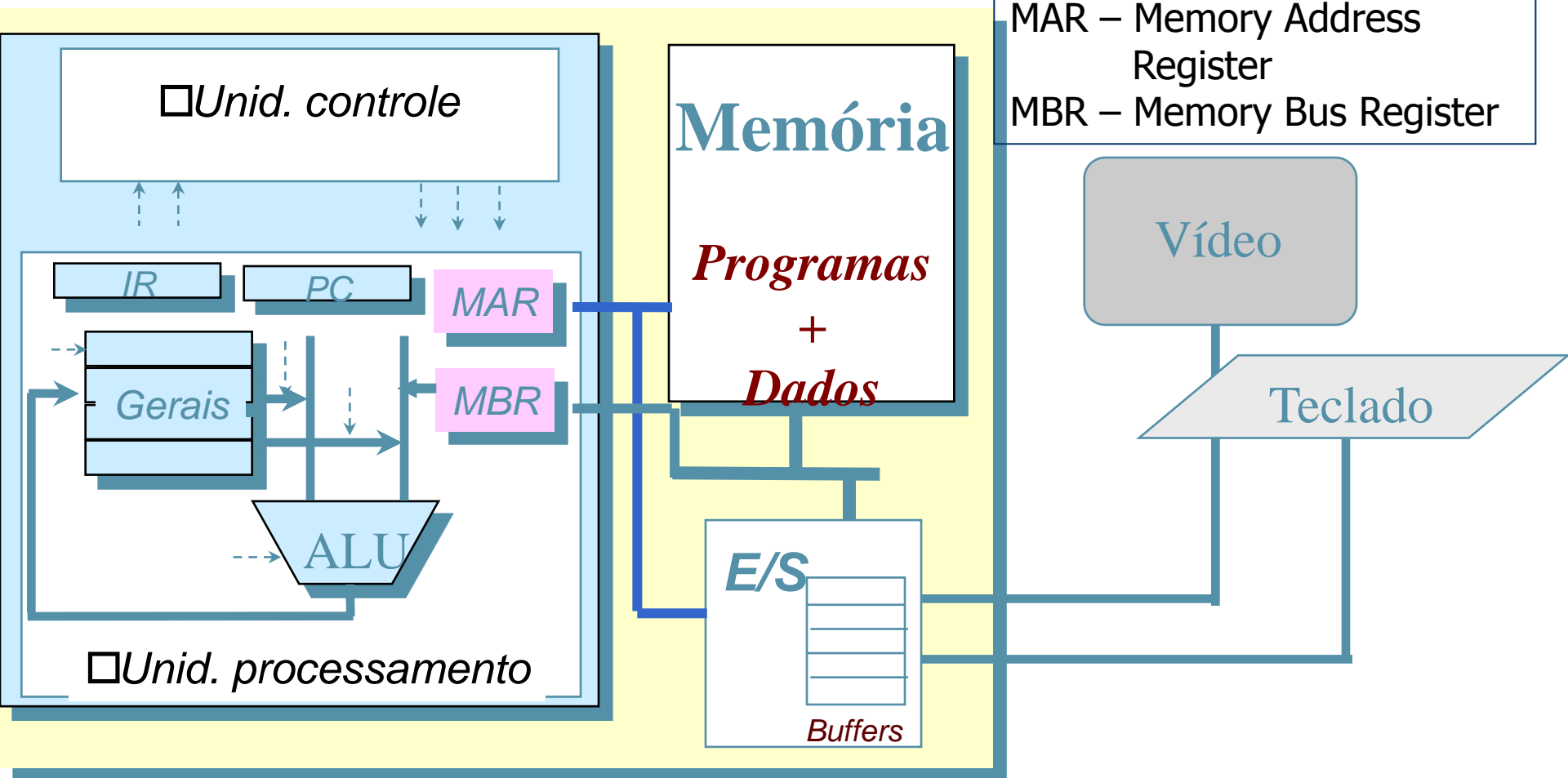
Unidade Central de Processamento



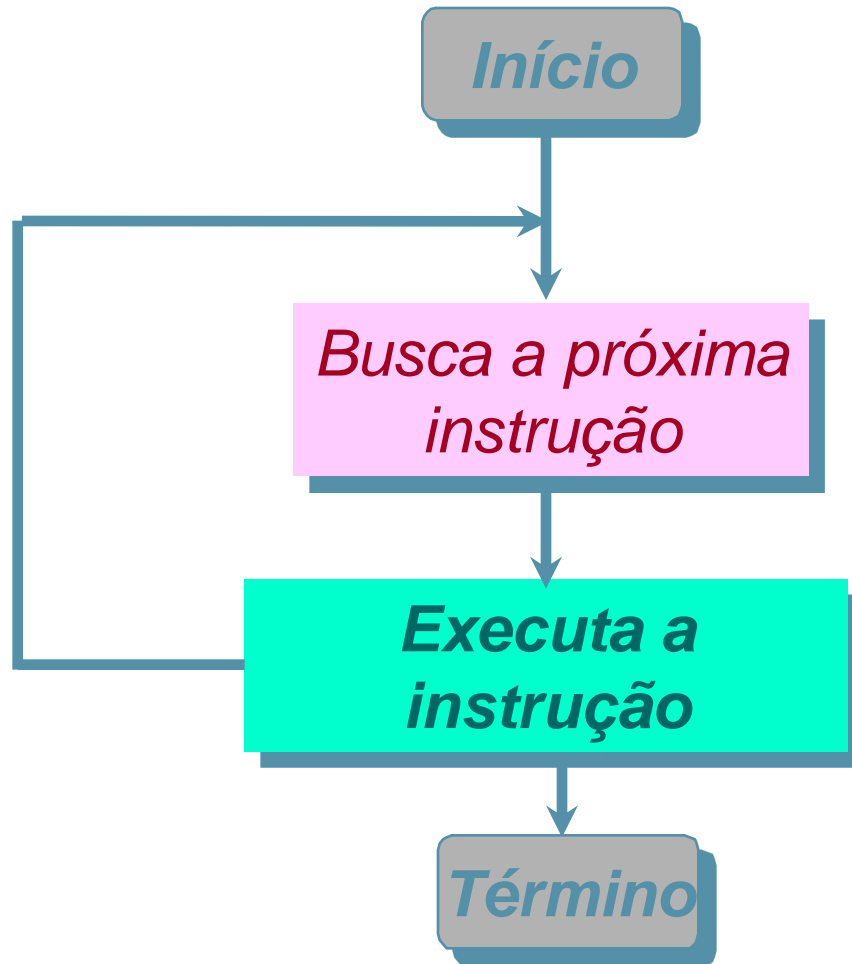
Sistemas Digitais

CIn-UFPE

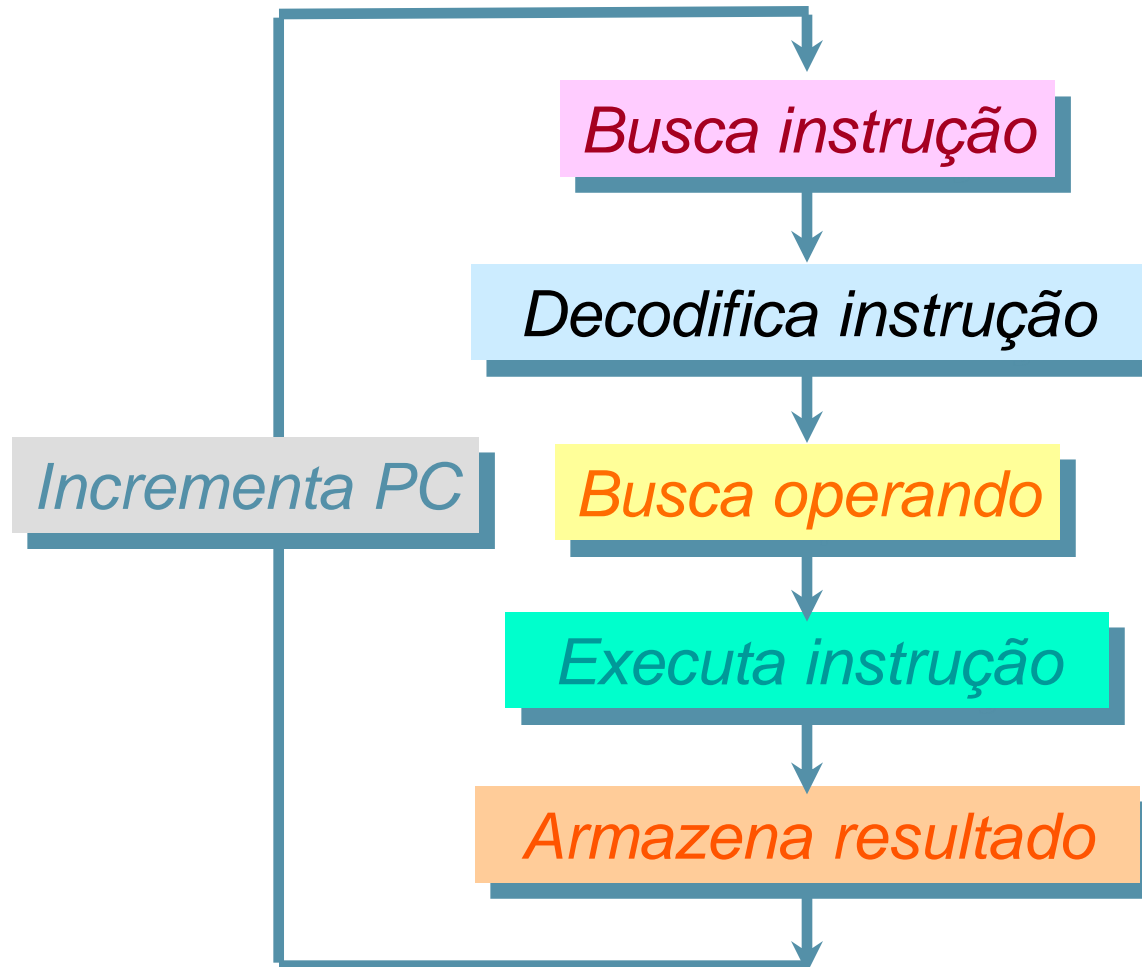
Como funciona um computador simples

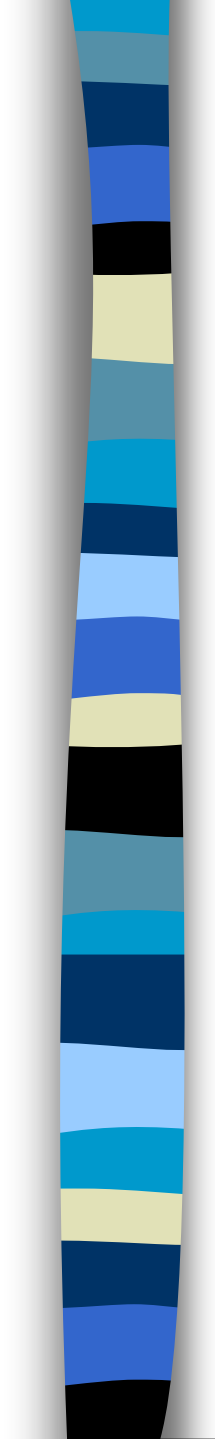


Como funciona um computador?



