

INF2270 Oblig2

Tormod Brændshøi/tormobr

April 4, 2017

I denne oppgaven skulle vi lage en mini cpu. Dette løste jeg ved hjelp av en counter, ett sett med programminnet, dataminnet, et register og logikk.

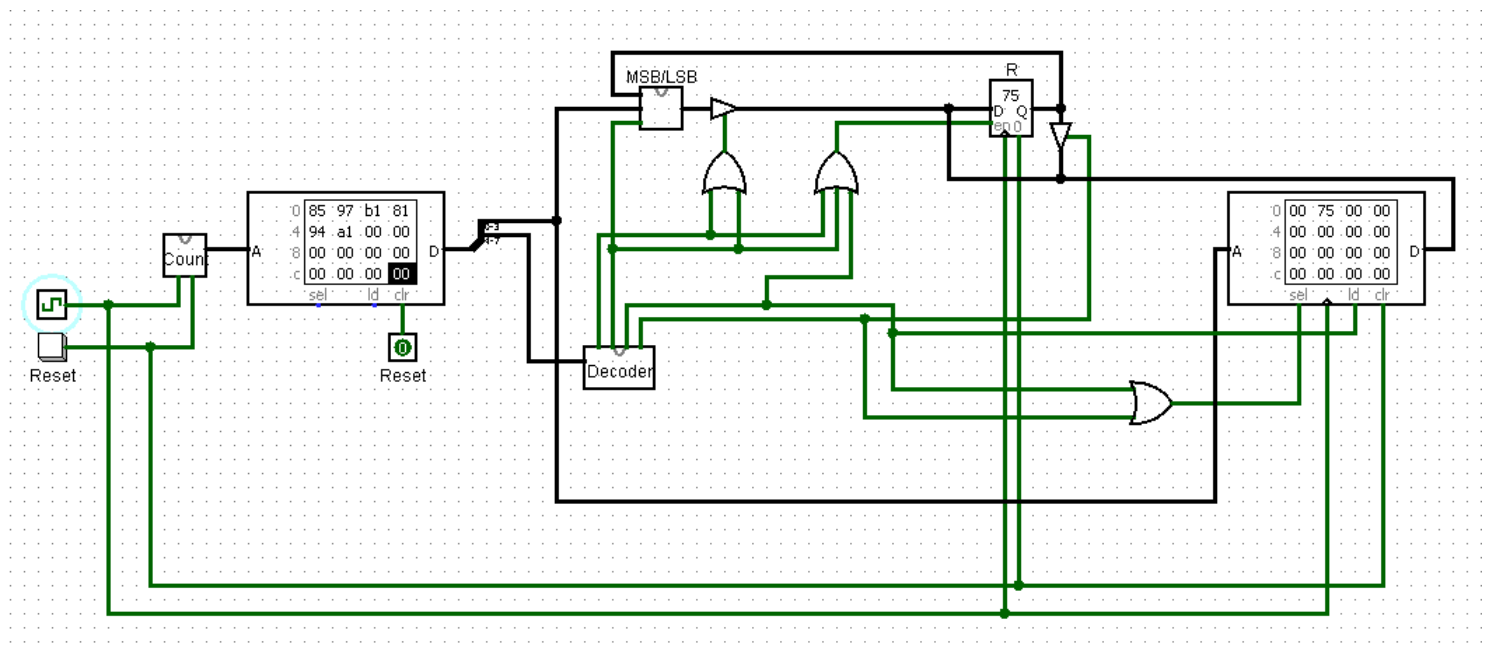
Hovedkomponentene i løsningen min er:

- Counter(En teller som fungerer som en adressepeker til programminnet)
- Programminnet(RAM hvor instruks lagres)
- Dekoder(Instruksene dekodes)
- LSB/MSB(Velger LSB eller MSB til å erstatte innholdet i registeret)
- Register
- Dataminnet(RAM som vi skriver til, og loader fra)

Steg for steg hva som skjer:

1. En counter økes for hvert klokkesignal. Når telleren øker beveger vi oss gjennom instruksene som ligger i RAM.
2. Når vi når en ny instruks i RAM sendes den videre. Vi bruker så en splitter hvor det ene signalet går inn i en dekode, og det andre går til LSB/MSB og inn som adresse-peker i dataminnet.
3. I dekode henter vi ut 4 signaler, hvor de ulike signalene er ulike instruks, som er gitt i oppgaven. På instruks 8 og 9(1000 og 1001) skal vi legge til data i registeret, men på instruks A og B(1010 og 1011) skal vi skrive eller lese fra dataminnet. I LSB/MSB henter vi det som er i registeret, og legger til det nye signalet som LSB eller MSB avhengig av instruksene som ble hentet fra dekode. På instruks 8 skal vi erstatte LSB og på 9 skal vi erstatte MSB.
4. Avhengig av instruksene blir det nye signalet/bits sendt inn i registeret, leses fra dataminnet til registeret eller skriver fra registeret til dataminnet.
5. Når vi har gått gjennom alle instruksene stopper telleren, og det skjer ikke noe mer.

På bildet under kan vi se resultatet av å kjøre program 1 fra oppgaven. Vi ser at resultatet blir: 0 75.



Clock(170,360)	Instruction	R	Data in memory
1	1000 1100	0000 0000	xxxx xxxx
0	1001 1101	0000 1100	xxxx xxxx
1	1001 1101	0000 1100	xxxx xxxx
0	1011 0000	1101 1100	1101 1100
1	1011 0000	1101 1100	1101 1100
0	1011 0011	1101 1100	1101 1100
1	1011 0011	1101 1100	1101 1100
0	1000 0001	1101 1100	xxxx xxxx
1	1000 0001	1101 1100	xxxx xxxx
0	1001 0001	1101 0001	xxxx xxxx
1	1001 0001	1101 0001	xxxx xxxx
0	1010 0000	0001 0001	1101 1100
1	1010 0000	0001 0001	1101 1100
0	1011 0100	1101 1100	1101 1100
1	1011 0100	1101 1100	1101 1100
0	0000 0000	1101 1100	xxxx xxxx
1	0000 0000	1101 1100	xxxx xxxx
0	0000 0000	1101 1100	xxxx xxxx
1	0000 0000	1101 1100	xxxx xxxx
0	0000 0000	1101 1100	xxxx xxxx
1	0000 0000	1101 1100	xxxx xxxx
0	0000 0000	1101 1100	xxxx xxxx
1	0000 0000	1101 1100	xxxx xxxx
0	0000 0000	1101 1100	xxxx xxxx
1	0000 0000	1101 1100	xxxx xxxx
0	0000 0000	1101 1100	xxxx xxxx
1	0000 0000	1101 1100	xxxx xxxx
0	0000 0000	1101 1100	xxxx xxxx
1	0000 0000	1101 1100	xxxx xxxx
0	0000 0000	1101 1100	xxxx xxxx
1	0000 0000	1101 1100	xxxx xxxx
0	0000 0000	1101 1100	xxxx xxxx