Višeprocesorska i višejezgrena računala

1. Višeprocesorski sustavi (MIMD)
- podjela s obzirom na memorijsku organizaciju

2. Višejezgreni procesori
- podjela s obzirom na izvedbu višedretvenosti

1. Višeprocesorska računala (MIMD)

MIMD arhitektura obilježena je višestrukim instrukcijskim tokom i višestrukim tokom podataka.

Svaki od procesora pribavlja i izvršava svoje vlastite instrukcije na svojim podacima.

Višeprocesorska računala iskorištavaju *paralelizam na razini procesa i dretvi*.

- svaki procesor može izvršavati njemu dodijeljen *proces*.
- svaki od procesa može imati više dretvi tako da se izvođenje jednog procesa s većim brojem dretvi može povjeriti većem broju procesora.

Višedretveni višeprocesorski sustavi dopuštaju istodobno izvođenje većeg broja procesa s izdvojenim adresnim prostorima i izvođenje više dretvi koje dijele adresni prostor.

Osnovna značajka višeprocesorskog računala jest veći broj procesora približno jednakih (vrlo često) identičnih obilježja koji na izvjestan način dijele zajednički memorijski prostor.

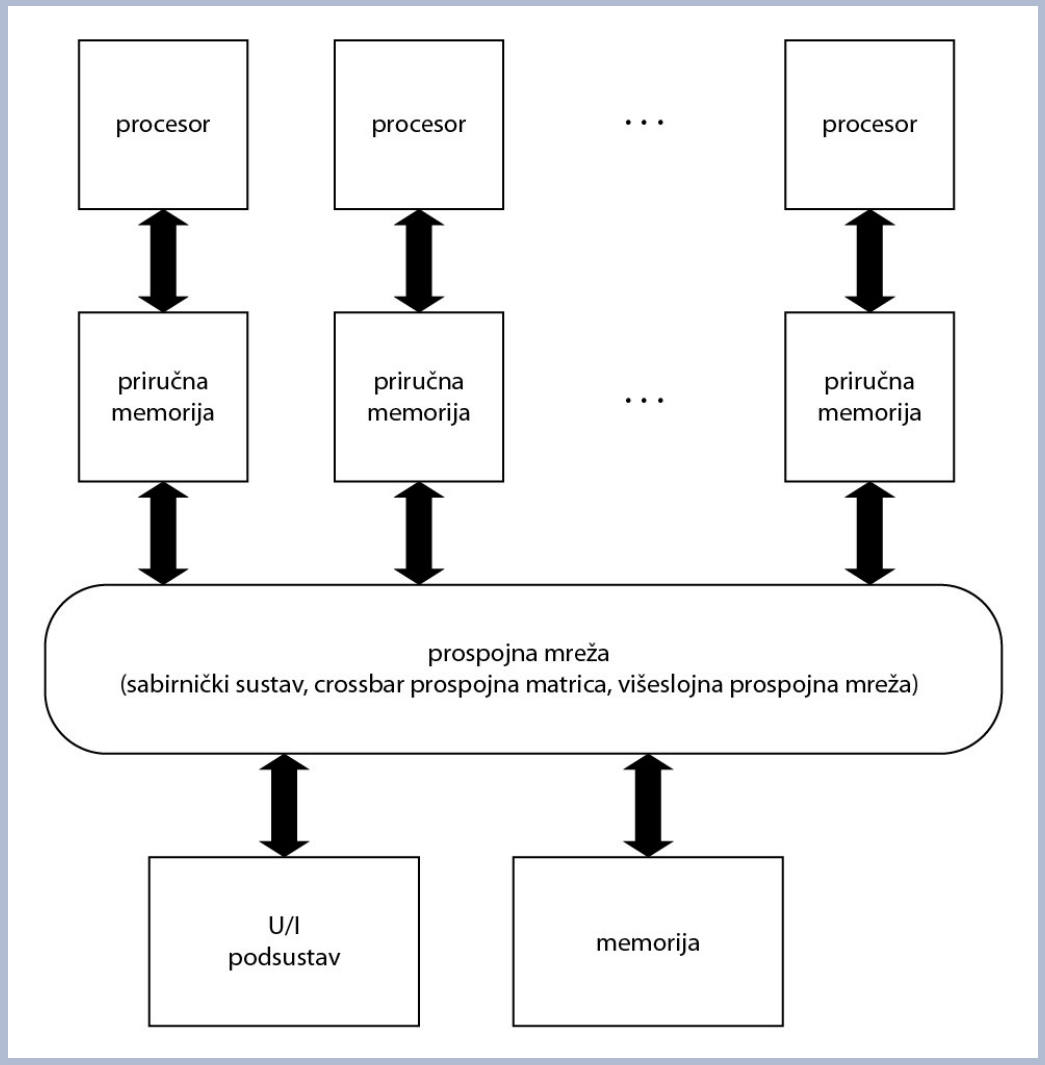
Ovisno o broju procesora i načinu organizacije memorijskog sustava višeprocesorska računala mogu se klasificirati u sljedeće skupine:

