CompSys 2022-23 Recap

Pipeline og ILP

Asbjørn Munk
asmu@di.ku.dk

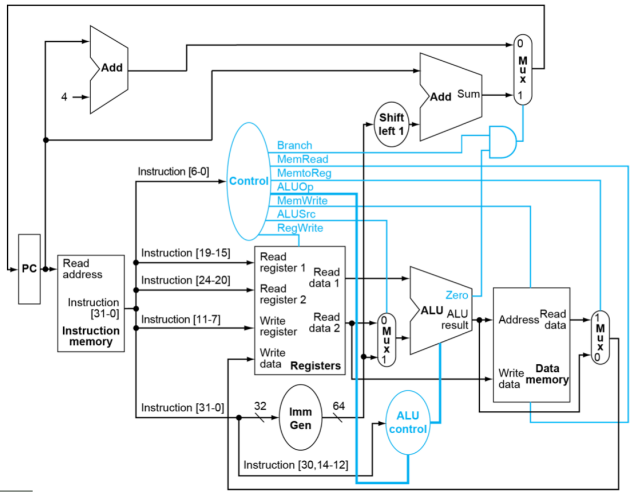
KØBENHAVNS UNIVERSITET

Plan

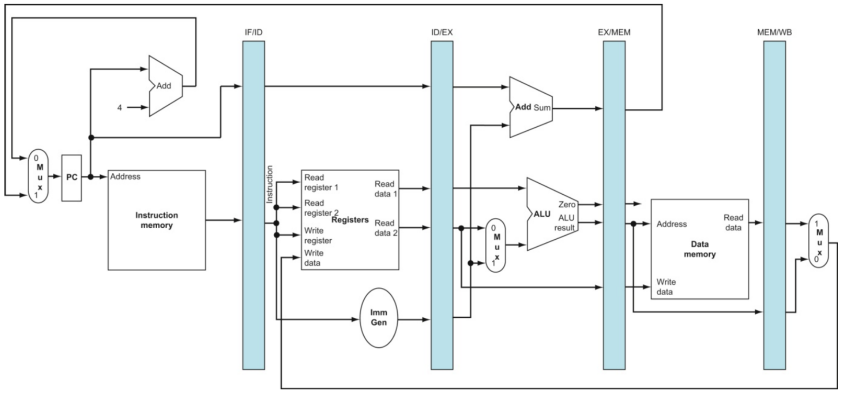
De næste 45 minutter

- Gennemgå lignende pipeline opgaver fra re-eksamen 2021, men med RISC-V kode.
- I må gerne stille spørgsmål løbende.
- Forbehold for mindre fejl. Hjælpe mig med at finde dem!

En simpel maskine



Pipelined mikroarkitektur



Vi betragter følgende RISC-V kode

```
L1: bge   x10, x12, end
    lw    x14, 0(x11)
    addi  x11, x11, 4
    sw    x14, 0(x10)
    addi  x10, x10, 4
    jal   L1
L1: bge   x10, x12, end
    lw    x14, 0(x11)
    addi  x11, x11, 4
    sw    x14, 0(x10)
    addi  x10, x10, 4
    jal   L1
```

Antagelser

Medmindre andet er angivet forudsætter følgende spørgsmål at:

- Alle hop er korrekt forudsagt.
- Al adgang til hukommelsen resulterer i et cache-hit.

Hvorfor er det sidste vigtigt?

Simpel 5 step pipelined

Question: Make an execution diagram, showing the execution of above code on a **simple 5-step pipelined machine**, as presented in the course note. Explain shortly any assumptions you may have to make.

Simpel 5 step pipelined

Question: Make an execution diagram, showing the execution of above code on a **simple 5-step pipelined machine**, as presented in the course note. Explain shortly any assumptions you may have to make.

