

## Oblig 2 - INF2270

Rune Hovde  
Brukernavn: runehovd

April 10, 2016

# Oppgaven

I denne obligen var oppgaven å lage en CPU vha Logisim som kunne lese og skrive til.

## Counter

På counteren brukte jeg en 4-bits adder og et register. Den er kobla opp sånn at outputen til registeret er koblet via en adder, som legger til en bit hver gang, som går inn i inngangen til registeret igjen. Hvis counteren teller opp til 15, vil "enable"-inngangen på registeret skru seg av i ett klokkesignal, og kretsen vil automatisk resette seg til 0 (0000).

## Programminne

På programminnet bruker jeg en RAM med "Data Interface" = One synchronous load/store point. Det er inn her vi laster inn instruksjonene. minnet får informasjon om hvilken seksjon den skal lese fra av counteren.

Testprogrammene vi fikk var slik:

1. 85 97 b1 81 94 a1 00 00 00 00 00 00 00 00 00
2. 8c 9d b0 b3 81 91 a0 b4 00 00 00 00 00 00 00 00

## Dataminne

Som dataminnet brukte jeg også en RAM, men denne gangen er "Data Interface" = Separate load and store point.

Resultatet vi får fra test-programmene:

1. 00 75
2. dc 00 00 dc dc

## Registeret

Inne i "Logic"-circuiten min ligger det et register. Der mellomlagrer cpuen data, og gjør mellomoperasjoner før det skal lagres til minnet.

## Dekoder

Jeg har en dekode, som tar inn tallene fra programminnet, og analyserer hva som skal skje. De forskjellige mulige operasjoner som decoderen analyserer:

1. Input har minterm 8: LSB
2. Input har minterm 9: MSB
3. Input har minterm 10: Load R
4. Input har minterm 11: Write R

Grunnen til at jeg har dette er for å sørge for at kun 1 source skriver på systembusen samtidig.

Table 1: Testprogram 1

Clock(150,350)	System bus	RAM(930,330)[0]	RAM(930,330)[1]	RAM(930,330)[2]	RAM(930,330)[3]	RAM(930,330)[4]	R
0	0000 1100	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 1100
1	1101 1100	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 1100
0	1101 1100	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	1101 1100
1	1101 1100	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	1101 1100
0	1101 1100	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	1101 1100
1	1101 1100	1101 1100	0000 0000	0000 0000	0000 0000	0000 0000	1101 1100
0	1101 1100	1101 1100	0000 0000	0000 0000	0000 0000	0000 0000	1101 1100
1	1101 0001	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	1101 1100
0	1101 0001	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	1101 0001
1	0001 0001	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	1101 0001
0	0001 0001	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	0001 0001
1	1101 1100	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	0001 0001
0	1101 1100	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	1101 1100
1	1101 1100	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	1101 1100
0	1101 1100	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	1101 1100
1	xxxx xxxx	1101 1100	0000 0000	0000 0000	1101 1100	1101 1100	1101 1100

Table 2: Testprogram 2

Clock(150,350)	System bus	RAM(930,330)[0]	RAM(930,330)[1]	RAM(930,330)[2]	RAM(930,330)[3]	RAM(930,330)[4]	R
0	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
1	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
0	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 1100	0000 1100
1	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 1100	0000 1100
0	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	1101 1100	1101 1100
1	1101 1100	0000 0000	0000 0000	0000 0000	0000 0000	1101 1100	1101 1100
0	1101 1100	0000 0000	0000 0000	0000 0000	0000 0000	1101 1100	1101 1100
1	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	1101 1100	1101 1100
0	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	1101 1100	1101 1100
1	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	1101 1100	1101 1100
0	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	1101 0001	1101 0001
1	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	1101 0001	1101 0001
0	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	0001 0001	0001 0001
1	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	0001 0001	0001 0001
0	1101 1100	0000 0000	0000 0000	1101 1100	0000 0000	1101 1100	1101 1100
1	1101 1100	0000 0000	0000 0000	1101 1100	1101 1100	1101 1100	1101 1100