- računala s uniformnim pristupom memoriji UMA (engl. *Uniform Memory Access*),
- računala s neuniformnim pristupom memoriji NUMA (engl. *Nonuniform Memory Access*),
- računala koja imaju samo priručnu memoriju COMA (engl. *Cache-Only Memory Architecture*).

U *UMA računalu* svi procesori dijele zajedničku fizičku memoriju i svi imaju jednako vrijeme pristupa svakoj od riječi u memoriji (uniformni, odnosno ujednačeni pristup memoriji).

Svaki od procesora može imati i svoju vlastitu priručnu memoriju organiziranu u jednu ili više razina.

Na sličan način kao što dijele memoriju, procesori dijele i resurse U/I podsustava.



UMA model višeprocesorskog računala

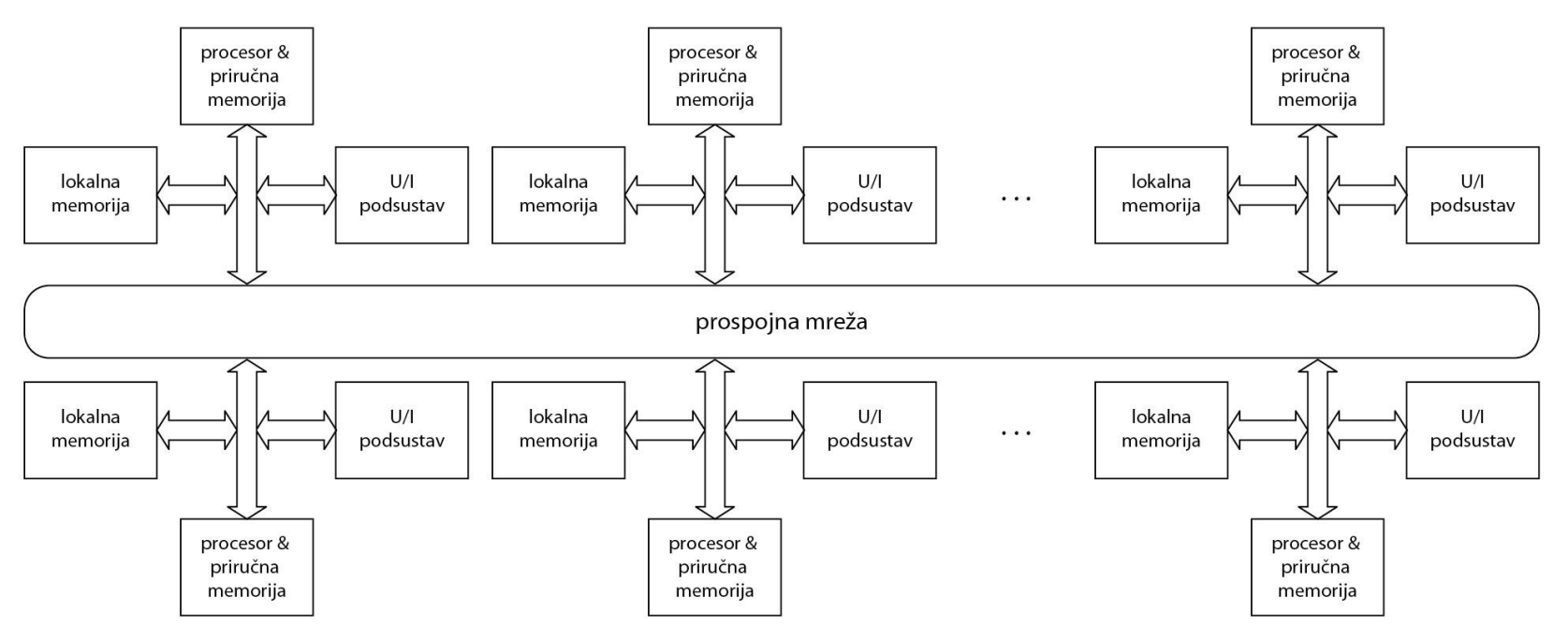
UMA model se još naziva i *višeprocesorsko računalo sa središnjom dijeljenom memorijom* (engl. *centralized shared-memory*) ili *simetrični multiprocesorski sustav s dijeljenom memorijom* SMP (engl. *symmetric shared-memory multiprocessor*) – zato što memorija ima isti ("simetrični") odnos spram svih procesora.

