IN1020 - oblig1

Mari Gilje Galta

Brukernavn: marigga

Oblig laget av:
Mari Gilje Galta
Universitetet i Oslo (UiO)
Informatikk: Robotikk og Intelligente Systemer

1 Introduksjon

Denne obligen går ut på å implementere en kombinatorisk krets i Logisim som multipliserer et vilkårlig positivt 4-bits binært tall $(A_3A_2A_1A_0)_2$ med $(10)_{dec}$. Obligen er en av fire obliger som må være godkjent for å bestå faget IN1020. For å løse denne oppgaven er tipsene gitt i [1] brukt.

2 Tips

Fremgangsmåten er funnet ved å basere seg på tipsene gitt i oppgaven. De første tipsene er å tenke over hvordan man multipliserer 10-tallsystemet med 10 og 2-tallsystemet med 2. Svaret på begge disse to er at man flytter tallet man har en til venstre og legger på en null som vist nedenfor.

$$\begin{aligned} 1_{dec} \cdot 10_{dec} = 10_{dec} \\ 0001_2 \cdot 0010_2 = 0010_2 \end{aligned} \tag{1}$$

Videre ser vi på den faktoriserte av 10_{dec}:

10	0
5	1
2	0
1	1

Table 1: Den faktoriserte av 10

Som viser at $10_{dec} = 1010_2$. Ved å bruke denne informasjonen sammen med tips nr. 4 (A + A + A + A + A + A + A + A + A + A = 10A), ser vi at det er mulig å skrive multiplikasjon av 10 som $10A = 2A + 8A$. Dette kommer godt med da både $2^1 = 2$ og $2^3 = 8$ er binære tall og har plassering 4 og 2 når 10_{dec} er faktorisert til binært.

Videre er finner man ut at 1111_2 er det høyeste tallet vi kan representere ved 4-bits tall. Ved å multiplisere dette tallet med 10_{dec}, får vi 10010110_2 som består av 8 bits. Av dette kan vi se at vi kommer til å trenge å gi 4 bits inn i kretsen, mens 8 bits skal komme ut av kretsen.

3 Fremgangsmåte

Av tipsene i avsnitt 2 ser vi at vi kan dele opp problemet vårt i to:

1. Multiplisere det binære tallet med $2_{dec} = 0010_2$
2. Multiplisere det binære tallet med $8_{dec} = 0100_2$

Dette gjør vi ved å først multiplisere $(A_3A_2A_1A_0)_2$ med 0010_2 som vist i første rad av tabell 2, og deretter multiplisere tallet med 0100_2 som vist i andre rad i tabell 2. Summen av disse regnes deretter ut for å få sluttproduktet.

Kolonne nr:	8	7	6	5	4	3	2	1
	?	?	?	?				
		0	0	A_3	A_2	A_1	A_0	0
+		A_3	A_2	A_1	A_0	0	0	0
=	?	A_3	A_2	$A_1 + A_3$	$A_0 + A_2$	A_1	A_0	0

Table 2: Utregning av et 4-bits binært tall multiplisert med 10 (8 + 2)

Spørsmålstegnene i regnestykket ovenfor er verdier i mente. De blir enten 0 eller 1, avhengig av veridene til A. Her ser vi at kolonne 1, 2 og 3 er konstanter som man ikke trenger å gjøre noe med. Kolonne 4 er ikke en konstant, men må summeres sammen med A_0 og A_2. Denne kan gi ut en verdi i mente, men mottar ingen mente fra kolonne 3. Kolonne 5, 6 og 7 kan både ta i mot en i mente og gi en i mente. Mens kolonne 8 kun er den siste mente-verdien (0 eller 1).

Da kolonne 6 og 7 legger sammen en verdi A_2/A_3, en null-bit og en i mente er det det samme som å se bort i fra nullen, og kun legge sammen to verdier. Altså har både kolonne 4, 6 og 7 to tall de legger sammen, og gir 0 eller 1 i mente. Dermed kan vi bruke en halvadder på begge disse. Kolonne 5 har tre verdier å legge sammen og gir 0/1 i mente, og trenger dermed en fulladder. Mens kolonne 1,2 og 3 er konstanter. Kolonne 8 tar man ut i fra menten fra kolonne 7.

Halvadder og fulladder er videre forklart i avsnittene 4 og 5. Den fullstendige kretsen er gitt i avsnitt 6.

4 Halvadder

En halvadder er en sub-krets som adderer sammen to bitter. Kretsen har to innganger og to utganger hvorav den ene utgangen er en 0 eller 1 bit i mente.

