

30. CPU. 8 biți  
16 biți adrese

reg. A 8 biți

reg. PC, X 16 biți

opcode  $\rightarrow$  1 octet = 8 biți

instr. are 3 octeți, 1 octet este opcode  $\Rightarrow 3-1=2$  ~~16 biți~~

0 procesor intern  $\rightarrow$  2 tacte  
[ extern  $\rightarrow$  4 tacte

fetch opcode  $\rightarrow$  6 tacte

nr tacte totale pt. instr. urm. <sup>deplasament</sup>  
ST A, (X).

1. Prima dată, facem fetch opcode 6

!!! forță de 2 ori pt. 6 procesor e pe 8 biți și mai avem 16 biți

2. fetch offset 1 <sup>6T</sup>

3. fetch offset 2 <sup>6T</sup>

4. calcul adresă  $(X + \text{offset})$  <sup>2T</sup>

5. scriere în memorie <sup>4T</sup>

20T op interogare = 100T

66 / 45 Tm

CPU 16 biți

32 biți de adresă

CPU 32 biți

f = 50 MHz

Pb. 36/11

a) interfața cu memorie de 30/5. <sup>tb interogare</sup>

Cât la sută din procesor necesită interogarea perifericului / din timpul total al procesorului?

timp interogare =  $\frac{\text{nr. interogări} \cdot \text{cât durează op. de interogare}}{f}$

f

①

$$1T_{\text{procesor}} = \frac{1}{f} = \frac{1}{50 \cdot 10^6}$$

$$t_{\text{imp interogare}} = \frac{30 \cdot 100}{50 \cdot 10^6} = \frac{3}{5 \cdot 10^4} = 0,6 \cdot 10^{-4}$$

→ Ca % det procent înmulțim cu 100  
 $\Rightarrow 0,6 \cdot 10^{-2} \% = 0,006 \%$

b) floppy → vizual transfer date spre CPU în unități de  
 16 bit și are o rată de transfer de 50 k/s

→  
 1 interogare --- 2 B  
 X int --- 50 KB

$$X = \frac{50 \text{ KB} \cdot 1 \text{ interogare}}{2 \text{ B}} = 25 \cdot 10^3 \text{ interogări}$$

$$t_{\text{int}} = \frac{n \cdot t_{\text{int}} \cdot \text{op int}}{f} = \frac{25 \cdot 10^3 \cdot 100 \text{ T}}{50 \cdot 10^6} = \frac{25}{50 \cdot 10} = 0,05 \text{ s}$$

c)

32 bit.

rata de transfer 2 Mo/s

1 int. --- 4 octeți

X int. --- 2 Mo

Procesorul poate transfera

într-o sec. 2 Mo

La o întrebare, procesorul a 4 octeți

$$X = \frac{2 \cdot 10^6}{4} = 0,5 \cdot 10^6 \text{ interogări}$$

$$t_{\text{int}} = \frac{n \cdot t_{\text{int}} \cdot \text{op int}}{f} = \frac{0,5 \cdot 10^6 \cdot 100}{50 \cdot 10^6} = \frac{5 \cdot 10^4}{5 \cdot 10^4} = 1$$

$$\Rightarrow 1 \cdot 100 = 100 \%$$

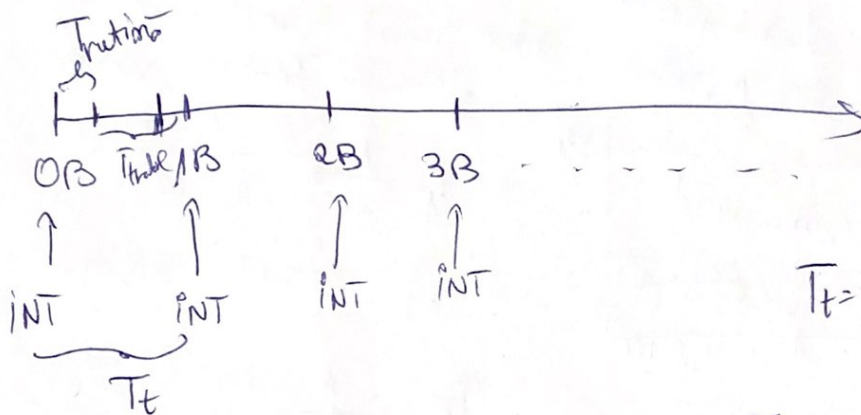
$$8b = 10$$

$$1KB \neq 1000B$$

$$1KB = 2^{10}B$$

$$1MB = 2^{10}KB \cdot 2^{10} \cdot 2^{10}B$$

42 Data  $t = 25 \cdot 10^4 b/s$   
Transfer dintre disp. periferic și CPU



$T_t$  = timp total dintre  
2 interrupturi

$$T_t = ?$$

$$1 octet \dots \times D$$

$$\dots 1 D.$$

$$\frac{25 \cdot 10^4}{8}$$

$$\rightarrow T_t = \frac{8}{25 \cdot 10^4} \quad \triangleright$$

$T_{rutina} = 2 \mu s$  (până la  
începutul de  
rutina de  
trebură)

