

Rapport

Lab 1

Absoluttverdikrets - portkretser

av

Even Wiik Thomassen
Broen van Besien

Gruppe 193

Lab utført: 8. september 2004
Rapport levert: 12. november 2004

Lab 1
Absoluttverdikrets - portkretser

Sammendrag

I løpet av denne labben har vi laget en krets som tar et fire bits tall, og gir oss absoluttverdien til det tallet, hvis tallet er negativt, får vi det den positive representasjonen av tallet. Hvis tallet er positivt, kommer det uforandret gjennom kretsen vår.

Vi hadde kun logiske kretser, *XOR* og *AND*, til rådighet for å bygge kretsen vår. Disse byggeblokkene var allerede satt opp på et kretskort, så utfordringen vår var å koble brikkene sammen slik at kretsen fungerte slik forventet.

En absoluttverdikrets for et binært tall på tos-komplementsform må gjøre to operasjoner. Først må alle bit'ene bli invertert, så må vi legge til en til det nye tallet. Men dette skal bare skje når vi har et negativt tall. Siden det første bit'et forteller fortegnet til et tos-komplement-tall, bruker vi det som styresignal til hele kretsen.

Vi har forenklet en *carry-ripple*-adderer, og vi har beregnet teoretisk forplantningsforsinkelse og stige- og falltid på både kritisk sti, og de forskjellige logiske blokkene vi har brukt. Deretter har vi beregnet de samme størrelsene på den faktiske kretsen, med oscilloskopet, for så å sammenligne disse verdiene.

Til slutt har vi gitt en mulig forklaring på avvikene mellom teoretisk og beregnede verdier, og diskutert om hvorvidt målene med labben har blitt oppfylt.

Innholdsfortegnelse

Sammendrag	iii
Innholdsfortegnelse	iv
1 Innledning	1
2 Teoridelen	1
3 Målemetode og arbeidsbeskrivelse	2
3.1 Forarbeidet	2
3.1.1 Oppgave 2	2
3.1.2 Oppgave 3	2
3.1.3 Oppgave 4	2
3.1.4 Oppgave 5	2
3.1.5 Oppgave 6	3
3.1.6 Oppgave 7	3
3.1.7 Oppgave 8	3
3.2 Labarbeidet	4
3.2.1 Fase 1	4
3.2.2 Fase 2	4
3.2.3 Fase 3	4
3.2.4 Fase 4	4
3.2.5 Fase 5	4
3.2.6 Fase 6	4
4 Utstyrlisten	9
5 Diskusjon og konklusjon	10
Referanser	11
A Veddlegg: Funksjonstabell for fire bit inverterer	12
B Veddlegg: Funksjonstabell for absoluttverdikretsen	13

1 Innledning

Denne rapporten beskriver arbeidet med å designe en absoluttverdikrets. Hensikten med arbeidet er todelt: Lære hvordan en absoluttverdikrets kan bygges opp. Danne en forståelse for digitalteknikk med fysiske portkretser.

2 Teoridelen

Absoluttverdikretsens oppgave er å gjøre negative tall positive. Positive tall blir ikke forandret av kretsen. Tallene er på tos-komplement, da angir det første bit'et fortegnet på tallet. Hvis det er '0' er tallet positiv, hvis det er '1' er tallet negativ. Derfor kan vi bruke det første bit'et for å styre styreinngangen. For å invertere tallet på tos-komplement inverterer man alle bit'ene og legger til 1.

For å implementere en tos-komplement inverterer skal vi bruke logiske blokken *XOR*, og for å addere 1 skal vi bruke en *ripple-carry* adderer. Siden vi bare skal legge til 1, kan vi forenkle *ripple-carry* addereren.

Ripple-carry adder består av forskjellige identiske blokker. Hver blokk representerer en bit, og har tre innganger og to utganger. Inngangene består av de to bit'ene som skal legges sammen og overflytsbit'et fra den forrige blokken. Utgangene består av en bit som har summen og et overflytsbit. Overflytsbit'et kommer når summen av de to bit'ene blir to, og da blir summen null. Blokkene er koblet sammen på en linje, der overflytsbit'et fra den forrige er koblet til inngangen på den neste. Se kapittel 5.2 side 170 i Gajski [1].

Den forenklete *ripple-carry* addereren har blokker med bare to innganger og to utganger. Det andre bit'et som skal adderes blir overflytsbit'et fra den forrige blokken. Fordi vi skal legge til én, er bare det minst signifikante bit'et én og derfor bruker vi bare den ene summasjonsinngangen i hver blokk, og lar den andre ta imot overflytsbit'et.

