Vektorski računari

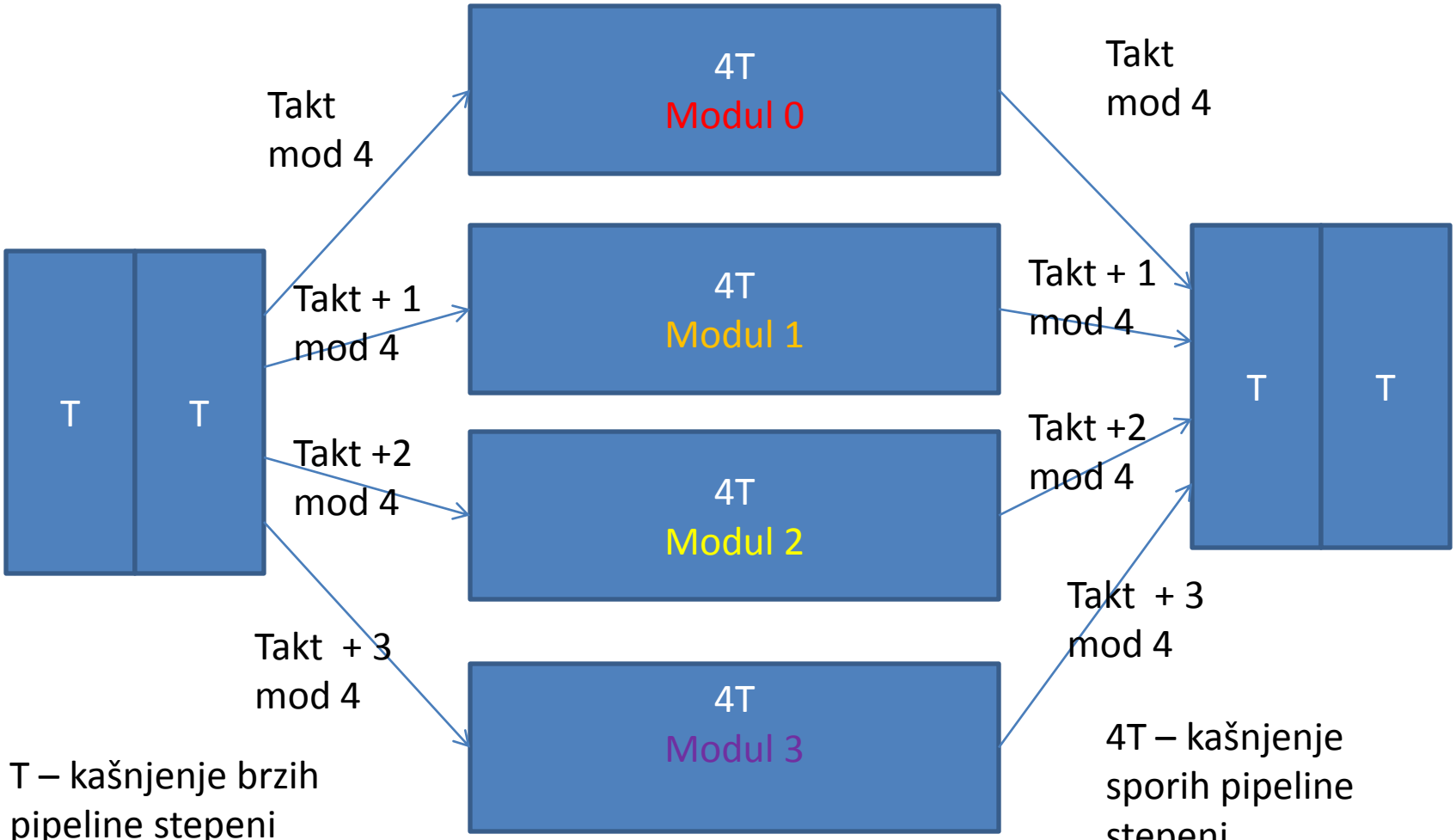
Zoran Jovanovic

Korišćeni slajdovi sa