

Rapport Lab 2

Absoluttverdikrets

av

Mats Lieungh
Andreas Nordal

Labgruppe 623

Lab utført: 2/10 2008
Rapport levert: 11/11 2008

Absoluttverdikrets

Sammendrag

Denne rapporten tar for seg hvordan man kan lage en digital krets som finner absoluttverdien av et tall. Absoluttverdikretsen skal selv finne ut om tallet som står på inngangen er negativt, og i så fall bytte fortegn ved å ta 2s komplement av det. Bruksområdet for denne absoluttverdikretsen skal være å finne lydstyrken i en lydkompressor [1]. Prosessen fra teori og forberedelser, til modellering og simulering i utviklingsverktøy og fysisk implementasjon vil bli gjennomgått. Tester og simuleringer viser at kretsen er i stand til å ta absoluttverdien av alle tall med 16 bit, med unntak av det minste tallet, men dette var forutsett.

Innhold

1. Innledning	1
2. Teori	1
2.1. Binære negative tall	1
2.1.1. 2s komplement	1
2.2. Invertererkrets	1
2.3. Addererkrets	1
3. Målemetode og arbeidsbeskrivelse	2
4. Utstyrliste	4
5. Resultater	4
5.1. Simulering av Krets_12	4
5.1.1. Negative tall	4
5.1.2. Overflyt	4
5.1.3. Alle kombinasjoner av EN1 og EN2	5
5.2. ABS-Krets	5
6. Diskusjon	5
6.1. Simuleringer	5
6.1.1. Alle negative tall	5
6.1.2. Overflyt	5
6.1.3. EN1 og EN2	5
6.1.4. ABS-Krets	6
6.1.5. Generelt	6
6.2. Problem: Det minste tallets absoluttverdi	6
6.2.1. Løsning 1: Ikke legg til 1	6
6.2.2. Løsning 2: Negativ absoluttverdi	6
7. Konklusjon	6
8. Vedlegg	7
8.1. Vedlegg 1	7
8.2. Vedlegg 2	8
9. Referanser	9
10. Tilbakemelding	9

1. Innledning

Opggaven er å lage en elektrisk digital krets som tar inn et 16 bits heltall med fortegn og gir ut absoluttverdien av tallet. I labforberedelsene skal det designes en 4-bits invertererkrets, en 4-bits addererkrets, en 4-bits absoluttverdikrets basert på disse to, samt den endelige absoluttverdikretsen. Det skal også planlegges hvordan kretsene skal testes. De to sistnevnte kretsene skal modelleres og simuleres med utviklingsverktøyet Quartus II [2]. Når disse fungerer feilfritt, skal den endelige absoluttverdikretsen lastes over på et Altera DE2 FPGA-kort [5] i form av en mikroprogrammering. Et grensesnitt til absoluttverdikretsen er allerede laget, slik at den påtrykkes stimuli via to registre, og utgangen vises på et firesifret sjusegmentdisplay. I siste test skal et tastatur koblet til DE2-kortet brukes til å skrive verdier heksadesimalt til registrene.

2. Teori

2.1. Binære negative tall

De to vanligste representasjonene av heltall i datamaskiner er fortegnstall (signed) og uten fortegn (unsigned). For at begge tallformatene skal kunne adderes og subtraheres på samme måte, brukes en teknikk for å representere tall med fortegn: Negative tall lagres som 2^s komplement av de tilsvarende positive. [3]

2.1.1. 2^s komplement

Det mest signifikante bitet av et tall på 2^s -komplement-form har negativ vekt, det vil si at alle tall som har 1 på denne posisjonen er negative. Å finne 2^s komplement av et tall kan gjøres ved å invertere det bitvis og så legge til 1. Dette tilsvarer å bytte fortegn på tallet.