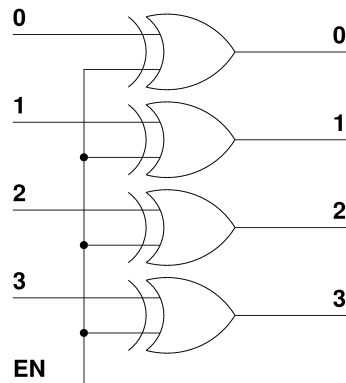
Implementasjonen skal gjøres på et lab-kort. Kortet har 2 syv-segment display som viser inngangs- og utgangsverdier. Kortet har også 3 logiske IC-kretser, og kortet har 32 tilkoblinger. Implementasjonen blir da å koble rett mellom de forskjellige tilkoblingene. Inngangstaller blir gitt ved å velge på kortets fire-bits giver med stiftlist med kortslutningsbøyler. Pinneparet på toppen gir det mest signifikante bit'et (msb). Stiftlistene står for de binære representasjonene av tallene som vi vil produsere.

3 Målemetode og arbeidsbeskrivelse

3.1 Forarbeidet

3.1.1 Oppgave 2

Vi brukte 4 logiske *XOR* for å implementere KRETS_1. KRETS_1 tar seg av inverteringen av tallet. Egenskapen til en *XOR*-krets er at hvis en av inngangenene er n, blir den andre inngangen invertert. n blir til null, og null blir til n. Derfor bestemte vi oss for å bruke 4 logiske *XOR* der vi brukte n inngang til styresignalet, og den andre til et av bit'et i tallet som vi skulle invertere. Se figur 1. Se vedlegg A for funksjonstabellen.



Figur 1: Fire bit invertorer

3.1.2 Oppgave 3

Vi skulle forenkle uttrykkene for *ripple-carry*-addereren, se under, og så skulle vi sette opp funksjonstabell for halv-adderer-blokkene, se tabellen under.

$$\begin{aligned} S_n &= A_n \oplus B_n \oplus C_n & \Rightarrow & S_n = A_n \oplus B_n \\ C_{n+1} &= A_n \cdot B_n + A_n \cdot C_n + B_n \cdot C_n & \Rightarrow & C_{n+1} = A_n \cdot B_n \end{aligned}$$

Tabell 1: Halv-adderer

A	B	S_n	C_{n+1}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

3.1.3 Oppgave 4

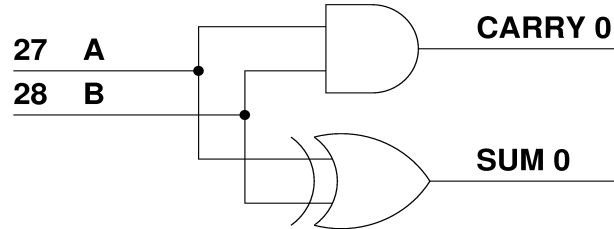
Med fire-bits binærtall får man 16 kombinasjoner. Siden tos-komplement har kun 1 representasjon for null, blir det en ujevn fordeling av tall, fra -8 til 7.

3.1.4 Oppgave 5

Vi skulle sette opp funksjonstabell for absoluttverdikretsen, se vedlegg B for resultatet.

3.1.5 Oppgave 6

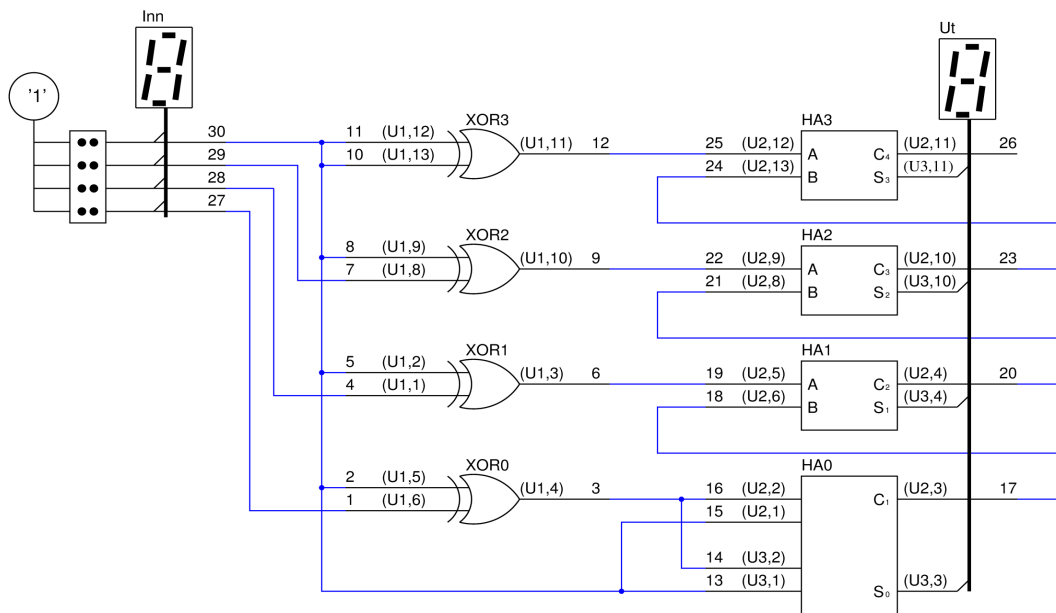
Her skulle vi sette opp sammenkoblingene for å lage en halv-adderer-blokk som vi kan teste, uten å ha resten av kretsen koblet sammen, se figur 2, for implementasjonen.



