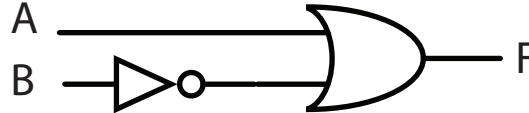


IN1020 - Ukeoppgave for andre uken av maskinvaredelen

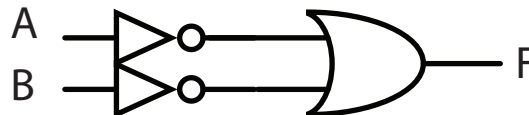
Omid Mirmotahari og Yngvar Berg

FASIT

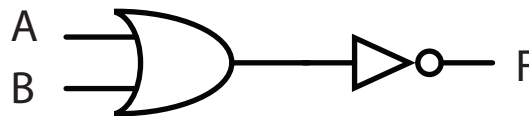
- 1) Hva er kretsdesignet for $F = A + B'$



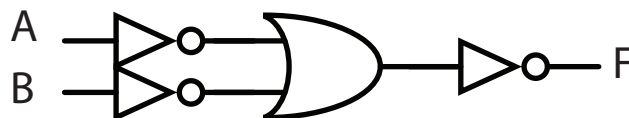
- 2) Hva er kretsdesignet for $F = A' + B'$



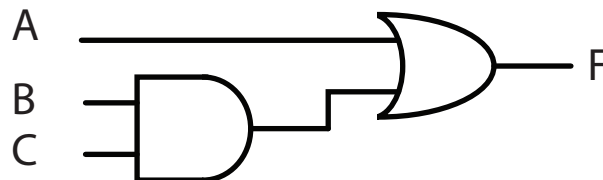
- 3) Hva er kretsdesignet for $F = (A + B)'$



- 4) Hva er kretsdesignet for $F = (A' + B')'$



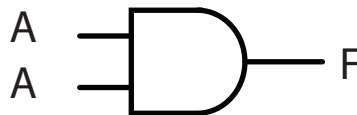
- 5) Hva er kretsdesignet for $F = A + BC$



- 6) NØTT! Hva er kretsdesignet for $F = A + B + C$

Denne kretsen kan løses enten ved å ha en 3-inputs OR-port eller å ha to 2-inputs OR.

- 7) Hva er kretsdesignet for $F = AA$



- 8) Vis gjennom en sannhetsverditabell at $F = A' + B' = (AB)'$

A	B	$A' + B'$	$(AB)'$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

- 9) Sjekk gjennom en sannhetsverditabell at $F = A'B' = A + B$

A	B	A'B'	A+B
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

Funksjonene er ikke like. De er motsatte av hverandre. $F = A'B' = (A+B)'$
 Det betyr at $F = (A'B')'' = (A'' + B'')' = (A + B)'$

10) NØTT! Sett opp sannhetsverditabellen for $F = A + AB$

A	B	A+AB
0	0	0
0	1	0
1	0	1
1	1	1

11) Kan du klare å forenkle uttrykket i oppgave 10?

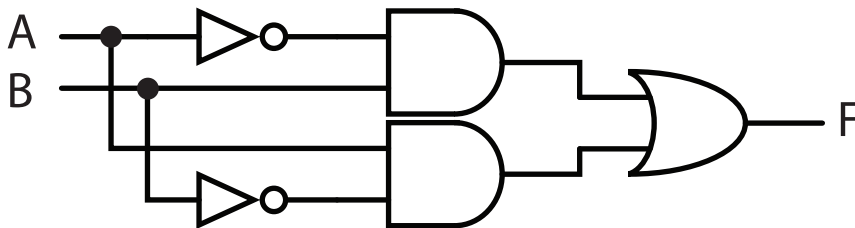
Denne funksjonen kan forenkles til å være A. Altså $F = A + AB = A(1+B) = A$

12) Hva er det forenklede uttrykket av $F = A + A'B$

Denne funksjonen blir $F = A + A'B = (A + A')(A + B) = A + B$. Sett gjerne opp sannhetsverditabell for å se at dette stemmer.

13) NØTT! Implementer en XOR port med kun å bruke AND, OR, NOT porter.