vodećih univerziteta u SAD

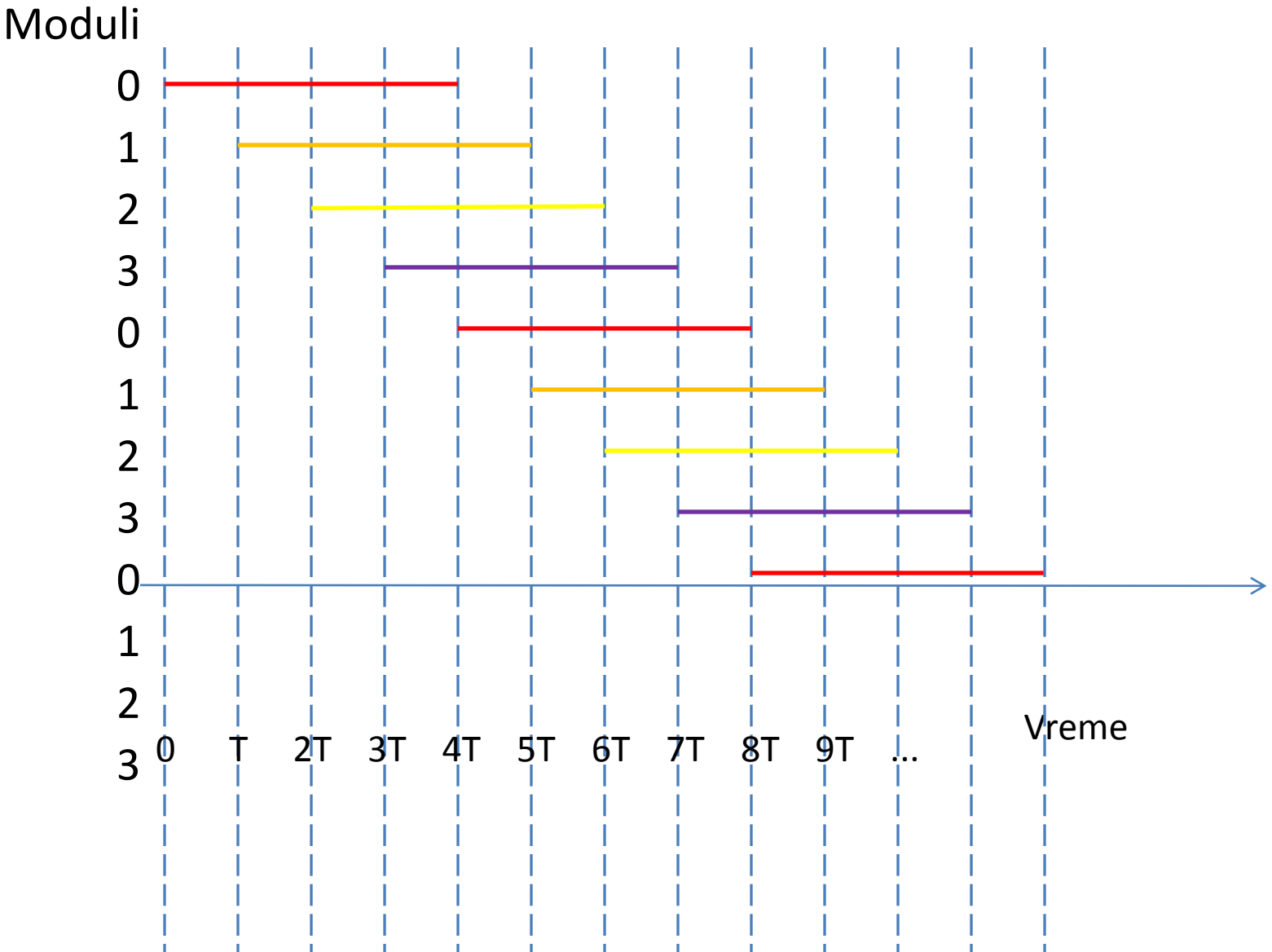
Paralelizam je u DoAll petljama?