Figur 2: Halv-adderer-blokk

3.1.6 Oppgave 7

Her skulle vi sette opp sammenkoblingene for hele kretsen, slik at vi fikk en 4-bits inverterer, og en 4-bits halv-adderer. Se figur 3 for sammenkoblingen av hele kretsen. Styresignalet (*msb*) kobles på de 4 *XOR* logikkportene i invertereren, og på den første halv-adderer-blokken.



Figur 3: Koblingen av hele kretsen

3.1.7 Oppgave 8

- Kritisk sti begynner på terminal 27 og 30, fordi addereren kan ikke begynne før den minst signifikante bit'et er invertert.
- Utgangen av kritisk sti er pinne $U3.11$ av halv-adderer-blokk 3.
- Den teoretiske forsinkelsen gjennom kretsen gjennom kritisk sti er 310ns.
($\rightarrow XOR$ -portene: $80ns \rightarrow HA_0 : 50ns \rightarrow HA_1 : 50ns \rightarrow HA_2 : 50ns \rightarrow HA_3 : 80ns = 310ns$)

•	Stigetid	Falltid
<i>XOR</i>	75ns	85ns
<i>AND</i>	45ns	55ns

3.2 Labarbeidet

3.2.1 Fase 1

Vi ønsket her å sjekke om halv-adderern fungerte. Vi startet med å koble opp kretsen som i figur 2 side 3. Vi koblet port 27 til port 13 og 15, så koblet vi port 28 til port 14 og port 16. Deretter koblet vi en likespenningskilde på +7 volt på terminal 31, og jord til terminal 32. Vi brukte de 2 nederste stiftlistparene for å gi riktig påtrykk til koblingen. Kretsen stemte overens med tabell 1 på side 2.

3.2.2 Fase 2

Her ønsket vi å sjekke om hele absoluttverdikretsen fungerte som forventet. Vi startet med å koble opp kretsen som i figur 3 side 3. Se figur for sammenkoblingen av kretsen. Vi testet med verdiene i tabell 3 på side 13, og resultatet stemte overens med tabellen. Ergo fungerer absoluttverdikretsens. Deretter fjernet vi alle stiftene.

3.2.3 Fase 3

For å fullføre resten av testene skulle vi koble til oscilloskopet. Vi koblet oscilloskopet til kretsen som framgangsmåten i labboken [2], side 22-23.

3.2.4 Fase 4

Vi ønsket å måle *signalgeneratoren* med oscilloskopet, slik at vi fikk en firkantpuls på 100kHz med 5 volt peak to peak. Vi koblet *signalgeneratoren* til oscilloskopet som i labboken [2], side 24. Da fikk vi følgende figur fra oscilloskopet, se figur 4, der vi ser kilden (*signalgeneratoren*).

3.2.5 Fase 5

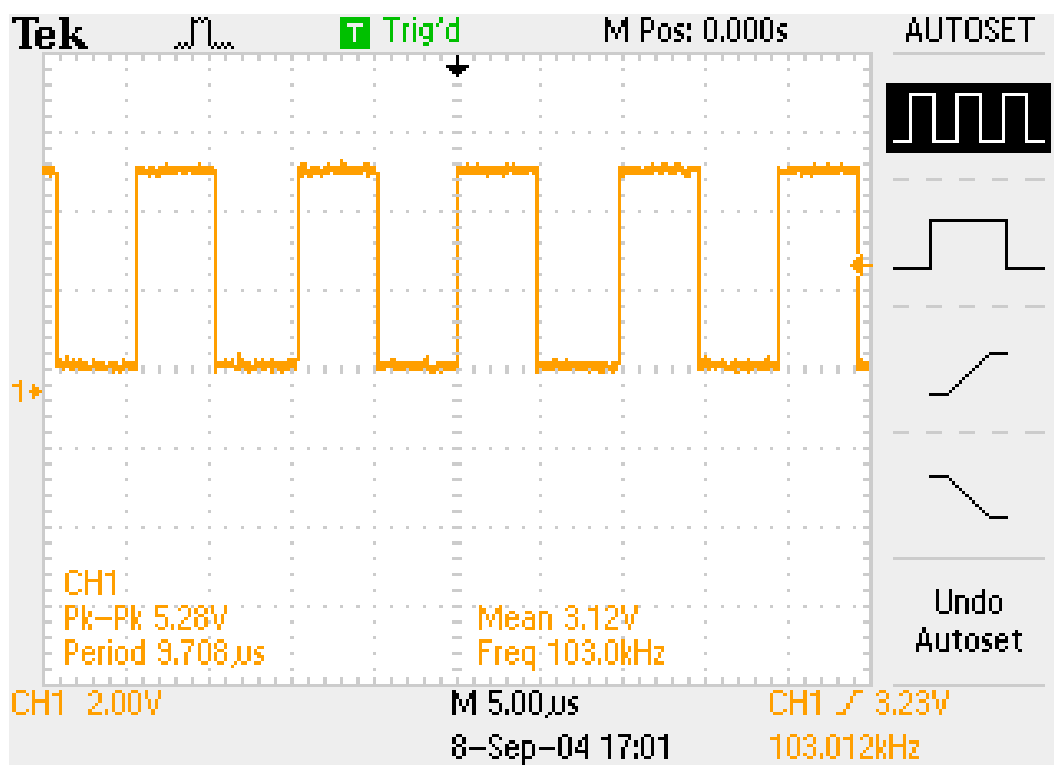
Her ønsket vi å beregne forplantningsforsinkelsen til den kritiske stien. Vi koblet *signalgeneratoren* til oscilloskopet og labkortet som i labboken [2], side 25-26. For å måle forplantningsforsinkelsen gjennom kritisk sti koblet vi proben direkte på halv-adderer-blokk 4, *U.11*. For å unngå støy satt vi jord fra proben på terminal 32. Så innstilte vi oscilloskopet som i labboken [2], side 26, for å måle forsinkelsen. Resultatet ble at forsinkelsen var 672.0ns, se figur 5.

