

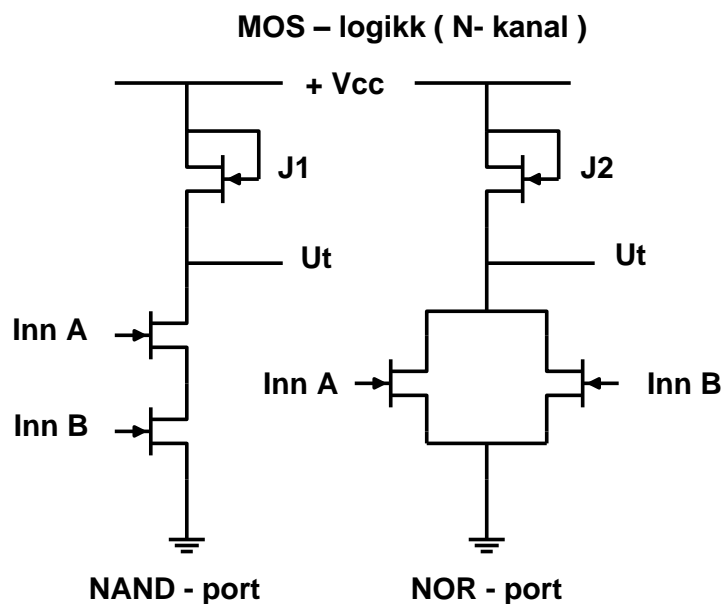
DIGITALE kretser og systemer

Binære systemer – består av kretser som bare arbeider med to mulige tilstander – "0" og "1"

Boolsk algebra er et system for matematisk analyse av binære systemer.

En Boolsk variabel kan ha en av to mulige tilstander, "0" og "1" ev. "false" og "true"

Binære systemer kan brukes til både logiske og aritmetiske operasjoner.



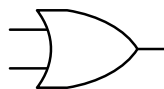
Grunnlaget for boolsk algebra bygger på 3 typer logiske porter: NOT, AND og OR

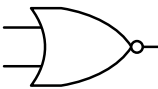
Disse 3 elementene er byggesteinene for alle digitale systemer. Ved en *negasjon* (invertering) av uttrykkene for AND og OR fremkommer portene NAND og NOR. Dette er kretser som lettere lar seg realisere kretsteknisk - samtidig som de kan realisere de samme Boolske likningene.

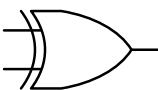
DIGITALE kretser og systemer

Sannhetstabeller


Sannhetstabeller gir sammenhengen mellom logiske nivåer på inngangene og utgangen til portene – legg merke til symbolene som brukes


OR 

NOR 

XOR 

OR , NOR og XOR				
A	B	OR	NOR	XOR
0	0	0	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	0	0

AND 

NAND 

AND og NAND			
A	B	AND	NAD
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

DIGITALE kretser og systemer

Tallsystemer

Desimalt system med grunntall 10.

Hva representerer de enkelte faktorene i tallet 321?

$$321 = 3 \cdot 10^2 + 2 \cdot 10^1 + 1 \cdot 10^0 = 300 + 20 + 1$$

Binært tallsystem med grunntall 2

Eksempel på hvordan vi regner om fra et binært tall. Gjør om 10011 til et desimalt tall

$$10011 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 16 + 0 + 0 + 2 + 1 = 19$$

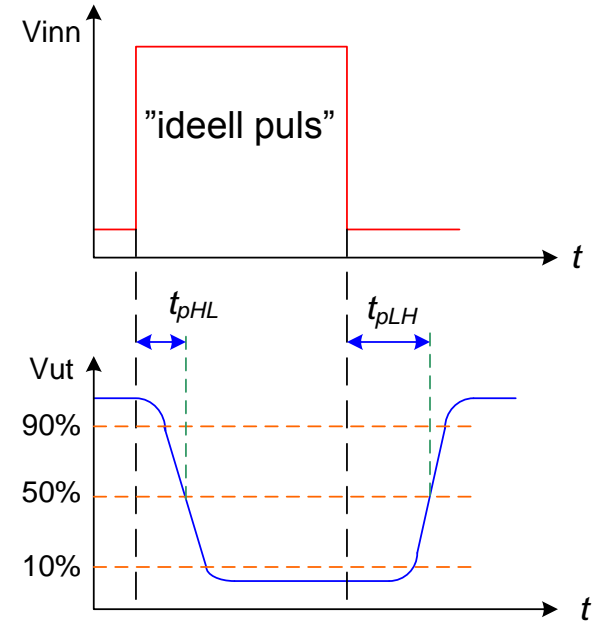
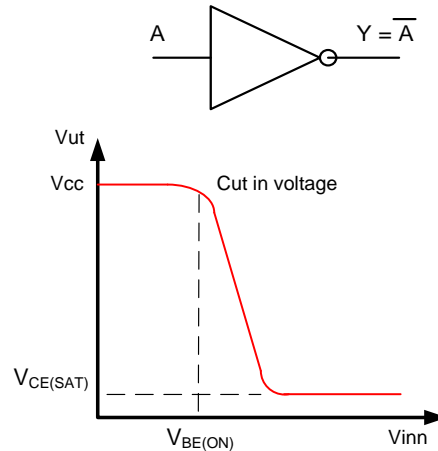
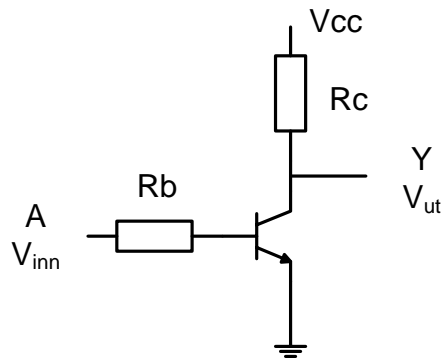
Desimalt	Binært	Desimalt	Binært
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