Komunikacija procesora sa zajedničkom memorijom i U/I podsustavom ostvaruje se prospojnom mrežom koja ovisno o zahtijevanoj propusnosti može biti ostvarena:

- sabirničkim sustavom,
- prospojnom matricom (crossbar switch) ili
- višerazinskom prospojnom mrežom (npr. Omega).

Multiprocesorski sustavi s neuniformnim pristupom memoriji NUMA

- *sustavi s porazdijeljenom memorijom*
  (engl. *distributed-memory multiprocessor*)
- umjesto centralizirane memorije, imaju memoriju pojedinih procesora.



Zbirka svih lokalnih memorija oblikuje globalni adresni prostor kojem mogu pristupiti svi procesori u sustavu. Zbog takve organizacije memorije razlikujemo dvije vrste pristupa memoriji:

- brzi pristup memoriji (kraće vrijeme pristupa) kada procesor pristupa svojoj lokalnoj memoriji,
- sporiji pristup (dulje vrijeme pristupa) kada procesor pristupa "udaljenoj" memoriji koja je, zapravo, lokalna memorija nekog drugog procesora; dulje vrijeme pristupa uzrokovano je dodatnim kašnjenjima jer se "udaljenoj" memoriji pristupa kroz prospojnu mrežu.

Multiprocesorski sustavi oblikovani u skladu s modelom NUMA imaju veliki broj procesora, npr. nekoliko stotina ili tisuća, i zato se za toliki broj procesora teško može realizirati središnja memorija sa zahtijevanom, odnosno prihvatljivom propusnosti (engl. *memory bandwidth*).

Svaki čvor u NUMA modelu sastoji od procesora, lokalne memorije, U/I podsustava i sučelja za pristup prospojnoj mreži. Ovisno o izvedbi NUMA modela, čvor može biti sastavljen od određenog broja procesora i lokalnih memorijskih modula tako da čini procesorsku nakupinu (engl. *processor cluster*).

Model COMA (cache only memory architecture) je sličan modelu NUMA; razlika je u tome što ovdje matična adresa podatka može migrirati na druge čvorove.

Logička adresa podatka ostaje ista, ali fizička adresa se mijenja. To je moguće provesti zbog adresnog preslikavanja (virtualni memorijski sustav).

Postoje i NUMA/COMA hibridi gdje neke stranice logičkog prostora započinju u NUMA načinu i po potrebi se izmijene u način rada prema modelu COMA.

2. Višejezgreni i višedretveni procesori

Višedretvenost (engl. *multithreading*) podrazumijeva da se više dretvi izvodi u jednom procesoru tako da se dretve međusobno isprepliću

- dretve se naizmjenično izvode u dodijeljenim funkcijskim jedinicama procesora uz nužno prospajanje konteksta sadržanog u tablici dretvi, i to nakon svake izmjene dretve.

Dva su glavna pristupa višedretvenosti:

- finozrnata višedretvenost (engl. *fine-grained multithreading*),
- grubozrnata višedretvenost (engl. *coarse-grained multithreading*).

Finozrnata višedretvenost podrazumijeva prospajanje dretvi nakon svake instrukcije.

- takvi procesori nazivaju se još i "bačvasti" (*barrel*) procesori
- susjedne instrukcije međusobno nezavisne jer pripadaju različitim dretvama tako da se protočna struktura djelotvorno iskorištava.