3.2.6 Fase 6

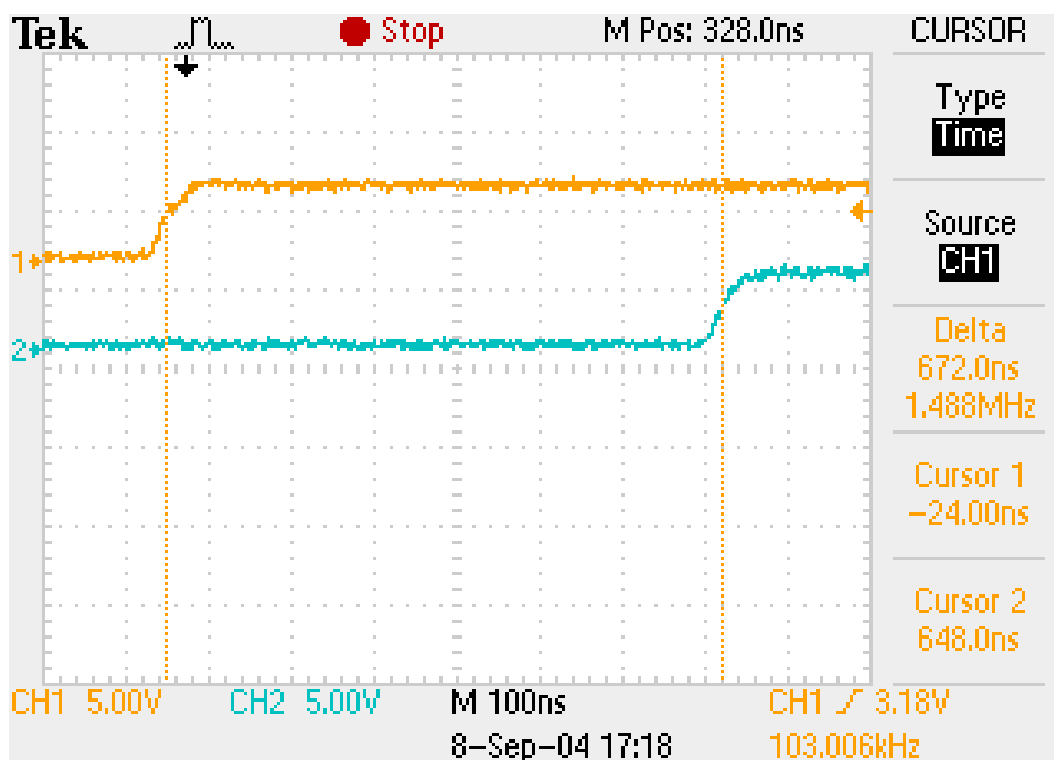
Vi ønsket å måle stigetiden og falltiden på den kritiske stien, og på *XOR* og *AND* blokkene. Først fjernet vi kabelen mellom oscilloskopet og signalgeneratoren. Så koblet vi proben på port U3.11 på HA3, for deretter å trykke AUTO SET på oscilloskopet. Så trykket vi på den tredje knappen, slik at vi fikk opp stigetiden til den kritiske stien. Den var 42.40ns, se figur 6 på side 6.

For å måle falltiden gjorde vi det samme som i punktet over, bare at vi brukte den fjerden knappen på oscilloskopet. Resultatet ble 60.80ns, se figur 7 på side 6.