Et binært siffer (0 eller 1)
kalles et BIT
8 BIT danner en BYTE
4 BIT danner en "nible"

DIGITALE kretser og systemer

BJT som bryter (inverter)

Innføring i noen begreper



For $V_{inn} < 0,5$ volt (cut in voltage) er transistoren "cut off"

Rise-time (t_r) og Fall-time (t_f)

Transisjonstid mellom logiske tilstander 90% - 10%

Problemet med forsinkelse skyldes ladningstransport inn-ut fra BASIS – samt interne kapasiteter i transistoren. Størst forsinkelse når transistoren skal stenges (skru av strømmen). Det tar tid å fjerne ladningsbærere fra basisområde. Spesilet hvis transistoren har gått i metning (Saturation) – vil det ta tid å gjenopprette sperresjiktet mellom basis og kollektor.

(saturation = begge diodene - base-emitter og base-kollektor er forspendte i lederetning). 4

Propagation delay t_p
High-Low Low-High

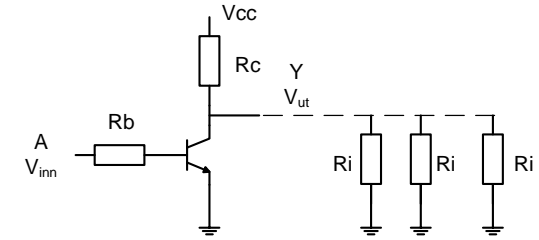
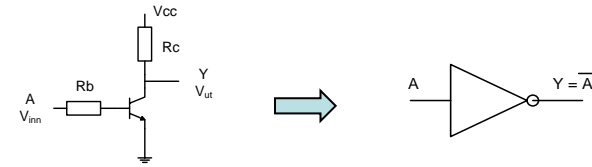
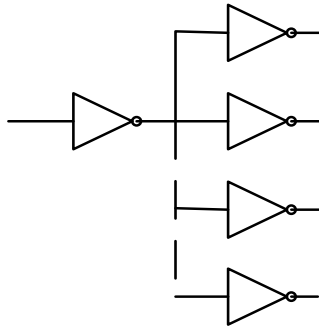
DIGITALE kretser og systemer

BJT som bryter (inverter)

Innføring i noen begreper

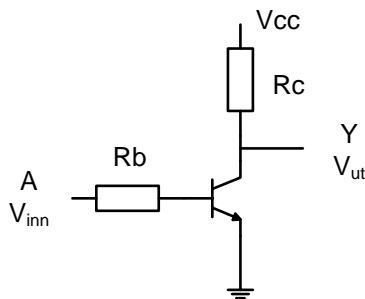
Fan – out

Hvor mange tilsvarende kretser klarer en utgang å drive ..



$$V_{ut\ HIGH} = V_{CC} \cdot \frac{R_{i1} \parallel R_{i2} \cdot \parallel R_{iN}}{R_C} \geq \text{Def. "1"}$$

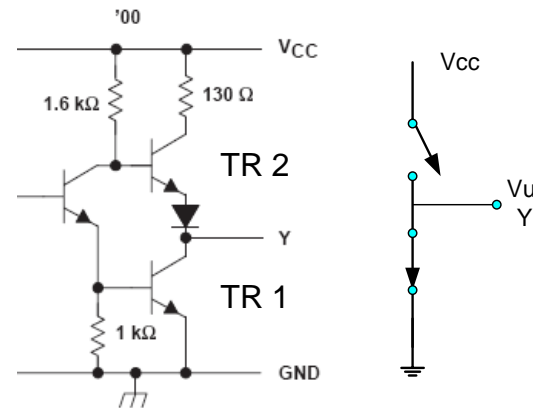
$Power = U \cdot I = R \cdot I^2$



Når transistoren er "PÅ" går det en stor strøm gjennom Rc. Kretsen blir "varm"
Alternativ koplning

Totem-pole output

TR2 sitter på toppen av TR1

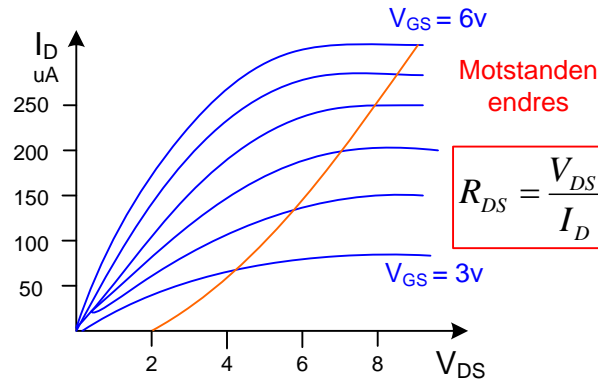
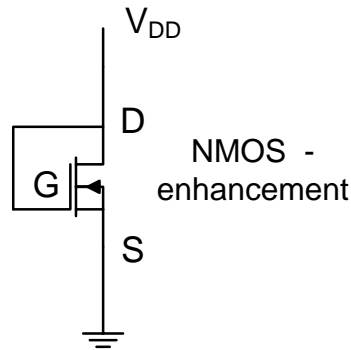