Executando um programa





Exemplo – Declarações simples

Muitas vezes o processamento de uma informação exige uma seqüência de transferência até que a tarefa seja terminada.

Exemplo 01:

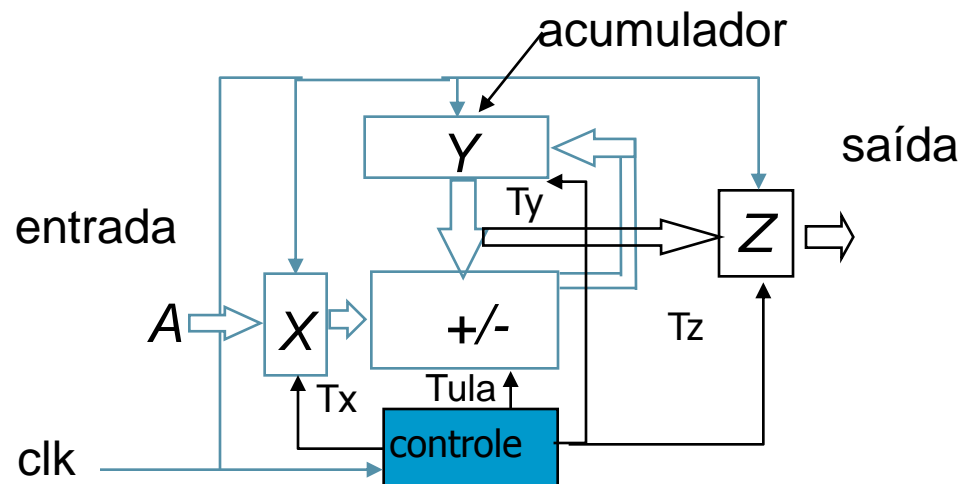
/* Programa que a função $(a+b-c)$ e mostra o resultado ($z = a+b-c$)

Sigad:	CLRLD	/* Clear acumulador Y, Z e carrega valor de A em X
	ADD	/*ADD 1o. valor, guarda resultado e lê novo valor
	ADD	/*ADD 2o .valor, guarda resultado e lê novo valor
	SUB	/*SUB 3o .valor, guarda resultado
DISP	Sigad	/*mostra resultado e repete cálculo em Z

Como construir uma **unidade de controle** que implemente este programa?