Višedretveno izvođenje povećava broj promašaja priručne memorije zbog narušavanja lokalnosti programa.

- obično se izvođenje dretvi u tom slučaju temelji na kružnom prioritetu (engl. *round-robin*) uz "preskakanje" dretvi koje su u stanju zastoja.

Višedretveni procesori imaju sklopovski podržano brzo uklapanje konteksta dretvi (eng. *hardware based fast context switching*) koje se temelji na tome da svaka dretva ima svoj skup fizičkih registara za pohranu konteksta izvođenja

Primjer:

Višedretveni procesor Tera (sada Cray MTA) podržava 128 dretvi, pri čemu je svakoj dretvi dodijeljen 41 64-bitni registar u procesoru.

- osim sklopovski podržanog brzog prospajanja konteksta, jezgre imaju i dinamičku izvedbu preimenovanja registara kojim se rješavaju hazardi WAW i WAR

Grubozrnata višedretvenost predviđa prospajanje dretvi samo onda kada nastupa dulji zastoj u tekućoj dretvi (npr. promašaj u priručnoj memoriji).

Za kraće zastoje, na primjer one izazvane hazardima, ne predviđa se izmjena dretvi te je to jedan od glavnih nedostataka grubozrnate višedretvenosti.

- takva odluka temelji se na procjeni cijene i količine posla za pražnjenje protočne strukture da bi se u nju mogao uputiti instrukcijski tok druge dretve
- izmjena dretvi i prospajanje njihova konteksta može se događati i pri svakoj *load* instrukciji (neovisno o tome je li se dogodio promašaj) ili nakon programskog odsječka (bloka instrukcija) koji pripadaju jednoj dretvi.

Grubozrnata i finozrnata višedretvenost može se kombinirati s paralelizmom na razini instrukcija ILP, ali i sa superskalarnosti (višestrukim protočnim strukturama u jednom procesoru).

Tri su procesorske konfiguracije moguće u tom slučaju:

- superskalarnost s grubozrnatom višedretvenosti,
- superskalarnost s finozrnatom višedretvenosti,
- superskalarnost sa simultanom višedretvenosti.

Superskalarni procesori s gubozrnatom višedretvenosti izvode istodobno veći broj instrukcija koje pripadaju istoj dretvi sve do trenutka kada nastupi dulji zastoj, u tom se trenutku prospaja kontekst dretvi i nastavlja se s izvođenjem druge dretve.

Superskalarni procesori s finozrnatom višedretvenosti isprepliću dretve nakon svake instrukcije

- to eliminira neke zastoje u izdavanju instrukcija uslijed hazarda
- budući da samo jedna dretva izdaje instrukcije tijekom perioda signala vremenskog vođenja, još uvijek postoje ograničenja u paralelizmu na razini instrukcija.

Npr. višejezgreni procesori tvrtke Sun nazvani T1 (Niagara 1) i T2 (Niagara 2) koriste finozrnatu višedretvenost.

