5b. Klasifikacije arhitektura

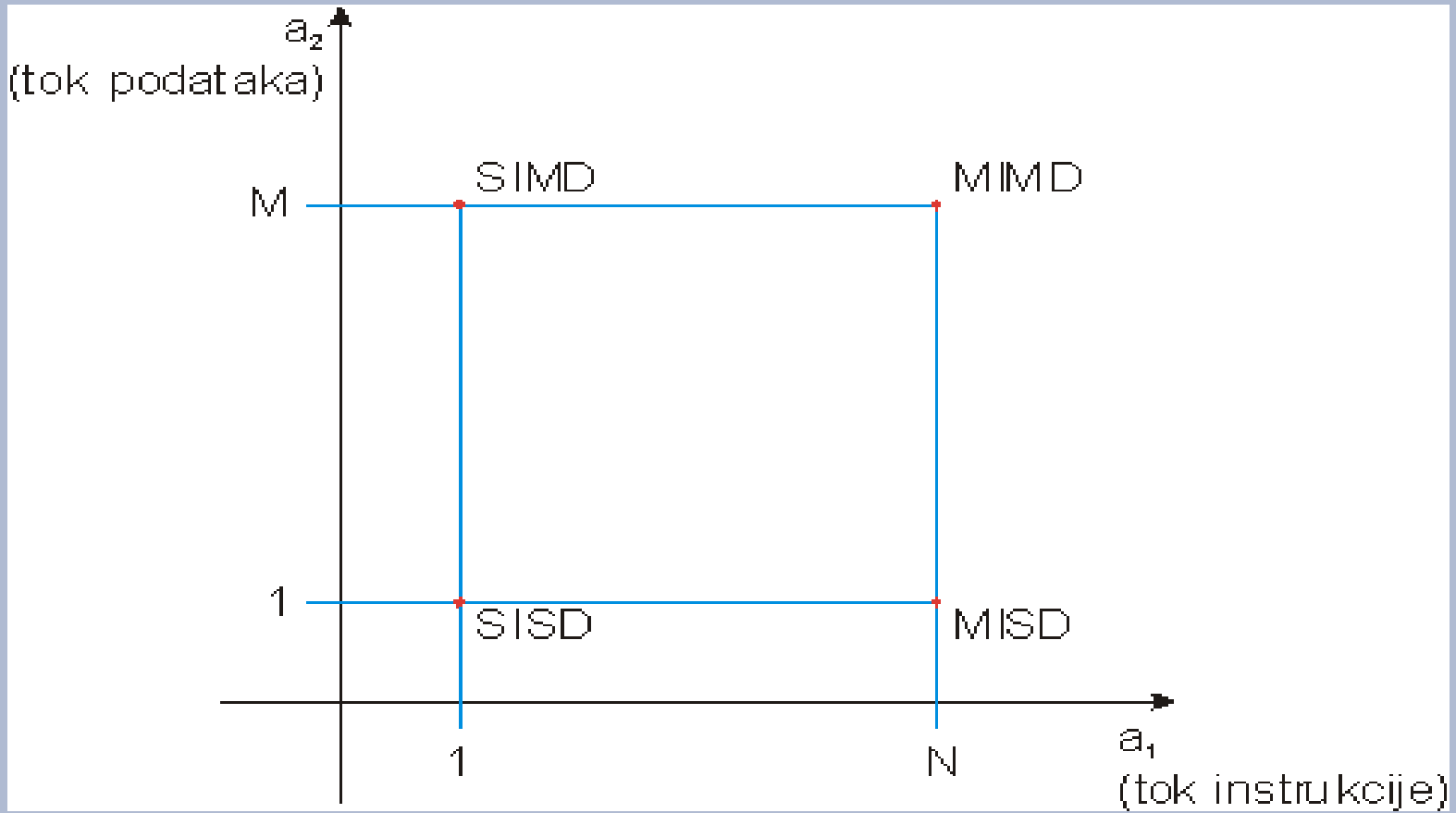
Flynnova klasifikacija prema **paralelizmu** (1966):

- razmatra se brojnost usporednih **instrukcijskih** (upravljačkih) i **podatkovnih** tokova

Četiri temeljne kategorije:

- **SISD**: jednostruki tokovi upravljanja i podataka (von Neumannova arhitektura)
- **SIMD**: jedan instrukcijski tok djeluje nad više tokova podataka (vektorska računala, MM ekstenzije)
- **MISD**: više instrukcija djeluje nad istim podatcima (redundantna računala tolerantna na pogreške)
- **MIMD**: višestruki instrukcijski tokovi djeluju nad nezavisnim podatcima (višeprocesorski sustavi)

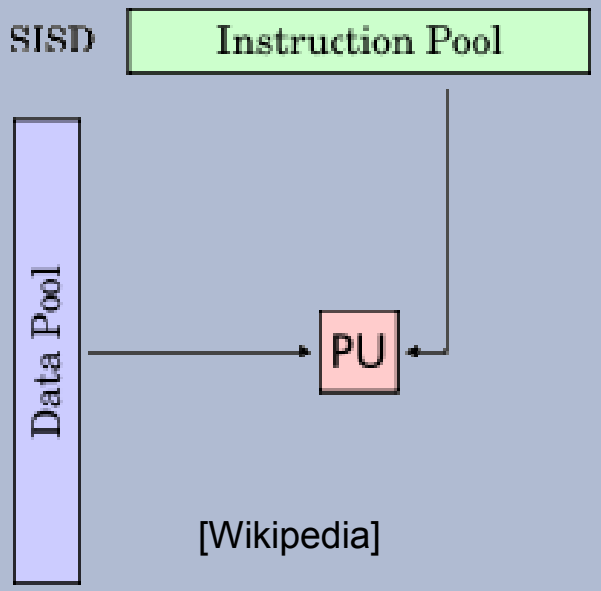
Flynnov dvodimenzionalni oblikovni prostor



[Ribarić]

SISD, A(1,1): kategorija na zalasku

- slijedna, von Neumannova arhitektura
- omogućava efikasno izvođenje **uobičajenih** algoritama (programeri razmišljaju **slijedno**)



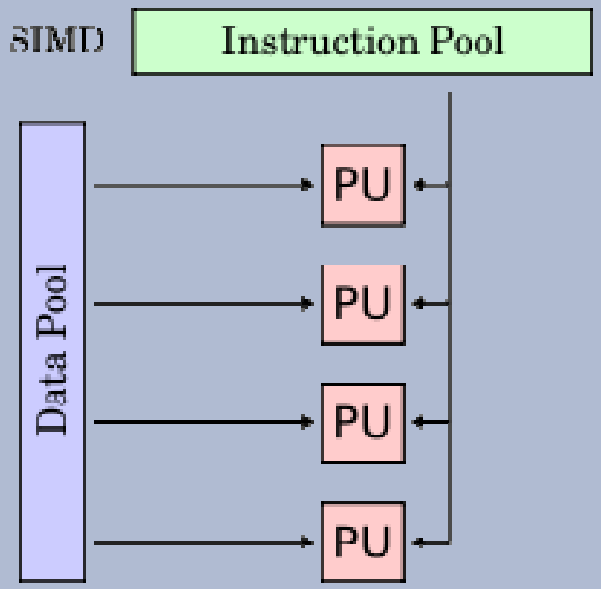
[Wikipedia]

Moderna SISD računala agresivno iskorištavaju paralelizam na razini instrukcije (engl. instruction level parallelism):

- protočnost
- superskalarnost (MIMD?)
- izvođenje izvan redoslijeda (dataflow?)
- spekulativno izvođenje
  - predviđanje grananja
  - predviđanje memorijskih međuovisnosti (RAW hazard)
  - (primjena načela favoriziranja čestog slučaja)

SIMD, A(1,m): vektorska računala

- koristi se paralelnost na razini podataka
- istovrsne operacije nad skupom podataka
- matrične operacije, obrada slike, grafika, simulacije, numerička analiza (PDE, integracija ...)
- nedostatak: ne mogu se vektorizirati svi algoritmi!



