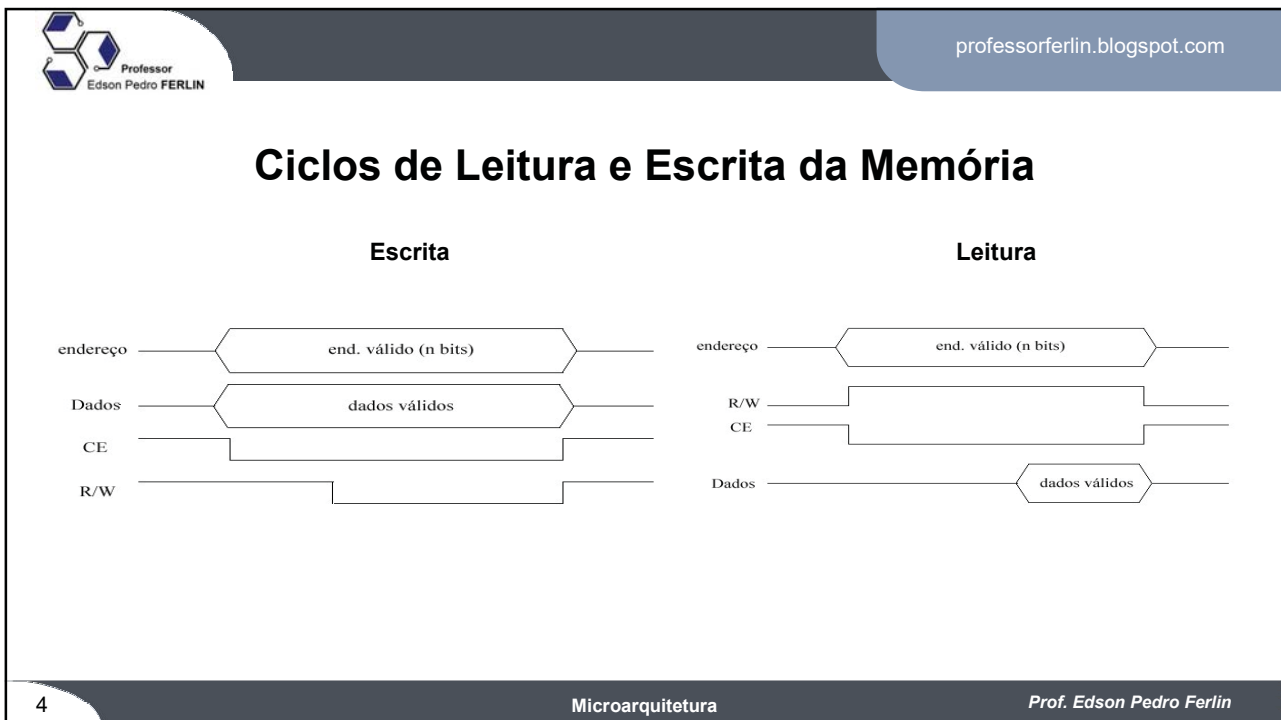
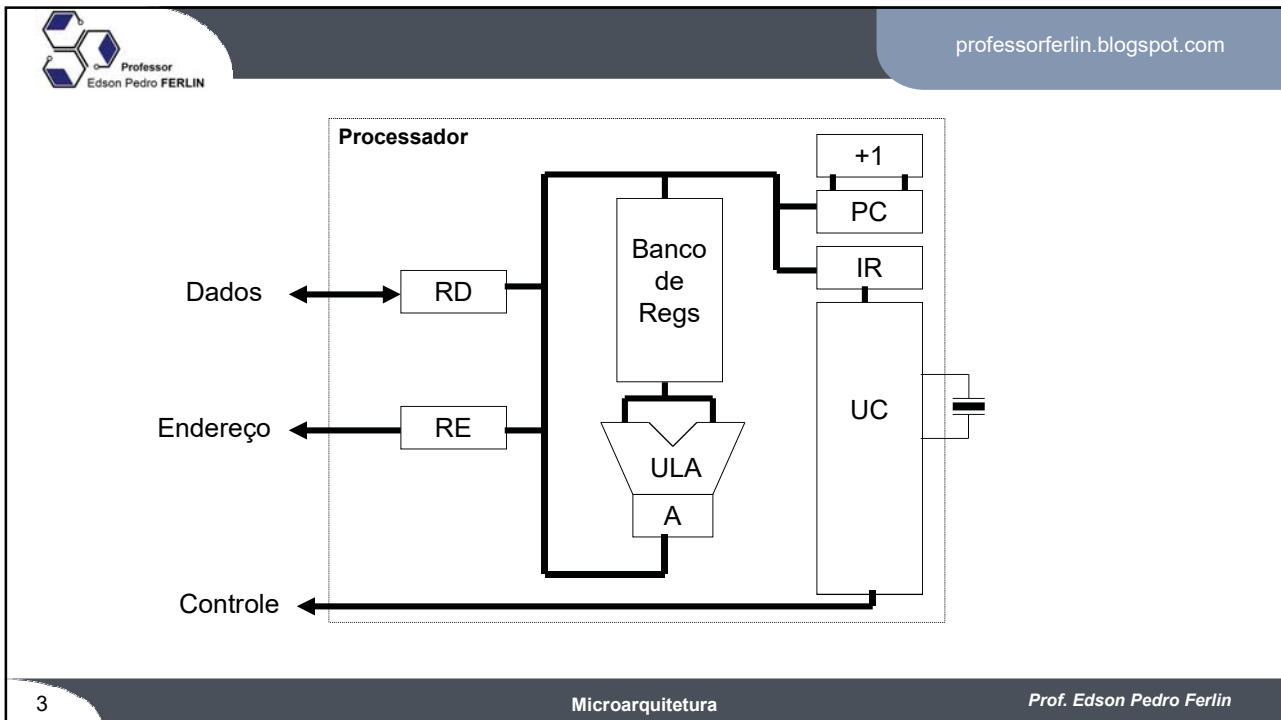