Tabell 1: Eksempler på representasjoner av ± 7 :

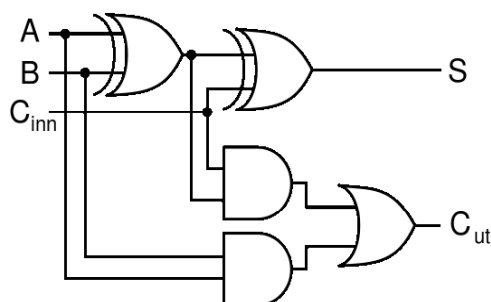
titallsystemet	4 bit	8 bit
7	0111	00001111
-7	1001	11111001

2.2. Invertererkrets

En invertererkrets skal ta et digitalt signal som består av et antall bit, og invertere hvert enkelt av disse. En enklest mulig invertererkrets kan bestå av en NOT-port på hver pinne, men den vil ikke kunne skrus av og på. I tabell 8.1 kan man se at å bruke funksjonen XOR mellom et bit og 1 vil gi det inverterte bitet ut, mens XOR mellom et bit og 0 ikke vil endre bitet. Altså kan XOR brukes som en kontrollerbar NOT-port. Ved å kontrollere disse portene med samme signal, har man en inverterkrets som er på når dette signalet er høyt, se venstre halvdel av figur 3.1.

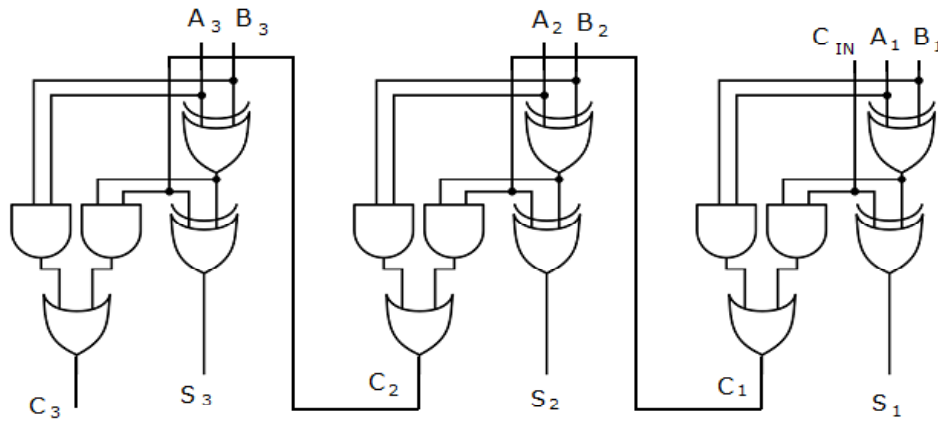
2.3. Addererkrets

En generisk addererkrets tar inn to binære tall og legger dem sammen. Dette kan gjøres ved å bruke en ripple-carry-adderer, som er satt sammen av flere like kretser kalt *fulladderere* [4]. Fulladdereren tar inn to bit A og B, samt et mente-inn-bit C_{inn} , og gir ut et summebit S og et mente-ut-bit C_{ut} .



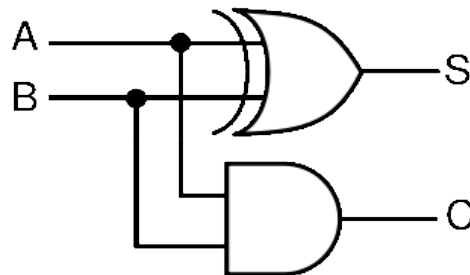
Figur2.2: Fulladderer

For å summere tall som består av flere bit, setter man flere slike fulladderere etter hverandre og får en ripple-carry-adderer, se figur 2.3.



Figur 2.3: Ripple-carry-fulladderer

En halvadderer er en enklere krets som tar inn to bit, og gir ut et summebit og et mente-ut-bit. En eventuell tolkning er at den ene inngangen utgjør mente inn.