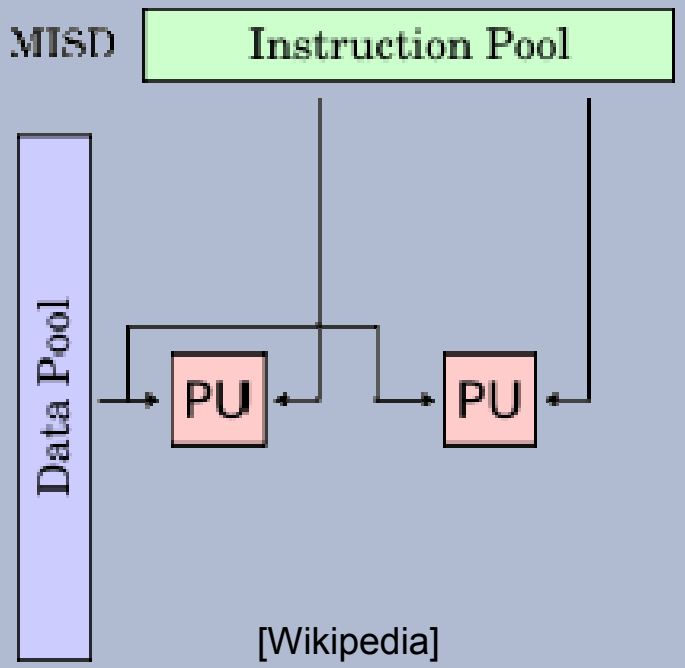
[Wikipedia]

Predstavnici računala sa SIMD arhitekturom:

- vektorska računala: ILLIAC IV, Cray, Cell SPE
- (GP)GPU: ATI (AMD), NVidia, Intel Larrabee
- MM ekstenzije
  - Intel: MMX (1997), SSE (1999), AVX (2010?)
  - Power PC: Altivec (1998)
  - AMD 3dNow (1998),SSE5 (2011)

MISD, A(n,1): manje korisna kategorija

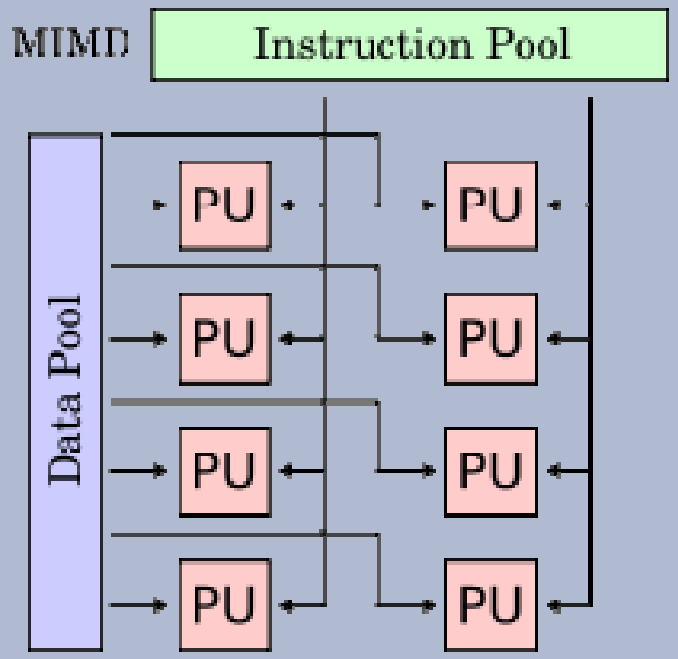
- tu se mogu svrstati redundantna računala (npr, računalo za upravljanje svemirskim brodom)
- također, po nekima, protočna računala i posebne paralelne arhitekture (sistolička polja)



[Wikipedia]

MIMD, A(n,m): potpuno paralelna računala

- distribuirani sustavi s nezavisnim procesorima
- usporedni tokovi izvođenja (dretve, procesi)
- podrška u jeziku (Erlang, Occam, Java), proširenju jezika (OpenMP: Fortran, C++) ili na razini biblioteke (<pthread.h>)
- šire područje primjene od SIMD, ali teško programiranje



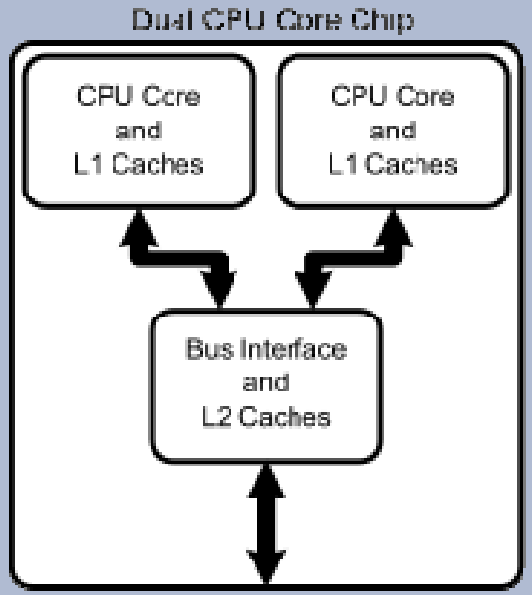
[Wikipedia]

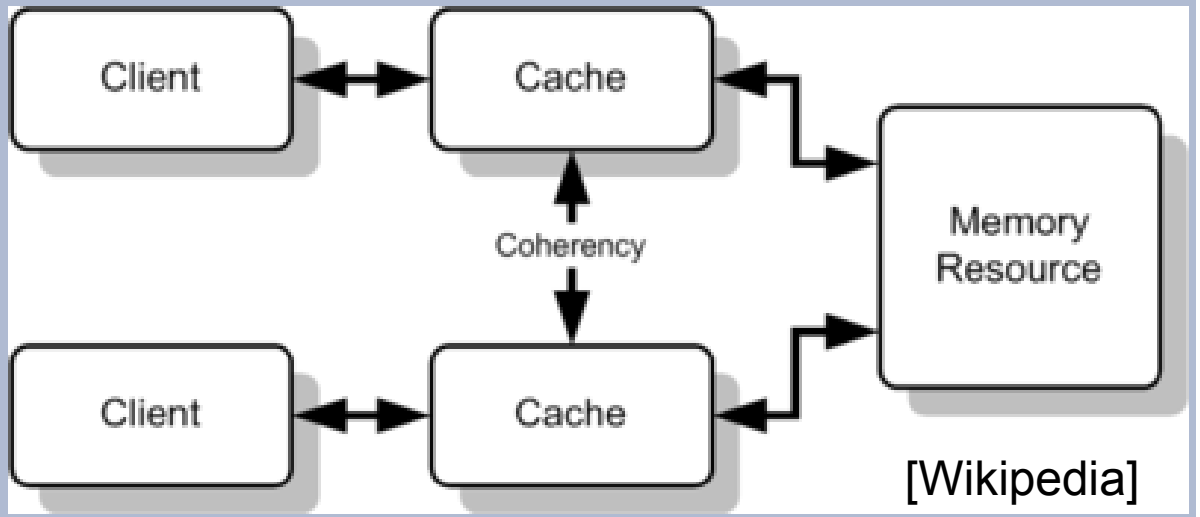
Predstavnici MIMD arhitektura:

- arhitekture s dijeljenom memorijom (SMP ili NuMA)
  - zajednička sabirnica (ili više jezgara)
  - efikasno dijeljenje podataka, teža nadogradnja
  - SUN, SGI, AMD, Intel, Cell
- arhitekture na lokalnoj mreži (grozd, cluster)
  - laka izmjena i dodavanje (jeftinih) elemenata
  - veća potrošnja energije
  - Google, BeoWulf, Solaris, Seti

Simetrična višeprocesorska računala (MIMD) (2000 →)

- pogodna za izvođenje nezavisnih procesa i paralelnih programa
- glavni problem: ograničen doprinos paralelizma (Amdahlov zakon)
- pristup s **dijeljenom sabirnicom** štedljiviji, brža komunikacija
  - procesori na jednom ili više integriranih sklopova
  - dijeljena priručna memorija: i prednost i komplikacija!
    - rješenja: npr, prisluškivanje sabirnice (snooping)
  - predstavnici: Intel Xeon, Core i7; AMD Opteron, Phenom





[Wikipedia]

Višejezgrena budućnost?

Usporedimo 10-godišnji napredak u performansi procesora:

- Pentium 1 (P54C, 1995) 100MHz, 5e6 tr. (600 nm): 100 VAX
- Itanium2 (Montecito, 2006) 1.6GHz, 1e9 tr. (90 nm): 6000 VAX
- Xeon (Wolfdale, 2007) 3 GHz, 5e8 tr. (45 nm): 9000 VAX

Ubrzanje ×90 uz ×100 više tranzistora i ×30 brži takt
→ performansa slijednih (SISD) računala ne prati Mooreov zakon
→ koliku bismo propusnost dobili kad bismo u jedan sklop stavili 100 procesora P1 na 1GHz?
→ zato se javlja višejezgreni trend!

- Intel: Xeon (6), AMD: Bulldozer (8), Power PC (3)
- SUN: Ultrasparc T2 (8)
- ARM11 (4), MIPS (16 - 1024)
- NVidia: Quadro, Tesla (128+)
- STI: Cell (8+1)

Klasifikacija s obzirom na redosljed izvršavanja instrukcija:

- računala upravljana **instrukcijskim** tokom
  - von Neumannova arhitektura (*instruction flow*)
  - redosljed izvršavanja određen redosljedom instrukcija

- računala upravljana **tokom podataka** (cca 1970)
  - instrukcija se izvršava kad svi ulazni parametri postanu raspoloživi (*dataflow*, nema programskog brojila!)
  - izvedba koristi asocijativne memorije za međurezultate i instrukcije (usko grlo)
  - bez uspješne implementacije opće namjene
  - koncept se primjenjuje kod **specijalnih** arhitektura (DSP, GPU, ...)