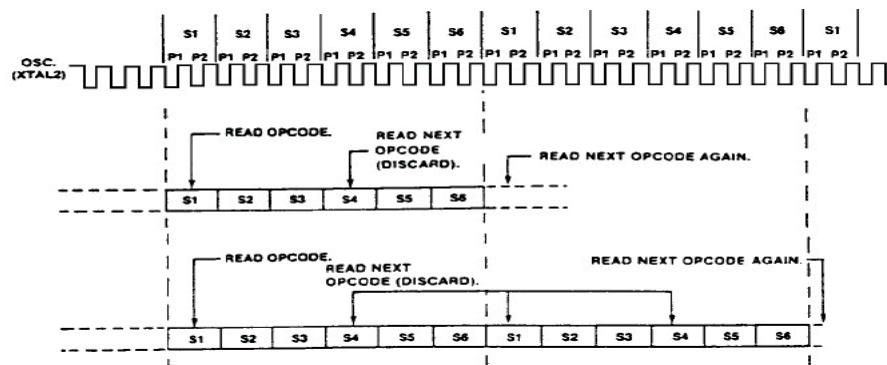
Microarquitetura

Prof. Edson Pedro Ferlin

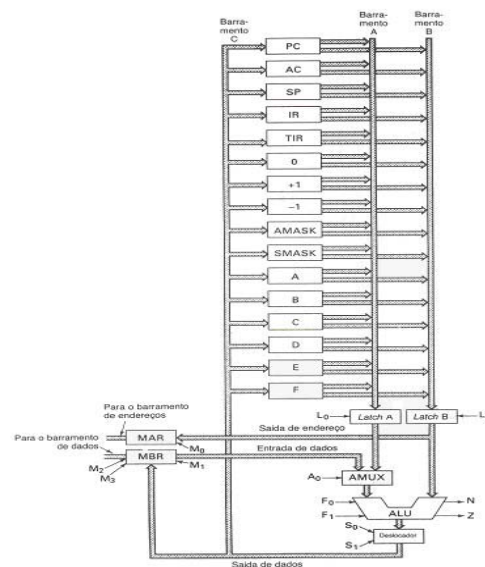
- **Objetivos**
 - Apresentar os conceitos de Microprogramação
- **Conteúdos**
 - Definições
 - Microarquitetura
 - Microprogramação
 - Macroarquitetura