- Odvojimo deo mašine za skalarni deo koda sa neregularnim paralelizmom i uradimo neku paralelizaciju (skalarni procesor)
- Deo mašine odvojimo samo za programske petlje, realno za DoAll i napravimo da postoji hardverska podrška za nizove (vektorski procesor)
- Svaki deo radi ono za šta je bolji, a kompajler ili čak programer odvaja kôd za skalarni ili vektorski procesor
- Kako napraviti vektorski procesor?

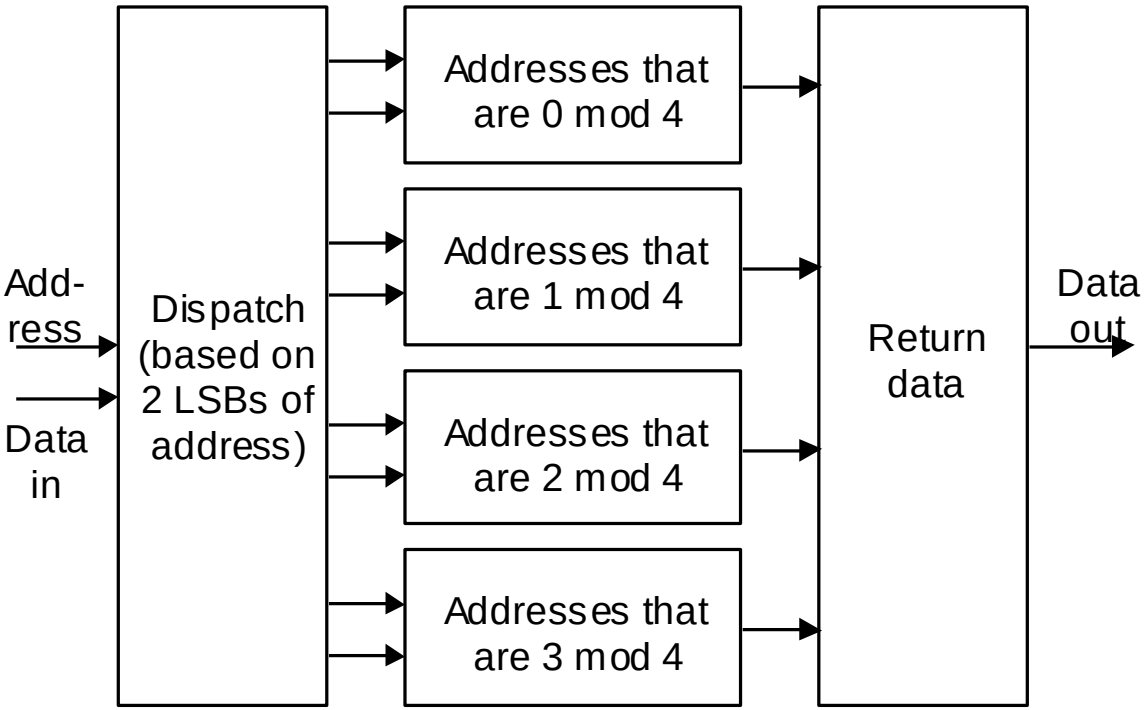
Protočni stepeni sa velikim kašnjenjem kombinacione logike u bržem pipeline-u