To transistorer i serie hindrer stor strøm ved "0" ut – TR 1 = PÅ TR 2 = AV



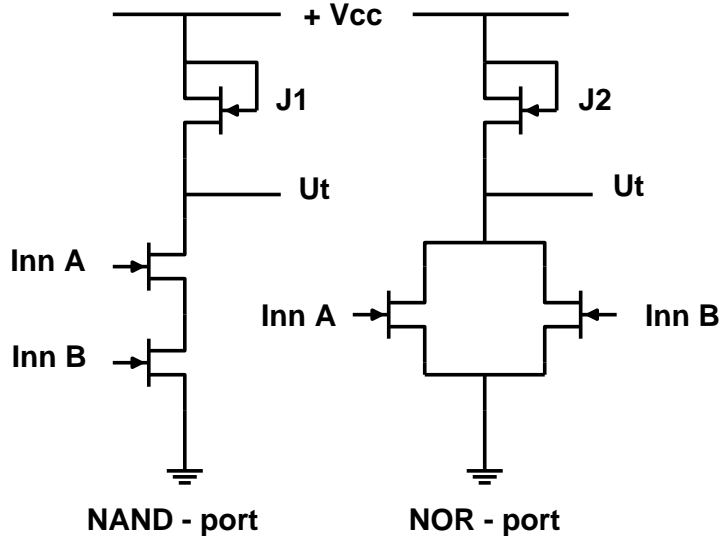
DIGITALE kretser og systemer

MOSFET som motstand



Når vi kopler Gate til Drain - opptrer forbindelsen Source - Drain som en ikkelineær motstand

MOS - logikk (N- kanal)



Vi fremstiller i dag de fleste "logiske kretser" i NMOS eller CMOS teknologi. Da vi kan bruke NMOS-transistorer som motstander forenkler dette realiseringen av mange kretser. Vi behøver ikke egne produksjonsmasker for å fremstille motstander. Figuren til venstre viser hvordan J1 og J2 blir brukt som lastmotstander når vi realiserer NAND- og NOR-funksjoner

DIGITALE kretser og systemer

Kretsfamilier

Bipolare komponenter (BJT) brukes i 3 typer logiske kretser: DTL, TTL og ECL.

DTL - Diode Transistor Logic (- historisk – første generasjon 1952-1963)

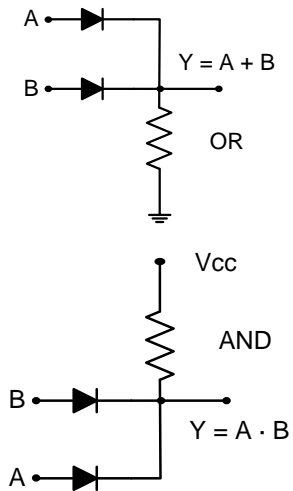
TTL – Transistor Transistor Logic (- brukes fortsatt i mange systemer – 2005)

ECL - Emitter Coupled Logic (- mye brukt i raske datamaskiner 1970 -1990)

Unipolare komponenter (FET) brukes i logiske kretser med **NMOS** og **CMOS** teknologi. Teknologien rundt FET utvikles kontinuerlig. Brukes i dagens prosessorer.

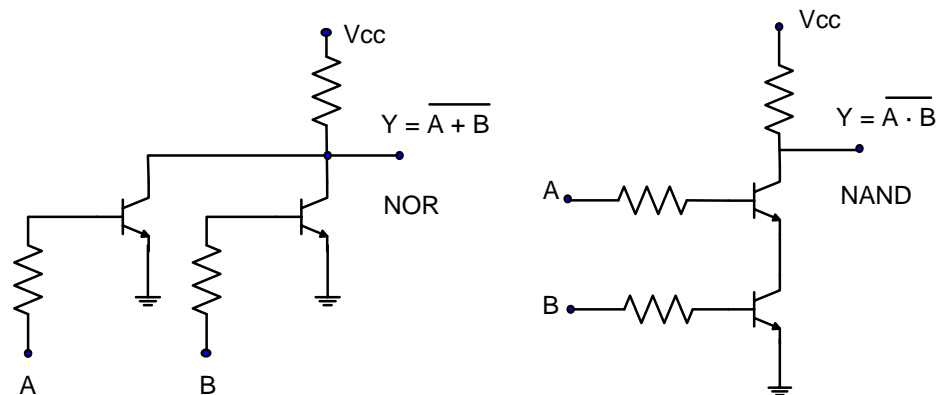
Historiske kretsfamilier

Diode Logikk (DL)



Diodelogikk ..
Egner seg dårlig til større logiske nettverk.
Godt egnet til enkle logiske funksjoner

Resistor Transistor Logikk (RTL)

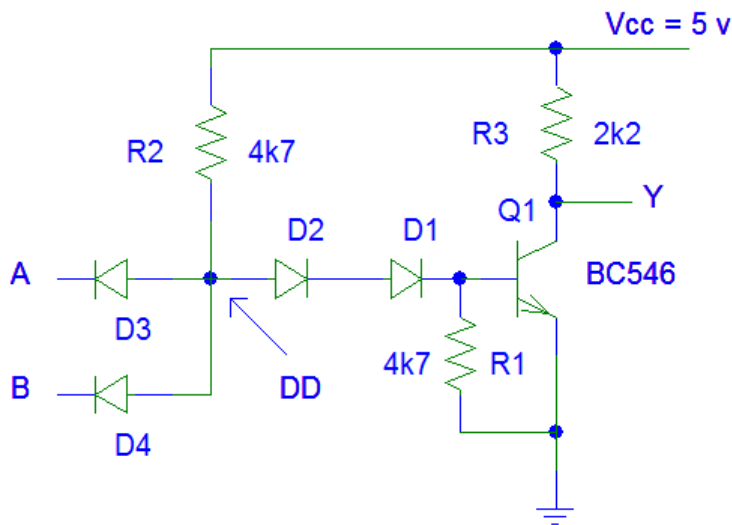


Kan brukes i større systemer - men trekker stor strøm.
Er "langsom" – store forsinkelser i t_{pHL} og t_{pLH}

