

Godkendelsesopgave 3

Pipeline-arkitektur

Maskinarkitektur, blok 1 2011

Introduktion

Dette er den tredje godkendelsesopgave på DIKUs bachelorkursus *Maskinarkitektur*. Den består af 2 delopgaver. Opgaven skal udarbejdes i grupper af 2-3 personer, afvigelser herfra kræver tilladelse fra jeres instruktør.

Aflevering

Opgaven skal afleveres af en repræsentant fra gruppen på Absalon senest onsdag d. 26 oktober kl 23:55. Der skal afleveres Logisim-filer som løser opgaven samt en rapport (i PDF-format), der dokumenterer jeres arbejde. Rapporten kan være på enten dansk eller engelsk.

Hjælp til opgaven

Alle spørgsmål vedrørende opgaven bedes stilles på kursets forum på Absalon for at undgå forfordeling mellem kursusedtagerne. Husk at benytte jer af øvelsestimerne, hvor opgaven vil blive gennemgået.

G3.1 Pipeline-arkitektur

I denne delopgave skal jeres enkeltcyklusarkitektur fra G1 videreudvikles så den understøtter *pipelining*. Datavejen i pipeline-arkitekturen skal understøtte *forwarding*, *stalling* og *flushing* for at kunne håndtere *hazards* (se figur 4.60, s. 375 i [1]). Hop-instruktionerne skal implementeres som følger

- *j*: Hop udføres i ID. Instruktionen i IF slettes.
- *jal*: Hop udføres i ID. Skrivning til registerbank sker i WB. Instruktionen i IF slettes.
- *jr*: Hop udføres i EX. Instruktioner i IF og ID slettes.
- *beq*: Hop udføres i EX. Instruktioner i IF og ID slettes hvis hoppet tages. Bemærk at betingede hop i tekstbogen foregår i ID - se bort fra dette.

Pipeline-arkitekturen skal ikke benytte *delay slots*. Dvs. at der skal altid bruges *stalling* til at undgå kontrol-hazards.

Udleveret skelet

For at hjælpe jer i gang medfølger et mikroarkitekturskelet i form af en fungerende enkeltcyklusarkitektur (svarende til løsningen af G2.1). Det står jer frit at ændre skelettet efter behag eller at bygge videre på jeres egen løsning af G2.1. Bemærk at programhukommelseskomponenten indeholder et program, der afprøver korrektheden af jeres løsning. Afprøvningen er dog ingenlunde udtømmende og det kan derfor være en god idé at I selv foretager yderligere kontrol.

BEMÆRK at skelettet først udleveres onsdag d. 12. oktober ved midnatstid, pga. G2-genafleveringsdeadlinen.

Råd og vink

- Bemærk at der i tekstbogen s. 369 er en fejl i udtrykket for hvornår der skal forwardes fra MEM/WB. Det rigtige udtryk er

```
if(MEM/WB.RegWrite
    && (MEM/WB.RegisterRd != 0)
    && !(EX/MEM.RegWrite
        && (EX/MEM.RegisterRd != 0)
        && (EX/MEM.RegisterRd == ID/EX.RegisterRs))
    && (MEM/WB.RegisterRd == ID/EX.RegisterRs)) ForwardA = 1;

if(MEM/WB.RegWrite
    && (MEM/WB.RegisterRd != 0)
    && !(EX/MEM.RegWrite
        && (EX/MEM.RegisterRd != 0)
        && (EX/MEM.RegisterRd == ID/EX.RegisterRt))
    && (MEM/WB.RegisterRd == ID/EX.RegisterRt)) ForwardB = 1;
```

- Stalling kan opnås ved at sætte Enable-indgangen til 0 på registerkomponenten. Dermed lader registret sig ikke opdatere selvom klokken slår ud.
- Registerbanken skal sættes til at trigge på *falling edge*. Dette gør at skrivning til registerbanken sker i første halvdel af clock-cyklen, således at den skrevne værdi kan udlæses i anden halvdel af clock-cyklen. Alternativt skal der implementeres et ekstra forwarding path rundt om registerbanken.
- Gør en indsats for at strukturere jeres kredsløb. Det betaler sig i det lange løb når kredsløbet skal afluses for fejl og efterhånden som mikroarkitekturen udvides.