For å lage denne kretsen trenger man en XOR-port som avgjør om det er en 0 eller 1 bit som skal stå på plassen der bittene sto. Da gir utgangen kun ut 1 hvis èn av bittene er 1. I tillegg trenger man en AND-port som avgjør om det blir 0 eller 1 i mente. Om begge bittene er 0, gir utgangene ut 0 med 0 i mente. Mens om begge er 1, gir utgangene ut 0 med 1 i mente. Mente ut er den øverste utgangen.

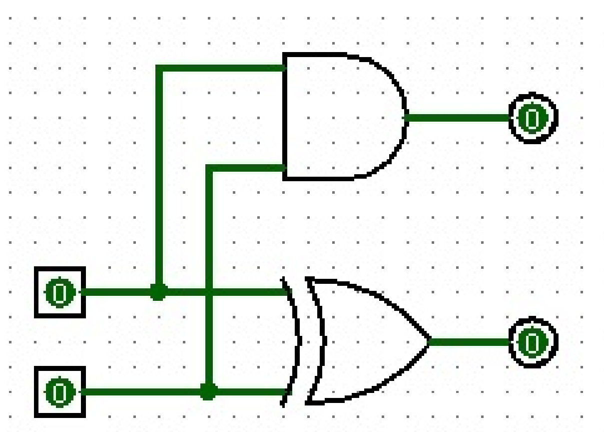

Figure 1: Halvadder i Logisim

5 Fulladder

En fulladder er en sub-krets som adderer sammen to bitter i tillegg til en i mente. Kretsen har altså tre innganger og to utganger hvorav den ene utgangen er en 0 eller 1 bit i mente.

For å lage denne kretsen trenger man tre ulike porter: AND, OR og XOR. I denne kretsen er de to bittene som skal inn i de to nederste inngangene, mens menten er den øverste inngangen. Mente ut er den øverste utgangen.

For å avgjør om det skal stå 0 eller 1 på plassen til bittene som legges sammen, må vi legge sammen både bittene og verdien i mente. Dette gjøres med to XOR-porter som vist i figuren. Da er det både en XOR-port som sjekker verdien til bittene, og lar dette bli null om de har samme verdi, og 1 om de har ulike verdier. Den andre XOR-porten sjekker verdien fra mente i forhold til dette. En AND-port brukes både for å sjekke om begge bittene er like. Om begge er 1 blir utganen lik 1, hvis ikke blir den 0. AND-port blir også brukt for å sjekke utfallet av XOR-porten bak bittene sammenlignet med den i mente. Om begge eller en av disse AND-portene gir ut 1 og ikke 0, så sier OR-porten i kretsen at det skal være 1 i mente ut. Om ikke blir menten 0.

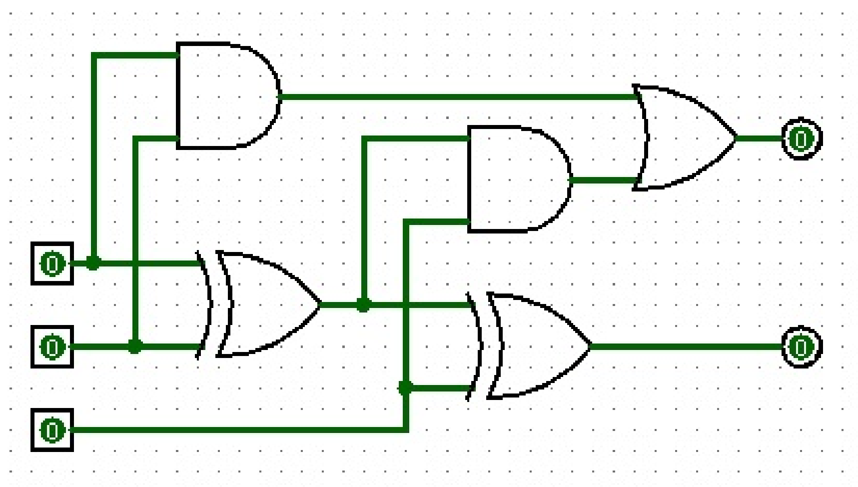

Figure 2: Fulladder i Logisim

6 Ferdig kombinatorisk krets

I figur 3 ser vi den ferdige kombinatoriske kretsen. I figuren har vi valgt å bruke $1111_2 * 10_{dec} = 10010110_2$ som eksempel. Som forklart tidligere tar den inn 4 bits og gir ut 8 bits. Bittene som blir gitt ut står i den rekkefølgen vi hadde skrevet dem på. Bittene som gis inn er referert ved A-verdiene deres.