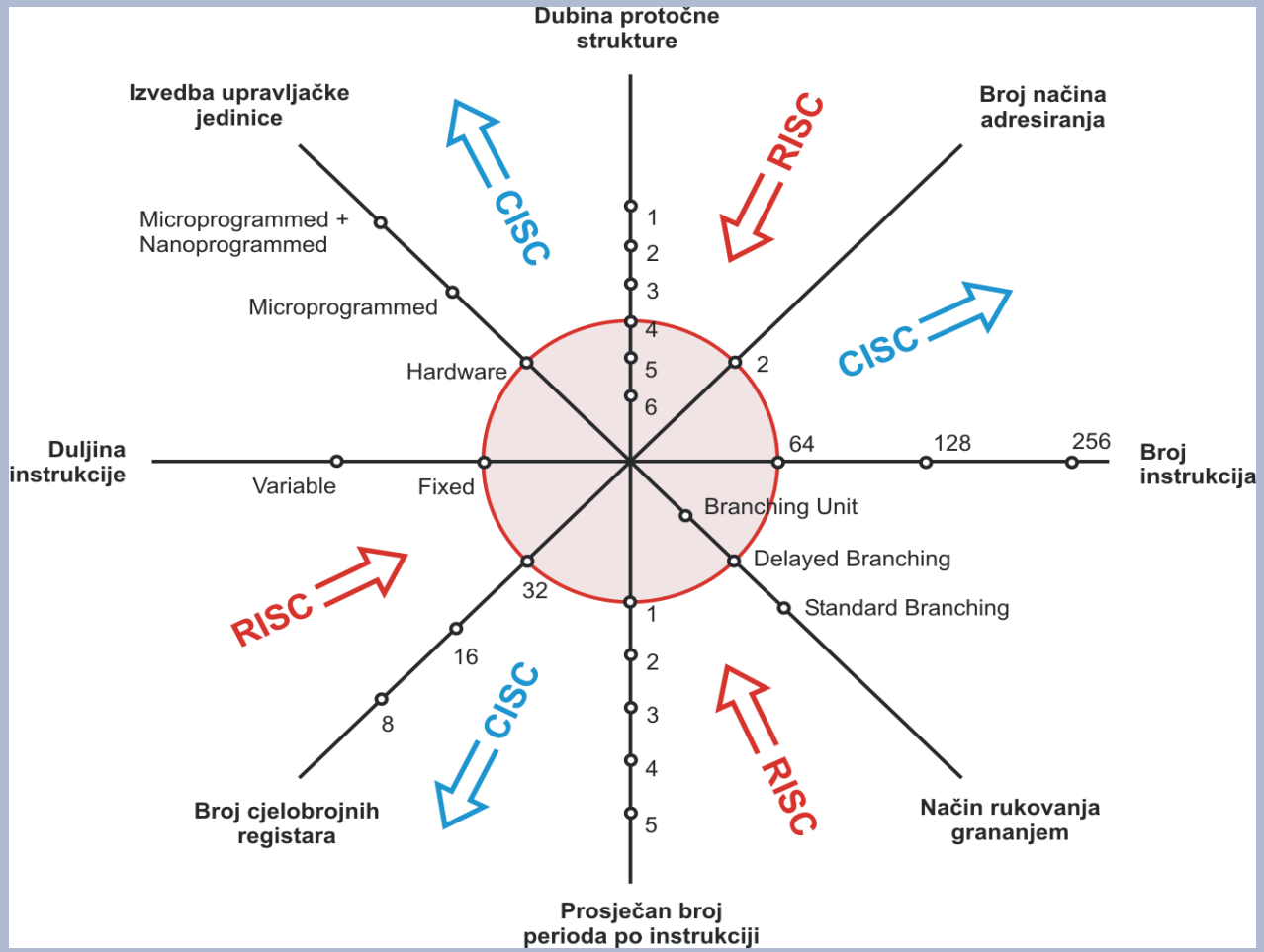
Ograničeno upravljanje tokom podataka

- izvođenje instrukcija izvan programskog redosljeda (*OoO - out of order execution, dynamic scheduling)*
- instrukcije unutar prozora izvođenja se dinamički raspoređuju po procesnim jedinicama kad ulazni parametri postanu raspoloživi
- prozor izvođenja (*execution window*): 30 – 200 instrukcija (ograničeno upravljanje tokom podataka, *restricted dataflow*)
- relativno malo procesnih jedinica (2-10)
- prestavnici: superskalarna računala s **dinamičkim raspoređivanjem**:
  - CDC 6600 (1964),
  - MIPS R10000 (1995)
  - Intel P6 (1995), AMD K5 (1996), Cyrix 6x86 (1996),
  - Alpha 21264 (1998)
  - PowerPC, …

Klasifikacija instrukcijskih arhitektura

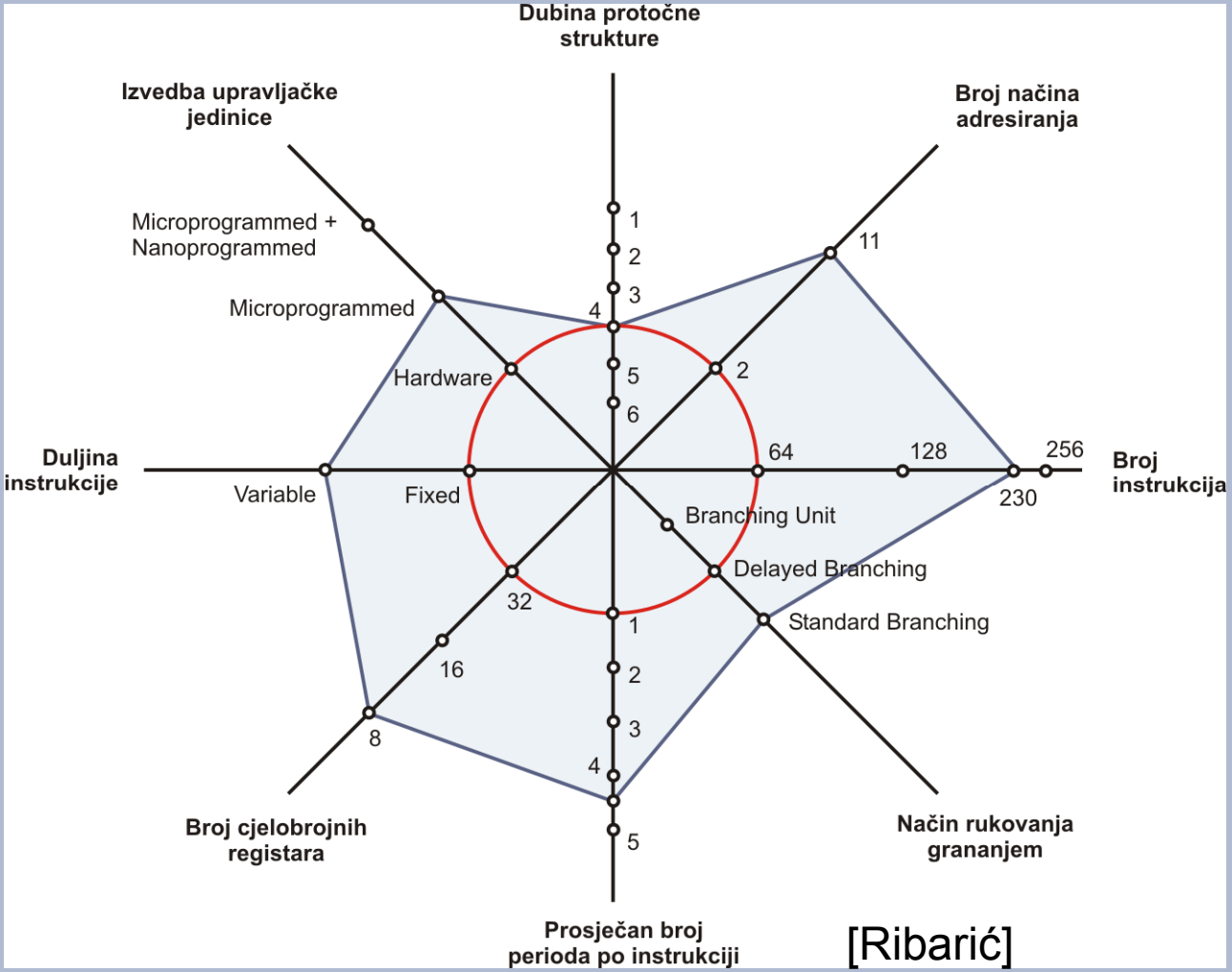
- **Arhitekture CISC**, 1970 →
  - complex instruction set architecture (x86, MC680x0, Z80)
  - 1950-1980: težnja smanjenja “semantičkog jaza” između strojnog i programskog jezika
  - brojni i raskošni načini adresiranja (od 12 do 24 i više);
  - veliki skup strojnih instrukcija s promjenjivim formatom
  - mali registarski skupovi i mikroprogramsko upravljanje
  - skalarna organizacija: jedna procesna jedinica (tipično, zbrajanje i posmak)
  - CPI od dva do nekoliko stotina

Zvjezdasti (kiviat) grafovi:



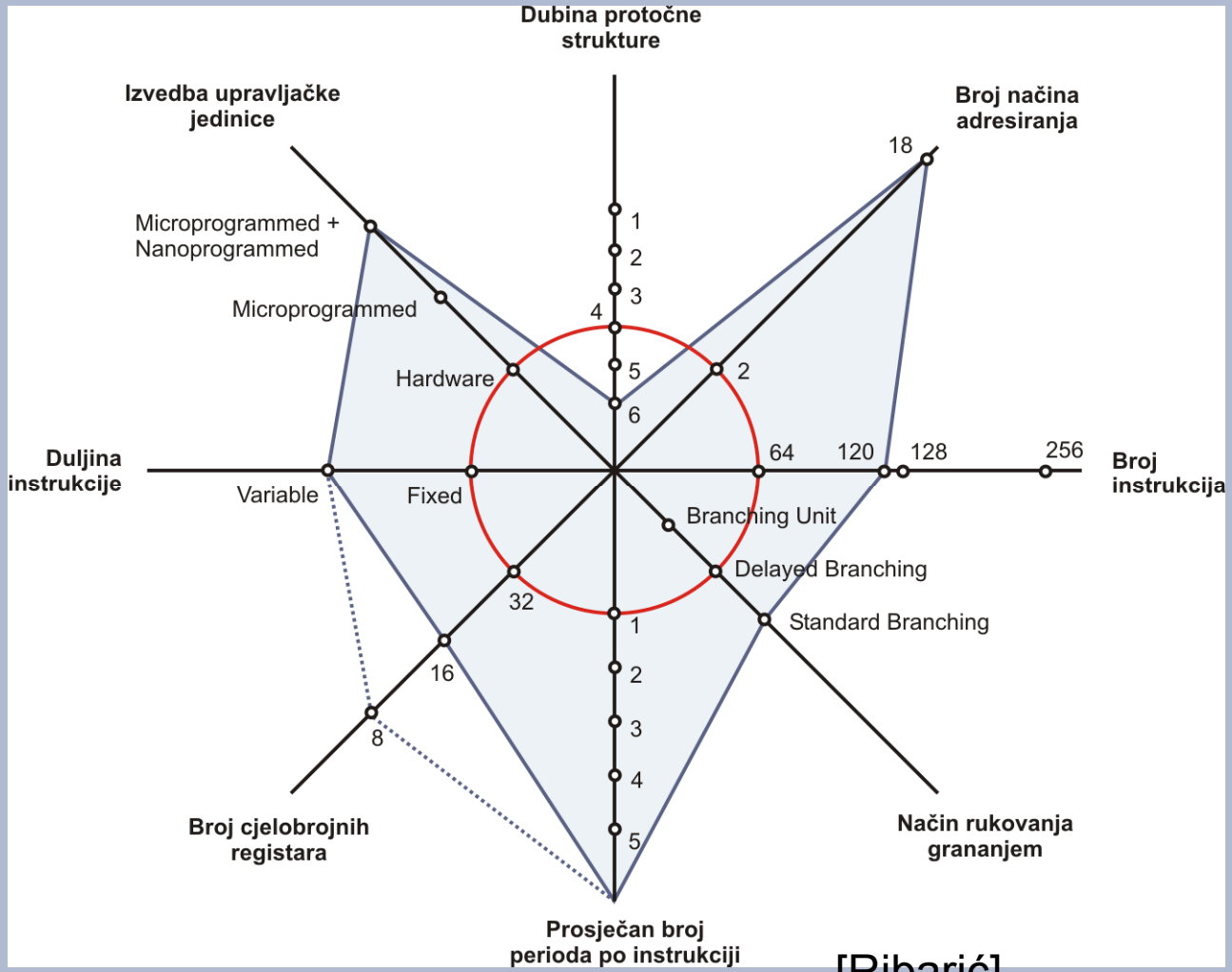
zvjezdasti graf (kiviat graf)

i486



[Ribarić]

MC 68040



[Ribarić]

- Glavna ideja pristupa CISC - povećati performansu smanjenjem i_c:
  - pristup nije preživio test vremena
    - prevoditelji ne uspijevaju iskoristiti složene instrukcije
    - protočna struktura slabo popunjena uslijed različitih instrukcija
    - arhitektura x86 je iznimka koja potvrđuje pravilo
  - bolji rezultati dobiveni jednostavnijim i pravilnijim instrukcijskim skupovima koji mogu raditi na većim frekvencijama – RISC
    - *favoriziraj česti slučaj!*
  - moderni CISC procesori (x86, AMD, Intel) imaju RISC jezgru, te dinamički prevode x86 u interni instrukcijski skup tipa RISC!

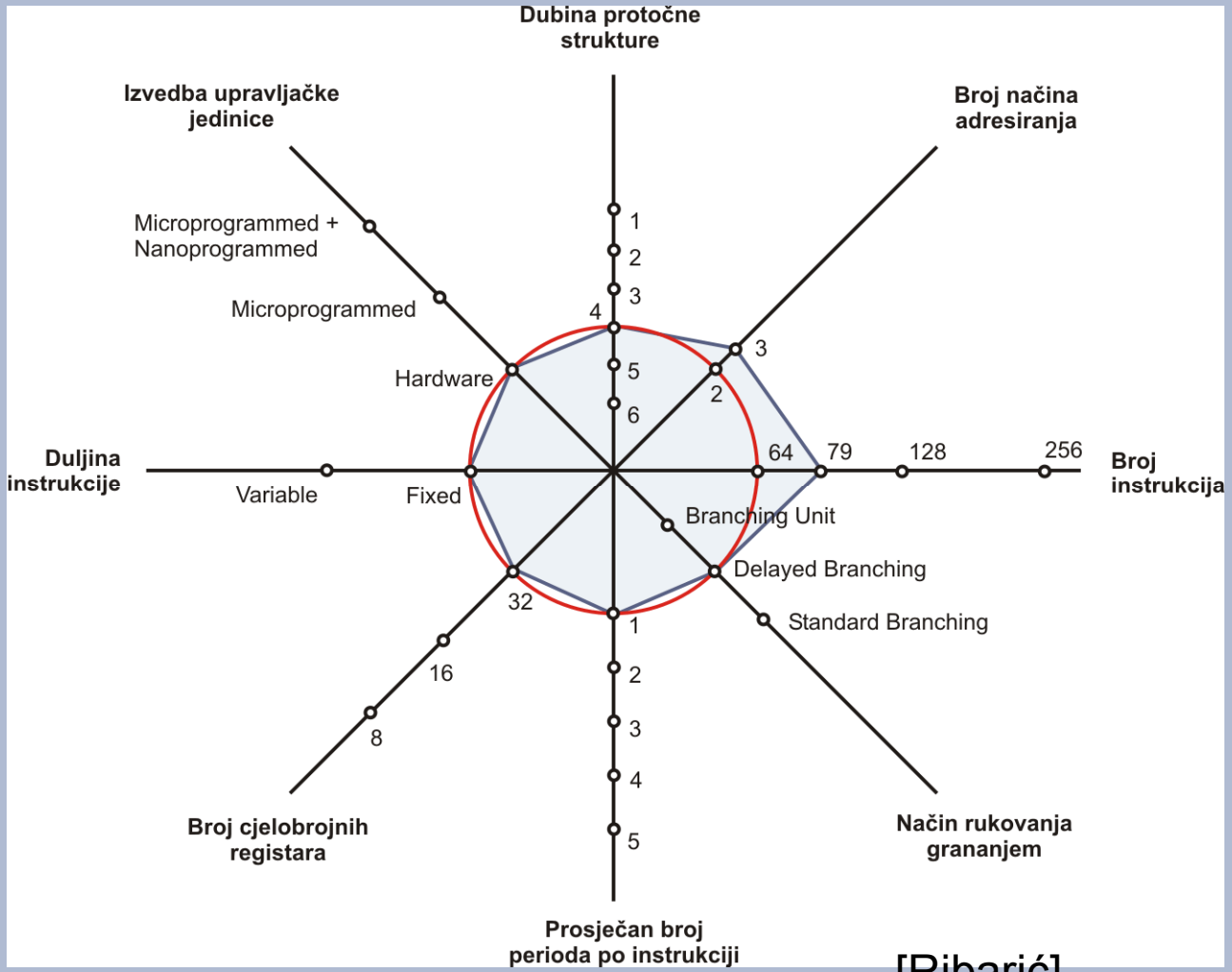
Klasifikacija instrukcijskih arhitektura (2)

- **Skalarne RISC arhitekture** 1980 →
  - RISC: reduced instruction set computer
  - MIPS, Sun Sparc, DEC Alpha, Arm, Power PC, ...
  - istraživanja pokazala: složene instrukcije se koriste **rijetko**, mnogo vremena se troši na spremanje **međuvrijednosti**
  - **ideja:** smanjiti složenost u korist brzine jednostavnih instrukcija (`add r1, r2,r3` ili `load r1, r2+offset`)
  - jednostavne instrukcije pogodne za protočnu izvedbu!

