

5. Klasifikacija arhitekture

Flynnova klasifikacija

5.1. SISD

5.2. SIMD

5.3. MISD

5.4. MIMD

Klasifikacija arhitekture prema načinu izvršavanja instrukcija

5.5. Upravljački tok

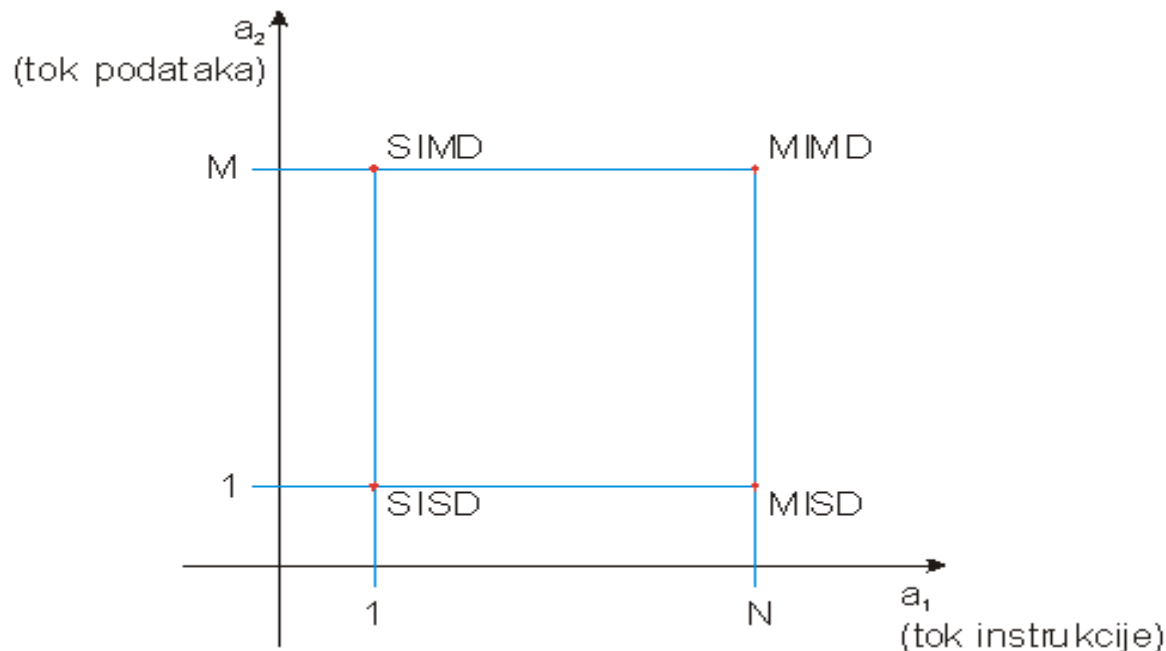
5.6. Tok podataka

5.7. Upravljanje zahtjevom

Različite klasifikacije arhitekture računala:

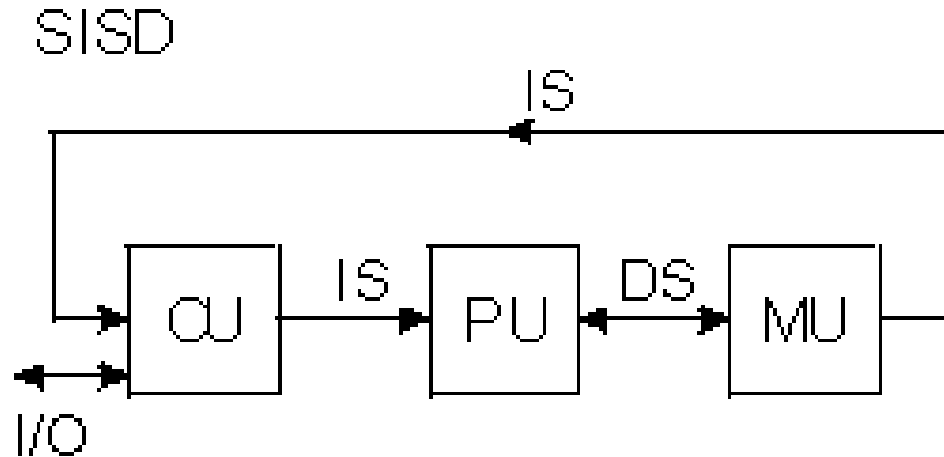
- G. A. Anderson i E. D. Jansen (1975): klasifikaciju arhitekture temelje na putovima za prijenos poruka te prospojnim i procesnim elementima;
- E. I. Organick (1979) klasificira arhitekturu na temelju načina izvršavanja instrukcija;
- S.I. Kartashev i S. P. Kartashev (1978) s obzirom na promjenjivost arhitekture;
- G. J. Myers (1978) kao kriterij predlaže načine premošćivanja jaza između stroja i viših programskih jezika;
- P. C. Treleaven (1984) temelji klasifikaciju na upravljačkim mehanizmima izvođenja instrukcija;

Jedna od opće prihvaćenih klasifikacija arhitekture je ona koju je predložio M. J. Flynn (1972. godine) i temelji se na **instrukcijskom i upravljačkom toku**