Som beskrevet i avsnitt 3, er de tre bakerste bittene konstante $(0, A_0, A_1)$. De resterende bittene blir kalkulert ved hjelp av halvaddere og fulladdere som sett i figur 3.

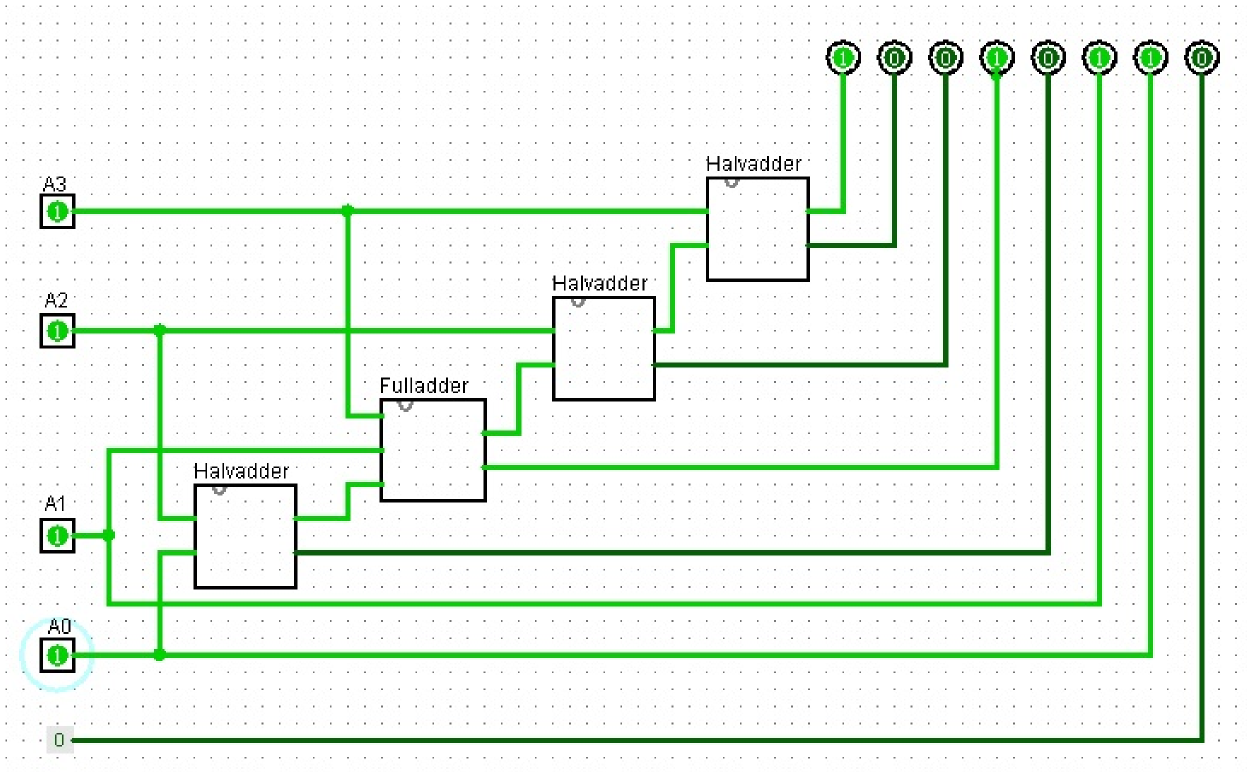

Figure 3: Den kombinatoriske kretsen i Logisim

7 Hva jeg synes om oppgaven

Jeg synes dette var en bra oppgave. Den va svært lærerik og testet studentens evne til å tenke binært enn å tenke slik vi alltid har tenkt før: med desimaltall. Jeg fikk med meg at det var flere som slet med å forstå hvordan de skulle bruke tipsene til å forstå hvordan systemet skulle settes opp. Så det er mulig det hadde komt godt med med litt ekstra øvingstimer for dem. Jeg klarte det pga hjelp og diskusjon med medstudenter. Hadde det ikke hvert for dem, hadde nok jeg også slitt med å utføre oppgaven.

En annen måte man kunne ha løst oppgaven på var å bruke (A + A + A + A + A + A + A + A + A + A = 10A) i stedet for (2A + 8A = 10A). Man kunne også ha laget færre eller flere subkretser.

References

[1] UiO. *IN1020: Obligatorisk oppgave 1*. Sept. 2017. URL: `http://www.uio.no/studier/emner/matnat/ifi/IN1020/h17/timeplan/omid/oblig_multiplikator_del1.pdf`.