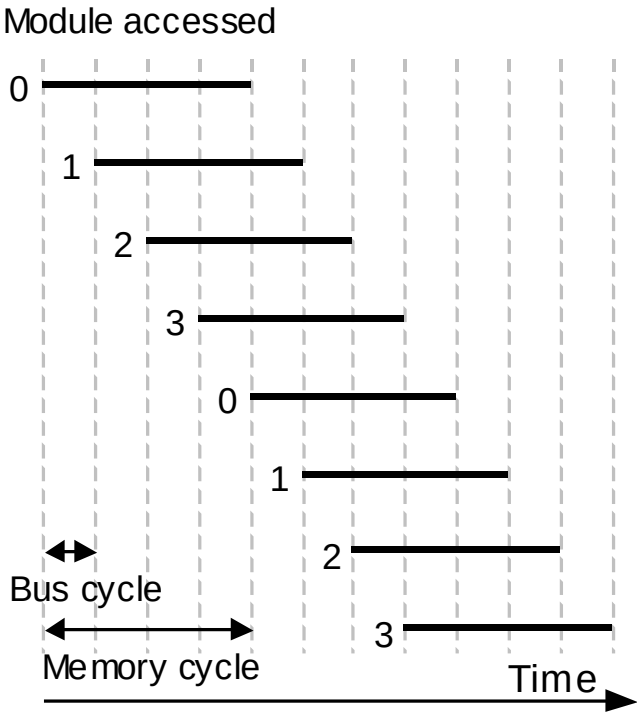


Ciklusi

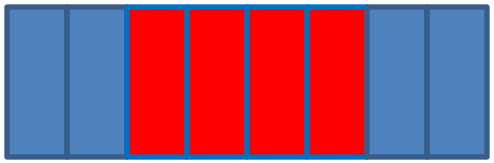


Memory Interleaving – memorija je spora!!!





Ako se sekvencijalno ide po adresama u memorijskim operacijama, iluzija je sledeća:

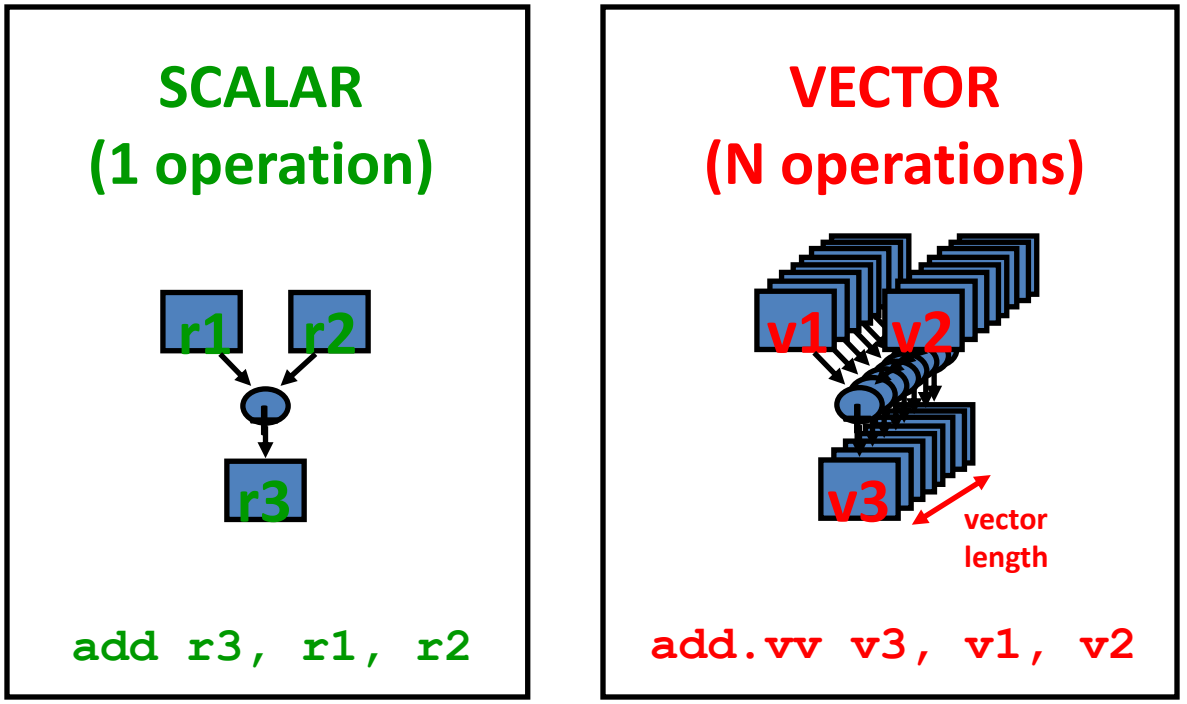


"Memorijski pipeline"

4* sporija memorija radi efektivno punom brzinom u pipeline-u ako čitamo ili upisujemo u blokove podataka!!!
CRAY 16* sporija memorija (mod 16)!!

