

Obligatorisk innlevering #1

INF2270 — Vår 2017

Thor Marius Kokvik Høgås
thor@roht.no & tmhogaas@ifi.uio.no

13. februar 2017

Innhold

1	Fremgangsmåte	1
2	Utrekning	1
3	Egentanker	2
4	Annet	2

1 Fremgangsmåte

Jeg startet først med å lage en krets som kunne gange et 4-bits binært tall med tallet 10. Den enkleste fremgangsmåten her—tenkte jeg—var å se på gangeoperasjonen med 10 som noe litt enklere, dvs. $10x = 2x + 8x$.

Disse to operasjonene er ekvivalente med å henholdsvis bitshifte én og tre plasser til venstre. Resultatet av disse to operasjonene addert med hverandre vil være resultatet av utregningen, gitt av vi bare skal gange det 4-bits binære tallet med +10.

2 Utrekning

Med fremgangsmåten klar ser vi på hvordan utregningen blir, for så å ta for oss funksjonsuttrykket for det positive gangeresultatet.

Bits	7	6	5	4	3	2	1	0
Inn:					A_3	A_2	A_2	A_1
$2x$:				A_3	A_2	A_2	A_1	
$8x$:		A_3	A_2	A_2	A_1			

Figur 1: Regnestykket

Vi kan da få følgende *positivt resultat*, som vi kan bruke videre for å kunne regne ut ev. regnestykke hvor vi ønsker å gange med -10 i stedet for $+10$.

$$\begin{aligned}
 P_0 &= S_0 = 0 \\
 P_1 &= A_1 \\
 P_2 &= A_2 \\
 P_3 &= A_2 \oplus A_1 & C_3 &= A_2 A_1 \\
 P_4 &= (A_3 \oplus A_2) \oplus C_{P_3} & C_4 &= (A_3 A_2) + ((A_3 \oplus A_2) C_3) \\
 P_5 &= A_2 \oplus C_{P_4} & C_5 &= A_2 C_{P_4} \\
 P_6 &= A_3 \oplus C_{P_5} & C_6 &= A_3 C_{P_5} \\
 P_7 &= C_{P_6}
 \end{aligned}$$

Figur 2: Funksjonsuttrykk for positiv ganging

Vi ser her at P_4 og C_{P_4} representerer en fulladder, altså en port som tar hensyn til hvorvidt vi også tar *inn* mente, i tillegg til det som skal adderes.

Fra disse 8 bits inverterer vi samtlige verdier med en rad med XOR-porter tilknyttet vår 1-bits pin, hvor vi velger om vi ønsker å gange med -10 , med andre ord hvorvidt resultatet skal være signed. Følgende fra de inverterte verdiene har vi en rad med halvaddere. Fra LSB tar vi og legger til én—det høye signalet fra vår sign-pin—i lag med det inverterte signalet, og eventuell mente forplanter seg bortover mot MSB, i den andre enden. Vi forkaster overflow fra denne addisjonsprosessen, ettersom det ikke har noen innvirkning på vår 9-bits resultat, hvorav 8-bits representerer utregningsresultatet fra gangingen, og hvor niende bit er vår sign-bit.

3 Eigentanker

Jeg er ikke helt fornøyd med det faktum at jeg ikke har satt opp komplette Karnaughdiagram for samtlige av funksjonsuttrykkene. Jeg har en magefølelse av at hele kretsen skal være mulig å gjøre vesentlig mer kompakt, men jeg endte opp med et veldig prøv-meg-fram-inspirert opplegg som jeg ikke er stolt av. Derfor gikk jeg tilbake til min opprinnelige framgangsmåte, dvs. den som er definert i dette dokumentet.

4 Annet

I Logisim implementerte jeg full- og halvadderne som underkretser for enkel inklusjon i sluttkretsen. De er synlig som *FA* og *HA* fra sidemenyen.

Merk—som nevnt i kretset—at displayet kun fungerer for unsigned multiplikasjon.