



G2 rapport

Mips pipeline Simulator i C

Simon van Beest

Kristian Høi

Mikkel Trolle Enevoldsen

October 6, 2015



1 Mips-simulator

Vi har skrevet en MIPS pipeline simulator, som understøtter følgende funktioner: add, addu, addi, addiu, and, andi, beq, bne, j, jal, jr, lw, lui, nor, or, ori, sll, slt, sltu, slti, sltiu, srl, sub, subu, sw, syscall. Når simulatoren har simuleret en given mips assembly kode, vil vise resultaterne i terminalen ved at printe givne register variable til terminalen og hvor mange instruktioner den kørte.

Design decisions

En beslutning vi tog var angående hvilken værdi, der skulle returneres ved fejl. Vores program har fem fejl koder: 1,2,3,4,5.

1 hvis man giver for få argumenter til simulatoren ved kørsel

2 hvis simulatoren ikke kunne læse .cfg filen.

3 og 4 hvis simulatoren ikke understøtter den givne assembly kode.

5 hvis SYSCALL bliver kaldt.

Vi har valgt at implementere vores i type instruktioner ved at benytte vores r type instruktioner. Ved et if-else statement skelner vi mellem immediates og rt værdier, dettes gøres i alu(). Med lui har vi dog valgt at indføre et særligt case, 'case 16', for denne i type instruktion.

Tests

Vi har skrevet en test til hver funktion vores simulator. Disse test ligger i mappen asm. Testen til hver funktion hedder det samme funktionen. Så testen til addu hedder addu.S osv. Da vores simulator ikke understøtter hazards, har vi mange nop operationer. De enkelte tests køres hver for sig som i nedenstående eksempel, og vi observerer hvorvidt de enkelte tests giver det forventede resultat. Det forventede resultat ligger som en kommentar i hver .S. Eksempelvis tester vi herunder srl:

```
archimedes@syracuse:/media/sf_Ark/2nd$ ./sim default.cfg asm/srl.elf
Executed 5 instruction(s).
Executed 6 cycle(s).
PC = 0x40002c
at = 0x0
v0 = 0xa
v1 = 0x0
t0 = 0x1
t1 = 0x1d
t2 = 0x0
t3 = 0x2
t4 = 0x29
t5 = 0x2a
t6 = 0x0
t7 = 0x0
sp = 0x4a0000
fp = 0x0
ra = 0x0
archimedes@syracuse:/media/sf_Ark/2nd$
```

Her får vi det forventede resultat, og således virker addi. Ligeledes er dette gældende for resten af instruktionerne.