$$F = A \text{ xor } B = A'B + AB'$$

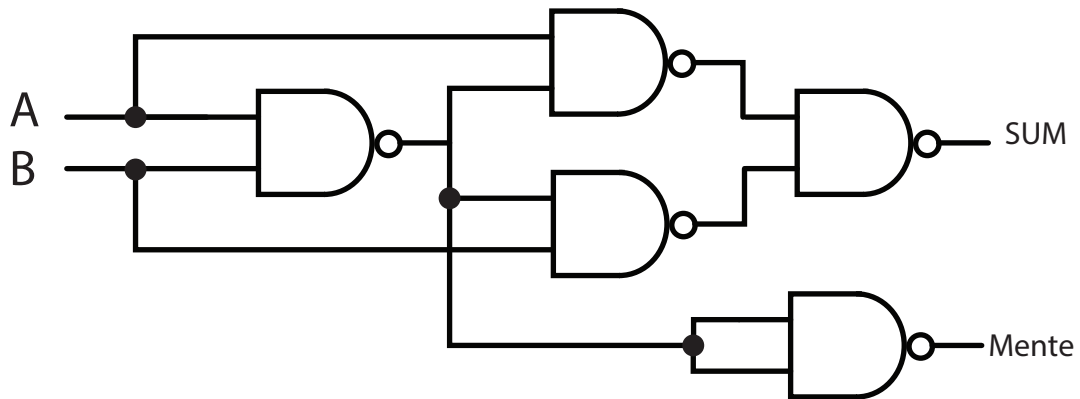


14) Hva er den totale forsinkelsen gjennom kretsen i oppgave 13

Gitt 5 ps portforsinkelse for enhver port. Kretsen i oppgave 13 har en logisk dybde på 3, som betyr at inngang til utgang går igjennom tre porter etter hverandre for den lengste veien

Den totale forsinkelsen er da 15 ps.

15) NØTTNØTTNØTT! Kan du implementere en Halv-adder med bare NAND-porter



16) Gitt en portforsinkelse på 5 ps for enhver port, hvilken frekvens vil en 1-bits halv-adder kunne operere på?

En portforsinkelse på 5 ps trenger en klokkeperiode på 10ps (forutsatt 50%-50% klokkesignal).

En klokkeperiode på 10ps gir 100 GHz i frekvens.

17) Diskuter fordeler og ulemper med å ha kretser som må synkroniseres.

Fordeler:

- Valide data
- Forutsigbarhet

Ulemper:

- Lavere hastighet

18) Diskuter fordeler og ulemper med å ha et klokkesignal som har en klokkeperiode som er 50% høy og 50% lav

Fordeler:

- Lett å lage
- Stabilt

Ulemper:

- Ikke optimalt for alle operasjoner

19) Diskuter fordeler og ulemper med å ha et klokkesignal som har en klokkeperiode som er 70% høy og 30% lav.

Fordeler:

- Ofte optimalisert for å gjennomføre mer komplekse operasjoner (større logisk dybde)
- Opplevs som raskere

Ulemper:

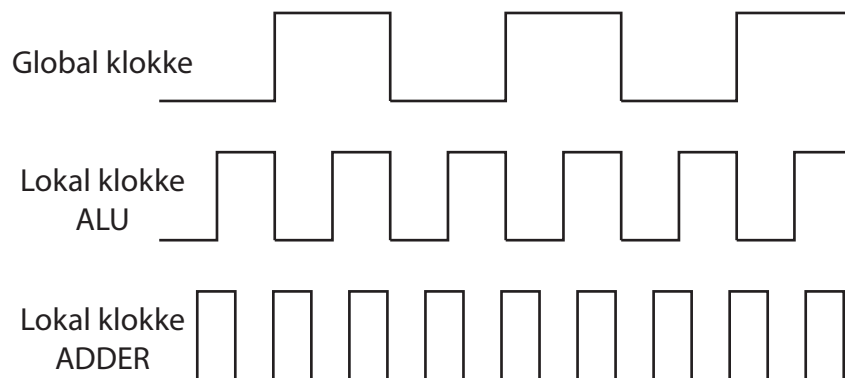
- Vanskelig å lage
- Ofte utsatt for stabilitetsutfordringer

20) NØTT! Hva er frekvensen for en port (5ps delay) når klokkeperioden er som i oppgave 19.

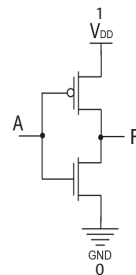
Her må man huske på at en port vil prosessere i tidsrommet klokken er høy. Hvis man da må ha minimum 5ps for når klokken er høy betyr det at $5ps = 70\%$ av hele klokkeperioden. Dette gir en klokkeperiode på $7,2 ps (5ps/70\%)$. Vi velger å avrunde det til $8 ps$ slik at vi får da en frekvens på $125GHz$.

21) NØTT! Sett opp hvordan tidsforløpet for følgende tre klokker og hvordan de skal synkroniseres. (1) Global klokke, (2) Lokal klokke for ALU, (3) Lokal klokke for ADDER.