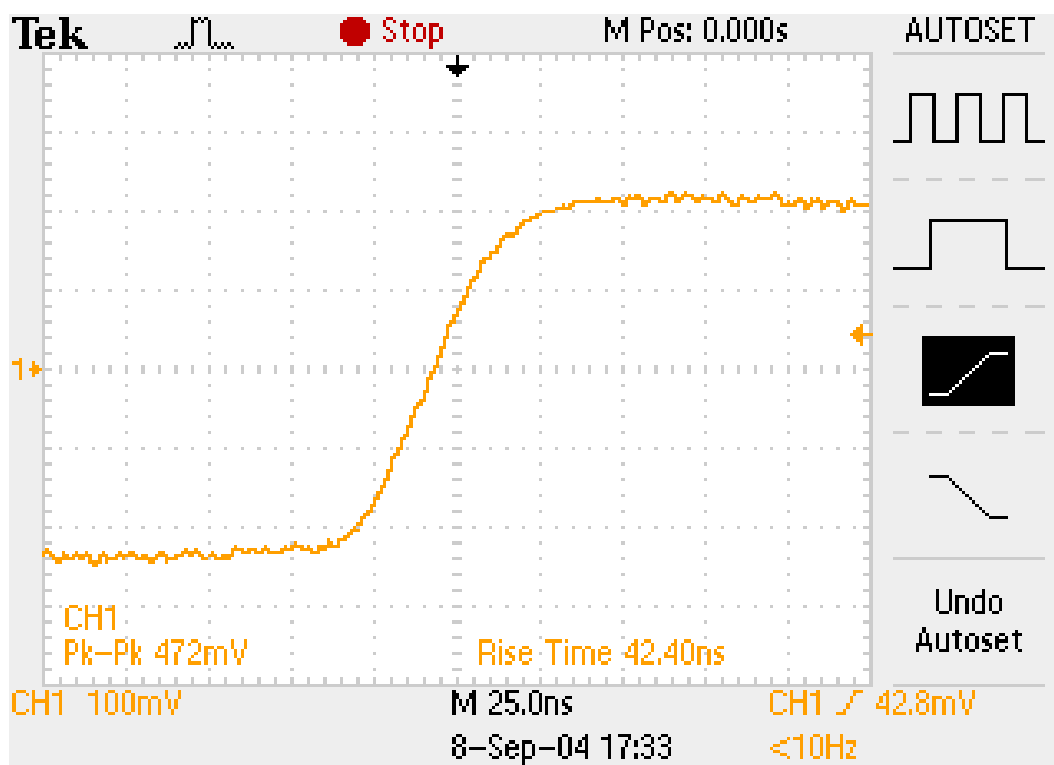
For å måle stigetiden til *XOR*-porten satte vi proben på U1.4 på XOR0. Dette er en av portene til invertereren. Resultatet ble 55.00ns, se figur 8 på side 7.



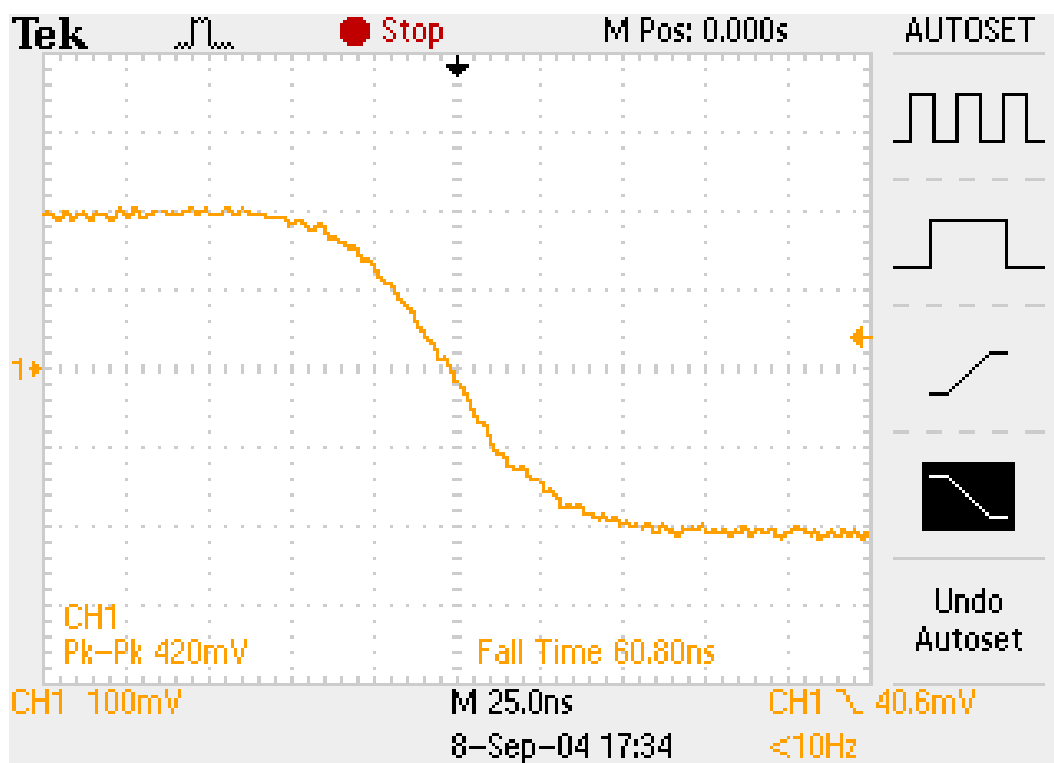
Figur 4: Inngangssignal fra signalgeneratoren