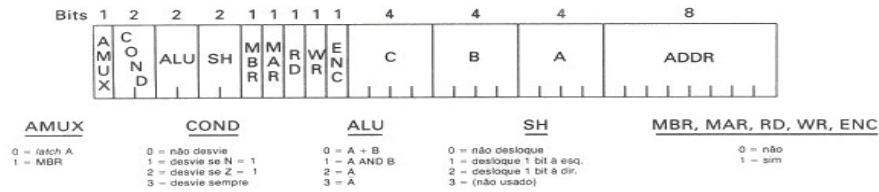
Ciclos de Máquina e de Clock



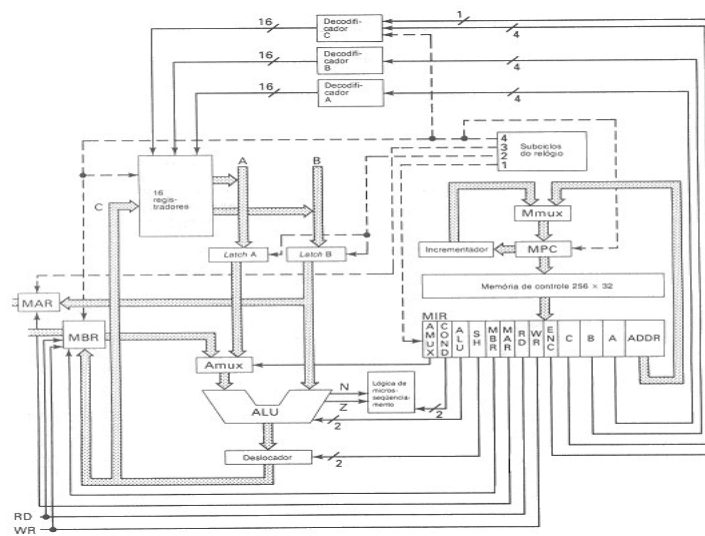
Fluxo de Dados



Palavra de Controle - Gabarito



Microarquitectura



Conjunto de Instruções

Binário	Mnemônico	Instrução	Significado
0000xxxxxxxxxxxx	LODD	Carrega direto	$ac := m[x]$
0001xxxxxxxxxxxx	STOD	Armazena direto	$m[x] := ac$
0010xxxxxxxxxxxx	ADDD	Adiciona direto	$ac := ac + m[x]$
0011xxxxxxxxxxxx	SUBD	Subtrai direto	$ac := ac - m[x]$
0100xxxxxxxxxxxx	JPOS	Desvia se positivo	if $ac \geq 0$ then $pc := x$
0101xxxxxxxxxxxx	JZER	Desvia se zero	if $ac = 0$ then $pc := x$
0110xxxxxxxxxxxx	JUMP	Desvia	$pc := x$
0111xxxxxxxxxxxx	LOCO	Carega constante	$ac := x$ ($0 \leq x \leq 4095$)
1000xxxxxxxxxxxx	LODL	Carga total	$ac := m[sp + x]$
1001xxxxxxxxxxxx	STOL	Armazena local	$m[x + sp] := ac$
1010xxxxxxxxxxxx	ADDL	Adiciona local	$ac := ac + m[sp + x]$
1011xxxxxxxxxxxx	SUBL	Subtrai local	$ac := ac - m[sp + x]$
1100xxxxxxxxxxxx	JNEG	Desvia se negativo	if $ac < 0$ then $pc := x$
1101xxxxxxxxxxxx	JNZE	Desvia se não zero	if $ac \neq 0$ then $pc := x$
1110xxxxxxxxxxxx	CALL	Chama procedimento	$sp := sp - 1$; $m[sp] := pc$; $pc := x$
1111000000000000	PSHI	Empilha indireto	$sp := sp - 1$; $m[sp] := m[ac]$
1111001000000000	POPI	Desempilha indireto	$m[ac] := m[sp]$; $sp := sp + 1$
1111010000000000	PUSH	Coloca na pilha	$sp := sp + 1$; $m[sp] := ac$
1111011000000000	POP	Retira da pilha	$ac := m[sp]$; $sp := sp + 1$
1111100000000000	RETN	Retorna	$pc := m[sp]$; $sp := sp + 1$
1111101000000000	SWAP	Troca ac, sp	$tmp := ac$; $ac := sp$; $sp := tmp$
11111100yyyyyyyy	INSP	Incrementa sp	$sp := sp + y$ ($0 \leq y \leq 255$)
11111110yyyyyyyy	DESP	Decrementa sp	$sp := sp - y$ ($0 \leq y \leq 255$)