DIGITALE kretser og systemer

Kretsfamilier **DTL** - Diode Transistor Logic

2-input NAND - laboppgave # 4



Hvis A og B ikke tilkoples noen ytre krets vil **transistoren Q1 være "on" : Y = "0"**.
 Det går en strøm opp igjennom Emitter Base - dioden, videre gjennom diodene D1 og D2 – og motstand R2 til Vcc.

Spenningen i punktet DD blir ca 2,1volt.
 ($V_{be} + V_{D1} + V_{D2} = 0,7v + 0,7v + 0,7v = 2,1volt$)

NAND

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

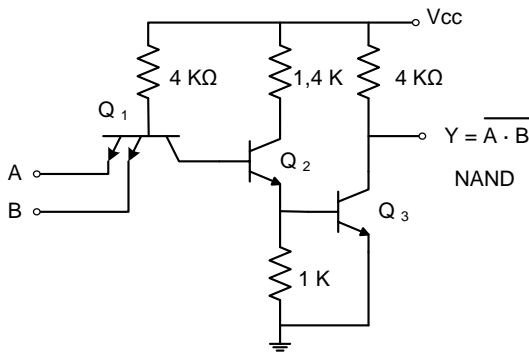
Hvis en av inngangene – A eller B legges til "jord" ("0") blir spenningen i DD = 0,7 volt. Det betyr at spenningen i DD ikke er høy nok til å løfte Vbe over 0,7v.

Transistor Q1 er "off" : Y = "1"

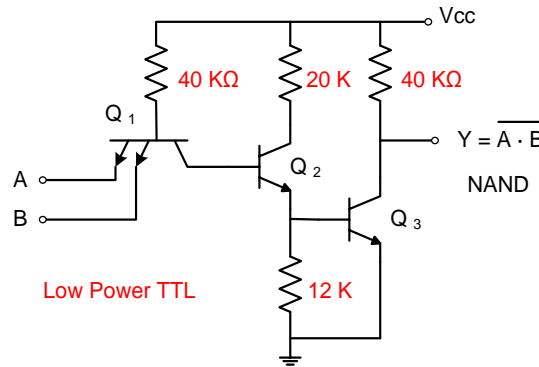
DIGITALE kretser og systemer

Kretsfamilier TTL – Low power (L) - Schottky (S) og (LS)

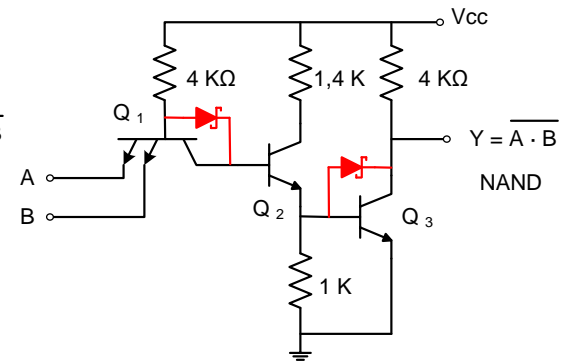
Problemer med hastighet og power (varme) – noen alternativer



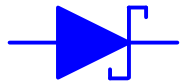
Standard TTL



74L00 – Low Power : Øker verdien på interne motstander. Low-Power men samtidig Low Speed



74S00 High Speed: Innfører Schottky-diode mellom Base og kollektor.



Schottkydioden (etter Tysk fysiker Walter H. Schottky) bruker en metall – halvleder junction. Ofte Al og n-dopet Si. Denne overgangen resulterer i en diode med meget rask "switching time" og meget lav diodespenning i lederetning - ca. 0,3volt.

Kopler vi en Schottkydiode mellom basis og kollektor på transistoren hindrer vi at den går i metning (saturation). Dioden begynner å lede alt ved 0,3volt – men BC-dioden krever 0,7volt før den leder. – Sperresjiktet over BC-dioden opprettholdes.

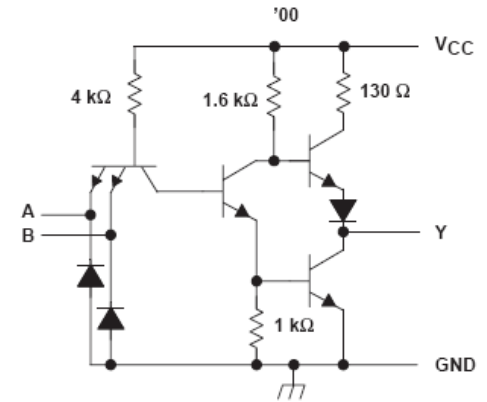
DIGITALE kretser og systemer

Kretsfamilier TTL – transistor-transistor-logikk

Kretsene viser koplingen internt og i SN7400 quadruple two input NAND gates. Datablad fra Texas Instruments.

