G2
Maskinarkitektur
Efterår 2011

Jens Fredskov
Naja Mottelson
Søren Pilgård

2. oktober 2011

Indhold

1 Indledning

Denne rapport dokumenterer gruppens arbejde med anden godkendelsesopgave i kurset Maskinarkitektur. Den indeholder, for såvidt vi kan se, korrekte besvarelser af begge underopgaver.

2 g2-1

Vores implementation af enkeltcykluskredsløbet følger i store træk samme mønster som den beskrevet i lærebogen. Udover at forlænge ALUop-wiren fra kontrollen til ALUkontrollen har vi dog tilføjet tre nye kontrolbits, som kort beskrives i figur 1

Navn	Effekt når lav	Effekt når høj
SignExtend	immediate-feltet fortegnudvides 16 pladser.	immediate-feltet nuludvides 16 pladser.
j	Ingen	PC'en sættes til (PC+4[31:28], address, 2'b0)
jal	Ingen	Register 31 sættes PC+4

Figur 1: Nye kontrolbits

De instruktioner der gennemgåes grundigt i lærebogen (lw, sw, beq) har vi implementeret efter samme algoritme der præsenteres dér. Vi vil derfor ikke bruge yderligere tid på at gennemgå dem udførligt her, udover at angive hvilke bits der er sat i kontrollen for de forskellige instruktioner:

lw: MemRead, MemToReg, ALUsrc, RegWrite, SignExtend.

sw: MemWrite, ALUsrc, SignExtend.

beq: Branch, SignExtend, ALUop = 101 (signalerer til ALU'en om at udføre subtraktion).

3 addu, subu, and, or, nor, slt, jr

Alle R-instruktioner har vi implementeret på grundlæggende samme måde: I kontrollen sættes Write-registeradressen til rd (vha. RegDst) og RegWrite sættes så det er muligt at skrive til registrene. Resten af kontrollogikken foretages i ALU-kontrollen som modtager en 3-bit wire indeholdende ALUop-signalet fra kontrollen samt de 6 bits der hentes ud fra operationens funct-felt. Ud fra dette input genererer ALU-kontrollen et output på 6 bits, hvoraf de fire mindst betydende sendes til ALU'en som signal om hvilken af de aritmetisk-logiske instruktioner den skal udføre, hvorefter outputtet fra ALU'en skrives til registerbanken. I tilfældet af de grundlæggende R-instruktioner benyttes de to sidste bits i ALU-kontrollens output ikke.

I tilfælde af instruktionen jr benyttes ALU'en ikke. Det eneste kredsløbet foretager sig ved denne instruktion er at sende et signal til en MUX der vælger imellem at skrive den standard-inkrementerede adresse til PC'en (PC+4 i dette tilfælde) eller outputtet fra register rs.

4 sll, srl, sra

Da skifteoperationerne ligeledes er R-instruktioner er de implementeret med samme kontrollogik som ovenfor. Her sættes den fjerde bit i ALU-kontrollens output dog, og gives som selector-input til en MUX der giver shamt som den første operand til ALU'en i stedet for register rs.

5 addiu, andi, ori, slti

I-instruktionerne er implementeret på samme facon som R-instruktionerne med de undtagelser at der skrives til register rt samt at ALUsrc-bitten sættes i kontrollen. ALUsrc gives som selector-input til en MUX som vælger hvilket input der gives til ALU'ens anden operand. Når ALUsrc er sat vælges operationens immediate-felt, som er blevet enten nul- eller fortegnsudvidet afhængigt af instruktionen. Nul- eller fortegnsudvidelse specificeres ved kontrollens SignExtend-bit.

6 j, jal

Som antydet i indledningen er J-instruktionerne implementeret med deres 'egne' bits i kontrollen, hhv. j og jal. Når j-instruktionen udføres er denne bit den eneste der er sat. Signalet herfra går til en MUX der vælger imellem at skrive den standard-inkrementerede adresse til PC'en eller den der specificeres af instruktionens adressefelt.
Ved jal bliver samme bit som ved j-instruktionen sat således at der hoppes til en given adresse fra adressefeltet. Endvidere sættes jal-bitten som vha. en MUX vælger konstanten 31 (1f i hexadecimal) og sender dette til rW, hvilket er det register der skal skrives til. Derudover vælger jal-bitten også (igen vha. en MUX) mellem at sende det normale input og PC+4 til W, altså den data der skal skrives. Slutteligt sættes regWrite også således at skrivningen af PC+4 til register 31 bliver udført.

7 g2-2

Vores implementation af underopgave 2 er at finde i den vedlagte fil g2-2.asm. Som dokumentation for denne henviser vi til de kommentarer der er anført i kildekodefilen selv.
Endvidere kan siges at koden giver de rigtige resultater (19 og 111 i decimal, dette samme som det udleverede C-program) både i MARS og på vores kredsløb (omend det sidste tager nogen tid).