xxxxxxxxxx é um endereço de máquina de 12 bits; na coluna 4, ele é chamado de x.
yyyyyyyy é uma constante de 8 bits; na coluna 4, ele é chamado de y.



```

0: mar := pc; rd;
1: pc := pc + 1; rd;
2: ir := mbr; if n then goto 28;
3: ir := lshift(ir + ir); if n then goto 19;
4: ir := lshift(ir); if n then goto 11;
5: alu := ir; if n then goto 9;
6: mar := ir; rd;
7: rd;
8: ac := mbr; goto 0;
9: mar := ir; mbr := ac; wr;
10: wr := goto 0;
11: alu := ir; if n then goto 15;
12: mar := ir; rd;
13: rd;
14: ac := mbr + ac; goto 0;
15: mar := ir; rd;
16: ac := ac + 1; rd;
17: a := inv(mbr);
18: ac := ac + a; goto 0;
19: ir := lshift(ir); if n then goto 25;
20: alu := ir; if n then goto 23;
21: alu := ac; if n then goto 0;
22: pc := band(ir, amask); goto 0;
23: alu := ac; if x then goto 22;
24: goto 0;
25: alu := ir; if n then goto 27;
26: pc := band(ir, amask); goto 0;
27: ac := band(ir, umask); goto 0;
28: ir := lshift(ir + ir); if n then goto 40;
29: ir := lshift(ir); if n then goto 35;
30: alu := ir; if n then goto 33;
31: a := ir + sp;
32: mar := a; rd; goto 7;
33: a := ir + sp;
34: mar := a; mbr := ac; wr; goto 10;
35: alu := ir; if n then goto 38;
36: a := ir + sp;
37: mar := a; rd; goto 13;
38: a := ir + sp;
39: mar := a; rd; goto 16;

```

Microprograma

```

40: ir := lshift(ir); if n then goto 46;
41: alu := ir; if n then goto 44;
42: alu := ac; if n then goto 22;
43: goto 0;
44: alu := ac; if x then goto 0;
45: pc := band(ir, amask); then goto 0;
46: ir := lshift(ir); if n then goto 50;
47: sp := sp + (-1);
48: mar := sp; mbr := pc; wr;
49: pc := band(ir, amask); wr; goto 0;
50: ir := lshift(ir); if n then goto 65;
51: ir := lshift(ir); if n then goto 59;
52: alu := ir; if n then goto 56;
53: mar := ac; rd;
54: sp := sp + (-1); rd;
55: mar := sp; wr; goto 10;
56: mar := sp; sp := sp + 1; rd;
57: rd;
58: mar := ac; wr; goto 10;
59: alu := ir; if n then goto 62;
60: sp := sp + (-1);
61: mar := sp; mbr := ac; wr; goto 10;
62: mar := sp; sp := sp + 1; rd;
63: rd;
64: ac := mbr; goto 0;
65: ir := lshift(ir); if n then goto 73;
66: alu := ir; if n then goto 70;
67: mar := sp; sp := sp + 1; rd;
68: rd;
69: pc := mbr; goto 0;
70: a := ac;
71: ac := sp;
72: sp := a; goto 0;
73: alu := ir; if n then goto 76;
74: a := band(ir, smask);
75: sp := sp + a; goto 0;
76: a := band(ir, smask);
77: a := inv(a);
78: a := a + 1; goto 75;