Figur 2.4: Halvadderer

Uansett om adderererkretsen består av full- eller halvadderere, vil mente-inn (C_{IN} i figur 2.3), inngå i summen. Hvis man aldri skal legge til noe annet enn 1 eller 0, er det hensiktsmessig å erstatte fulladdererne i ripple-carry-kretsen med halvadderere, og bruke mente-inn som den ene inngangen, se høyre halvdel av Figur 3.1.

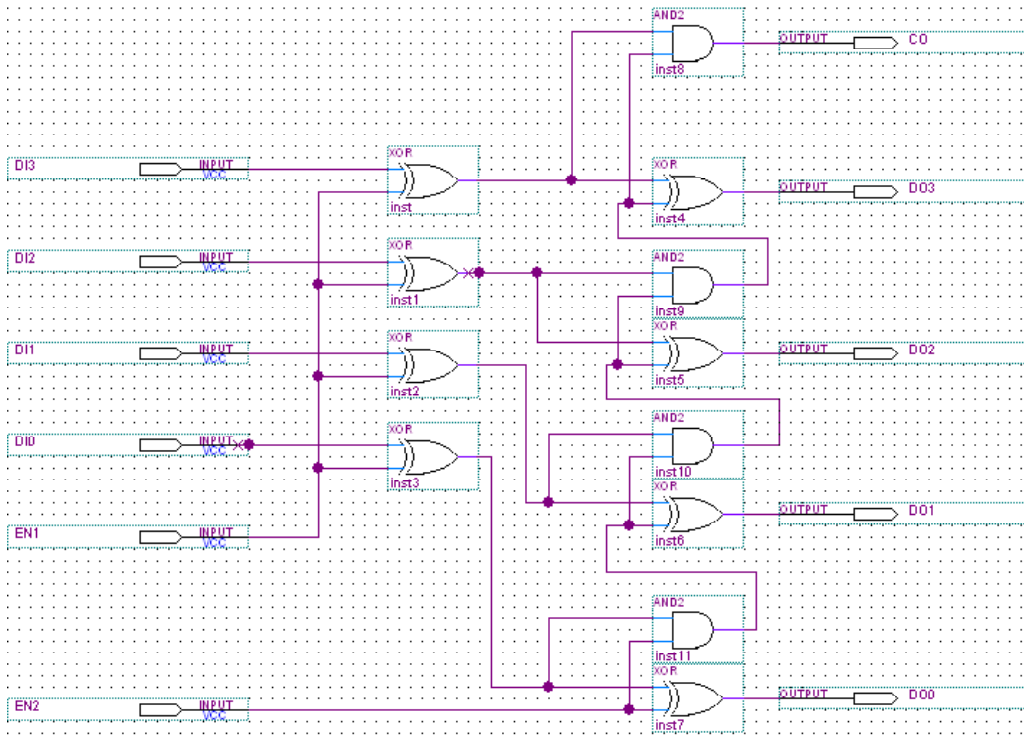
Fordi det tar tid for mente-informasjon å forplante seg gjennom en full- eller halvadderer, vil tiden brukt for hele addisjonen være gitt av stien fra minst signifikante bit på en inngang til mest signifikante bit i svaret. Dette er kritisk sti.

3. Målemetode og arbeidsbeskrivelse

Absoluttverdikretsen ble laget ved å først designe mindre deler av kretsen, for så å sette disse sammen. Den første delkretsen, Krets_1, er en invertererkrets som tar inn et 4-bits binærtall og et *enable*-signal, og gir ut det inverterte tallet dersom *enable*-signalet er satt. Dette ble gjort ved å bruke fire XOR-porter, en mellom *enable*-signalet og hvert av inngangsbitene.