- SISD – Single Instruction Stream Single Data Stream [$\mathbf{A}(1,1)$]
- MISD – Multiple Instruction Stream Single Data Stream [$\mathbf{A}(N,1)$]
- SIMD - Single Instruction Stream Multiple Data Stream [$\mathbf{A}(1,M)$]
- MIMD - Multiple Instruction Stream Multiple Data Stream [$\mathbf{A}(N,M)$]

5.1. SISD – jednostruki instrukcijski tok – jednostruki tok podataka

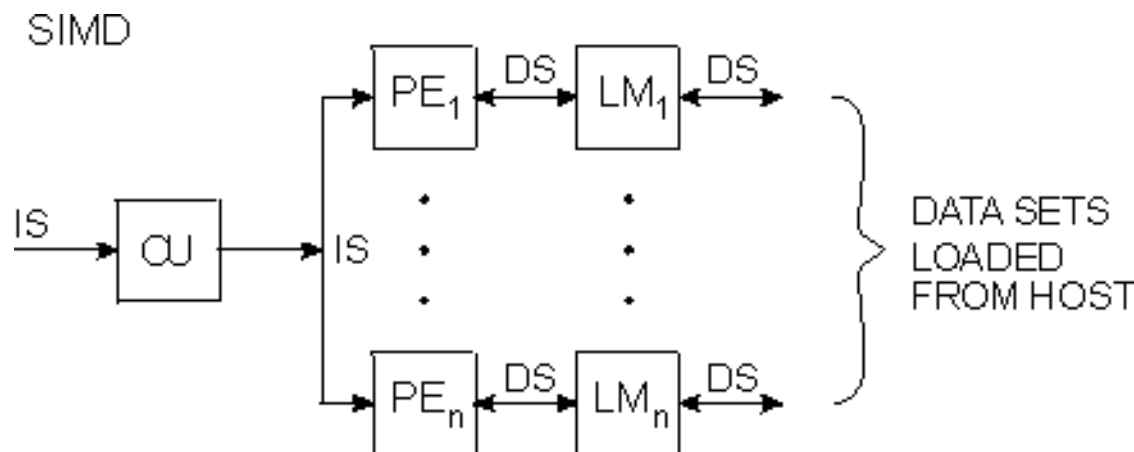


CU – upravljačka jedinica
PU – procesna jedinica
MU – memorijska jedinica

Računalni sustavi:

- CDC 3170
- PDP 11

5.2. SIMD – jednostruki instrukcijski tok – višestruki tok podataka



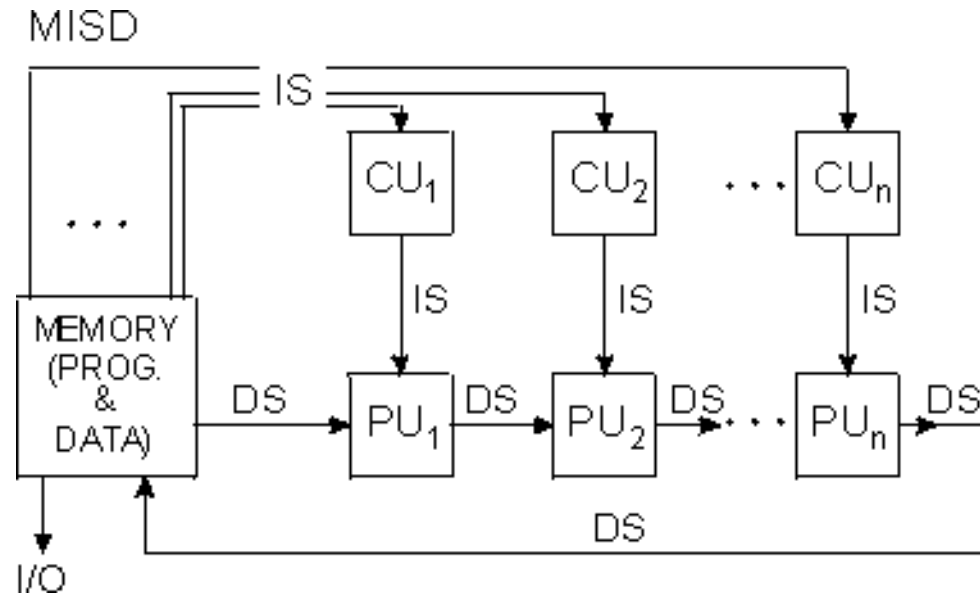
LM – lokalna /privatna memorijska jedinica

PE – procesni element

Računalni sustavi:

- ILLIAC IV
- MPP (Massively Parallel Processor)
- STARAN

5.3. MISD – višestruki instrukcijski tok – jednostruki tok podataka



Računalni sustavi:

- IBM 360/91
- ASC TI
- (Intel 8086, Z 80000)

Opaska: Računala kategorije MISD se strogo teorijski gledano **ne mogu** fizički realizirati