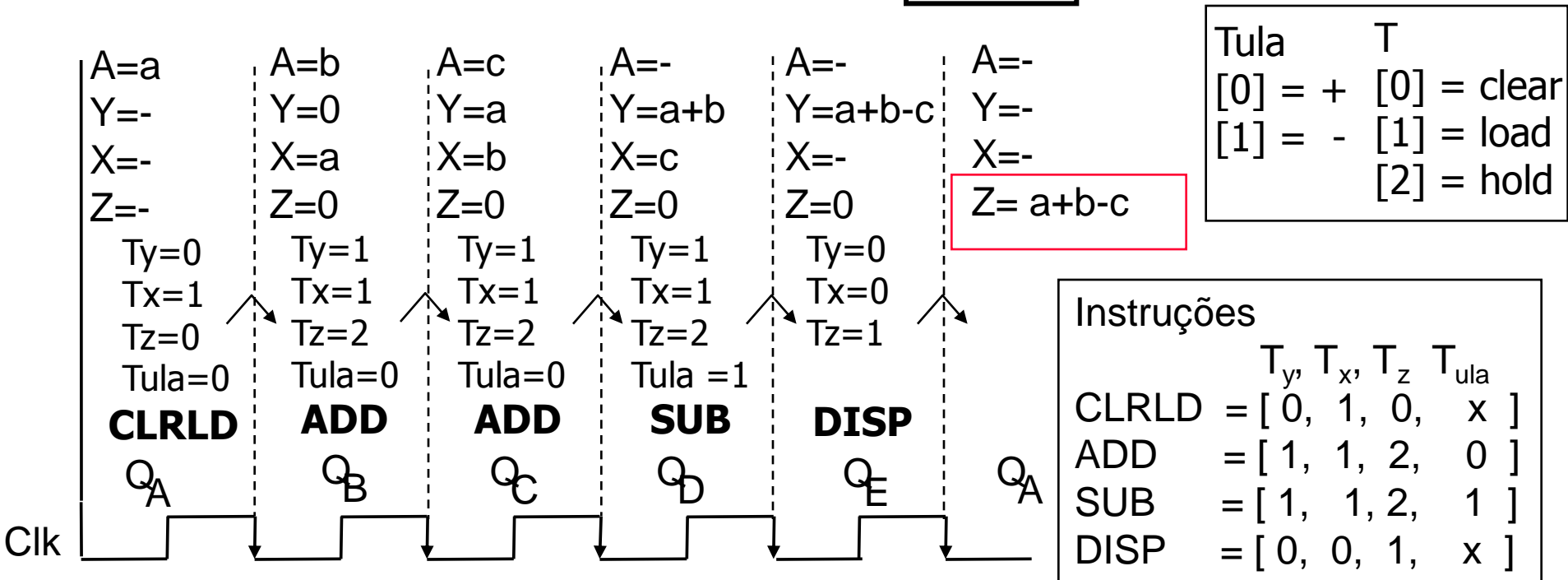
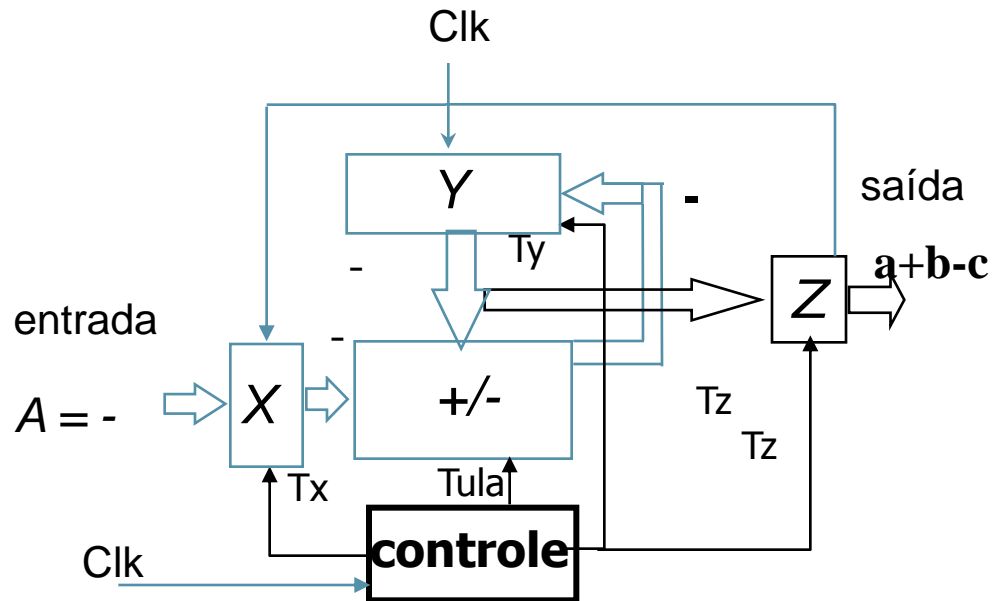
Estado Estado/saída