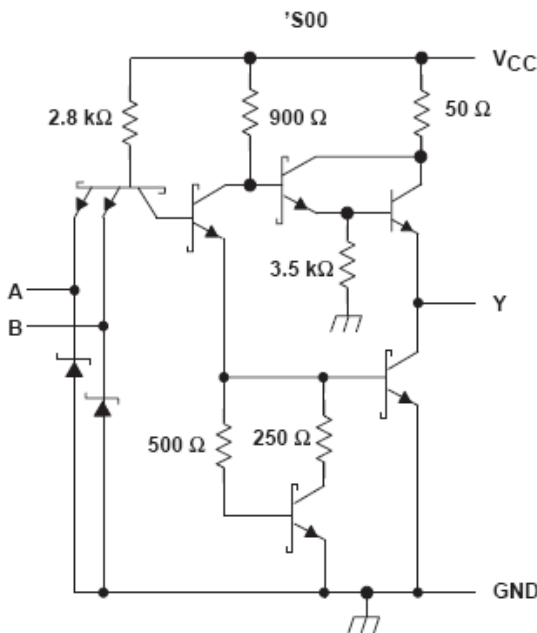
3 utgaver –

TTL 7400, Schottky 74S00, Low Power Schottky 74LS00

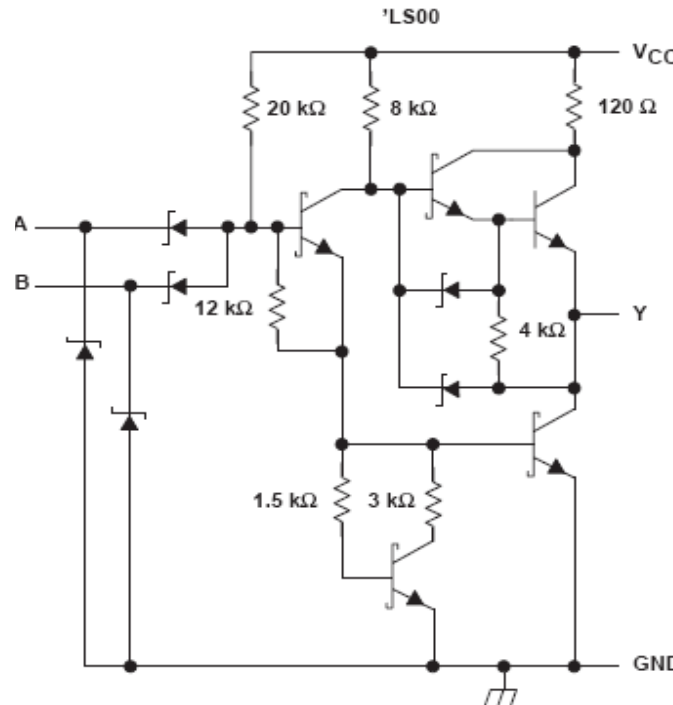


FUNCTION TABLE
(each gate)

INPUTS		OUTPUT
A	B	Y
H	H	L
L	X	H
X	L	H

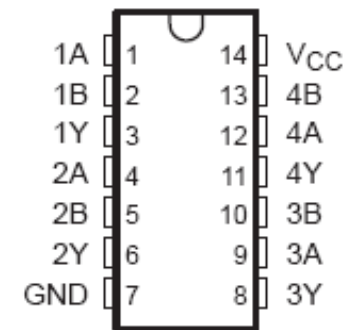


Schottky



Low Power Schottky
Bemerk motstandsverdiene

(TOP VIEW)



Standard TTL – DIL
I 1964 standardiseres
7400-serien.

DIGITALE kretser og systemer

Kretsfamilier TTL – transistor-transistor-logikk

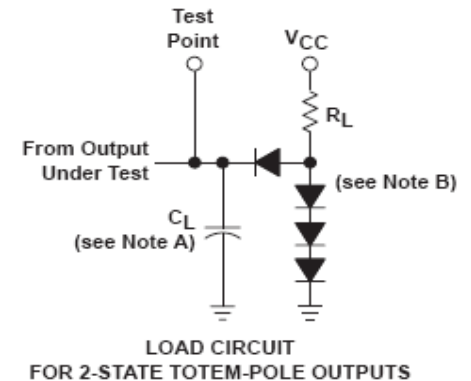
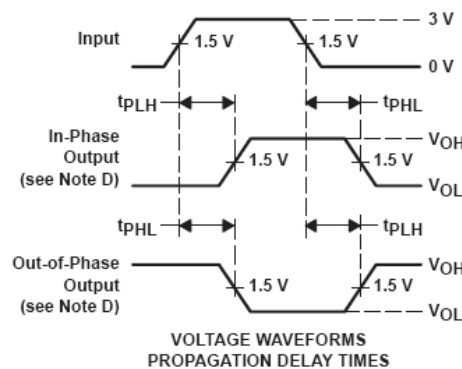
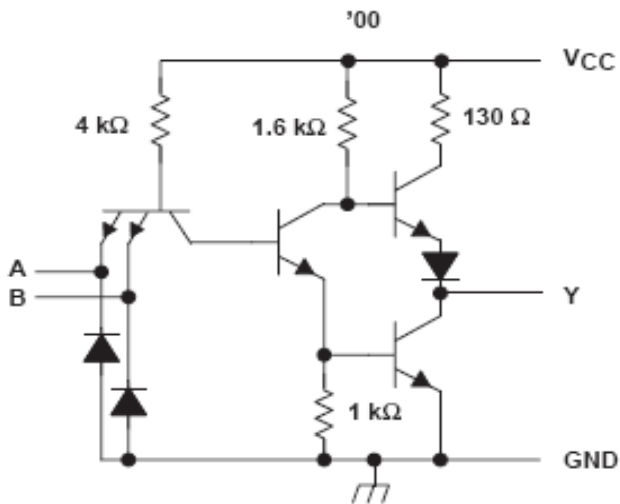
SN7400 quadruple two input NAND gates. Datablad fra Texas Instruments.