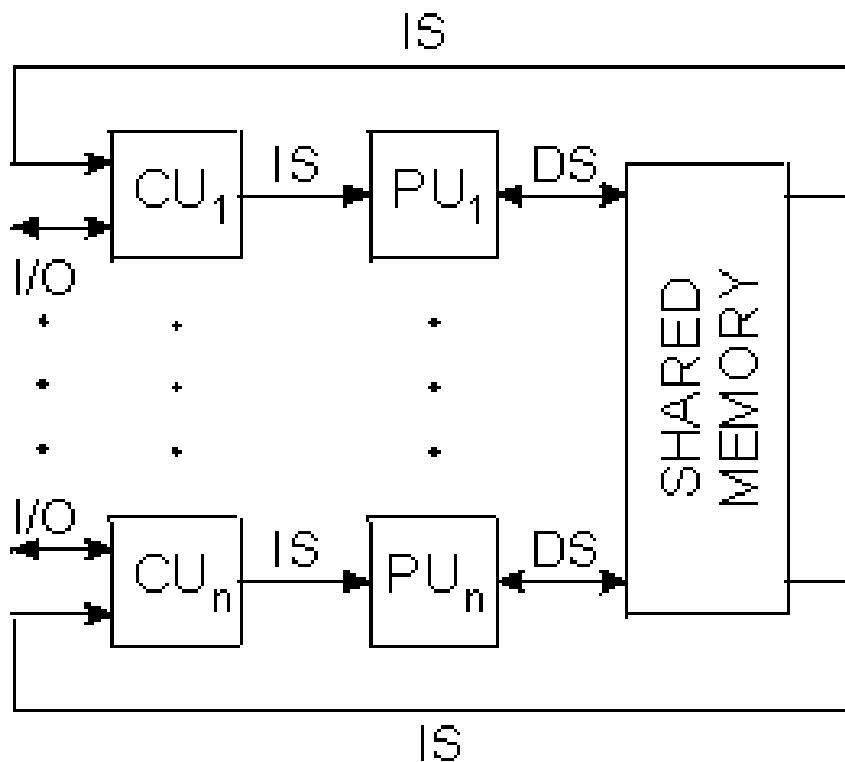
MISD kategorija računala:

- Protočna računala (RISC i CISC)
- Sistolička polja

5.4. MIMD - višestruki instrukcijski tok - višestruki tok podataka

MIMD – višeprosorski (multiprosorski) sustavi

MIMD



Računalni sustavi:

- Univac 1108
- C.mmp
- ZMOB

Zadatak 1.

Pročitati poglavlje 1. Arhitektura računala iz knjige S. Ribarić, Arhitektura računala, ŠK Zagreb 1990./2004. str. 15 – 26.

Klasifikacija arhitekture prema načinu izvršavanja instrukcija (P. C. Treleven):

- arhitektura računala s upravljačkim tokom (engl. control-flow)
- arhitektura računala upravljano tokom podataka (engl. data - flow)
- arhitektura računala upravljano zahtjevom (engl. demand-driven)

Model računala (P. C. Treleven)

Instrukcija u modelu ima oblik:

(arg 0 arg 1 arg 2 ... arg n)

$\text{arg } i \in \{\text{operatori, vrijednosti_operanda, adrese_operanada, adrese_rezultata, adrese_sljedeće_instrukcije}\}$

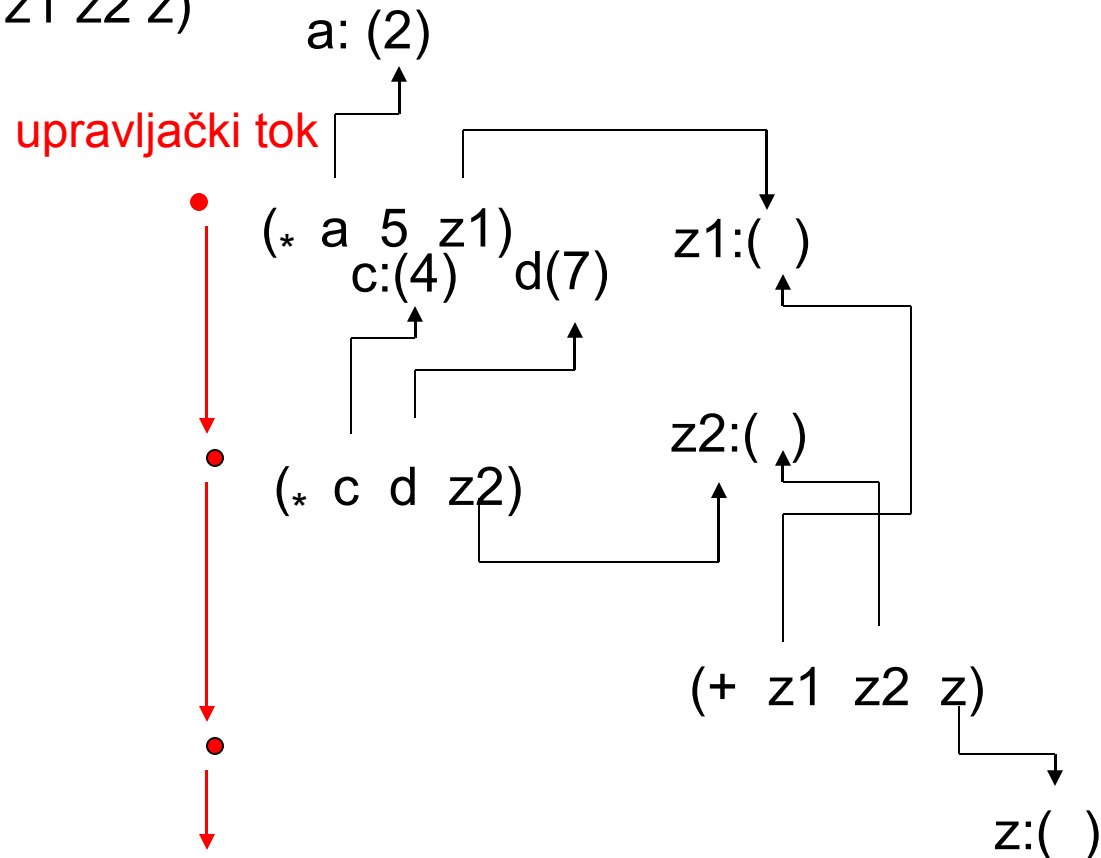
1. Primjer izvršavanja programa u računalu s upravljačkim tokom