Q_A	Sigad: Q_B /CLRLD
Q_B	Q_C /ADD
Q_C	Q_D /ADD
Q_D	Q_E /SUB
Q_E	Q_A /DISP



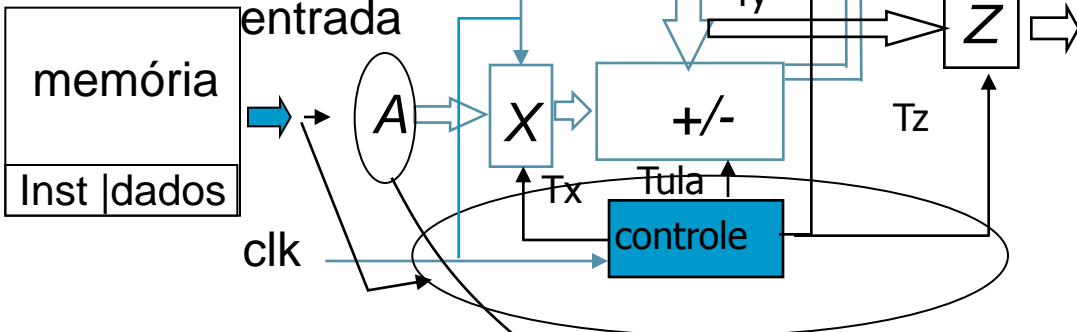
$$Z = a+b-c$$

Q_A	Sigad: Q_B/CLRLD
Q_B	Q_C /ADD
Q_C	Q_D /ADD
Q_D	Q_E /SUB
Q_E	Q_A /DISP

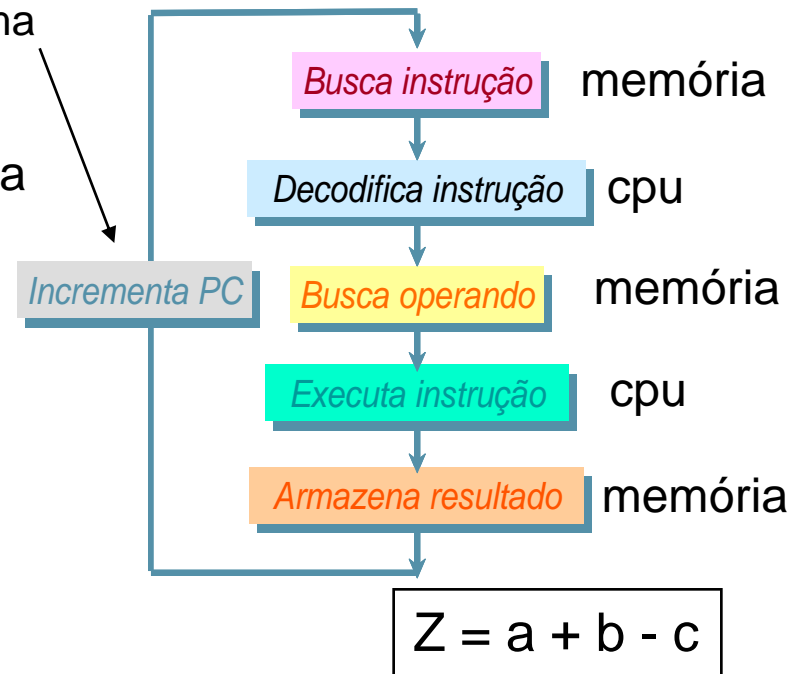


Instruções	T_y	T_x	T_z	T_{ula}
CLRLD	0	1	0	x
ADD	1	1	2	0
SUB	1	1	2	1
DISP	0	0	1	x

$$Z = a + b - c$$



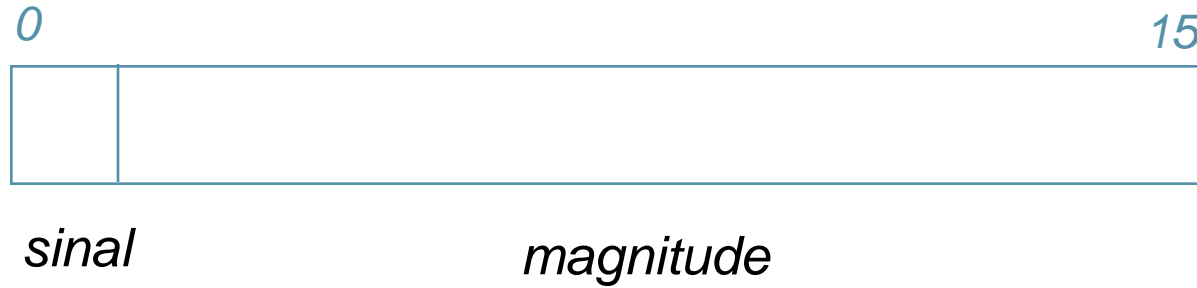
Execução de um programa



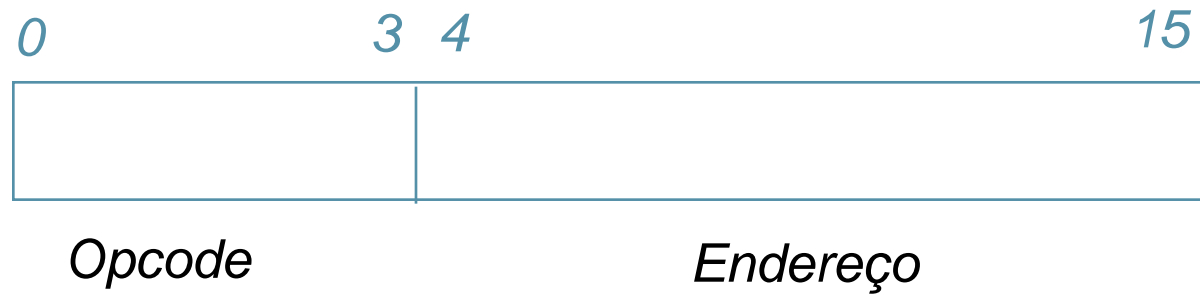
Contador do Programa (PC)	Memória:	cpu
	Instruções	
	end(dados)	
00	[0,1,0,x] CLRLD end+0(a)	X = -; Y = -; Z = -;
01	[1,1,2,0] ADD end+1(b)	X = a; Y = 0; Z = 0;
02	[1,1,2,0] ADD end+2(c)	X = b; Y = a; Z = 0;
03	[1,1,2,1] SUB -	X = c; Y = a+b; Z = 0;
04	[0,0,1,x] DISP -	X = -; Y = a+b-c; Z = 0;
05	-	X = -; Y = a+b-c; Z = a+b-c;