```

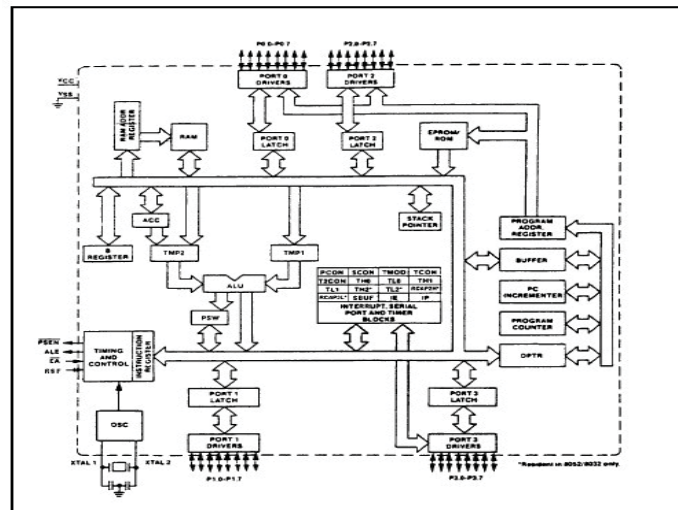
Macroarquitetura (ISA – *Instruction Set Architecture*)

- Conjunto de Instruções
- Registradores
- Modelo de Memória
 - Endereçamento
 - Pilha
 - Memória Cache
 - Unidades Funcionais