Antagelser:
Instruktionsudførsel:

```
alle: "Fe De Ex Mm Wb"
```

Ressourcer:

```
Fe: 1, De: 1, Ex: 1, Mm: 1, Wb: 1
```

Simpel 5 step pipelined

```
                          0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17
L1: bge   x10, x12, end
    lw    x14, 0(x11)
    addi  x11, x11, 4
    sw    x14, 0(x10)
    addi  x10, x10, 4
    jal   L1
L1: bge   x10, x12, end
    lw    x14, 0(x11)
    addi  x11, x11, 4
    sw    x14, 0(x10)
    addi  x10, x10, 4
    jal   L1
```

Simpel 5 step pipelined

```
                          0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17
L1: bge   x10, x12, end   Fe De Ex Mm Wb
    lw    x14, 0(x11)           Fe De Ex Mm Wb
    addi  x11, x11, 4              Fe De Ex Mm Wb
    sw    x14, 0(x10)                 Fe De Ex Mm Wb
    addi  x10, x10, 4                    Fe De Ex Mm Wb
    jal   L1                                Fe De Ex Mm Wb
L1: bge   x10, x12, end                        Fe De Ex Mm Wb
    lw    x14, 0(x11)                              Fe De Ex Mm Wb
    addi  x11, x11, 4                                 Fe De Ex Mm Wb
    sw    x14, 0(x10)                                    Fe De Ex Mm Wb
    addi  x10, x10, 4                                       Fe De Ex Mm Wb
    jal   L1                                                   Fe De Ex Mm Wb
```

Hvad er IPC?

2-vejs superskalar

Question: Make an execution diagram, showing the execution of above code on a **2-way superscalar machine**, as presented in the course note. Explain shortly any assumptions you may have to make.

2-vejs superskalar

Question: Make an execution diagram, showing the execution of above code on a **2-way superscalar machine**, as presented in the course note. Explain shortly any assumptions you may have to make.

Antagelser:

```
load:  "Fe De Ag Mm Wb"
store: "Fe De Ag Mm"
andre: "Fe De Ex Wb"
```

```
Fe:2, De:2, Ex:2, Ag:1, Mm:1, Wb:2
```

Bemærk vi kun har en `Ag` og en `Mem`, da vi kun tillader en cache-tilgang pr. clock cykel. Husk kun en instruktion *for hvert register* fra `Ex` og frem.

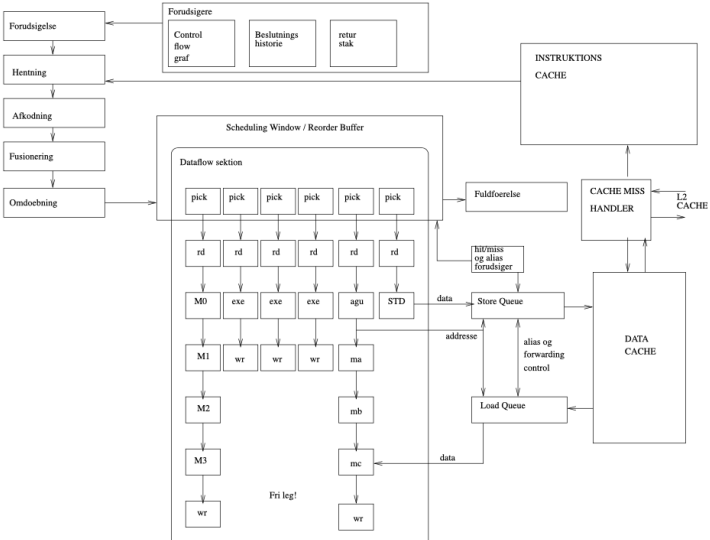
2-vejs superskalar

```
                          0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20
L1: bge   x10, x12, end
    lw    x14, 0(x11)
    addi  x11, x11, 4
    sw    x14, 0(x10)
    addi  x10, x10, 4
    jal   L1
L1: bge   x10, x12, end
    lw    x14, 0(x11)
    addi  x11, x11, 4
    sw    x14, 0(x10)
    addi  x10, x10, 4
    jal   L1
```

