

Labrapport - Lab 4

Sondre H. Elgaaen & Eirik Wittersø

14. april 2019

TFE4101 Krets og Digitalteknikk
Norges Teknisk-Naturvitenskapelige Universitet

Sammendrag

Innhold

Sammendrag	i
Innhold	ii
1 Innledning	1
2 Teori	2
2.1 Tallrepresentasjon	2
2.2 Absoluttverdi	2
2.3 Modifisert Ripple Carry adder	2
2.4 Tidsforsinkelse og kritisk sti	3
3 Vårt Arbeid	4
3.1 Utstysrliste	4
3.2 Forarbeid	4
3.3 Kobling av absoluttverdikrets	6
4 Resultater	7
5 Diskusjon	8
6 Konklusjon	9

1 Innledning

2 Teori

2.1 Tallrepresentasjon

Det er 3 vanlige måter å representere heltall binært: Magnitude uten fortegn, magnitude med fortegn, og toerkomplement. I denne laboppgaven har tallene blitt representert med toerkompliment. På toerkomplimentsform representeres positive tall som magnitude uten fortegn. Negative tall representeres ved å ta inversen til absoluttverdien av tallet, og legge til 1. Å ta inversen til et binærtall gjøres ved å gjøre om alle 1-ere til 0, og alle 0-ere til 1. Man kan se om et tall er negativt ved å se på det MSB (det mest signifikante bittet). Hvis det er 1 er tallet negativt, hvis det er 0 er tallet positivt.

2.2 Absoluttverdi

En absoluttverdikrets skal ta inn et tall, og gi ut absoluttverdien av tallet. Å ta absoluttverdien av et tall på toerkomplementsform gjøres i 3 steg.

1. Sjekke fortegnet til tallet ved å se på MSB.
 - Hvis tallet er positivt, gi tallet som output.
 - Hvis tallet er negativt, utfør punkt 2 og 3.
2. Ta inversen av tallet.
3. Legg til 1 til tallet og gi det som output.

2.3 Modifisert Ripple Carry adder

En vanlig ripple carry adder består av en serie med heladdere som utfører addisjon på bitnivå, og sender ut et summ-bit til output-bussen, og et carry bit som går videre til den neste heladder. En slik adder tar inn 2 tall på toerkomplementsform eller som magnitude uten fortegn, og legger dem sammen. Den Modifiserte ripple carry adderen som har blitt brukt i denne laboppgaven skal derimot ta inn ett tall på toerkomplimentsform og et bit. Det den skal gi ut er tallet plus det ene bittet. Adderen er bygget opp av halvaddere. En halvadder har 2 inputs, (A og B), og 2 outputs, (S og C). Halvadderer legger sammen de to bitsene, og sender signal på S hvis summen blir 1, og signal på C hvis summen blir 2. I den modifiserte ripple carry adderen er halvadderene koblet sammen slik at et av carryen fra det forrige bittet går til et av inputtene på det neste bittet.

2.4 Tidsforsinkelse og kritisk sti

Tidsforsinkelsen til en krets er tiden det tar fra inputsignalene endrer seg til utsignalene har fått sine endelige verdier. Tidsforsinkelsen avhenger hoved sakelig av hvilke porter signalet må gå igjennom, og hvor mange av dem det er. Den veien signalet bruker lengst tid på å forplante seg igjennom kretsen er den kritiske stien. Tidsforsinkelsen igjennom den kritiske stien er det viktig å ta hensyn til når man designer kretser. Denne tiden avgjør hvor lenge man må vente etter å ha gitt input før man kan endre det igjen. Denne tiden brukes til å bestemme hvilken klokkehastighet kretsen kan kjøre på, og det er lurt å se på den kritiske stien hvis man skal optimalisere kretsen for å kjøre på høyere klokkehastigheter.