Exemplos de Microarquiteturas



Arquitetura do 8051

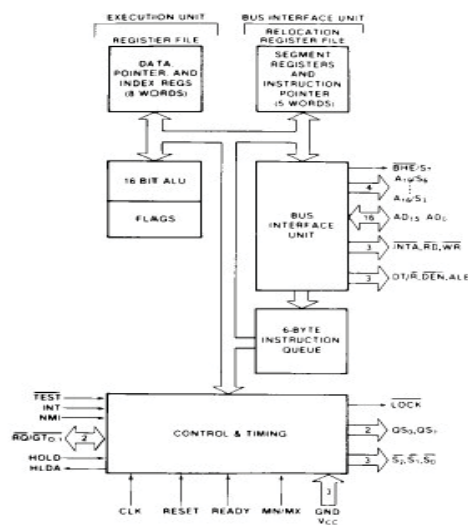


13

Microarquitetura

Prof. Edson Pedro Ferlin

Arquitetura do 8086

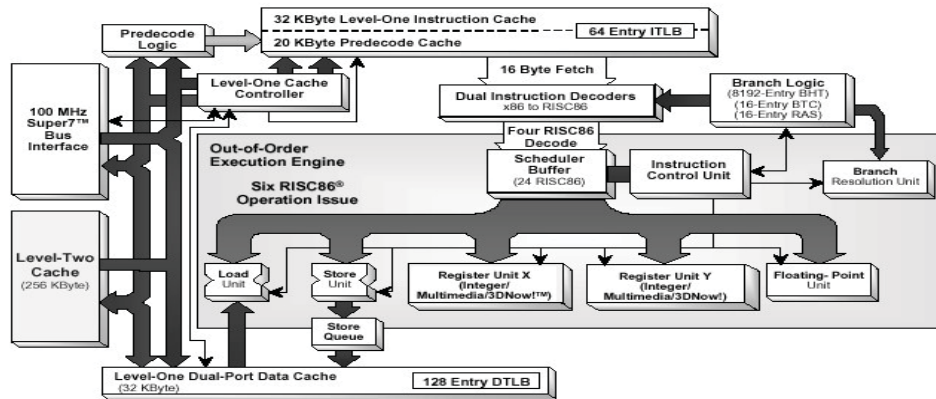


14

Microarquitetura

Prof. Edson Pedro Ferlin

Arquitetura do AMD K6 - III

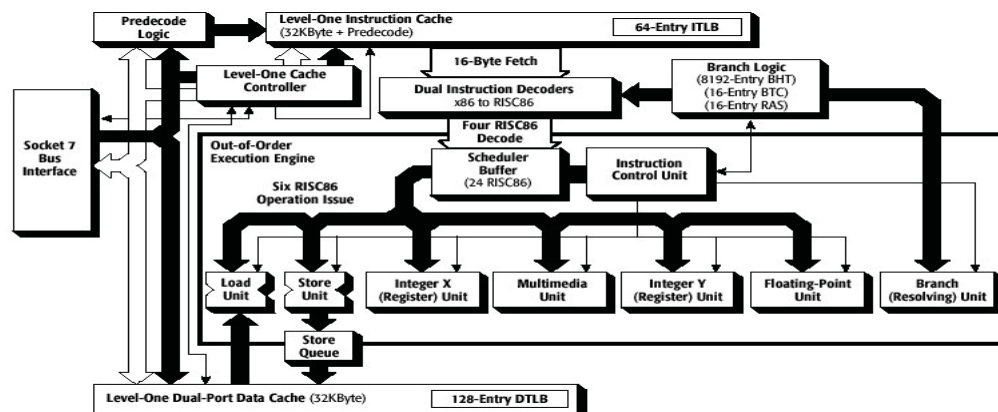


17

Microarquitetura

Prof. Edson Pedro Ferlin

Arquitetura do AMD K6 - MMX

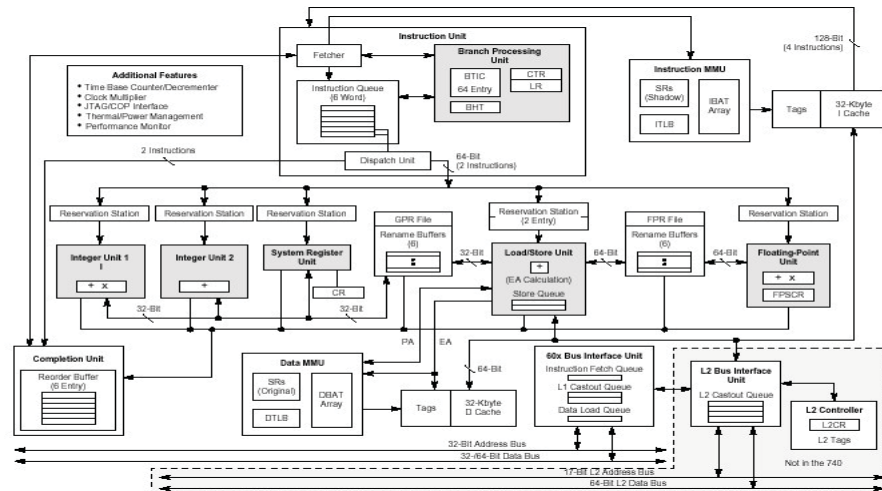


18

Microarquitetura

Prof. Edson Pedro Ferlin

Arquitetura do PowerPC 750

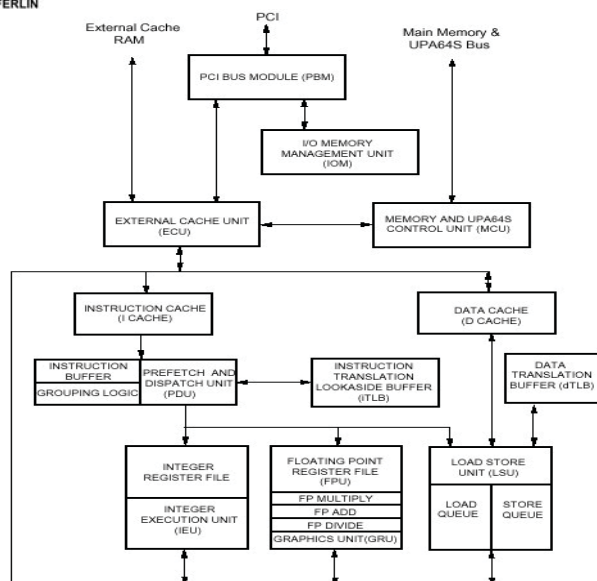


19

Microarquitetura

Prof. Edson Pedro Ferlin

Arquitetura do Ultra-SPARC III



20

Microarquitetura

Prof. Edson Pedro Ferlin

Atividade

- Resolver os exercícios.