switching characteristics, $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$ (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	SN5400 SN7400			UNIT
				MIN	TYP	MAX	
t_{PLH}	A or B	Y	$R_L = 400\ \Omega$, $C_L = 15\text{ pF}$	11	22		ns
t_{PHL}				7	15		

		SN7400			UNIT
		MIN	NOM	MAX	
V_{CC}	Supply voltage	4.75	5	5.25	V
V_{IH}	High-level input voltage	2			V
V_{IL}	Low-level input voltage	0.8			V
I_{OH}	High-level output current	-0.4			mA
I_{OL}	Low-level output current	16			mA
T_A	Operating free-air temperature	0	70		$^\circ\text{C}$



DIGITALE kretser og systemer

Kombinatoriske kretser – Addisjon av binære tall

Hvert bit har to mulige tilstander/verdier
- 0 eller 1 Dvs. 4 mulige kombinasjoner:

$$0 + 0 = 0$$

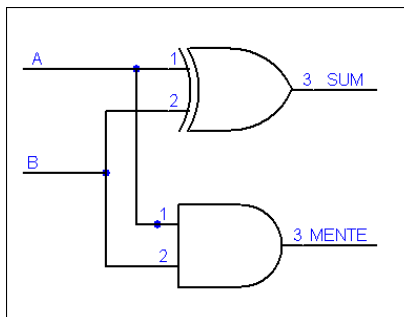
$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 + \text{mente (1)}$$

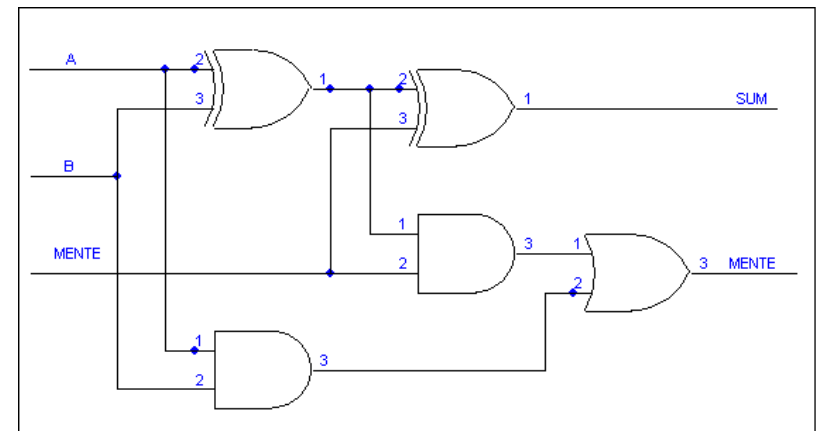
Den siste kombinasjonen, 1 + 1 viser at addisjonen kan ha et resultat på 2 bit.

Kretsen under mangler mente inn – kalles derfor en "Halv-adder"



Halv-adder mangler mente inn

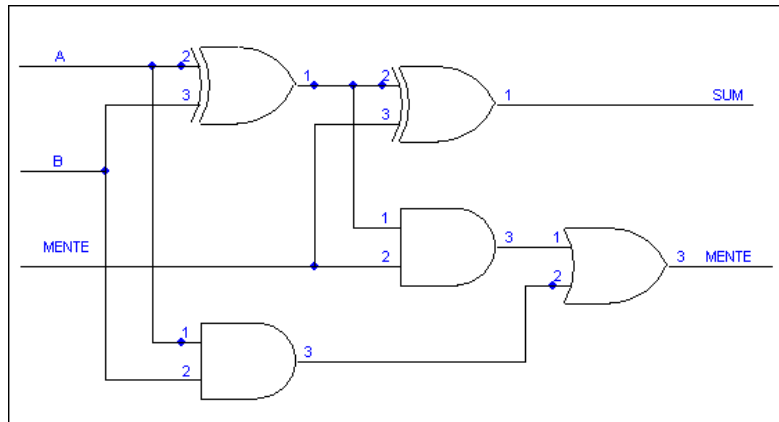
INPUT		OUTPUT	
A	B	MENTE	SUM
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



Full adder – mente inn

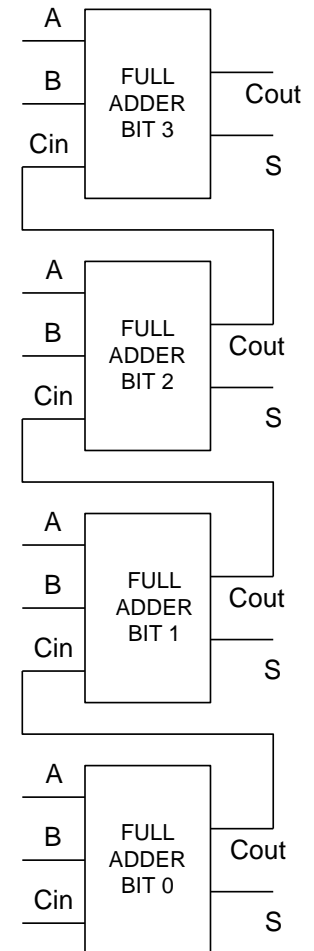
DIGITALE kretser og systemer

Kombinatoriske kretser – Addisjon av binære tall



INPUTS			OUTPUTS	
A	B	Mente inn	Mente ut	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Distribusjonen av mente. Minst signifikant bit nederst i rekka. Vi må vente med avlesning av resultatet til vi er helt sikker på at "mente" har forplantet seg opp gjennom alle kretsene..

