

INF2270 – Obligatorisk oppgave 2

”En mini-CPU”

Hovedmål

I denne obligatoriske oppgaven skal dere bruke Logisim til å implementere og teste en liten CPU som bruker 4 maskinkode instruksjoner.

Oppgaven

Dere skal bruke en arkitektur som har separat programminne og dataminne (kalles Harvard Architecture). Det å bruke Harvard Arkitektur med dedikert instruksjonsbus og små raske minneelementer har fordeler med at man trenger egentlig ikke en instruksjons register (IR), men kan bruke utgangen (output) av instruksjonsminne direkte.

- For minneelementene kan dere bruke RAM. Vi skal kun bruke 16 adresser (altså 4-bit) av minneelementet.
- Instruksjonene utføres sekvensielt. Dette vil medføre en rekke konsekvenser, der i blant:
 - At vi kan bruke en teller (counter) som instruksjonspeker/program teller (IP/PC) som teller opp hele tiden uten avbrytelser (interrupt). Konstruer telleren på en slik måte at en pulse switch kan resette telleren og bruk kombinatorisk logikk for å unngå at telleren teller til mer enn 16.
 - Denne telleren skal dere implementere selv med en 4-bit adder og et register.
 - Ettersom alle instruksjonene blir ferdig innen en klokkesykel så trenger vi ikke å ha en tilstandsmaskin som blir trigget av instruksjoner. Enkel kombinatorisk logikk vil holde i dette tilfellet.
- Det holder med et 8-bit data register til CPUen. Vi kaller den for R.
- Det skal være en single system bus (merket som S7....S0) i tillegg til instruksjonsbus(en) som kobler sammen data minne, (deler av) instruksjons-bus(en) og utgangen (output) av R til inngangen av dataregisteret R.
 - Dere må sørge for at det kun er en aktiv kilde (source) som driver hver enkelt linje (line) av denne bus(en), slik at alle kilder som potensielt kan drive en linje er tilkoblet en ”tri-state”.
 - For å koble instruksjons bus(en) og utgangen av R til inngangen av R så burde dere bruke to sett av 4-bit tri-state buffere. Komponenten ‘Controlled Buffer’ kan brukes til dette.
- De fire 8-bit instruksjonene er beskrevet i tabellen under. ”DDDD” i maskin koden refererer til ”data”, mens ”AAAA” er data i minneadressen.

Forklaring	Maskinkode
Load de 4 LSB(ene) av R med DDDD	1000DDDD
Load de 4 MSB(ene) av R med DDDD	1001DDDD
Load R fra dataminne gitt adressen AAAA	1010AAAA
Write R til dataminne gitt adressen AAAA	1011AAAA
Alle andre operasjonskoder skal være NOP	(no operation = ikke gjør noe)

Dere kan load(e) programmer med inntil 16 instruksjoner i programminnet. Høyreklikk, så velg 'Load image...'.

Retningslinjen/konvensjonen er at de laveste adressene er to bytes (hexadesimale) helt til venstre, med de 4 MSB(ene) til høyre og de 4 LSB(ene) til venstre. Slik at følgende linje:

```
v2.0 raw
85 97 B1 81 94 A1 00 00
00 00 00 00 00 00 00 00
```

blir eksekvert slik:

```
85
97
B1
81
94
A1
```

HINT! I forhold til å kunne fullføre alle instruksjonene innen en klokkesykel, burde dere bruke falling edge av klokkesignalet til å trigge loading(en) av data registeret R.

Testprogram

Program 1:

```
v2.0 raw
85 97 B1 81 94 A1 00 00
00 00 00 00 00 00 00 00
```

Resultat:

```
v2.0 raw
0 75
```

Program 2:

```
v2.0 raw
8C 9D B0 B3 81 91 A0 B4
00 00 00 00 00 00 00 00
```

Resultat:

Her kan det oppstå et problem om WE er lav gjennom hele klokkeperioden: etter 'write 3' kan WE forbli lav mens instruksjonen allerede skiftes til den neste instruksjon som kan føre til at data ved adressen 1 overskrives også.

Riktig resultat:

```
v2.0 raw
dc 0 0 dc dc
```

Mulig feil resultat:

```
v2.0 raw
dc dc 0 dc dc
```

Verktøyet

Dere har nå hatt anledning til å knytte bekjentskap til Logisim i Obligatorisk oppgave 1. Hvis dere ønsker mer dokumentasjon kan dette finnes her:

<http://ozark.hendrix.edu/~burch/logisim/docs/current/guide/tutorial/index.html>

Innlevering

Innleveringen skal skje gjennom en skriftlig rapport hvor dere inkluderer logging av kretsen ved kjøring av testprogrammene.

OBS! Vær nøye med å sette label på relevante innganger/utganger.

Viktig at dere også leverer inn design filen (.circ filen)

Personlig vil jeg oppfordre dere alle til å gjøre innleveringen så oversiktlig som mulig slik at dere kan få gode tilbakemeldinger fra deres gruppelærer. Det er også viktig at dere følger retningslinjene deres gruppelærer gir.

Lykke til!