Značajke arhitekture RISC

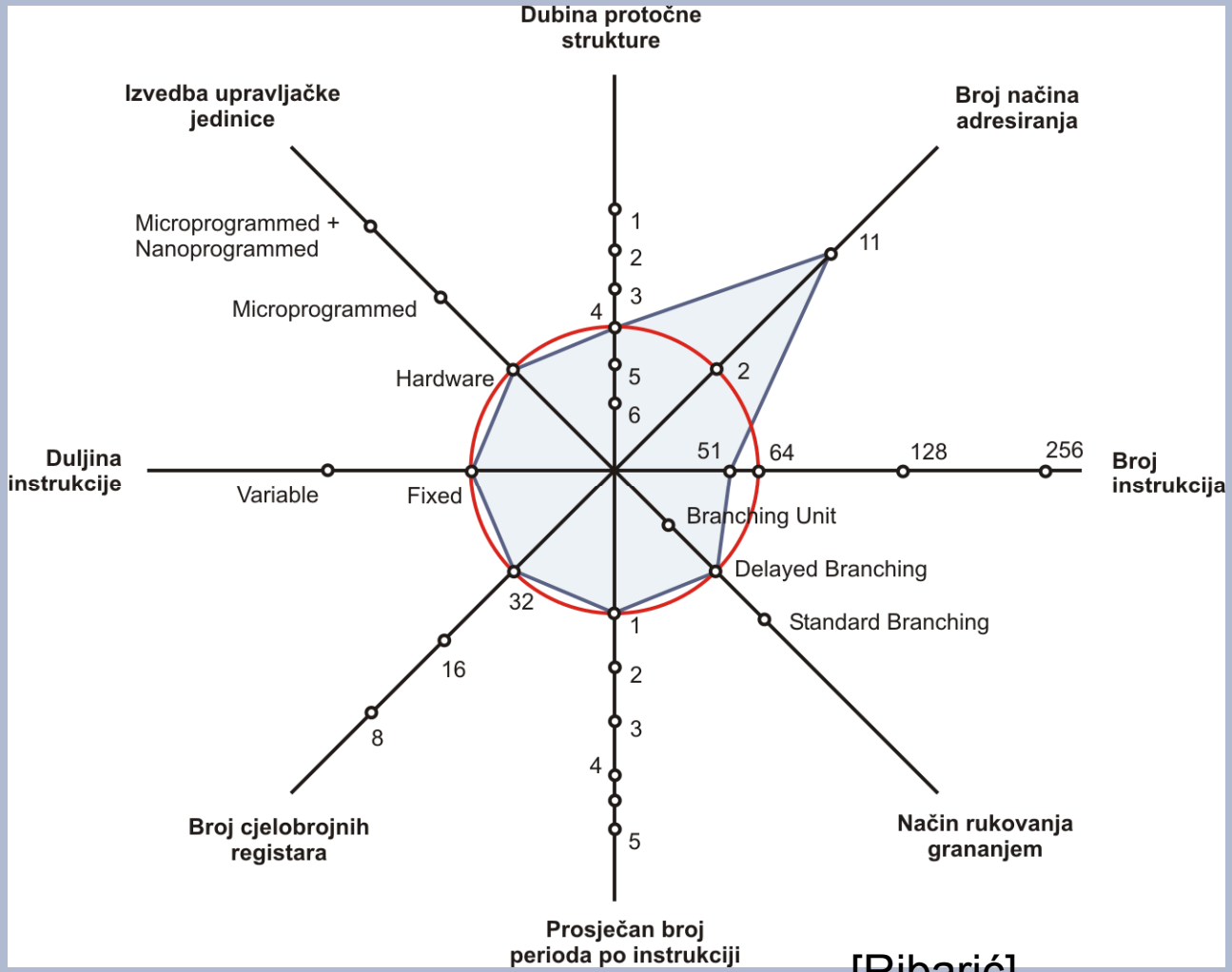
- **jednostavni** načini adresiranja
- **ortogonalni** instrukcijski format (čvrsta polja za operande)
- **spartanski** instrukcijski skup (duljina instrukcije 32 bita)
- mnogo **registara** (32+)
- sklopovsko upravljanje
- jedine memorijske instrukcije su **load** i **store**
- CPI ≈ 1 – jedan period dovoljan za većinu instrukcija;
- radna frekvencija tipično veća od odgovarajućeg CISC-a

i860



[Ribarić]