Memória

■ *Dado*



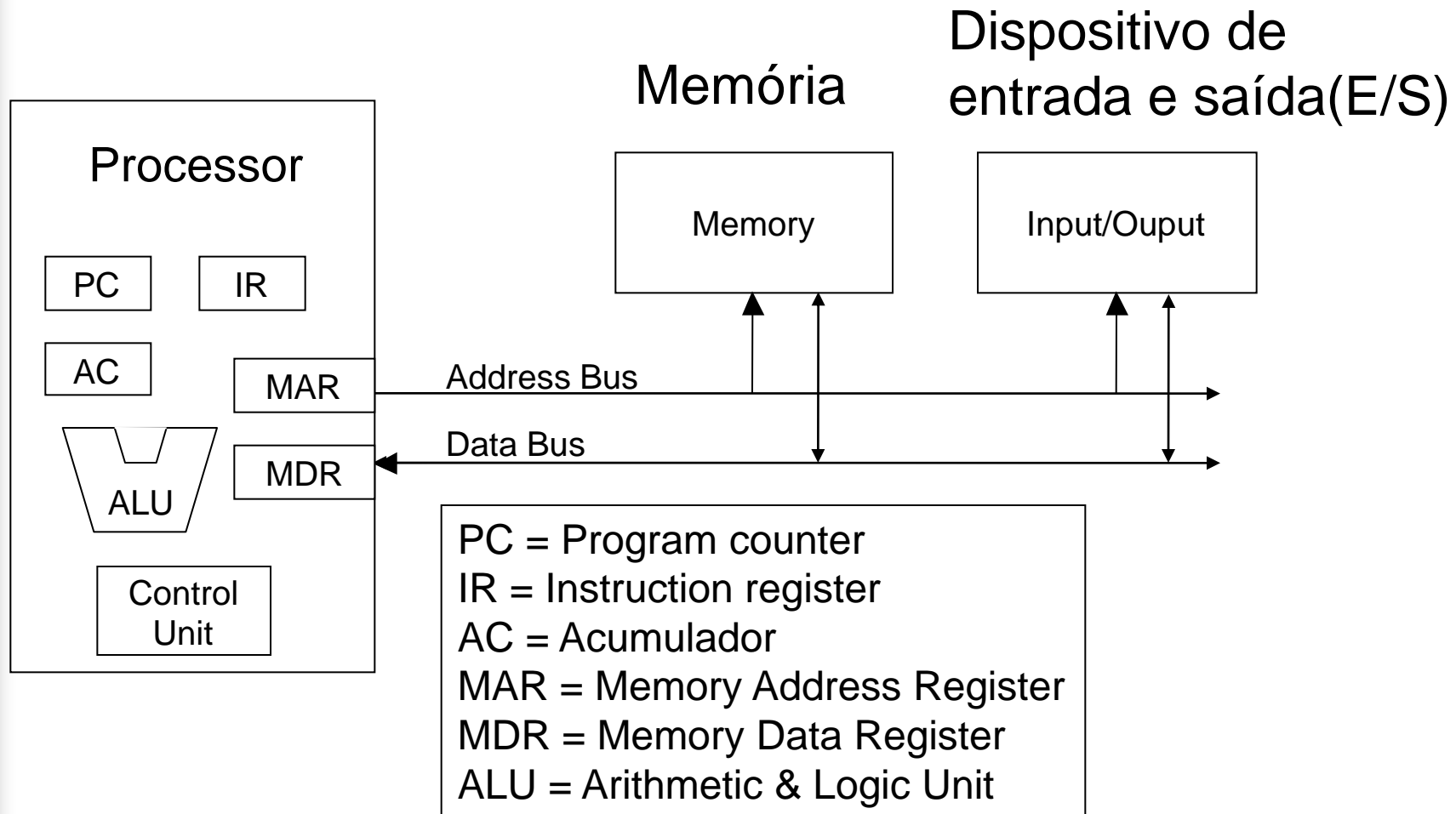
■ *Instrução*



Opcode

0001	AC \leftarrow Mem.
0010	Mem. \leftarrow AC
0101	AC \leftarrow AC + Mem.

Arquitetura do computador





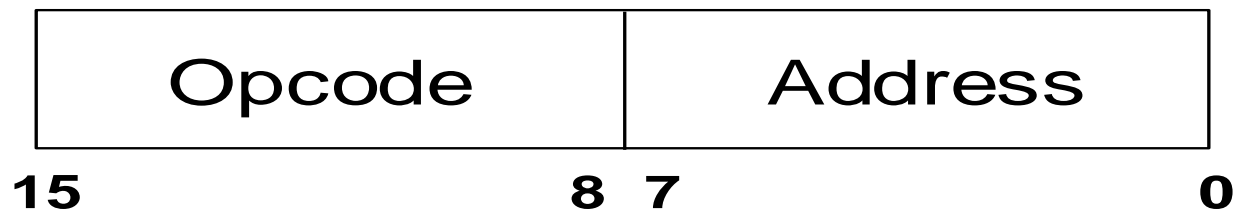
Arquitetura do computador

- PC - program counter – indica a próxima instrução a ser executada.
- IR - Instruction register – recebe a instrução a ser decodificada pela CPU.
- AC – Acumulador (registrador auxiliar). Guarda temporariamente valores sendo calculados.
- MAR – Registrador que indica a próxima posição de memória a ser referenciada. Conectado ao barramento de endereços.
- MDR – Registrador usado para receber ou transmitir dados. Conectado ao barramento de dados.
- ALU (ULA) – Unidade Lógica