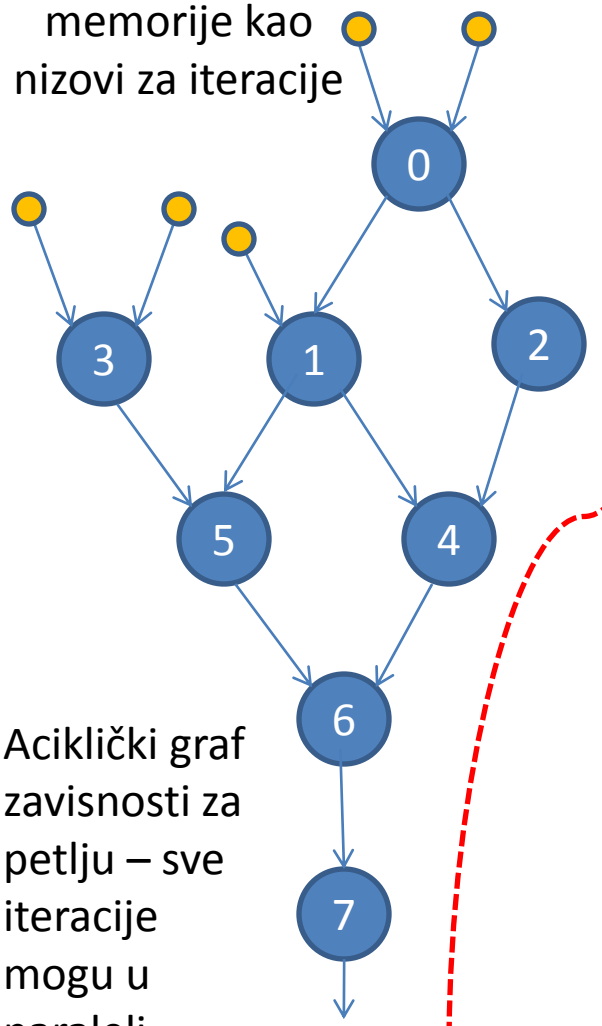
Model za petlje koje obrađuju nizove podataka – naročito DoAll

- Iskoristiti Memory interleaving da se napune vektorski registri (sadrže cele nizove za sve ili deo operacija data ready na vrhu grafa)
- Međurezultati se takođe čuvaju u vektorskim registrima