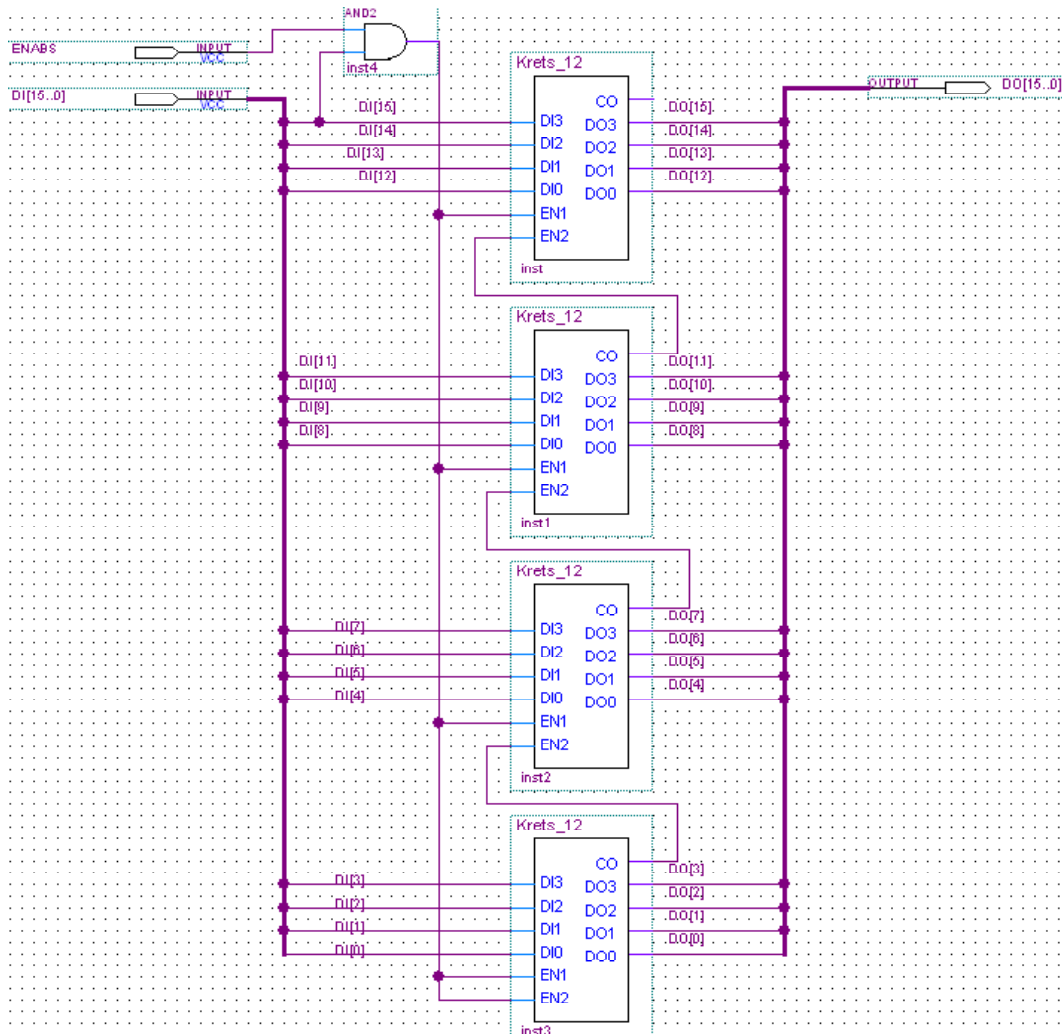
Den andre delkretsen, Krets_2, er en modifisert adderererkrets som tar inn et 4-bits binærtall og et *enable*-signal. Denne kretsen legger til 1 dersom *enable*-signalet er satt. Dette ble gjort ved å bruke fire halvadderere koblet slik at mente-ut fra den ene gikk inn på den ene inngangen på den neste.

Krets_1 og Krets_2 ble kun tegnet som en del av forarbeidet; bare sammensetningen av disse to ble tegnet i Quartus II. Denne sammensatte kretsen, Krets_12, er en krets som tar inn et 4-bits binærtall, og som tar 2s komplement av tallet dersom både det mest signifikante bitet og *enable*-signalet er 1. Figur 3.1 viser Krets_12, hvorav venstre halvdel utgjør Krets_1, og høyre halvdel utgjør Krets_2.