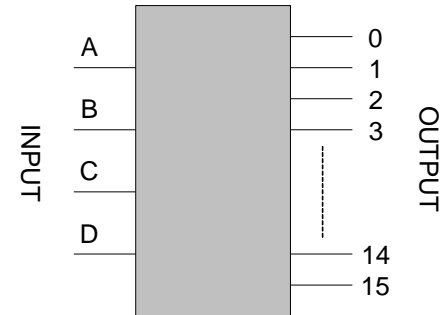


DIGITALE kretser og systemer

Kombinatoriske kretser – Dekodere / enkodere

Dekoder

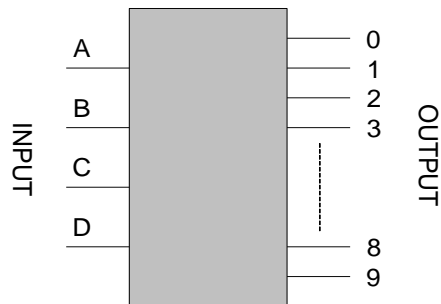
System som mottar en N-bit kode og som etablerer en (og bare en) "1" på 2N utgangslinjer.



Figur 4. 4 BIT til 16 linjers dekode

BCD til desimal dekode

BCD = Binær kodet desimal. En "8 4 2 1 kode" hvor alle binærkodene - opp til 1001 aksepteres. Se Tabell 1 under.



Figur 5. 4 bit BCD til desimal dekode

A	B	C	D	Ut
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

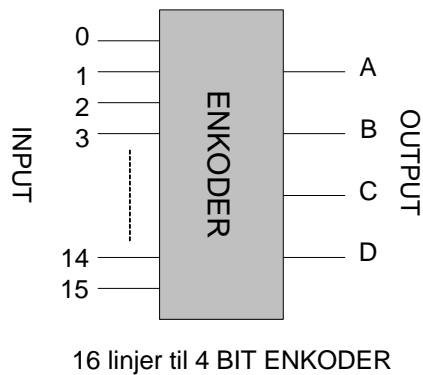
Tabell 1

DIGITALE kretser og systemer

Kombinatoriske kretser – Dekodere / enkodere

Enkoder

Den inverse prosess til dekodning. For hver "1" på en av N linjer genereres en N-bit kode. Se Tabell 2.



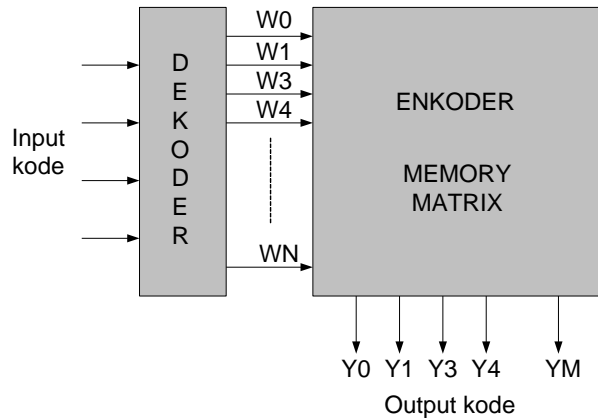
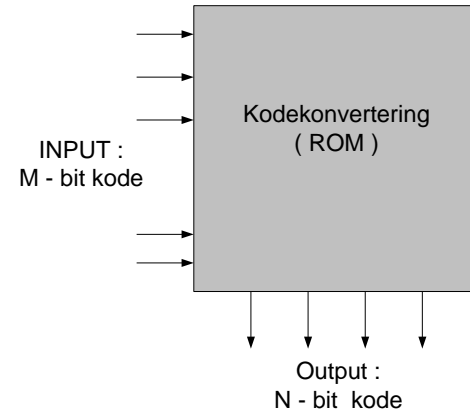
															A	B	C	D
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1
8	0									1						1	0	0
9	0										1					1	0	1
A	0											1				1	0	1
B	0												1			1	0	1
C	0													1		1	1	0
D	0														1	1	0	1
E	0															1	1	1
F	0																1	1

Tabell 2

DIGITALE kretser og systemer

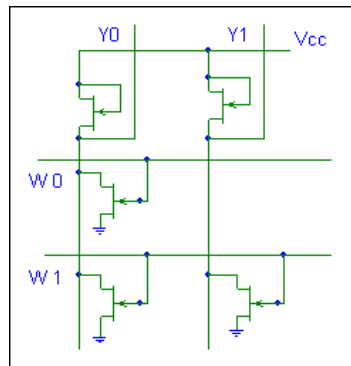
Kombinatoriske kretser –

READ – ONLY MEMORY (ROM)



Den funksjonelle relasjon mellom “output” og “input” er bygget inn i ”hardware”. Informasjonen er lagret permanent.

Den interne organiseringen er ofte gjort som vist på figuren – Først en DEKODER som så setter opp en entydig forbindelse inn til en ENKODER MEMORY MATRIX.