Instruções básicas

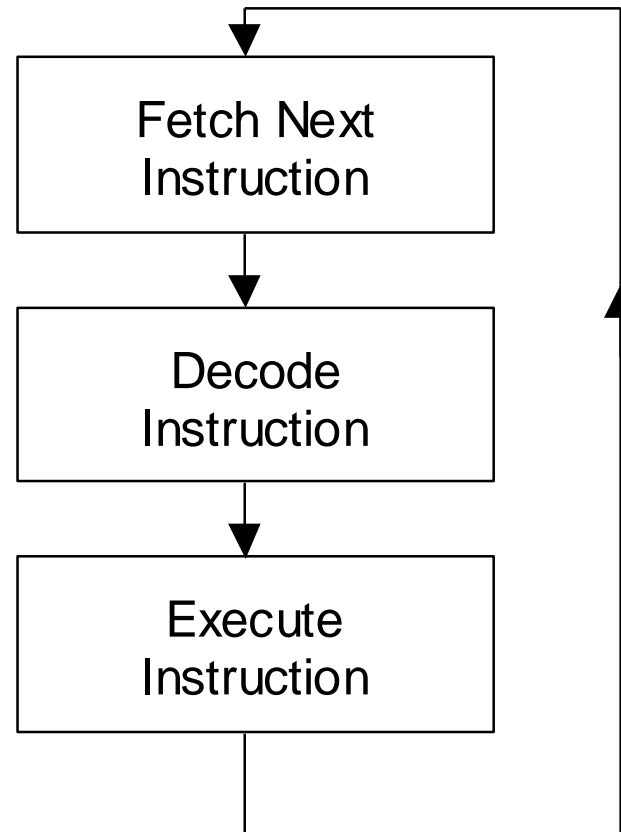
ADD implementa $Y = X + Y$ (acumulador)
SUB implementa $Y = Y - X$
CLRLD faz $X = \text{adr}(\text{memória})$, limpa Y e Z
MULT2 faz $Y = Y * 2$
DIV2 faz $Y = Y / 2$