Figur 3.1: Krets_12: 4-bits absoluttverdikrets

Den ferdige absoluttverdikretsen, ABS-Krets, skal behandle lydsamplere på 16 bit. Denne settes sammen av fire Krets_12. For at den skal være en absoluttverdikrets, skal den kun bytte fortegn når inngangen er negativ, det vil si at mest signifikante bit er 1. Den skal heller ikke gjøre noe med mindre enable-bitet er 1, derfor brukes en AND-port mellom disse bitene.



Figur 3.2: ABS-Krets: Hele 16-bits absoluttverdikretsen

For å luke ut eventuelle feil på et tidlig stadium, ble Krets_12 testet før den ble brukt videre i oppbygningen av ABS-Krets. Testingen av begge kretsene ble gjort med et simuleringstøytøy i Quartus II. Til slutt ble hele systemet lastet over i DE2-kortet. Utfra en testplan utarbeidet i forarbeidet ble absoluttverdikretsen testet ved å sette de to registrene som utgjorde inngangene på ABS-Krets ved hjelp av tastatur. Utgangen ble lest av sjusegmentdisplayet og sammenliknet med testplanen.

4. Utstyrliste

1. Altera DE2 FPGA-kort
2. Quartus II (Version 8.0 Build 231 07/10/2008 SJ Full Version)
3. Microsoft Windows (Version 5.1 Build 2600.xpsp_3_gdr.080814-1236)
4. USB-kabel
5. Batterieliminatør
6. Tastatur med USB-til-PS2-plugg
7. Håndlenker (type jordet)

5. Resultater

5.1. Simulering av Krets_12

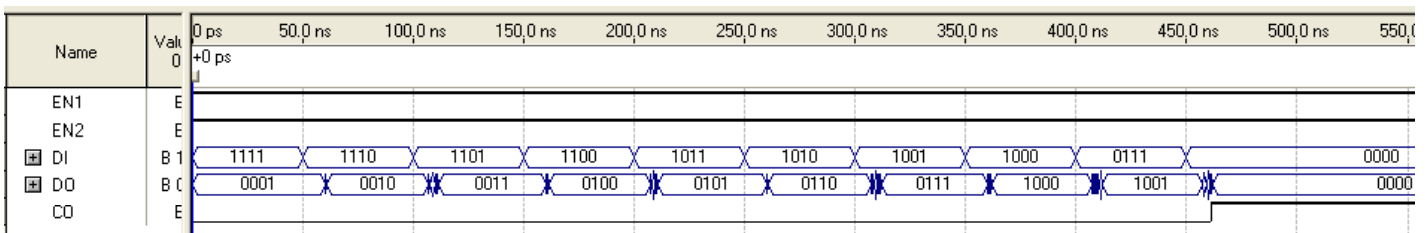
Tabell 5.1: Inn- og utganger av Krets_12 i simulering

Navn	Beskrivelse
EN1	enable-signalet til Krets_1
EN2	enable-signalet til Krets_2
DI	påtrykt 4-bits tall
DO	utgangen av Krets_12
CO	mente-ut fra Krets_12

Hver av figurene 5.1-3 viser et skjermbilde av en simulering i Quartus II. I venstre rute vises navnene på signalene som inngikk i simuleringen. I høyre rute vises tidsforløpet av signalene.