b) Von Neumannovo računalo (control-flow)

$$z = (a * 5) + (c * d)$$

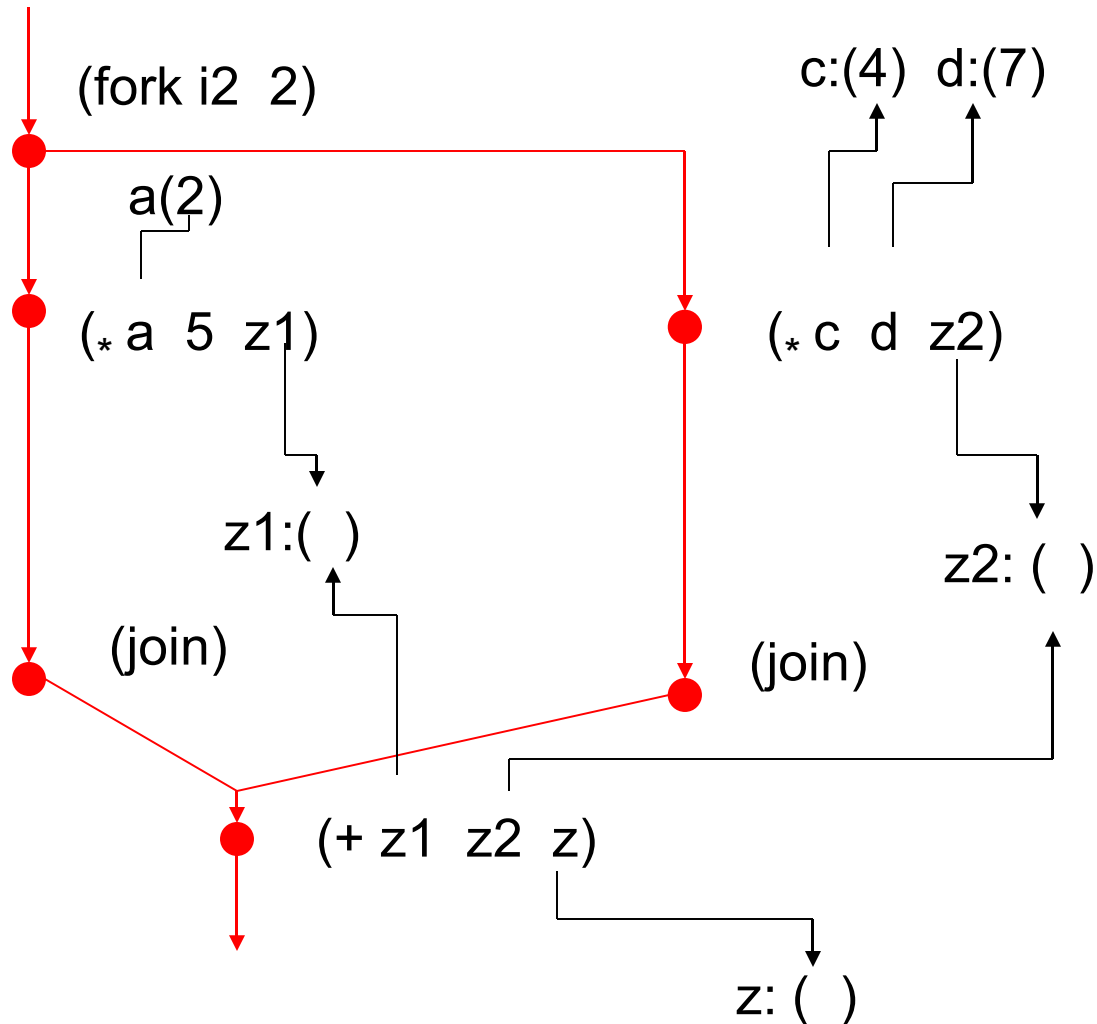


(* a 5 z1) (* c d z2) (+ z1 z2 z)



b) Računalo paralelne arhitekture (control-flow)