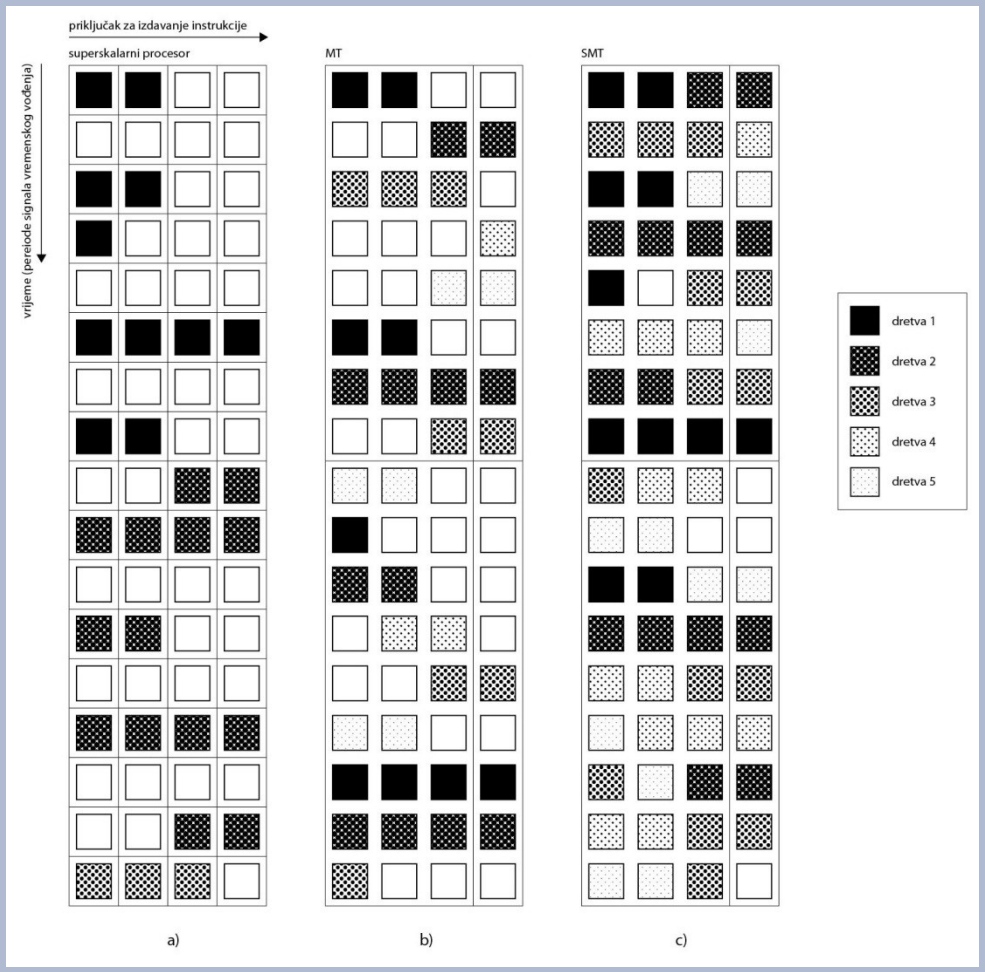
Simultana višedretvenost SMT (engl. *simultaneous multithreading*) zasniva se na činjenici da suvremeni (superskalarni) procesori imaju paralelizam na razini funkcijskih jedinica veći od onog koji jedna dretva može iskoristiti.

SMT računala dinamičkim raspoređivanjem *istodobno* izdaju više instrukcija iz *nezavisnih* dretvi u istoj periodi signala vremenskog vođenja.

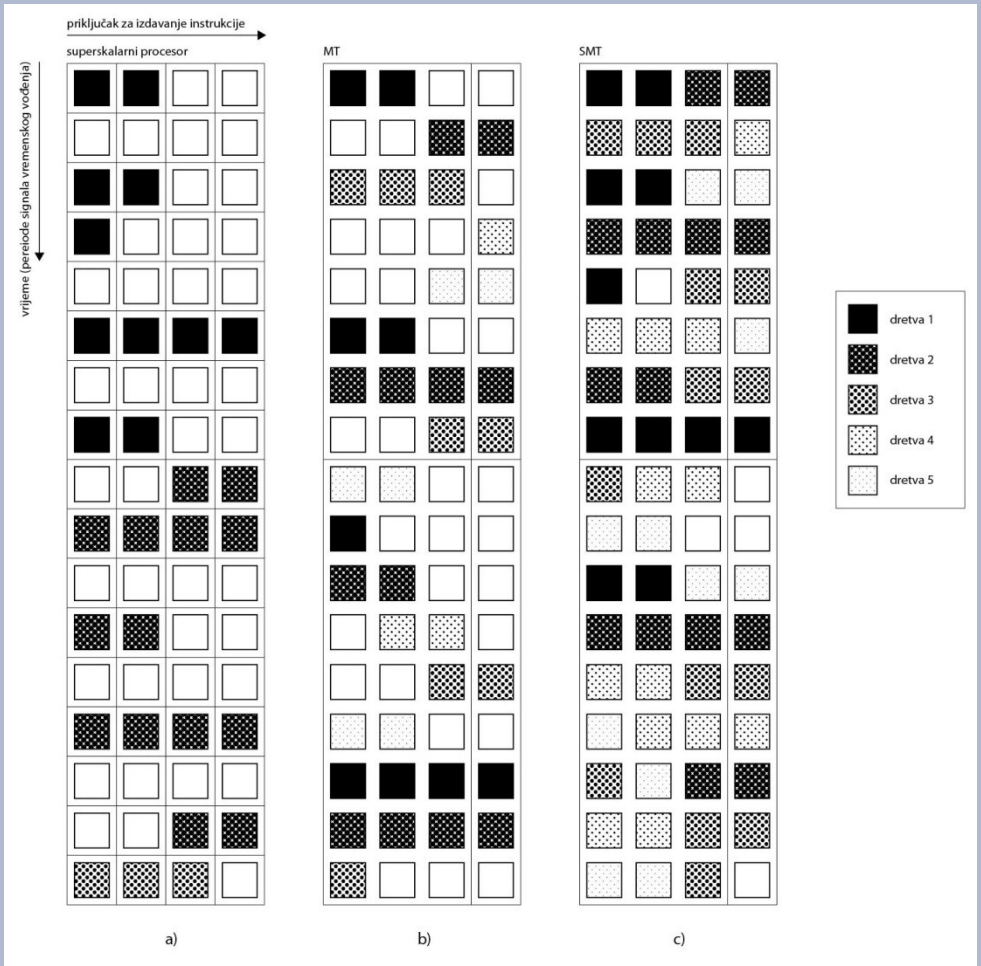
U SMT-u se iskorištava paralelizam na razini dretvi i paralelizam na instrukcijskoj razini u kombinaciji s izdavanjem više instrukcija u jednom periodu signala vremenskog vođenja.