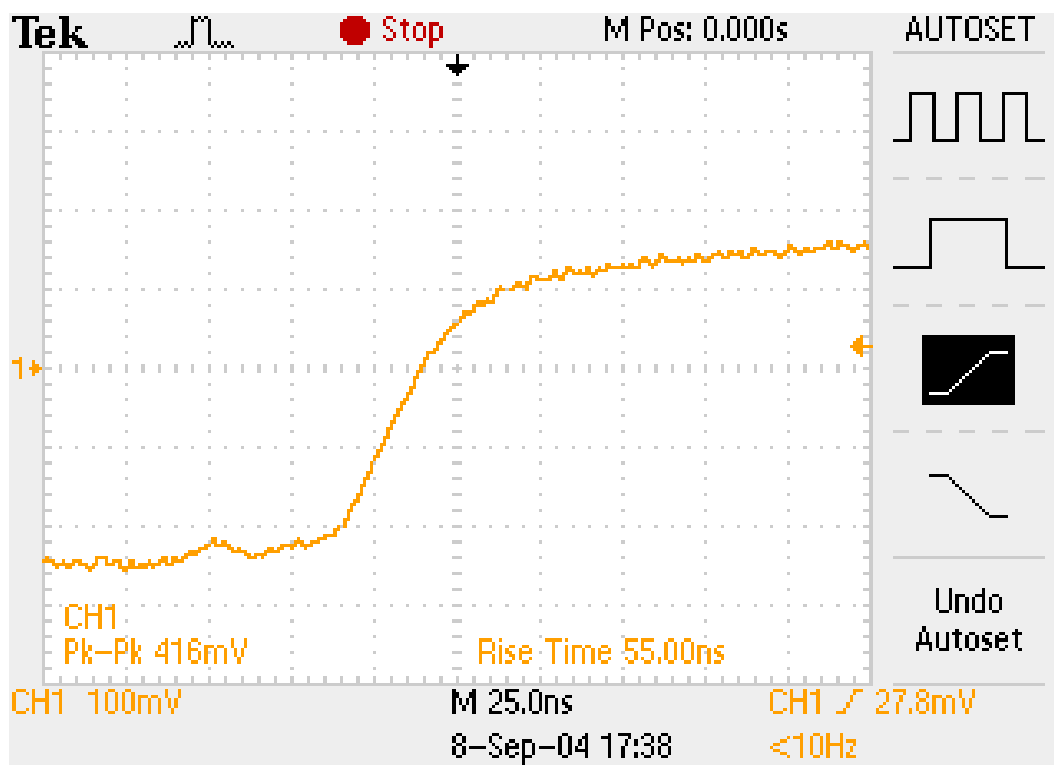
Figur 5: Forplantningsforsinkelsen i kritisk sti