3 Vårt Arbeid

3.1 Utstysrliste

Veroboard, 32 sockets

Kretskort gitt til lab i TFE4101

Digital Oscilloskop - Rohde & Schwarz RTB2004

Signalgenerator - Rohde & Schwarz HMF2525

Spenningskilde - Rohde & Schwarz HMC 8042

Banankabler

BNC-BNC kabler

BNC-banankabel

BNC-splitter

Prober

3.2 Forarbeid

Design av absoluttverdikrets

I forarbeidet skulle det designes en 4-bits absoluttverdikrets. For å ta absoluttverdien av et binært tall kan man invertere det og addere 1. En absoluttverdikrets kan dermed bygges opp av inverterkretser og av halvadderkretser. Vi startet med å designe en inverterkrets, altså en krets som inverterer hvert bit som kommer inn, hvis kretsen er aktivert.

Det blir gjort tydelig av tabell 1 at en slik krets kan lett implementeres som en XOR-port.

In	En	Out
0	0	1
0	1	0
1	0	1
1	1	0

Tabell 1: Sannhetstabell for inverterkrets

Deretter designet vi en halvadderkrets. En halvadderkrets adderer to tall og har to utganger, en for summen og en for mente, så lenge kretsen er aktivert.

In	Carry-In	Sum	Carry-Out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Tabell 2: Sannhetstabell for halvadderkrets

Etter å ha designet disse komponentene, måtte vi sette de sammen til å bli en 4-bits absoluttverdikrets. Da lagde vi blokker ved å sette inverterkretsen og halvadderkretsen i serie, og satte fire av disse blokkene i parallell. Videre brukte vi MSB som enable signal for inverterne og som mente inn for første halvadder, slik som i figur [REF HER].

Beregning av kritisk sti og maksimal klokkehastighet

Kritisk sti fant vi ved å se på det scenarioet hvor det blir flest menteforplantninger, nemlig overgangen fra 1000 til 1000, hvor det skjer menteforplantninger gjennom hele kretsen. Da går kritisk sti gjennom 2 XOR porter og 3 AND porter. For å beregne forsinkelsen gjennom kritisk sti fant vi verdiene for forsinkelse gjennom de forskjellige portene ved 5V som maks spenning, som er som gitt i Tabell 3

Port	Forsinkelse
AND	125ns
XOR	140ns

Tabell 3: Forsinkelsestid gjennom porter

Gitt disse verdiene kan vi regne oss frem til forsinkelse gjennom kritisk sti.

$$2 \cdot 140ns + 3 \cdot 125ns = 655ns$$

Dette gir oss at maksimal klokkehastighet er... SJEKK VERDI

$$1/655ns = 1.53 \text{ MHz}$$

3.3 Kobling av absoluttverdikrets

4 Resultater

Ved testing av en enkelt halvadder fikk vi følgende resultater.

Tabell 4: Halvadder resultater

Verdier fra forarbeid				Målte verdier	
A	B	Sum	Carry	Sum	Carry
0	0	0	0	0	0
0	1	1	0	1	0
1	0	1	0	1	0
1	1	0	1	0	1

Ved testing av ferdigkoblet absoluttverdikrets fikk vi følgende resultater.

Tabell 5: Absoluttverdikrets resultater

Verdier fra forarbeid					Målte verdier	
Binary	Hex	Decimal	Binary(Abs)	Hex(Abs)	Binary(Input)	Hex(Output)
0111	0x7	7	0111	0x7	0111	0x7
0110	0x6	6	0110	0x6	0110	0x6
0101	0x5	5	0101	0x5	0101	0x5
0100	0x4	4	0100	0x4	0100	0x4
0011	0x3	3	0011	0x3	0011	0x3
0010	0x2	2	0010	0x2	0010	0x2
0001	0x1	1	0001	0x1	0001	0x1
0000	0x0	0	0000	0x0	0000	0x0
1111	0xF	-1	1111	0x1	1111	0x1
1110	0xE	-2	1110	0x2	1110	0x2
1101	0xD	-3	1101	0x3	1101	0x3
1100	0xC	-4	1100	0x4	1100	0x4
1011	0xB	-5	1011	0x5	1011	0x5
1010	0xA	-6	1010	0x6	1010	0x6
1001	0x9	-7	1001	0x7	1001	0x7
1000	0x8	-8	1000	0x8	1000	0x8

5 Diskusjon

6 Konklusjon