2-vejs superskalar

```
                          0  1  2  3  4  5  6  7  8  9  10 11 12
L1: bge   x10, x12, end   Fe De Ex Wb
    lw    x14, 0(x11)           Fe De Ag Mm Wb
    addi  x11, x11, 4           Fe De Ex Wb
    sw    x14, 0(x10)              Fe De Ag Mm
    addi  x10, x10, 4              Fe De Ex Wb
    jal   L1                          Fe De Ex Wb
L1: bge   x10, x12, end               Fe De Ex Wb
    lw    x14, 0(x11)                       Fe De Ag Mm Wb
    addi  x11, x11, 4                       Fe De Ex Wb
    sw    x14, 0(x10)                          Fe De Ag Mm
    addi  x10, x10, 4                          Fe De Ex Wb
    jal   L1                                      Fe De Ex Wb
```

Out-of-order


Forudsigelse
Hentning
Afkodning
Fusionering
Omdoebning
Forudsigere
Control flow graf
Beslutnings historie
retur stak
INSTRUKTIONS CACHE
Scheduling Window / Reorder Buffer
Dataflow sektion
pick
rd
M0
exe
agu
STD
M1
wr
ma
M2
mb
M3
mc
Fri leg!
Fuldfoerelse
CACHE MISS HANDLER
L2 CACHE
hit/miss og alias forudsiger
data
Store Queue
addresse
alias og forwarding control
DATA CACHE
Load Queue

2-vejs out-of-order

Question: Make an execution diagram, showing the execution of above code on a **2-way out-of-order machine** with 2 instructions in the dataflow section.

2-vejs out-of-order

Question: Make an execution diagram, showing the execution of above code on a **2-way out-of-order machine** with 2 instructions in the dataflow section.

Antagelser:

```
ALU-op: Fa Fb Fc De Fu Al Rn Qu pk rd ex wb Ca Cb
branch: Fa Fb Fc De Fu Al Rn Qu pk rd ex Ca Cb

load:  Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wb Ca Cb
store: Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc -- Ca Cb   // addresse
       -                    Q* -- -- -- -- pk rd st C*      // data

inorder: Fa, Fb, Fc, De, Fu, Al, Rn, Qu, Ca, Cb
outoforder: pk, rd, ex, wb

Resourcer: Fa:2, Fb:2, Fc:2, De:2, Fu:2, Al:2, Rn:2 [Qu-Ca]:2, Ca:2, Cb:2
```

Bemærk! Cache tager nu 3 clock cyckler at tilgå.

2-vejs out-of-order

```
                          0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20
L1: bge   x10, x12, end
    lw    x14, 0(x11)
    addi  x11, x11, 4
    sw    x14, 0(x10)
    addi  x10, x10, 4
    jal   L1
L1: bge   x10, x12, end
    lw    x14, 0(x11)
    addi  x11, x11, 4
    sw    x14, 0(x10)
    addi  x10, x10, 4
    jal   L1
```

2-vejs out-of-order

```
                          0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22
L1: bge   x10, x12, end   Fa Fb Fc De Fu Al Rn Qu pk rd ex Ca Cb
    lw    x14, 0(x11)        Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wb Ca Cb
    addi  x11, x11, 4        Fa Fb Fc De Fu Al Rn Qu pk rd ex wb -- -- -- Ca Cb
    sw    x14, 0(x10)           Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc -- Ca Cb
                                                     Q* -- -- -- -- pk rd st C*
    addi  x10, x10, 4           Fa Fb Fc De Fu Al Rn Qu pk rd ex wb -- -- -- Ca Cb
    jal   L1                       Fa Fb Fc De Fu Al Rn Qu pk rd ex wb -- -- -- Ca Cb
L1: bge   x10, x12, end            Fa Fb Fc De Fu Al Rn Qu pk rd ex -- -- -- -- Ca Cb
    lw    x14, 0(x11)                 Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wb Ca Cb
    addi  x11, x11, 4                 Fa Fb Fc De Fu Al Rn Qu pk rd ex wb -- -- -- Ca Cb
    sw    x14, 0(x10)                    Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc -- Ca Cb
                                                              Q* -- -- -- -- pk rd st C*
    addi  x10, x10, 4                    Fa Fb Fc De Fu Al Rn Qu pk rd ex wb -- -- -- Ca Cb
    jal   L1                                Fa Fb Fc De Fu Al Rn Qu pk rd ex wb -- -- -- Ca Cb
```

Bemærk at commit-fasen sker in-order!