Figur 6: Stigetiden til kritisk sti

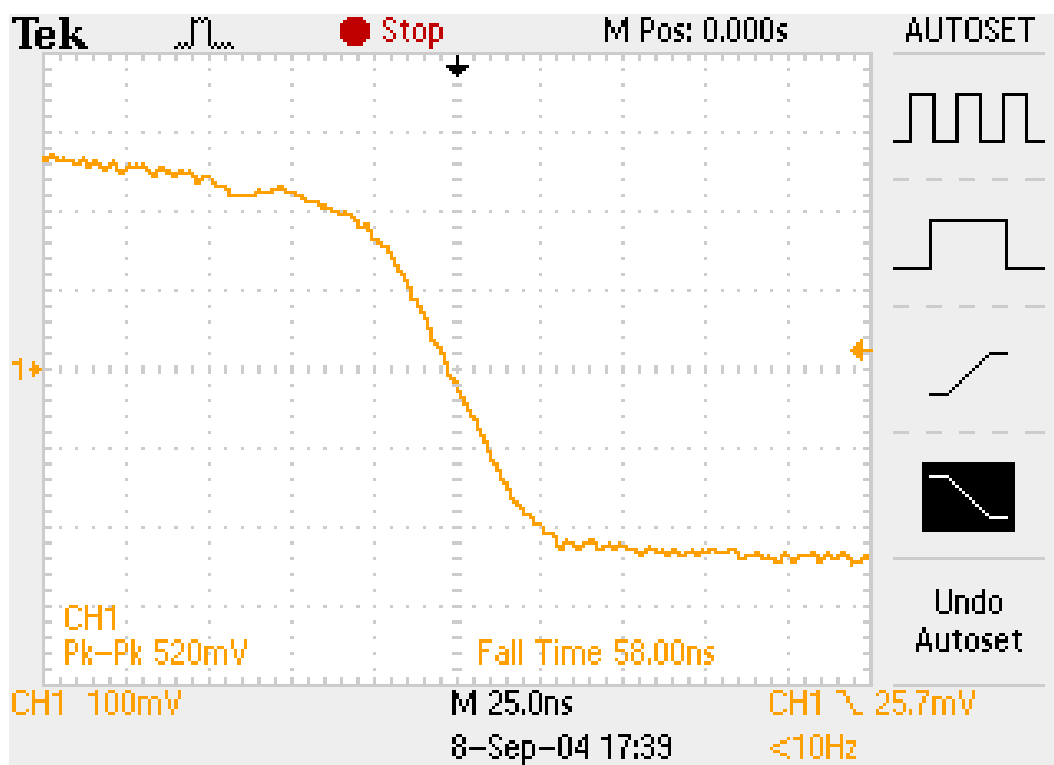


Figur 7: Falltiden til kritisk sti

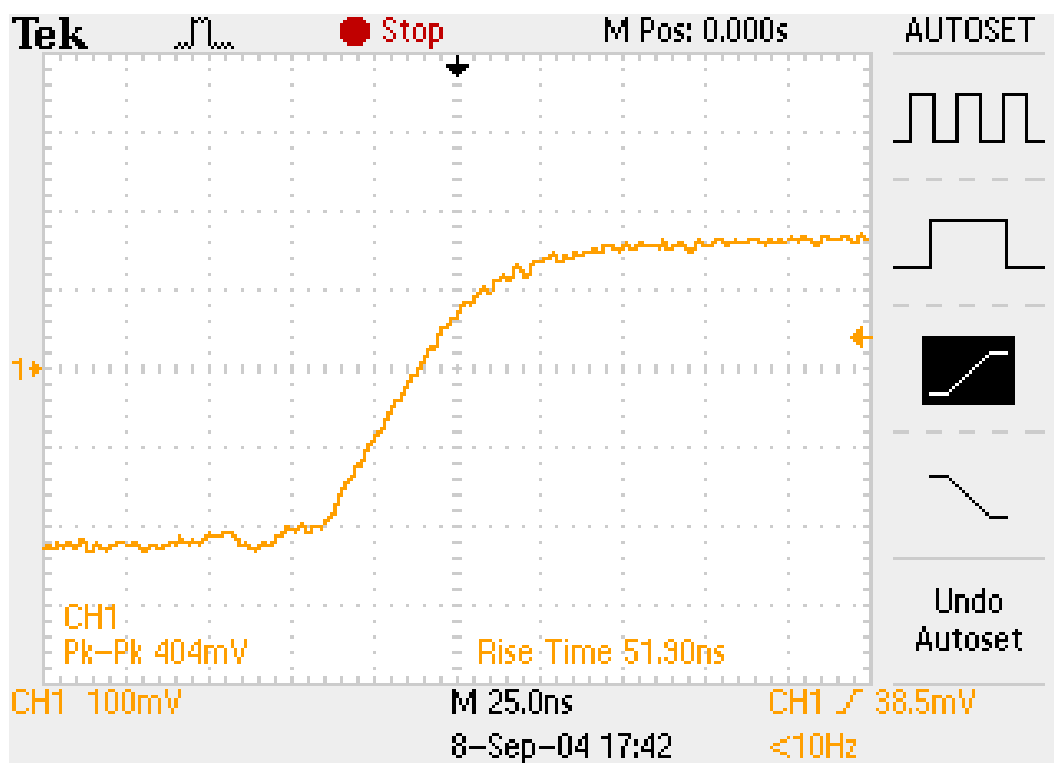
Figur 8: Stigetiden til *XOR*

For å måle falltiden til *XOR*-porten gjorde vi likt som i forrige punkt, men trykte på knapp 4 på oscilloskopet istedenfor. Den ble 58.00ns, se figur 9 på side 8.

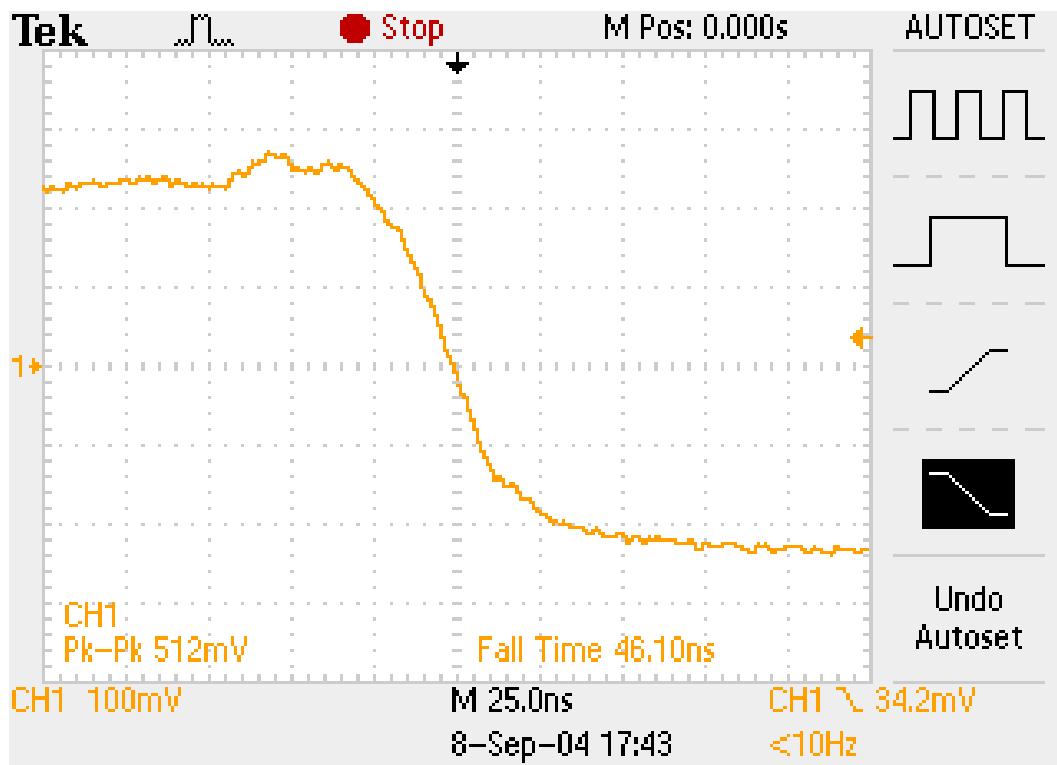
AND-portene målte vi på samme måte som *XOR*. Stigetiden ble 51.90ns, se figur 10 på side 8, falltiden ble 46.10ns, se figur 11 på side 9.



