

Computer Arkitektur og Operativ Systemer

Denne forelæsning optages og gøres efterfølgende tilgængelig
på Moodle

MEDDEL VENLIGST UNDERVISEREN, HVIS DU IKKE ØNSKER, AT
OPTAGELSE FINDER STED

This lecture will be recorded and afterwards be made available
on Moodle

PLEASE INFORM THE LECTURER IF YOU DO NOT WANT
RECORDING TO TAKE PLACE

Computer Arkitektur og Operativ Systemer

Instruktionsniveau parallelisme og program optimering

Lektion 7
Brian Nielsen

*Credits to
Randy Bryant & Dave O'Hallaron (CMU)*

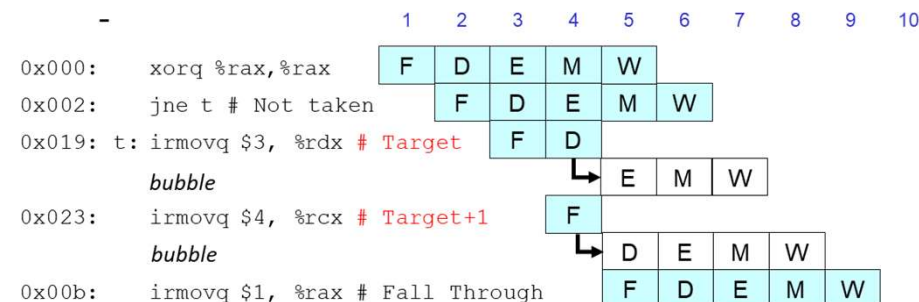
Mål

- Hvordan afvikler moderne processorer programmer?
- Hvordan vi får et program til at køre 40 gange så hurtigt?

Metode	Integer (CPE)		Double FP (CPE)	
Operation	Add	Mult	Add	Mult
Combine1 uden optimering	22.68	20.02	19.98	20.18
Combine m. bedste LxK udfoldning (8*8/10*10)	0.54	1.01	1.01	0.52

Instruktions-niveau parallelisme

- Pipelining, på papiret nemt, men
 - God og balanceret opdeling i trin?
 - Håndtering af data-afhængigheder mellem instruktioner
 - Data forwarding og "load-use stalls"
 - Håndtering af kontrol-afhængigheder mellem instruktioner
 - Spekulativ udførelse og branch-prediction



- Moderne processorer er super-skalare og anvender out-of-order afvikling

Program-optimering

- Basale og almindelige teknikker
 - Flyt beregninger uden for løkker
 - Undgå unødigt genberegning af samme værdi
 - Genbrug resultat af del-udtryk
- Compileren er normalt ret god til visse mikro/lokale optimeringer
- Forhindres ofte af uheldigt/dårligt skrevne programmer
 - Indeholder "optimization blockers"
 - Ofte kan bare en lille omskrivning af koden resultere i enorme gevinster:
 - Opsamling af mellemresultater i lokal variabel, ikke i hukommelsen
 - Inlining af små funktioner
 - Minimer løkke-overhead vha. udfoldning af løkker
- Øget instruktionsniveau parallelisme:
 - Multiple variable til opsamling af mellemresultater ("akkumulator" variable)
 - Udfoldning af løkker
- Næste gang: cache-venlig kode

Øvelserne

- Balancering og optimering af en pipeline
- Hukommelses-aliasering
- Sammenligne 2 metoder for beregning af polynomier
 - Hvilken er (optimeringsmæssigt) bedst?
 - Kritisk sti
- Foretag en loop unrolling
- Challenge 8:
 - Hvor meget kan I optimere et lille program?
 - Hvor langt kan I nå med vektor instruktionerne?
 - Hvilken effekt har loop-unrolling og multiple opsamlingsvariable i Java/C#?

Brug Hjælpelærer assistancen! Evt også til spm. mm lektionen