Primjeri procesora koji se temelje na SMT zamislima su Intelovi i AMD-ovi višejezgreni procesori (Core, Ryzen) te IBM Power.

Razlike između superskalarnosti, višedretvenosti (MT) i simultane višedretvenosti (SMT) ilustrirat ćemo na Eggersovom modelu obrade.



x os – priključci za izdavanje instrukcija (izvršne jedinice)
y os – vrijeme (periodi signala vremenskog vođenja rastu prema dolje)



Nekorišteni priključci za izdavanje instrukcija mogu se promatrati kao "horizontalno neiskorišteni priključci" i kao "vertikalno neiskorišteni priključci"

Pod horizontalno neiskorištenim priključcima podrazumijevamo slučaj kada su u jednom retku jedan ili više priključaka neiskorišteni (ali ne svi!).

Vertikalno neiskorišteni priključci događaju se kada su tijekom jedne periode *svi* priključci neiskorišteni – to nastupa zbog dulje latencije (duljeg zastoja u izvođenju) instrukcije, npr. pristupa memoriji, kada je privremeno spriječeno daljnje izdavanje instrukcija.

SMT procesor u svakom periodu takta "izabire" instrukcije za izvođenje iz svih dretvi.

- iskorištava se paralelizam na razini instrukcija izborom instrukcija iz bilo koje dretve
- nakon toga procesor dinamički raspoređuje resurse procesora između instrukcija i osigurava visoku razinu iskoristivosti sklopovskih resursa
- ako neka od dretvi ima visok stupanj paralelizma na razini instrukcija, onda se te instrukcije izvode i pritom je vrlo malo horizontalno neiskorištenih priključaka.

Ako paralelizam na razini instrukcija za jednu dretvu nije dovoljan, onda se izabire još jedna dretva (ili više njih) s nižim stupnjem paralelizma koja će popuniti prazne horizontalne priključke.

Na taj se način postiže uklanjanje i vertikalnih neiskorištenih priključaka i u velikoj mjeri horizontalnih neiskorištenih

Pretpostavite da je procesor superskalarni konvencionalne arhitekture i da može izdavati do četiri instrukcije u jednoj periodi signala vremenskog vođenja.
Pretpostavite da protočne strukture nisu specijalizirane i da se trebaju izvesti sljedeće tri dretve prikazane na slici:

dretva1:

1	1		
1			
1	1	1	
1	1		
1	1	1	1
1			
1	1	1	
1	1		

dretva2:

2	2	2	
2	2		
2			
2			
2			
2	2	2	2
2	2		
2	2		

dretva3

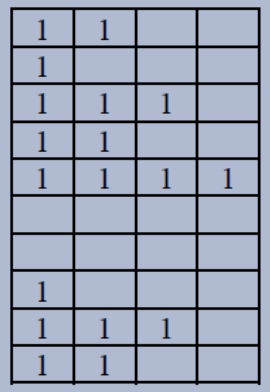
3	3	3	
3	3		
3			
3			
3	3	3	

rješenje:

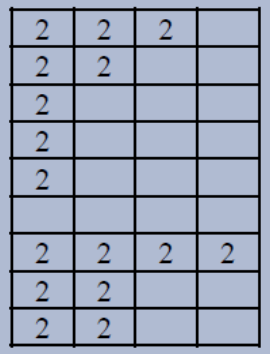
1	1		
1			
1	1	1	
1	1		
1	1	1	1
1			
1	1	1	
1	1		
2	2	2	
2	2		
2			
2			
2			
2	2	2	2
2	2		
2	2		
3	3	3	
3	3		
3			
3			
3	3	3	

Za tri dretve prikažite izvođenje za višedretveni superskalarni procesor (izdaje do četiri instrukcije) koji koristi finozrnatu dretvenost.
Pretpostavite da protočne strukture nisu specijalizirane i da se trebaju izvesti sljedeće tri dretve prikazane na slici:

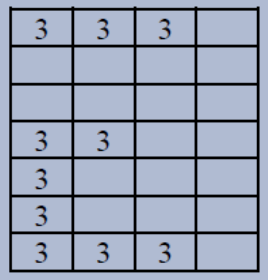
dretva1:



dretva2:

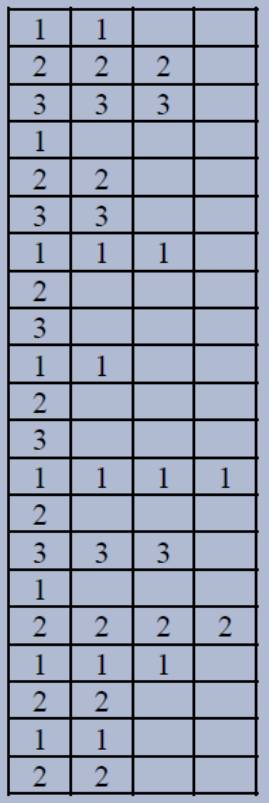


dretva3:



Finozrnata višedretvenost podrazumijeva prospajanje dretvi nakon svake instrukcije. Te su instrukcije međusobno nezavisne jer pripadaju različitim dretvama tako se protočna struktura djelotvorno iskorištava.

rješenje:



Prikažite izvođenje za višedretveni superskalarni procesor (izdaje do četiri instrukcije) koji koristi grubozrnatu dretvenost. Pretpostavite da se prospajanje dretvi događa kod zastoja koji odgovara vertikalno neiskorištenim priključcima. Također pretpostavite da prospajanje dretvi u ovom slučaju zahtijeva jednu periodu signala vremenskog vođenja.

dretva1:

1	1		
1			
1	1	1	
1	1		
1	1	1	1
1			
1	1	1	
1	1		

dretva2:

2	2	2	
2	2		
2			
2			
2			
2	2	2	2
2	2		
2	2		

dretva3:

3	3	3	
3	3		
3			
3			
3	3	3	

rješenje:

1	1		
1			
1	1	1	
1	1		
1	1	1	1
2	2	2	
2	2		
2			
2			
2			
3	3	3	
1			
1	1	1	
1	1		
2	2	2	2
2	2		
2	2		
3	3		
3			
3			
3	3	3	

Prikažite izvođenje u superskalarnom simultanom višdretvenom procesoru SMT (izdaje do četiri instrukcije) uz pretpostavku da protočne strukture nisu specijalizirane.

rješenje:

dretva1:

1	1		
1			
1	1	1	
1	1		
1	1	1	1
1			
1	1	1	
1	1		

dretva2:

2	2	2	
2	2		
2			
2			
2			
2	2	2	2
2	2		
2	2		

dretva3:

3	3	3	
3	3		
3			
3			
3	3	3	

1	1	2	2
2	3	3	3
1	2	2	
1	1	1	2
3	3	1	1
2	3	1	1
2	3	1	1
3	3	3	
2	2	2	2
1	2	2	
1	1	1	2
1	1	2	