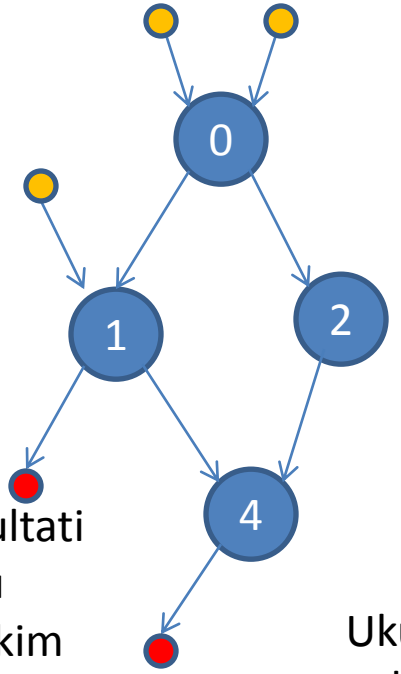
DoAll i vektorski registri

Argumenti iz memorije kao nizovi za iteracije



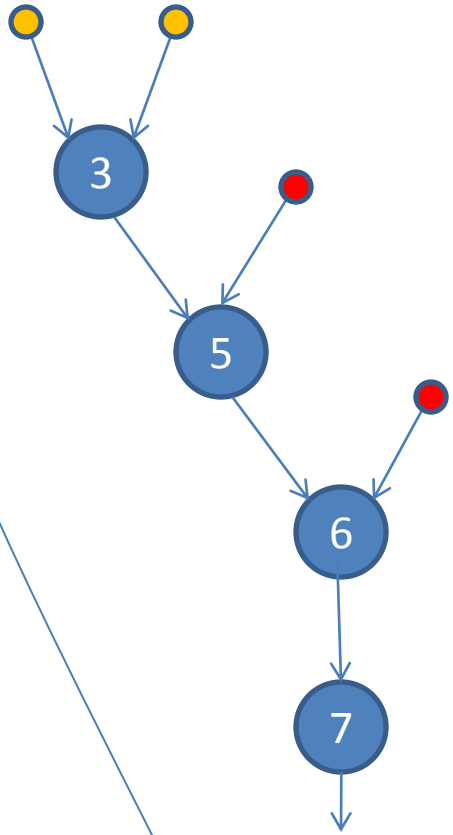
Aciklički graf zavisnosti za petlju – sve iteracije mogu u paraleli

Ideja: prvo uraditi za sve iteracije deo grafa (~4 operacije se započinje po ciklusu)



Ovi rezultati ostaju u vektorskim registrima

Zatim uraditi sledeći deo (~4 operacije po ciklusu) ...



Ukupno 9 vektorskih registara

Vektorska jedinica i ulančavanje instrukcija

- Iz acikličkog grafa otkidamo čitave podgrafove slobodne na vrhu i povezujemo ih u složene pipeline-e prema grafu zavisnosti po podacima.
- Data ready nizove ubacujemo u vektorske registre – **Interleaving. Nema cache za nizove!!!**
- Veličina podgrafa zavisi od broja raspoloživih ALU i broja vektorskih registara
- Veza rezultat-argument ostvaruje se preko istog vektorskog registra
- Vektorski registri mogu da ubace kašnjenja (pointer(i) za read i write pomereni) da se sinhronizuje stizanje argumenata za ulančani pipeline

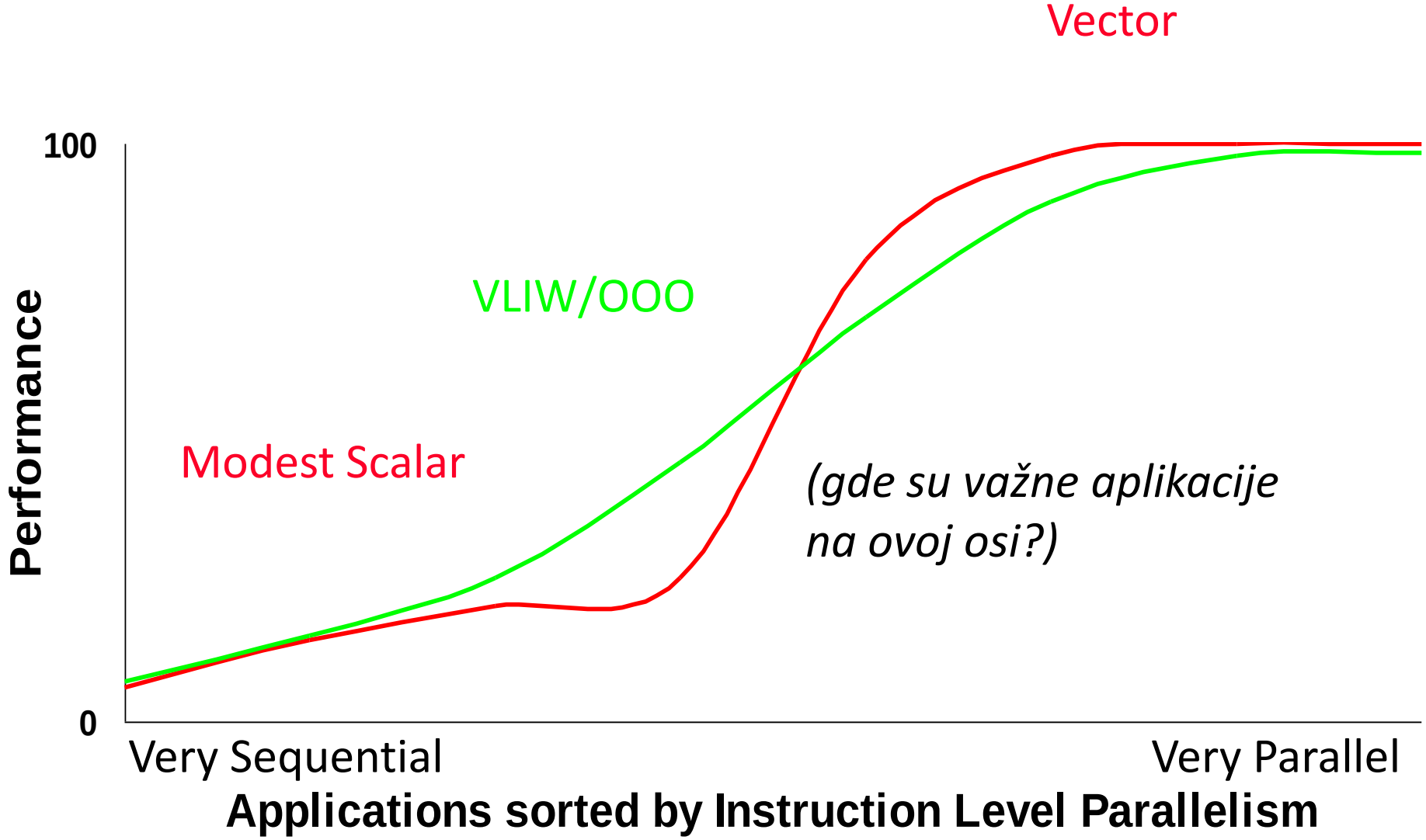
Vektorska jedinica i ulančavanje instrukcija (2)