Formato da instrução do computador



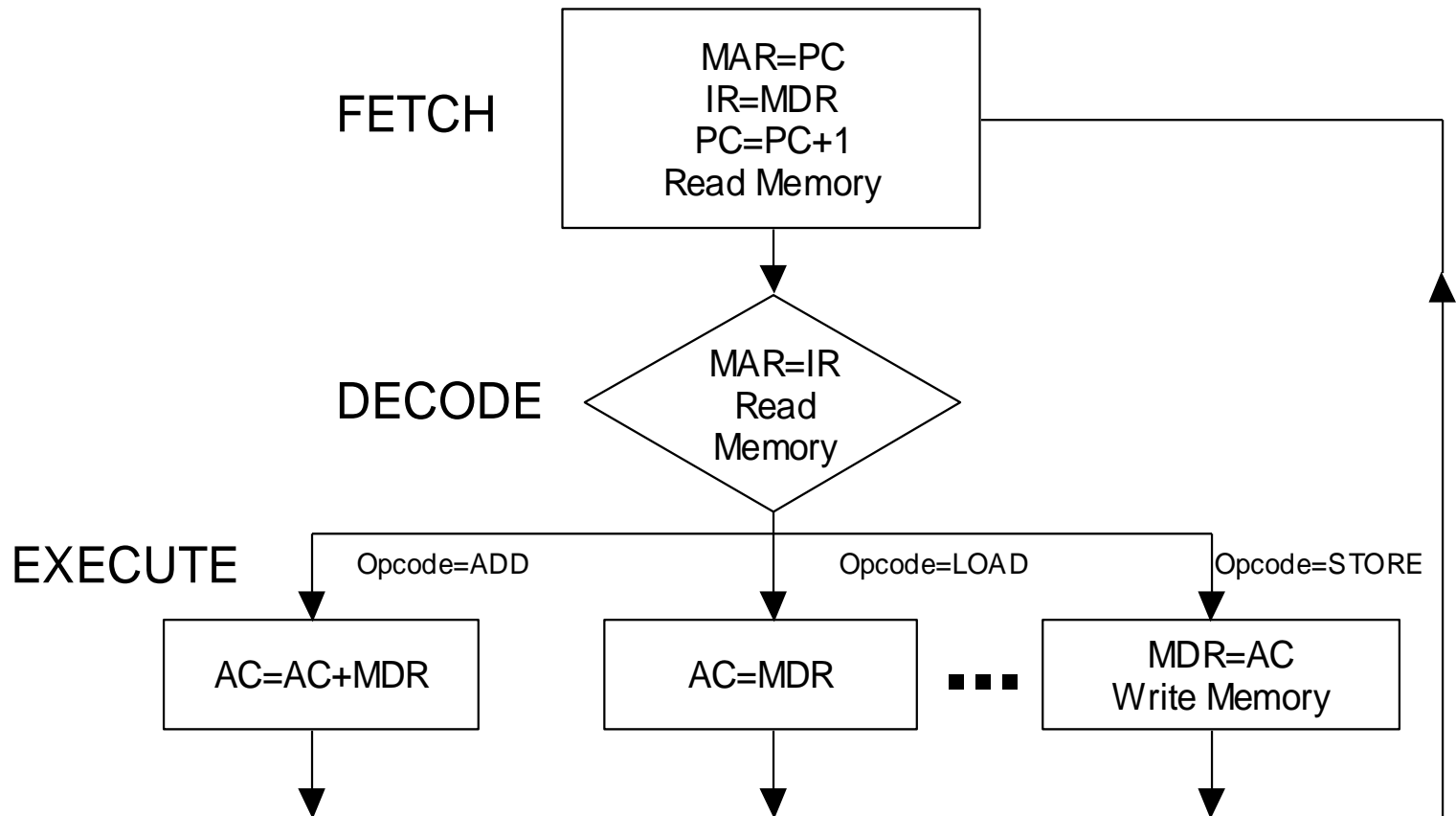
CPU - μ P1

- Fluxo de execução de instruções

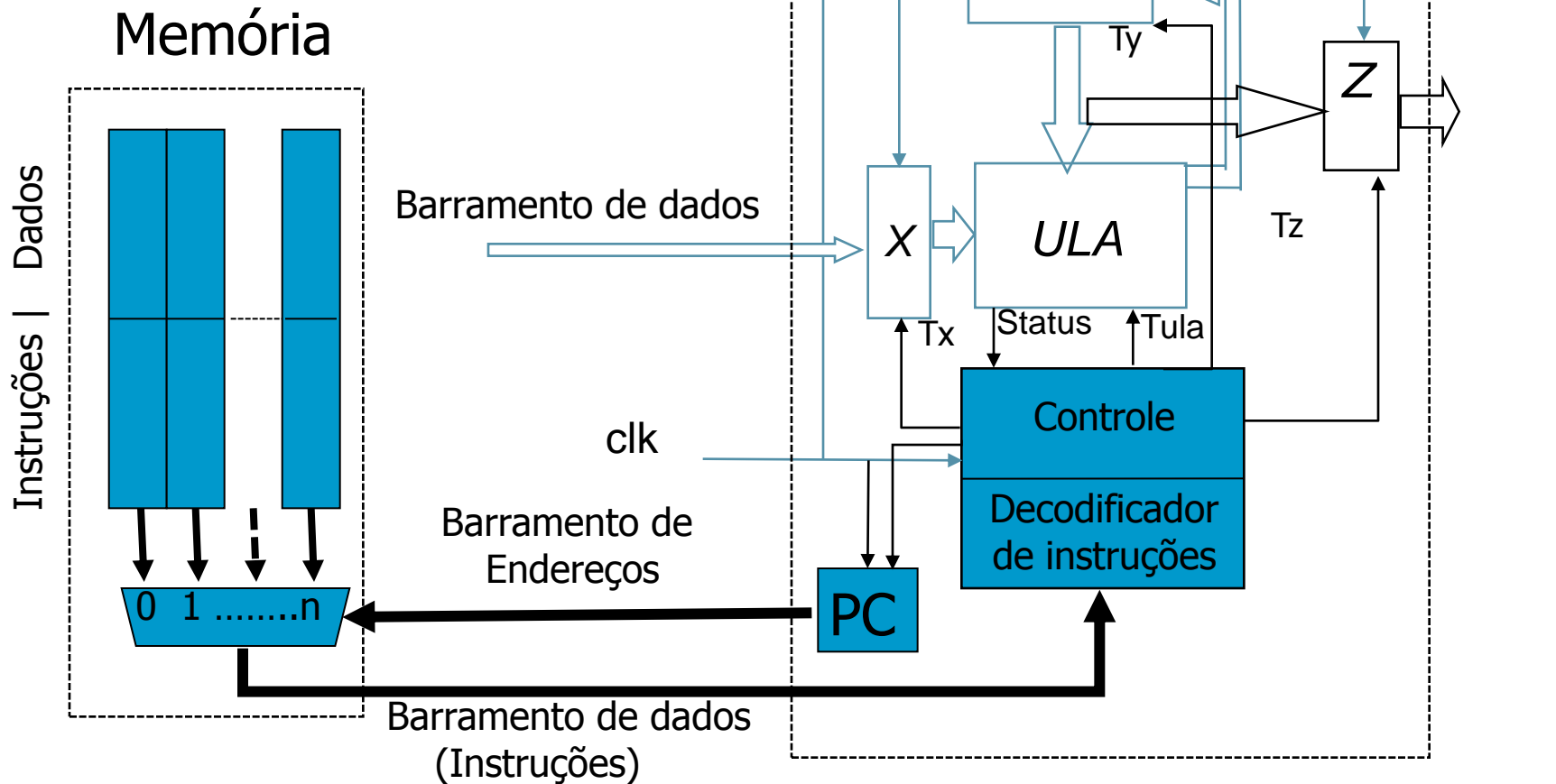


CPU - $\mu P1$

- Fluxo de execução de instruções detalhada



Instrução	Dados
4 bits	4 bits





Projeto

- Implementar as seguintes funções na arquitetura computacional projetada
 1. $Z = (a+c-d)*2$
 2. $Z = (a+b)/2$
 3. $X=a; Y=b; \text{if}(X>Y) \text{ then } Z= Y \text{ else } Z=X;$
a e b, c e d são variáveis