Figur 9: Falltiden til XOR



Figur 10: Stigetiden til AND

Figur 11: Falltiden til *AND*

4 Utstysrlisten

Utstyret vi brukte iløpet av labben var:

- Elektronisk oscilloskop: Tektronix tds 2014, s# g04-0158
- Signalgenerator: GG-4035
- Labkort: se beskrivelse i teoridelen.
- Datamaskin med Microsoft Windows XP(tm)
- Dataprogram for å ta skjermbilder fra oscilloskopet: "TDS2014"

5 Diskusjon og konklusjon

Beskrivelsen	Teoretisk	Målt	Avvik	Avvik (%)
Forplantningsforsinkelsen	310ns	672ns	362ns	116%
XOR stigetid	75ns	55ns	20ns	27%
XOR falltid	85ns	58ns	27ns	32%
AND stigetid	45ns	52ns	7ns	16%
AND falltid	55ns	46ns	9ns	16%

Den store forskjellen i forplantningsforsinkelsen kan skyldes at kretsen ble koblet opp på en holder, der vi koblet sammen alle punktene med kobberkabler. Det burde gi oss en del mer forsinkelse enn en ideell krets.

Når vi målte stige- og falltiden hos *XOR*- og *AND*-portene så varierte verdiene kraftig (+/- 15ns). Derfor kan vi si at avvikene på stige og falltiden er innenfor rimelighetens grenser.

Målet med labben var å lage en absoluttverdikrets som tok ett tall på fire bit ved hjelp av en inverterer og en modifisert *ripple-carry* adderer. Som vi har vist i denne rapporten har vi klart det.

Det andre målet med labben var å lære hvordan man lager en krets ved hjelp av logiske blokker, og det har vi vist her i rapporten at vi har lært.

Referanser

- [1] Gajski, D. D.: *Principles of Digital Design* (1997), Prentice Hall, New Jersey (USA).
- [2] NTNU, TFE4105 Digitalteknikk of Datamaskiner LAB Høst 2004.

A Veddlegg: Funksjonstabell for fire bit inverterer

Tabell 2: Funksjonstabell for fire bit inverterer

DI_1	EN_1	DO_1
0000	0	0000
0001	0	0001
0010	0	0010
0011	0	0011
0100	0	0100
0101	0	0101
0110	0	0110
0111	0	0111
1000	0	1000
1001	0	1001
1010	0	1010
1011	0	1011
1100	0	1100
1101	0	1101
1110	0	1110
1111	0	1111

DI_1	EN_1	DO_1
0000	0	1111
0001	0	1110
0010	0	1101
0011	0	1100
0100	0	1011
0101	0	1010
0110	0	1001
0111	0	1000
1000	0	0111
1001	0	0110
1010	0	0101
1011	0	0100
1100	0	0011
1101	0	0010
1110	0	0001
1111	0	0000

B Veddlegg: Funksjonstabell for absoluttverdikretsen

Tabell 3: Funksjonstabell for absoluttverdikretsen

<i>BIN</i>	<i>HEX</i>	<i>DEC</i>	<i>BIN(ABS)</i>	<i>HEX(ABS)</i>
0111	0x7	7	0111	0x7
0110	0x6	6	0110	0x6
0101	0x5	5	0101	0x5
0100	0x4	4	0100	0x4
0011	0x3	3	0011	0x3
0010	0x2	2	0010	0x2
0001	0x1	1	0001	0x1
0000	0x0	0	0000	0x0
1111	0xF	-1	0001	0x1
1110	0xE	-2	0010	0x2
1101	0xD	-3	0011	0x3
1100	0xC	-4	0100	0x4
1011	0xB	-5	0101	0x5
1010	0xA	-6	0110	0x6
1001	0x9	-7	0111	0x7
1000	0x8	-8	1000	0x8