

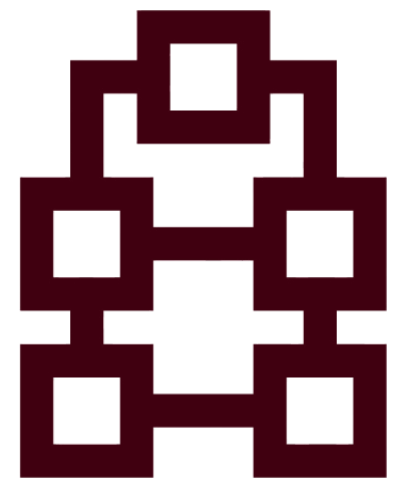
ELEKTROTEHNIČKI FAKULTET, UNIVERZITET U BEOGRADU
PARALELIZAM U ARHITEKTURI RAČUNARA I PROGRAMSKIM PREVODIOCIMA

TOMASULO ALGORITAM

doc. dr Maja Vukasović
as. ms Nikola Stanković

Tomasulo algoritam

- **Ugradjena *dataflow* mašina** za ograničen broj instrukcija (instrukcijski prozor) u procesoru.
- Eliminisanje antizavisnosti i izlaznih zavisnosti.
- Uloga *dataflow* mašine – izračunavanjem rezultata neke instrukcije, sve instrukcije koje zavise od nje postaju *data-ready* za taj argument.
- Rezultat operacije koja je izvršena se mora rasporediti do svih operacija koje čekaju.
- Hardverski se mora odrediti kada su argumenti i operacije *data-ready*.
- ***Tomasulo*** rešenje:
 - Emitovanje rezultata preko zajedničkog *data bus*-a do svih primalaca.
- Minimizacija hardvera i kontrole.
- Instrukcije – primaoci argumenata osluškuju *data bus* i prepoznaju podatak na osnovu identifikatora.



Data bus

- Sve instrukcije u instrukcijskom prozoru su povezane na zajednički *data bus*.
- Šalju se:
 - Rezultat operacije
 - Identifikator operacije koja emituje rezultat (jedinstven na nivou instrukcijskog prozora)
- Nakon emitovanja rezultata:
 - Sve zavisne instrukcije pokupe rezultat
 - Instrukcija koja je emitovala rezultat uklanja se iz dinamičkog DDG ugradjene *dataflow* mašine

Rezervacione stanice

- Jedna instrukcija u prozoru – jedna **rezervaciona stanica**.
- Poredi identifikator instrukcije koja emituje rezultat sa identifikatorom za argument svoje instrukcije koje nisu *data-ready*.
- Ako je identifikator identičan, pokupi podatak sa *data bus*-a i zapamti kao odgovarajući argument i postavi argument kao *data-ready*.
- Kada su svi argumenti *data-ready*, instrukcija postaje *data-ready*.
- Kada je instrukcija *data-ready*, ide na izvršavanje.
- Kada se instrukcija izvrši, šalje svoj rezultat na *data bus* zajedno sa identifikatorom rezervacione stanice u kojoj se instrukcija nalazi.
- Oslobadja se rezervaciona stanica za novu instrukciju.
- Veličina DDG-a je ograničena brojem rezervacionih stanica.

Uslov za ulazak nove operacije u instrukcijski prozor

- Na odgovarajućem dinamičkom tragu je.
- Prethodne instrukcije na tragu su ubačene u instrukcijski prozor.
- Postoji slobodna rezervaciona stanica.
- Moguće je ubaciti i paket instrukcija po ciklusu.

Identifikacija rezultata operacije

- Formiranje DDG se radi na osnovu predviđenog dinamičkog traga.
- Instrukcije su na tragu poredjane u programskom redosledu (*inorder*).
- Instrukcije se ubacuju u prozor u *inorder* poretku.
- Prilikom ubacivanja, radi se preimenovanje argumenata i rezultata -> jednoznačna definicija pravih zavisnosti po podacima i eliminacija antizavisnosti i izlaznih zavisnosti.
- Jedna instrukcija – jedan preimenovani rezultat – jedna rezervaciona stanica.
- Jedinstveni identifikator operacije na *data bus*-u – adresa rezervacione stanice.

Identifikacija argumenata

- Mapiranje argumenata operacije koja se ubacuje u prozor na izvor rezultata tih argumenata.
 - Jedinstven identifikator operacije koja proizvodi vrednost argumenta.
- Umesto adrese arhitekturnih registara za argumente – adrese rezervacionih stanica instrukcija koje proizvode te argumente.

Postupak ubacivanja u instrukcijski prozor

- Jedan arhitekturni registar se može preslikati na više rezervacionih stanica.
- Kada se ubacuje nova instrukcija, za argumente se posmatra poslednje takvo preslikavanje.
 - Preslikavanje se radi na osnovu arhitekturnog stanja registra.
- Ubacivanjem nove instrukcije u prozor, menja se arhitekturno stanje.
- Postoji tabela preslikavanja adresa arhitekturnih registara argumenata u lokacije rezervacionih stanica instrukcija u arhitekturnom stanju koje izračunavaju argumente.

Instrukcijski prozor

- Da li su sve instrukcije koje treba da pokupe rezultat neke operacije u instrukcijskom prozoru?
- Ako jesu, rezultat se ne pamti dodatno u registrima, nego se prosledjuje preko *data bus*-a – *forwarding*.
- Ako nisu, sadržaj *data bus*-a se pamti u registarskom fajlu – **registarski fajl nepreimenovanih vrednosti**.