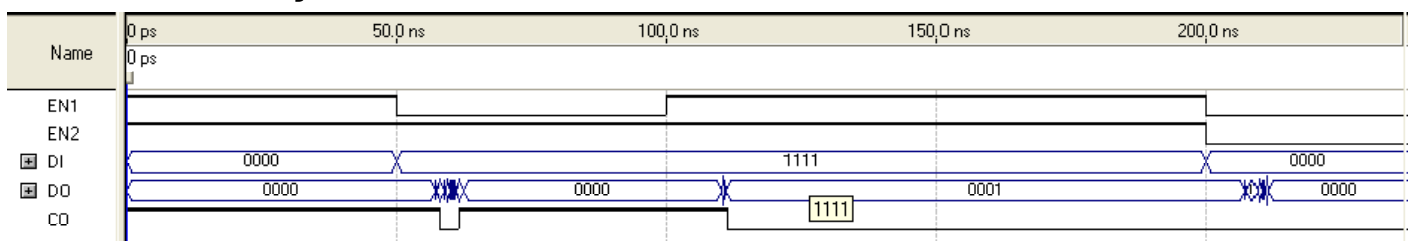
5.1.1. Negative tall

Fordi dette skal være en absoluttverdikrets, må den kunne snu fortegn på alle negative tall.



Figur 5.1: Alle negative tall

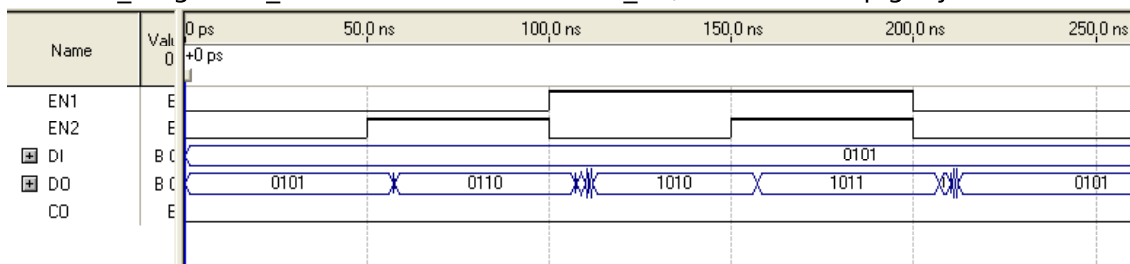
5.1.2. Overflyt



Figur 5.2: Mente-ut går høy

5.1.3. Alle kombinasjoner av EN1 og EN2

Selv om Krets_1 og Krets_2 er satt sammen til Krets_12, kan de foreløpig styres individuelt.



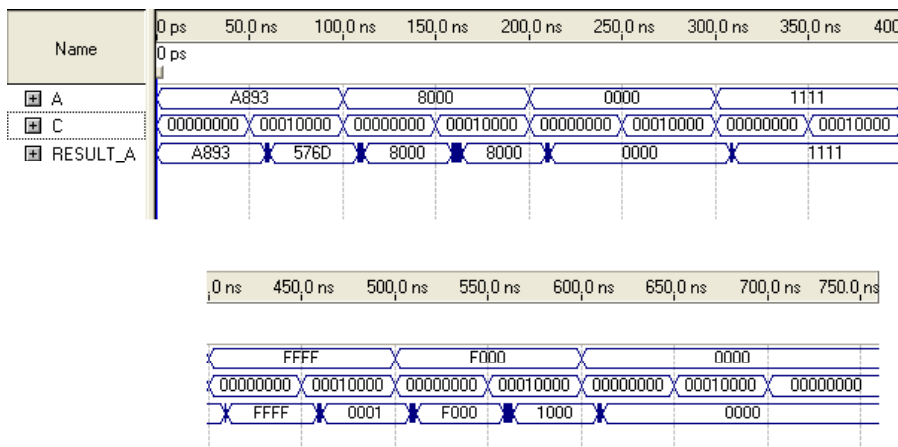
Figur 5.3: Individuell styring av Krets_1 og Krets_2

5.2. ABS-Krets

Tabell 5.2: Inn- og utganger av ABS-krets i simulering

Navn	Beskrivelse
A	inngangen til ABS-Krets
C	Inneholder ABS-Krets' enable-signal (ENABS i figur 3.2)
RESULT_A	utgangen av ABS-Krets

Dette er simuleringen av den totale absoluttverdikretsen. Alle tall er simulert med ENABS både på og av.