MC 88100

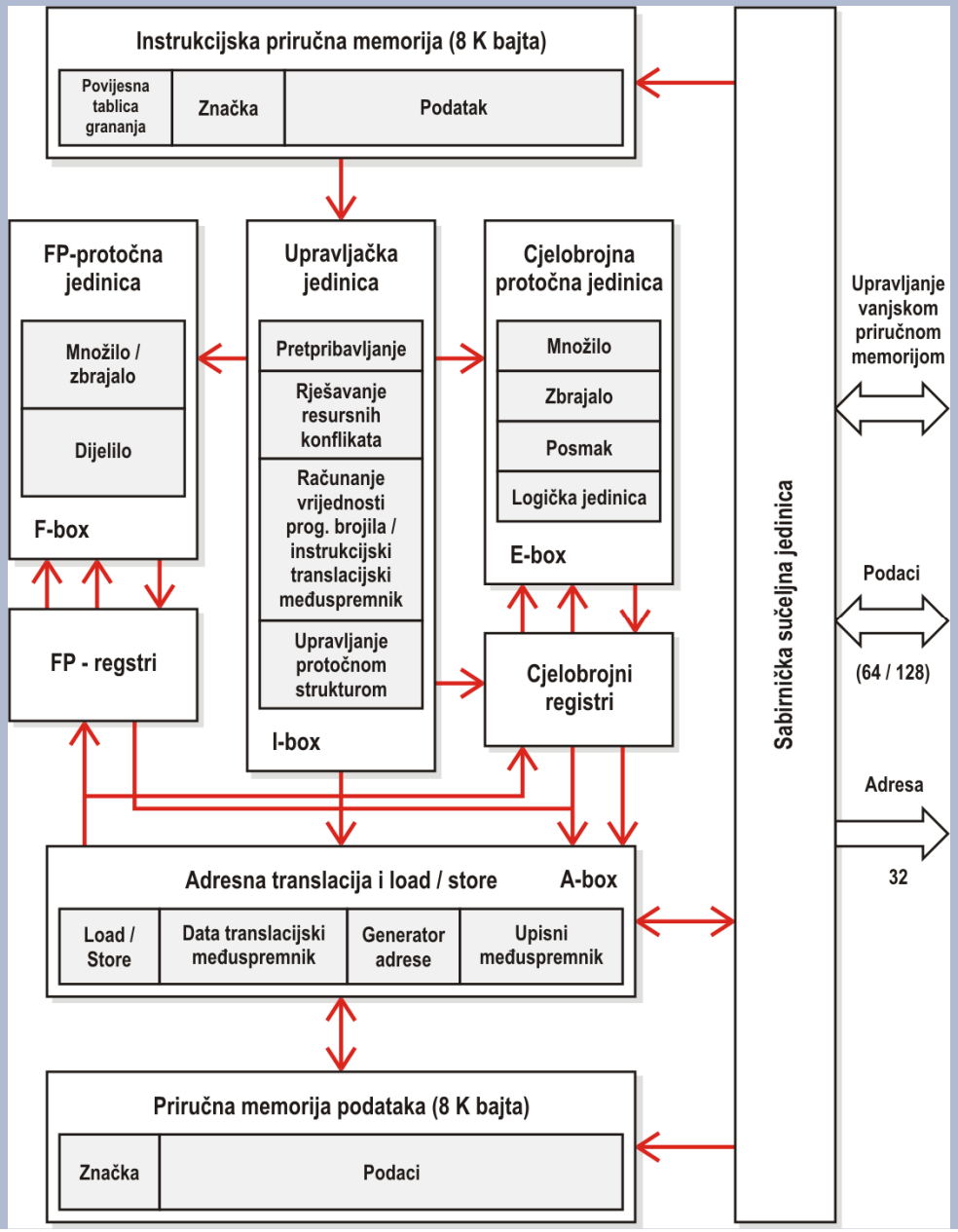


[Ribarić]

Klasifikacija instrukcijskih arhitektura (3)

- **Superskalarne RISC arhitekture** 1980 →
  - MIPS, Sun Sparc, DEC Alpha, Power PC
  - tijekom jednog perioda takta mogu izvršavati više instrukcija!
    - **agresivno** iskorištavanje instrukcijskog paralelizma
  - rani modeli izvode instrukcije **u skladu s redosljedom** (in order)
  - kasniji modeli (1995 →) dinamički razrješavaju međuovisnosti u prozoru izvođenja (30-200 instrukcija)
    - ograničeno upravljanje temeljeno na toku podataka (restricted dataflow control)
    - jednostavne instrukcije pogodne za takav model izvršavanja

Alpha 21064



[Ribarić]

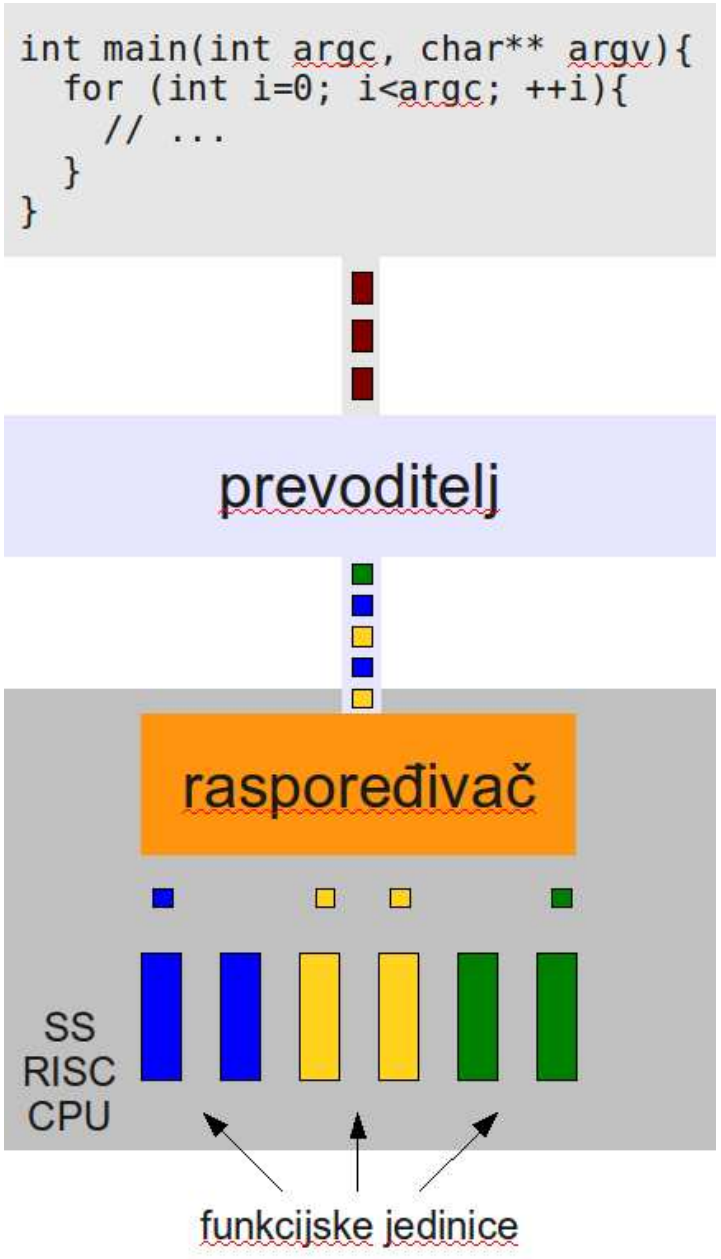
- Arhitektura x86 i dalje aktualna
  - prekretnica se dogodila 1995. godine:
    - PentiumPro, Cyrix 6x86 i AMD K5
    - CISC ISA, RISC organizacija
    - dinamičko prevođenje CISC instrukcija (x86) u interne RISC instrukcije
  - razlog: veliki **moment** uslijed rasprostranjenosti, Mooreov zakon
  - prednost “čistih” arhitektura (MIPS, Sparc, Alpha) poništena kompetitivnom tehnologijom izrade (bolja tehnologija je dostupna uslijed velikih serija)
  - današnji predstavnici: AMD Opteron, Intel Core i7

Instrukcijski paralelizam u superskalarnom RISC procesoru

Izvođenje može biti u skladu s programskim redosljedom ili izvan redosljeda (OoO)

Jednostavne instrukcije (RISC):

- olakšavaju izvedbu protočnih funkcijskih jedinica
- olakšavaju analizu međuovisnosti kod dinamičkog raspoređivanja
- impliciraju finu zrnatost programa koja vrlo dobro izlaže instrukcijski paralelizam



Odsječak programa za MIPS R10000 (1995):

```
l.d   f6, 34(r2)
l.d   f2, 45(r3)
mul.d f0, f2,f4
sub.d f8, f6,f2
div.d f10, f0,f6
add.d f6, f8,f2
```

konflikti: RAW, WAR

- **add.d** izvršen prije **div.d** jer su mu operandi postali prije dostupni (latencija **mul** > latencije **sub**)
- upis rezultata **add.d** čeka da **div.d** pročita operand **F6**

Redosljed izvođenja (dinamičko raspoređivanje):

1.
```
l.d   f6, 34(r2)
l.d   f2, 45(r3)
```

2.
```
mul.d f0, f2,f4
sub.d f8, f6,f2
```

3.
```
add.d f6, f8,f2
```

4.
```
div.d f10, f0,f6
```

[Hennessy07]

