# Afviklings Plot (Execution graph)

*af Finn Schiermer Andersen og Michael Kirkedal Thomsen, DIKU, 2019*

Denne lille note introducerer afviklingsplot.

Et afviklingsplot er en idealiseret illustration af hvordan en mikroarkitektur afvikler en strøm af instruktioner. Men det er også et redskab som kan bruges til at bestemme CPI (cycles per instruction), et vigtig mål for en mikroarkitekturs ydeevne for en strøm af instruktioner. (Det andet vigtige mål er selvfølgelig maskinens clock-frekvens)

Vi vil lave en gradvis opbygning og langsomt øge kompleksiteten. Dvs. vi starter her med en kort beskrivelse, eksemplificeret på en enkelt-cyklus maskine. Derefter vil vi beskrive, hvordan det fungerer på en simpel pipeline maskine, hvilket vil give en dybere forståelse. Derefter vil vi bevæge os over superskalar arkitekturer til en mere avanceret pipeline mikroarkitektur. Vi vil her også stifte bekendtskab med kontrol instruktioner.

## Idè

Under afvikling af hver instruktion på en given mikroarkitektur gennemgår instruktionen forskellige faser. Et afviklingsplot angiver tidspunktet for hver væsentlig fase en instruktion gennemløber. Instruktionsstrømmen angives yderst til venstre, oppefra og ned. Tiden angives i clock-perioder fra venstre mod højre.

### Eksempel: Enkelt-cyklus mikroarkitektur

Her er for eksempel afviklingen af 4 instruktioner på en enkelt-cyklus (single cycle) mikroarkitektur

```
                0123
movq (r10),r11  X
mulq $100,r11    X
movq r1,(r10)     X
subq $1,r10        X
```

Her er kun en enkelt fase, kaldet `X` for eXecute, da alle instruktioner kan udføres på en enkelt clock-periode. Vi har altså den sekventielle model, som den vi forstår når vi læser et assembler program; først indlæser vi noget fra hukommelsen, derefter lægger vi en værdi til dette, hvorefter vi skriver det tilbage til hukommelsen.

Hvis vi ønsker at finde denne arkitekturs CPI, kan man gøre dette ved at tælle antallet af instruktioner; i ovenstående tilfælde altså 4 clock perioder for 4 instruktioner, eller en CPI på 1.

# CPI

Vi vil hele vejen gennem bruge CPI, som et mål for ydelsen af et program. CPI står for Clocks cycles Per Instruction, altså et mål for hvor mange clock perioder en udførsel af et specifikt program tager. Det kan derfor beregnes ved at tælle hvor mange clock perioder det tager for sidste instruktion at blive afsluttet og dele det med antallet af instruktioner. For overstående er det 4 clock perioder delt med 4 instruktioner; altså `CPI = 1`, hvilket måske ikke er overraskende.

Det er dog vigtigt at notere præcist hvad dette betyder. For det første siger det meget lidt om hvordan en specifik arkitektur generelt opfører sig. Det udregner CPI for udførslen af et specifikt program, som måske kan representere en klasse af programmer, men man er nødt til at have meget stor benchmark af mange forskellige programmer for at sige noget generelt om mikroarkitekturen.

For det andet skal man være påpasselig med sammenligne CPI for et specifikt program mellem forskellige arkitekturer. CPI nævner f.eks. ikke noget om længden på clock perioden. Vi vil stadig gøre det, men er altid nødt til at have overstående med i vores overvejelse.

Det er også vigtigt et noteret at de plots vi laver starter på hvad vi kan kalde en "kold" maskine. Dvs. vores plots vil ikke have nogen instruktioner til at ligge i pipelinen i forvejen. Det kan betyde at især korte programmer vil have en bedre opførsel end hvis vores pipeline var fyldt i forvejen.