Figur 5.4: Simulering i henhold til testplan (tidsaksen er delt for å få plass i bredden).

6. Diskusjon

6.1. Simuleringer

6.1.1. Alle negative tall

Når EN1 og EN2 er høye, inverterer Krets_12 alle bitene og legger til 1. Som Figur 5.1 viser, ble det testet alle negative tall fra -1 til og med -8, samt det positive tallet 7 for ordens skyld. I alle tilfeller er det tallet med motsatt fortegn som kommer ut.

6.1.2. Overflyt

Mente-ut er en utgang som går høy ved overflyt, det vil si når halvaddereren prøver å legge sammen 1111_2 og 1_2 . Det skjer bare når Krets_1 sin utgang er 1111_2 , som kan skje på nøyaktig 2 måter, hvorav begge vises i figur 5.2. I den endelige absoluttverdikretsen vil ikke EN2 være høy uten at EN1 er det, så bare det første tilfellet er av betydning videre.

6.1.3. EN1 og EN2

Her vises at både invertererkretsen og addererkretsen fungerer individuelt som de skal. Invertererkretsen ble påtrykt tallet 5, mens addererkretsen ble påtrykt både 5 og 10.

6.1.4. ABS-Krets

Dette gikk helt etter testplanen. De positive tallene forble positive, mens de negative tallene ble positive da ENABS var høy, bortsett fra 8000_{16} . Dette er det minste tallet. Binært er dette 100000000000000_2 . Først inverteres dette og blir 011111111111111_2 . Når man så legger til 1 får man en menteforplantning helt til det mest signifikante bitet og får det samme som inngangen.

6.1.5. Generelt

Det kunne vært en fordel å simulert Krets_12 med alle positive tall mens EN1 og EN2 er lave; da dette skal bli en absoluttverdikrets skal kun de negative tallene endre fortegn. Riktignok viser de andre testene av denne kretsen at tallene 0 og 5 slipper uendret gjennom når EN1 og EN2 er lave. I simuleringen av ABS-Krets var også de positive tallene uendret.

Alle fire simuleringer viser at utgangen DO skifter raskt mellom flere verdier før den stabiliserer seg. Årsaken kan være at serien av porter som bestemmer hvert bit ikke endrer seg på likt. Ved nærmere ettersyn viste det seg (ikke synlig på figurene) at bare et bit endres om gangen. Menteforplantning kan være årsaken til at det i noen overganger ble spesielt mange kortvarige utgangsverdier, for eksempel i ABS-Krets ved påtrykk 8000_{16} på DI.

6.2. Problem: Det minste tallets absoluttverdi

Antall negative tall som kan representeres på 2s-komplement-form er ett mer enn antall positive tall som kan representeres. I vårt tilfelle, med 16 bit, er det plass til alle tall mellom -2^{15} og $+2^{15}-1$. Tar man 2s komplement av det minste tallet, så får man det samme til svar. Følgelig vil tallet -2^{15} slippe uendret gjennom vår krets.

6.2.1. Løsning 1: Ikke legg til 1

Det kommer selvsagt an på hva kretsen skal brukes til, siden dette strengt tatt gir feil svar. I et signalbehandlingssystem hvor man vil analysere signalstyrke, blir dette kanskje vel så riktig: Intervallet av mulige verdier har sentrum mellom 0 og -1, mens konseptet om absoluttverdi bygger på at sentrum av tallaksen er 0. Ved å kun invertere alle bitene oversettes nedre halvdel av intervallet til øvre halvdel. Det er først når man så legger til 1 at man kan få overflyt og havne på bunnen igjen. Denne løsningen implementeres lett ved å droppe addererkretsen.

6.2.2. Løsning 2: Negativ absoluttverdi

Ved å ta 2s komplement av positive tall i stedet for negative, kan alle tall uten problem gjøres negative. Det er bare å invertere enable-signalet som går til både inverter- og addererkretsene.