G3.2 Hopforudsiger

Udvid mikroarkitekturen beskrevet i G3.1 med en hopforudsiger. Mikroarkitekturen fra G3.1 er ikke særlig hurtig til at afvikle betingede hop (beq). Betingede hop afgøres i EX-trinnet, men forudsiges altid 'ikke-taget', idet pipelinen fortsætter med at behandle de efterfølgende instruktioner efter

hoppet, indtil det afgøres. Hvis hoppet i EX afgøres som taget, fjernes de to instruktioner i IF og ID.

Antager vi at 60% af alle hop tages bliver den gennemsnitlige CPI for betingede hop på 2.2 i G3.1-mikroarkitekturen. I G3.2 skal vi tilføje en hopforudsiger som hjælper på to måder.

1. Tagne hop kan behandles ligeså hurtigt som ikke tagne hop.
2. Hoppene kan forudsiges bedre end i G3.1.

Antager vi at forudsigeren kan forudsige 80% af alle betingede hop korrekt bliver den gennemsnitlige CPI for betingede hop 1.4 for den forbedrede mikroarkitektur.

Hopforudsigelse

Udvidelsen skal ske dels i IF, hvor vi forudsiger hop, dels i EX, hvor vi afgør hop. I IF tilføjes en *hopforudsigelsestabel*. Det er et lager med 32 celler indeholdende forudsigelsesinformation. Bit 2-6 i PC skal bruges til at indexere hopforudsigelsestabellen. Ved læsning fra hopforudsigelsestabellen fås tre værdier:

Tag	Et 25-bit tal som skal matche bit 7-31 i PC'en. Hvis Tag'en matcher har vi en gyldig forudsigelse.
P	To bit der koder for de fire mulige forudsigelser: 00 = WNT (weakly not taken) 01 = SNT (strongly not taken) 10 = WT (weakly taken) 11 = ST (strongly taken)
Target	Adressen der skal hoppes til, hvis instruktionen forudsiges taget.

Hopforudsigelsestabellen tilgås parallelt med læsningen fra programlageret i IF-trinnet. Forudsigelserne bruges til at ændre PC i IF-trinnet. Ændringen bestemmes af

```
if (Tag == PC
    && (P == WT || P == ST)
    && !changePC) PC = Target;
```

, hvor !changePC angiver at ingen trin senere i pipelinen signalerer ændring af PC.

Hopafgørelse

Betingede hop afgøres i EX som beskrevet i G3.1. Hvis et hop er fejlagtigt forudsagt skal instruktioner i IF og ID slettes som i G3.1. Efter afgørelse af et hop skal hopforudsigelsestabellen opdateres med Tag, P og Target. Beslutningen herom og opsætningen af skrivningen skal ske i EX. I samme ombæring skal

P ændres, så forudsigelsen ændrer sig efter udfaldet af det betingede hop. P kan kun skifte til en naboltilstand i rækkefølgen SNT, WNT, WT, ST eller blive stående hvis forudsigelsessikkerheden ikke kan blive stærkere.

Udleveret skelet

I skal arbejde videre på jeres løsning af G3.1. Det udleverede skelet til G3.1. indeholder en hopforudsigelsestabel som I skal benytte. Desuden medfølger et ikke-udtømmende testprogram.

Råd og vink

- For at kunne opdatere hopforudsigelsestabellen i EX skal I huske at opsamle velvalgte værdier fra IF og lade dem følge instruktionerne gennem pipelinen frem til EX.
- Det antages at hopforudsigelsestabellen kan både læses og skrives i samme cyklus (ligesom for registerbanken).

Litteratur

- [1] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design, Fourth Edition, Fourth Edition: The Hardware/Software Interface*. Morgan Kaufmann Publishers, 4th edition, 2008.