- Deo argumenata za nove podgrafove koji postaju slobodni na vrhu su već u vektorskim registrima nakon završetka prethodnog podgrafa!!
- Nema adresiranja memorije ili cache-a i povećava se dubina pipeline-a
- Broj lokacija vektorskih registara za elemente nizova je isti
- Ako je broj lokacija svakog vektorskog registra manja od ukupnog broja iteracija, mora se raditi manji broj iteracija - malo manje iteracija od broja lokacija vektorskih registara

Komponente Vektorskih Procesora

- *Vektorski Registri*: Fiksne veličine za jedan vektor (niz)
  - ima najmanje 2 read i 1 write port
  - tipično je 8-32 vektorska registra, svaki sa 64-128 64-bit elemenata
- *Vektorske Funkcionalne Jedinice (FJ)*: pipeline koji započinje novu operaciju svakog ciklusa
  - tipično 4 do 8 FJ: npr. 2x FP add, 2x FP mult, FP reciprocal (1/X), integer add, logical, shift;
- *Vector Load-Store jedinice (LSUs)*: memorijski pipeline interleaved memorije
- *Skalarni registri*: po jedan element za FP skalar ili adresu
- Cross-bar da poveže FJ , LSU, registre

VLIW/Out-of-Order prema Modest Scalar+Vector



Vektorski procesori - mane

- Posvećeno suviše pažnje DoAll, ignorišući pripremu za vektorske operacije i podrazumevajući veliki broj iteracija (pipeline je sa velikim brojem pipeline stepeni pa traži puno iteracija da se isplati !
- Povećanje vektorskih performansi bez posvećivanja dovoljno pažnje skalarnim performansama (Amdahl-ov zakon)
- Nedovoljna je propusnost memorije u odnosu na performanse vektorskog procesora

Prednosti vektorskih procesora

- Lako se dobijaju visoke performanse; N operacija se simultano (u dubokom pipeline-u) izvršava za DoAll petlje:
- Skalabilno (dobijaju se bolje performanse sa porastom HW resursa)
- Kompaktno: N operacija se prikazuje jednom jednostavnom instrukcijom (nasuprot VLIW)
- Prediktabilne (real-time) performanse prema statističkim performansama (cache)
- Multimedia, spremno i može se birati N * 64b, 2N * 32b, 4N * 16b, 8N * 8b
- Zrela i razvijena tehnologija kompajlera