$$z = (a * 5) + (c * d)$$

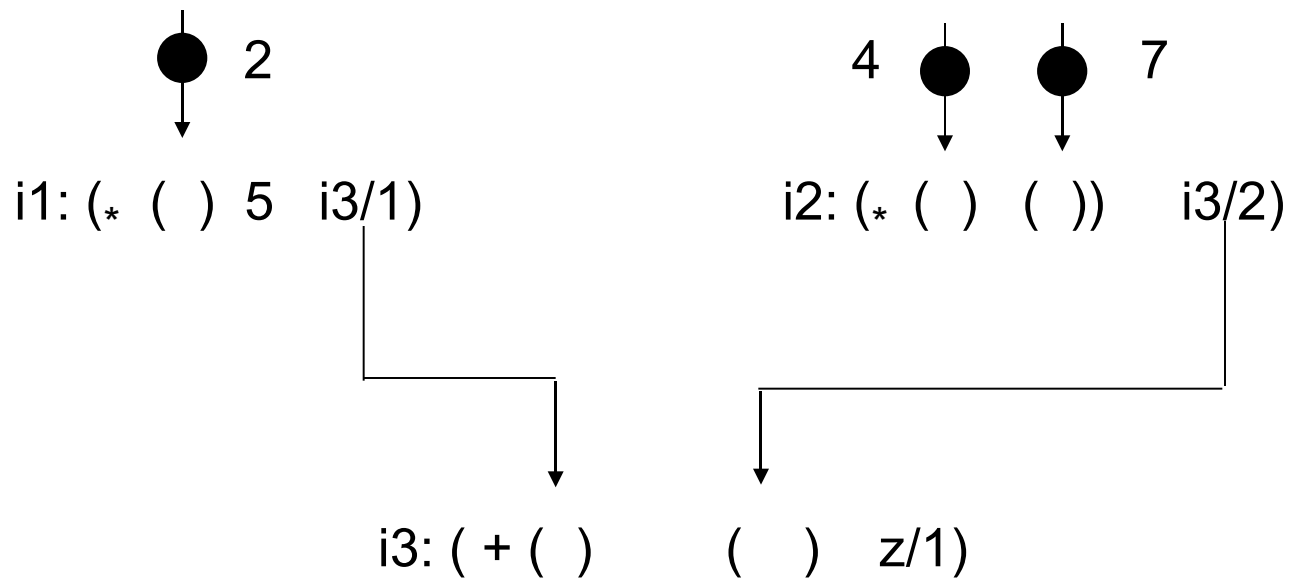


Značajke izvršavanja programa u računalu s upravljačkim tokom

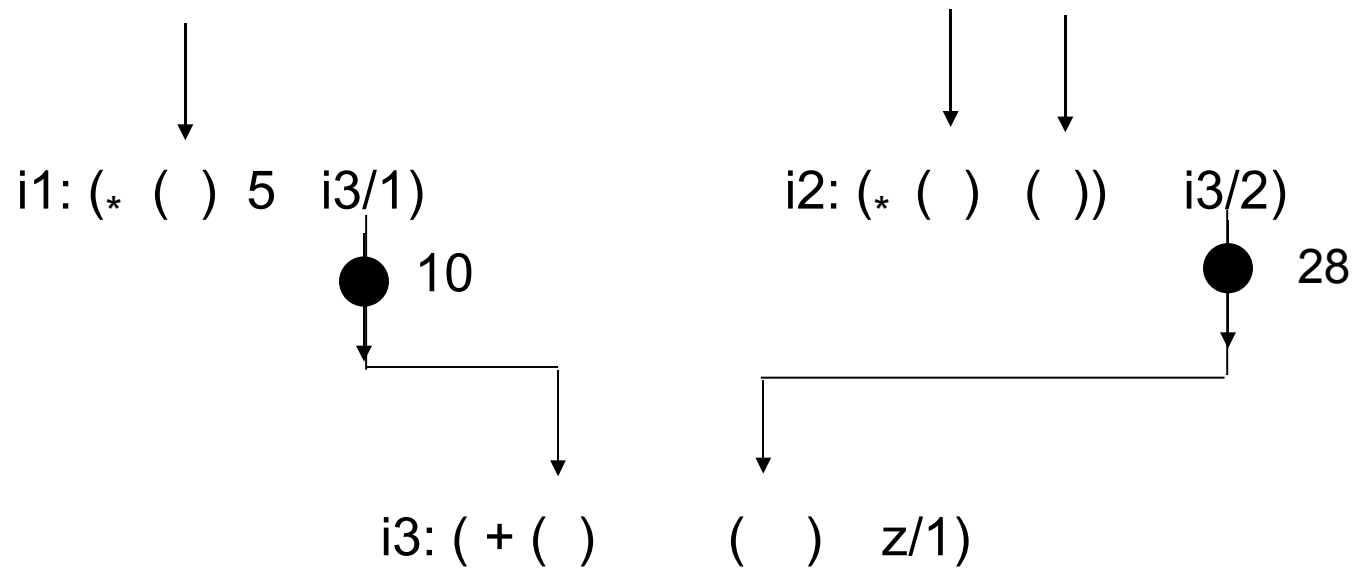
- tok upravljanja je slijedan
- slijed upravljanja implicitno određen (programsko brojilo)
- učestalo referenciranje memorije
- upotrebom posebnih operatora može se upravljati tokom izvođenja programa (jump, goto)
- kod računala s paralelnom arhitekturom tok upravljanja se uvišestručuje uporabom specijalnih operatora (fork, join)
- tok ili tokovi upravljanja i tok (ili tokovi) podataka odvojeni (SISD, SIMD, MISD, MIMD)
- podaci se proslijeđuju između instrukcija POSREDNO putem memorijskih lokacija koje instrukcije međusobno dijele