7. Konklusjon

Ved systematisk arbeid med enklere kretser ble det designet en 16-bits absoluttverdikrets. Absoluttverdikretsen er modulbasert; ved å sette sammen et antall Krets_12 vil man kunne lage liknende kretser som kan behandle så store tall man vil. Dessuten: Selv om bare et fåtall av alle de 2^{16} mulige kombinasjonene av inngangsbit ble testet, kan man ved generalisering være nokså sikker på at kretsen oppfører seg riktig for andre tall også, ettersom byggesteinen, Krets_12, ble testet grundig. Grundig testing er lettere å gjøre med en liten krets, for eksempel ble Krets_12 testet på alle 4-bits negative tall. Alle resultater av simuleringer og tester samsvarer med teorien fra forarbeidet. Det er altså liten grunn til å tro at det var noen feil i designet.

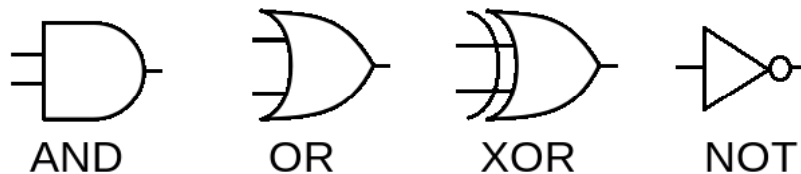
Dessverre er ikke absoluttverdien av det minste tallet representerbart, så det finnes ingen måte kretsen kan behandle dette tallet riktig på. Derimot kan problemet omgås med en av flere mulige tilpasninger til et gitt formål. Hvis absoluttverdikretsen skal brukes til å finne styrken av et signal, kan en ren invertererkrets være mer egnet, i og med at eksakt matematisk riktighet da ikke er poenget, og at man unngår problemet med absoluttverdien av det minste tallet. Uten addisjonsdelen, ville kretsen bli enklere og raskere.

8. Vedlegg

8.1. Vedlegg 1

Tabell 8.1: Sannhetstabell for logiske porter: AND, OR og XOR

a	b	a AND b	a OR b	a XOR b	NOT a
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0



Figur 8.1: Noen logiske porter

8.2. Vedlegg 2

Tabell 8.2: Testplan for Krets_12

DI	EN1	EN2	DO	CO
5	0	0	5	0
5	0	1	6	0
5	1	0	-6	0
5	1	1	-5	0
-1	1	1	1	0
-2	1	1	2	0
-3	1	1	3	0
-4	1	1	4	0
-5	1	1	5	0
-6	1	1	6	0
-7	1	1	7	0
-8	1	1	-8	0
7	1	1	-7	0
0	1	1	0	1
1	0	1	0	1

Tabell 8.3: Testplan for ABS-Krets

ENABS	DI	DO
0	0xA893	0xA893
1	0xA893	0x576D
0	0x8000	0x8000
1	0x8000	0x8000
0	0x0000	0x0000
1	0x0000	0x0000
0	0x1111	0x1111
1	0x1111	0x1111
0	0xFFFF	0xFFFF
1	0xFFFF	0x0001
0	0xF000	0xF000
1	0xF000	0x1000

9. Referanser

1. http://en.wikipedia.org/wiki/Dynamic_range_compression
2. http://en.wikipedia.org/wiki/Quartus_II
3. Daniel D. Gajski, Principles of Digital Design, New Jersey, 1997, s.35
4. Daniel D. Gajski, Principles of Digital Design, New Jersey, 1997, s.168
5. <http://www.altera.com/education/univ/materials/boards/unv-de2-board.html>

10. Tilbakemelding

Faget er et av de mer praktiske vi har hatt. Det er interessant å lære om moderne teknologi for en gangs skyld, ikke bare teori, og ikke altfor grunnleggende. Passe vanskelig. Studassene er flinke, men (forståelig nok) har ikke alle like mye peiling på DE2-kortet.