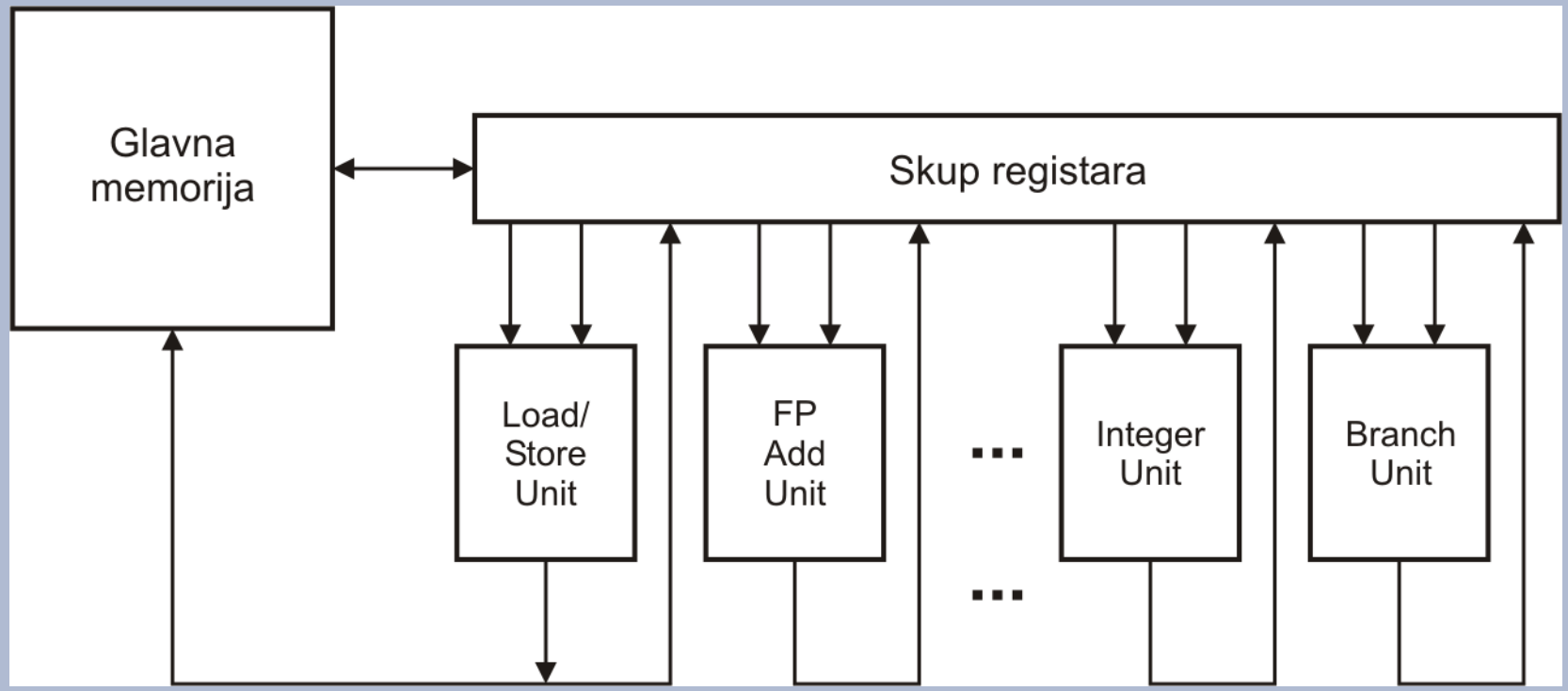
Klasifikacija instrukcijskih arhitektura (4)

- **VLIW** (very long instruction word, cca 1980):
  - specijalizirani ugrađeni procesori, TransMeta Crusoe, Intel Itanium
  - detekciju instrukcijskog paralelizma vrši prevoditelj

- Tipične značajke:
  - izravno upravljanje sklopovskim resursima (duga riječ, analogno horizontalnom mikroprogramiranju)
  - vrlo dobro iskorištenje sklopovskih resursa, jednostavnije upravljanje u odnosu na OoO RISC
  - višestrukost funkcijskih jedinica, superskalarna obrada
  - instrukcije unaprijed posložene od strane prevodioca
    - slabe reakcije na dinamičke događaje (npr, promašaj priručne memorije, krivo predviđeno grananje)
    - upitna prenosivost izvršnog koda preko generacija procesora
    - OoO RISC ima prednost kod računala opće namjene

- EPIC: explicitly parallel instruction computing
  - koncept zamišljen kao poboljšanje VLIW-a (HP, 1997)
  - relativno slabi rezultati u praksi (Itanium)

VLIW



Format instrukcije:

Load / Store	FP Add	FP Multiply	Branch	. . . .	Integer

[Ribarić]

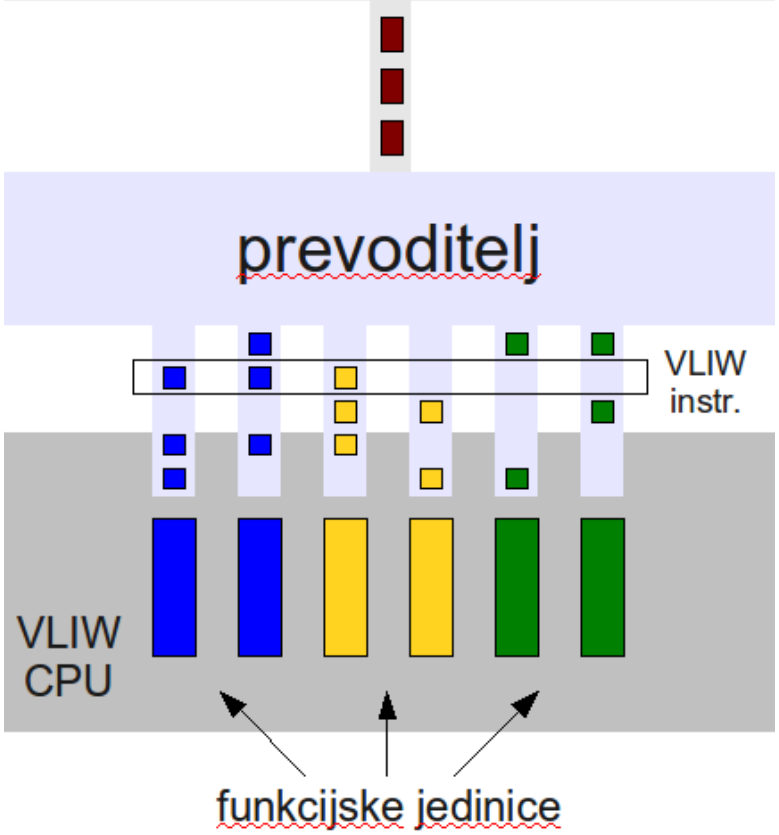
Iskorištavanje instrukcijskog paralelizma u procesoru tipa VLIW

U odnosu na OoO RISC:

- bolje iskorištenje paralelizma (prevodioc ima više informacija i više vremena od sklopovlja)
- (puno) jednostavnije upravljanje
- slabije reakcije na dinamičke događaje poput promašaja PM
- rezultati u tipičnim primjenama usporedivi s višestruko jeftinijim Intelovim i AMD-ovim procesorima

Prednost u specijaliziranim primjenama kad se kod može pažljivo optimirati

```
int main(int argc, char** argv){
  for (int i=0; i<argc; ++i){
    // ...
  }
}
```



Explicitly Parallel Instruction Computing
EPIC