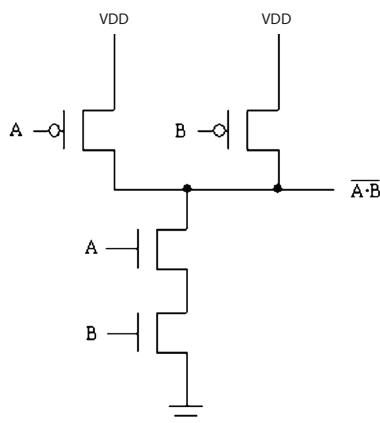
Global klokke må ha en periode som er minst dobbelt av lokal klokke for ALU. Og tilsvarende må en lokal klokke ha en periode som er minst dobbelt av en lokal klokke for ADDER. Eksempelvis:



22) Tegn opp en inverter med transistor



23) NØTTNØTTNØTT! Tegn opp en NAND med transistorer (ikke pensum)



24) NØTT! Diskuter fordeler og ulemper med å ha lavere forsyningsspenning enn 1V

Fordeler:

- Lavere effektforbruk (muligens... (sett bort fra lekkasje))

Ulemper:

- Tregere kretser
- Potensielt problem med å skru av transistorene godt nok til å unngå lekkasje

25) Diskuter hva som begrenser den globale klokkefrekvensen

Det er mye. Men i korte trekk:

Teknologien

Kretsdesign og logisk dybde

Pipeline og antall trinn

26) Når man skruer av datamaskinen, hvilke av nivåene i minnehierarkiet vil miste din data?

All primærminne som Register, Cache og RAM.

27) Gitt en forsinkelse på 100 ps for å lese ut data fra et register, hva er den høyeste frekvensen vi kan bruke for utlesing av et register

Det er to måter å tenke dette på. Hvis vi sier at det skal enten leses fra minne eller skrives til minne så kan vi beregne frekvensen som $1/200\text{ps} = 5\text{GHz}$

Litt mer komplisert er om man må kunne aksessere minne, både skrive og lese i samme synkroniseringsløyfe som gjør at man da får en frekvens på $1/400\text{ps} = 2.5\text{GHz}$

28) Gitt en forsinkelse på 1 ns for å lese ut fra cache (L1), hva er den høyeste frekvensen vi kan bruke for utlesing av cache (L1)

Hvis vi er avhengig av at en klokke synkroniserer utlesingen så trenger vi $1/2\text{ns}$ som da gir en frekvens på 500 MHz.

Om vi ikke er avhengig av en klokke synkronisering, så vil det bli 1 GHz

29) Gitt en forsinkelse på 50ns for å lese ut fra RAM, hva er den høyeste frekvensen vi kan bruke for utlesing av RAM

Hvis vi er avhengig av at en klokke synkroniserer utlesingen så trenger vi $1/100\text{ns}$ som da gir en frekvens på 10 MHz.

Om vi ikke er avhengig av en klokke synkronisering, så vil det bli 20 MHz

30) Gitt at dataen du ønsker å bruke ligger på harddisken (SSD) og lesing fra harddisken tar 500 ns. Hva er den høyeste frekvensen vi kan bruke for å lese ut fra harddisken?

Hvis vi er avhengig av at en klokke synkroniserer utlesingen så trenger vi $1/1000\text{ns} = 1/1\mu\text{m}$ som da gir en frekvens på 1 MHz.

Om vi ikke er avhengig av en klokke synkronisering, så vil det bli 2 MHz.

31) NØTT! Normalt tar det lengre tid å skrive til minne enn å lese ut fra minne. Regn ut hva frekvensen vil være for oppgave 27,28,29,30 når det tar 25% lengre tid for å skrive til de ulike minnene.

Denne er litt komplisert. Men vi kan forenkle og øke tiden med 25%. Da får vi

Oppgave 27: 100ps blir 125ps som gir 40 GHz

Oppgave 28: 1ns blir 1,25 ns som gir 400 MHz

Oppgave 29: 50ns blir 62,5 ns som gir 8 MHz

Oppgave 30: 500ns blir 625ns som gir 0,8 MHz.

Hvis man tenker at man først må lagre før man kan lese ut i neste ledd, så blir det VESENTLIG tregere frekvens.....

32) NØTTNØTTNØTT! CPUen trenger å prosessere data som ikke finnes i register, cache eller RAM, men ute på harddisken. Gitt at hvert ledd av minnehierarkiet må lagre dataen før de kan sende den videre. Hvor lang tid vil det ta fra harddisken til CPUen?

LAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAANG tid.....

Veldig forenklet så kan man anta at hastigheten fra register til cache blir 10 ganger forsinket. Cache til RAM 50 ganger forsinket og RAM til harddisk med 10 ganger. Hvis man må vente til at det er lagret før man leser da blir det dobbelt av hvert ledd som gir da

Lese fra

register:	100 ps
cache:	1 000 ps
RAM:	50 000 ps
SSD:	500 000 ps

Svaret blir da:

$(100 + 1.000 + 50.000) * 2 + 500.000 = 602.200 \text{ ps}$

Altså

Ca 6.000 ganger så lang tid som det vil da lese ut fra et register.

Hvis man legger til grunn at en prosessor kan gjennomføre en operasjon pr klokkeperiode, så taper man 6.000 operasjoner for å hente noe ute på harddisken.