$$T_{tratare} = 10 \mu s$$

$$T_{tranz} = T_t - T_{rutina} - T_{tratare}$$

$$T_t = 0,32 \cdot 10^{-4} = 32 \cdot 10^{-6} s = 32 \mu s$$

$$T_{tranz} = 32 - 2 - 10 = 20 \mu s$$



|    | Clock $\mu$ | CPI A | CPI B | CPI C | CPI D |
|----|-------------|-------|-------|-------|-------|
| P1 | 2,5 GHz     | 1     | 2     | 3     | 3     |
| P2 | 3 GHz       | 2     | 2     | 2     | 2     |

CPI = clock per instruction

↓  
indicator de performanță

inversul

IPC = gate instr  
se poate calcula  
immediat de procesor

Se execută  $10^6$  instr.

- 10% clasa A  $\rightarrow 10^5$  instr.
- 20% clasa B  $\rightarrow 2 \cdot 10^5$  instr.
- 50% clasa C  $\rightarrow 5 \cdot 10^5$  instr.
- 20% clasa D  $\rightarrow 2 \cdot 10^5$  instr.

Core P2 e mai rapid

pt. P1:  $10^5 \text{ instr.} \cdot 1 \cdot 2,5 \cdot 10^{-9}$

+

pt. P1:  $10^5 \cdot 1 \rightarrow \text{CPI}$   
 $+ 2 \cdot 10^5 \cdot 2 + 5 \cdot 10^5 \cdot 3 + 3 \cdot 10^5 \cdot 3$   
 $10^9 \cdot 2,5$

$P_1 = \frac{10^5 (1 + 4 + 15 + 6)}{10^9 \cdot 2,5} = \frac{26}{2,5 \cdot 10^4} = 10,4 \cdot 10^{-4} \text{ s}$

pt. P2:  $\frac{10^5 (2 + 4 + 2 + 2)}{3 \cdot 10^9} = 0,66 \cdot 10^{-3} \text{ s}$

P2 mai rapid

! trebuie să vedem

timpul în care  
se execută cele  $10^6$  instr.

$\rightarrow$  dacă transferăm procesul în perioade  
afirmăm un test de procesor

P1:  $T_1 = \frac{1}{f} = \frac{1}{2,5 \cdot 10^9} =$   
 $T = 2,5 \cdot 10^{-9} = 2,5 \text{ ns}$

$T_{P2} = \frac{1}{3 \cdot 10^9}$

## Temă Labs

24/308.

1. Nivel de citire a operanzilor din setul de registre (RD)
2. Nivel de scriere a rezultatelor în setul de registre (WR)

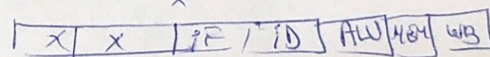
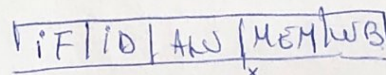
Răspuns: Nivelul de scriere (WR) este mai prioritar datorită necesității evitării hazardurilor RAW între instrucțiuni succesive.

29/309.

- a) 5 niveluri (IF, ID, ALU, MEM, WB)  
1 ciclu de tact / nivel  
instr. salt a cărei adresă se află la finele nivelului ALU.

delay slots  $\Rightarrow$

Instr. set:



DS=2

- b) procesor RISC (pipeline)

de ce se preferă implementarea unor busuri și memorii care separează pe inst. respectiv date?

Răspuns: Pt. a evita coliziunile de memorie în cazul fazelor de IF și MEM.

50/315

- a) Sumatorul este activat de o instrucțiune de branch. Acesta este activat pt. calculul adresei de salt.



b) Răspunsul de la ex. 24/308.

fi scopul este: dacă operația de adresă este după una de scriere, atunci sunt furnizate ca op de citire și are nevoie de un registru în care să se scrie imediat rezultatul.

c) În cazul unei instr. de tip LOAD, unitatea ALU are rolul de a calcula adresa

d)  $ST(R_4)05, R_2$

În latch-ul EX/HEM memorizat val.  $(R_4+05)$ .

$(R_4+05)$  provine din ALU.