2. Primjer izvršavanja programa u računalu upravljanom tokom podataka

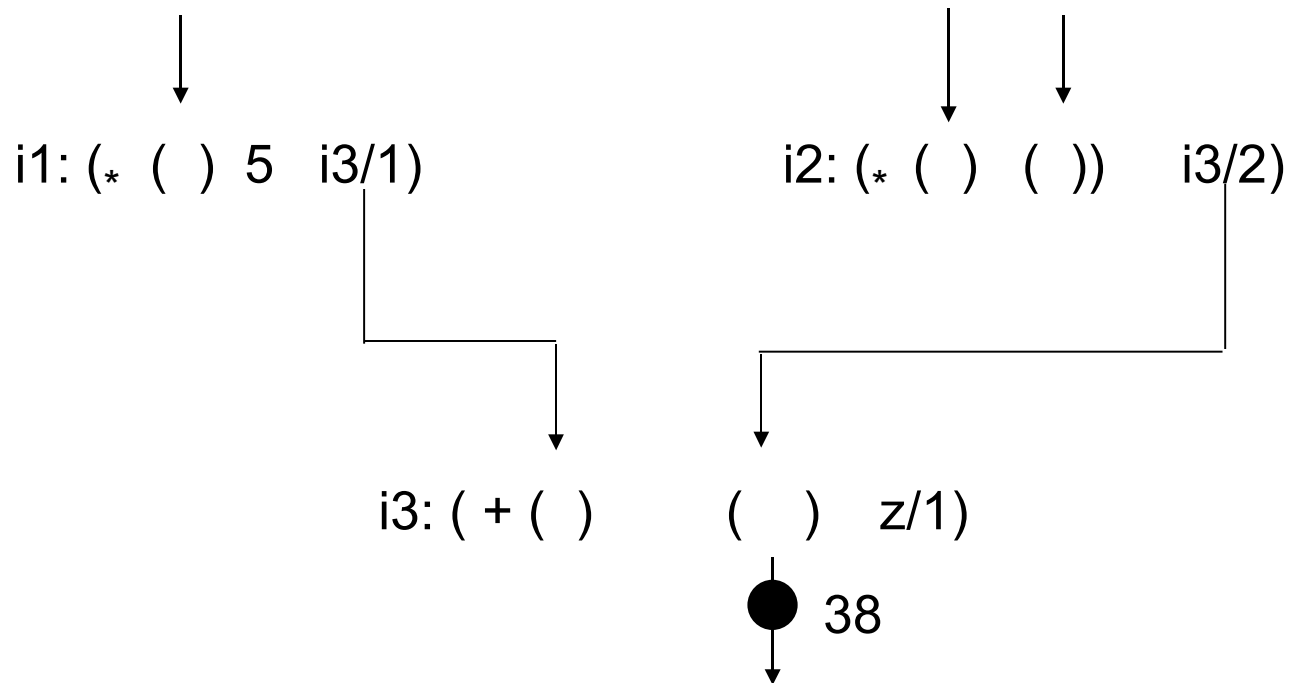
$$z = (a * 5) + (c * d)$$



2.korak



3. korak



Značajke obrade računalom upravljanim tokom podataka:

- upravljački tok je povezan i ujedinjen s tokom podataka
- instrukcije se počinju izvršavati u trenutku kada su im raspoloživi svi potrebni operandi (podaci)
- međurezultati se izravno prosljeđuju između instrukcija
- nakon izvršavanja instrukcije – ulazni podaci nisu više raspoloživi

Računala upravljana tokom podataka – razvoj

D. Adams (1970) – uporaba usmjerenih grafova za prikaz računanja
Model M

R. M. Karpa, R. E. Miller (1973)

J. E. Rodrigues (1980)

Zadatak 2.

Pročitati 6. Računala upravljana tokom podataka i računala upravljana zahtjevom, str. 111 –140. u knjizi S. Ribarić, Arhitektura računala, Školska knjiga, Zagreb, 1990. ili 2004.

Usporedba: control-flow – data-flow

Cijeli postupak računanja:

- Faza izbora
- Faza ispitivanja
- Faza izvršavanja

Data-flow

Faza izbora

Dodjeljivanje specijaliziranih procesora svakoj instrukciji

Faza ispitivanja

Donosi se odluka o instrukcijama koje su izvršljive

Faza izvršavanja

Mijenja lokalna stanja (stanje instrukcije i skup izvršljivih instrukcija)

Instrukcije “troše” svoje argumente

Control-flow

Faza Pribavi

Ø

Faza Izvrši

Mijenja stanje stroja (sadržaj memorije, registara)

3. Izvršavanje programa u računalu upravljanom zahtjevom

$$z = (a * 5) + (c * d)$$

- zahtjev za rezultatom pobuđuje izvršavanje operacije koja će ga generirati
- vrijednost za z se dobiva redukcijom:

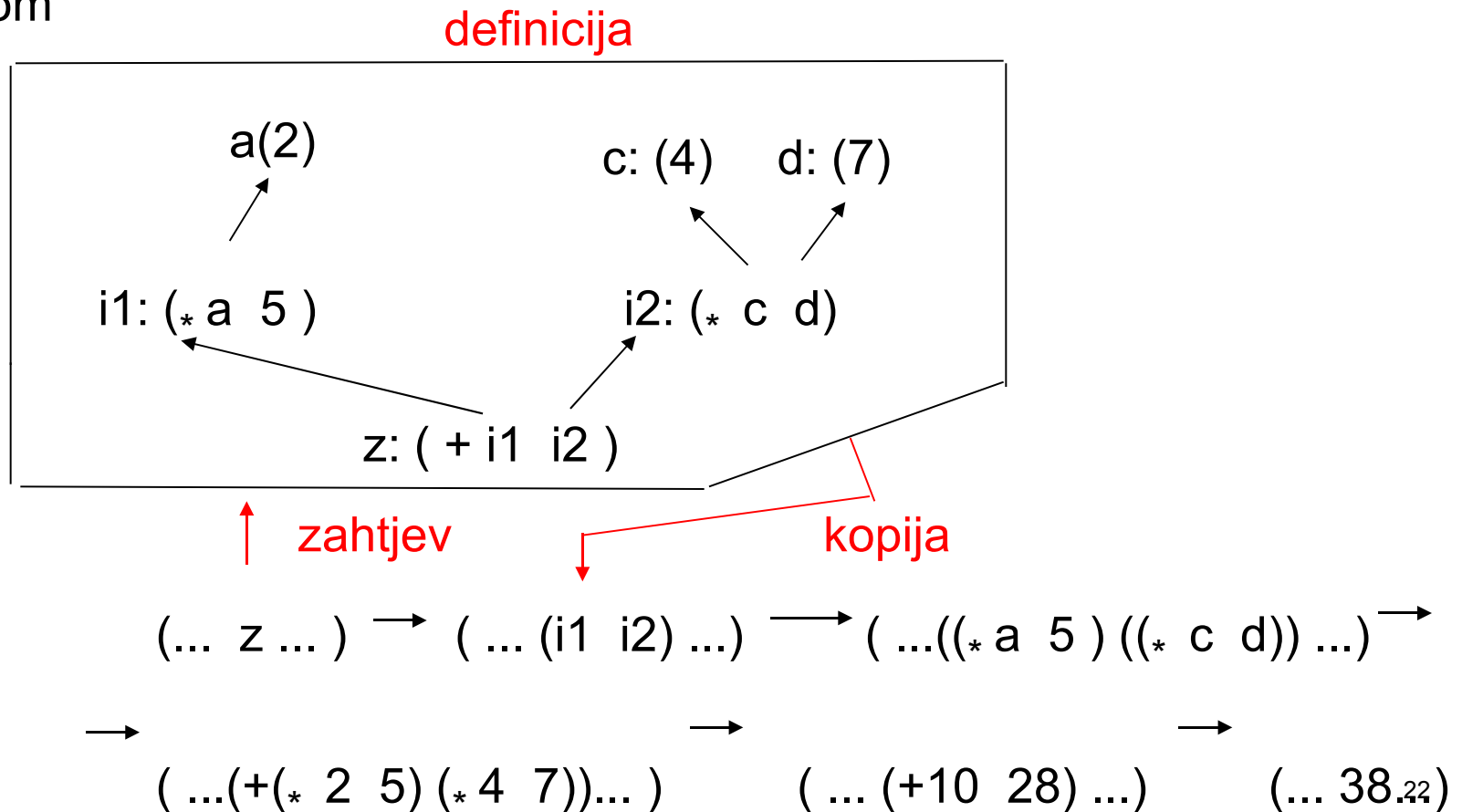
$$(2 * 5) + (4 * 7) \longrightarrow 10 + 28 \longrightarrow 38$$

Prema načinu rukovanja argumentima u instrukcijama postoje dva oblika redukcije:

- redukcija niza
- redukcija grafa

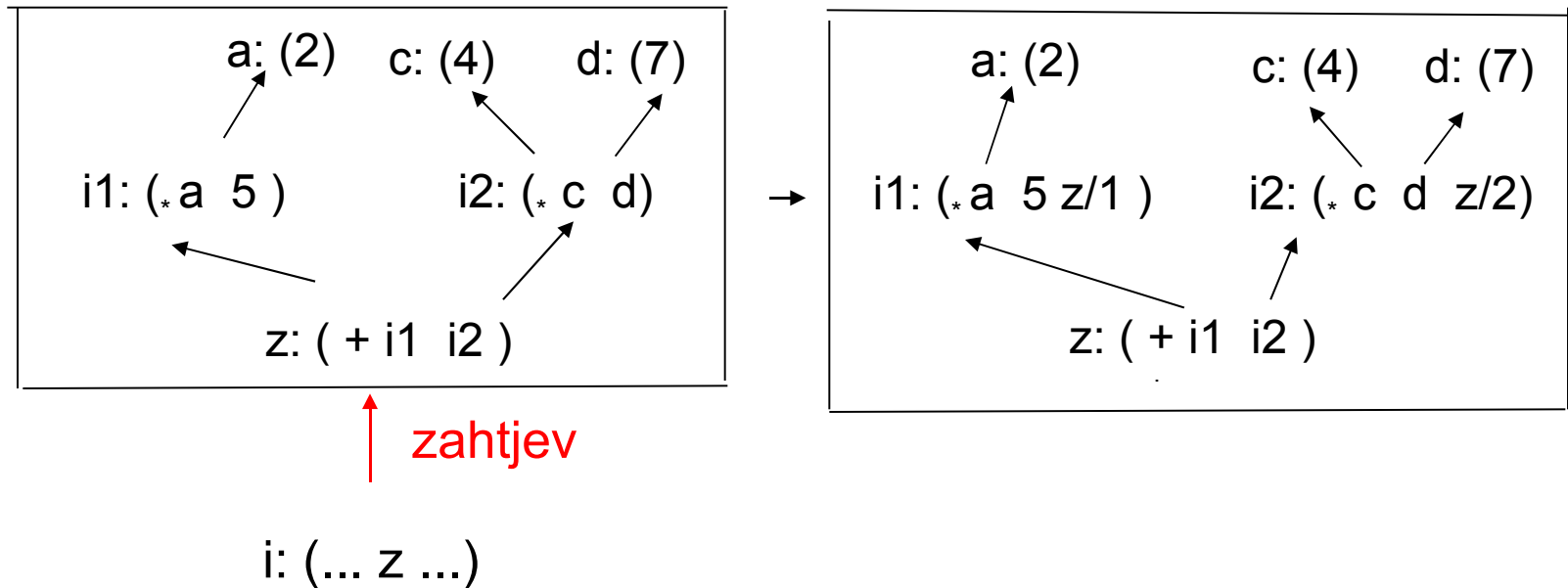
3.1. Redukcija niza

Osnovna značajka: svaka instrukcija pribavlja definiciju i rukuje njenom kopijom

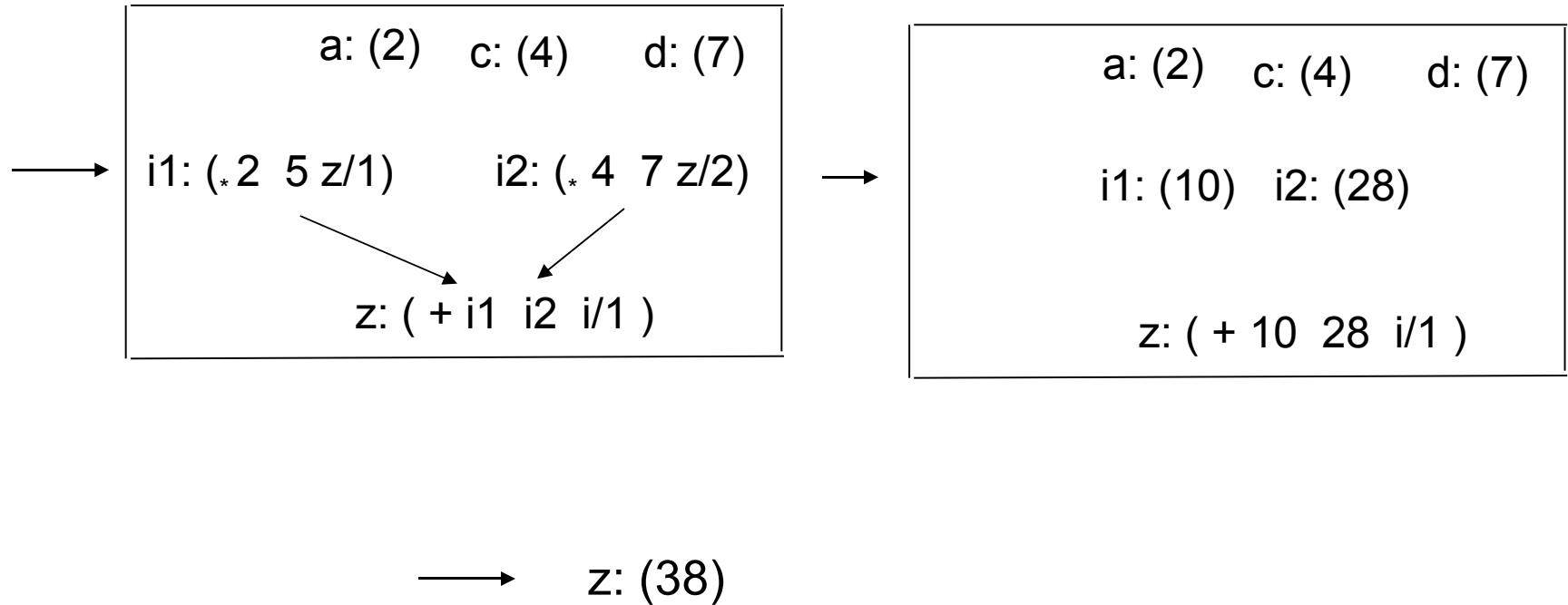


3.2. Redukcija grafa

Instrukcija rukuje definicijom umjesto kopijom definicije, ali tako da u definiciji upotrebljava argumente – kazaljke.



3.2. Redukcija grafa (nastavak)



Osnovne značajke izvršavanja programa u računalu upravljanom zahtjevom:

- nema dodatnih ograničenja u pogledu upravljanja slijedom izvršavanja instrukcija osim onih koja su postavljena zahtjevima za rezultatom
- ne upotrebljava se koncepcija pohranjivanja i obnavljanja vrijednosti (nema varijabli)
- dopušteno je gniježđenje zahtjeva za rezultatima

Primjeri računala

c) Data-flow:

- MIT data-flow
- Texas Instrument data-flow
- NTT Eddy
- Manchester data-flow
- NEC TIP
- Irvine/MIT

k) Demand-driven

- GMD
- North Carolina Cellulat Tree
- Cambridge SKIM
- Newcastle Reduction Machine
- Kent S-K