W0	W1	Y0	Y1
0	0	1	1
0	1	0	0
1	0	0	1

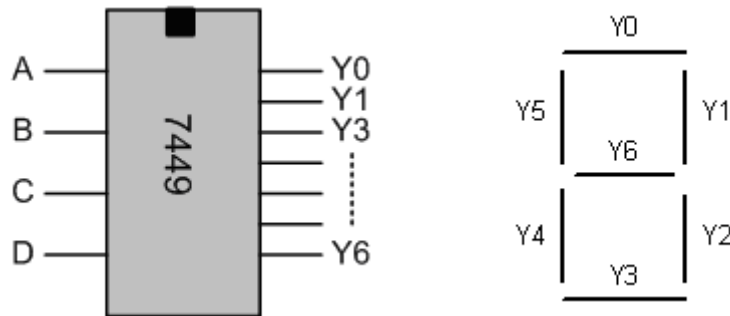
En encoder matrise kan være koplet som en permanent sammenkopling mellom transistorer som vist i figuren til venstre. Tabellen viser sammenhengen mellom input W og output Y

DIGITALE kretser og systemer

Kombinatoriske kretser –

ROM applikasjoner

ROM kan brukes til oppslagstabeller for trigonometriske funksjoner logaritmer etc. Et typisk eksempel vil være en driver for de 7 lysdiodene i et siffer – display:



DIGITALE kretser og systemer

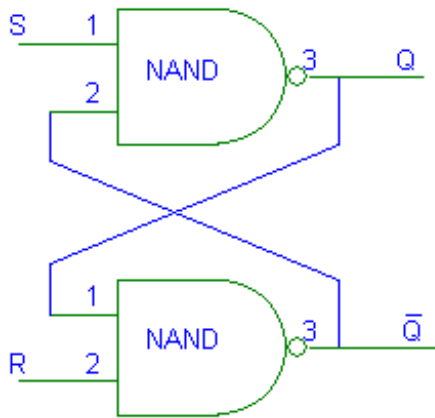
Sekvensielle systemer

1-bit lager (LATCH)

Skal en logisk krets "huske" en tilstand – og opprettholde denne tilstanden - må kretsen inneholde en form for tilbakekopling (feedback).

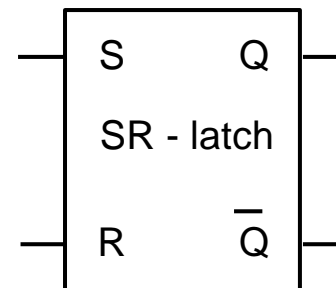
En latch er en krets som kan eksistere i en av to stabile tilstander, enten med utgangen $Q = 1$ som kalles "1" tilstand, eller med $Q = 0$ som kalles "0" tilstand. En latch kan realiseres vha. både NAND og NOR-porter.

Kretsen under er en NAND- latch.



Inngangene er ofte tilordnet bokstavene "S" og "R" som står for henholdsvis "SET" og "RESET". Tabellen under viser sammenhengen mellom kontrollsignaler og tilstanden på utgangene.

Input		Output	
R	S	Q	\bar{Q}
0	0	Q	\bar{Q}
1	0	0	1
0	1	1	0
1	1	Ubestemt	



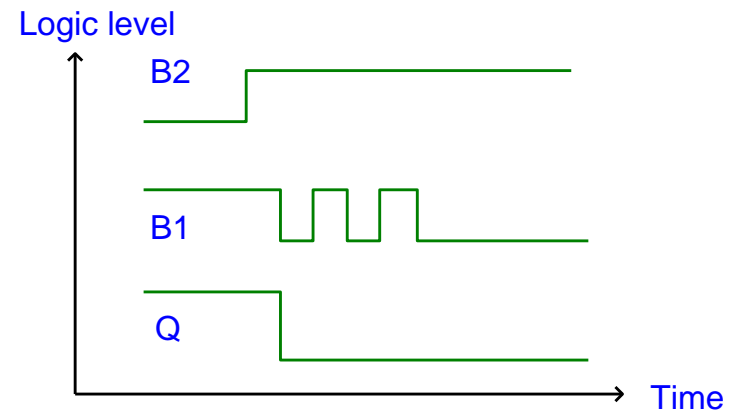
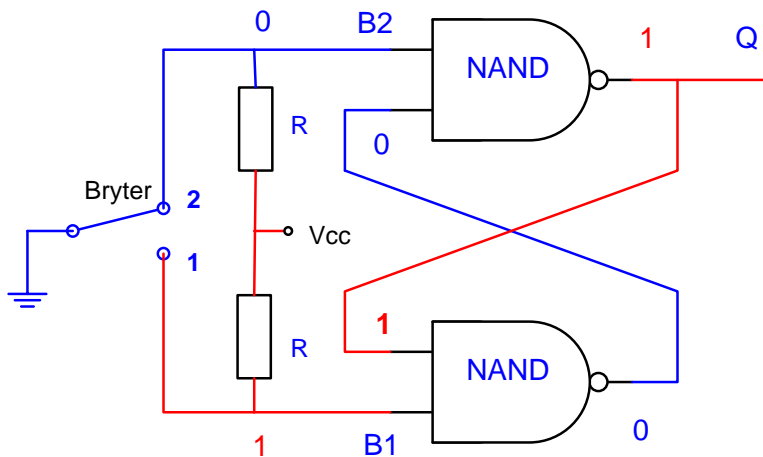
DIGITALE kretser og systemer

Sekvensielle systemer

1-bit lager (LATCH)

Bruksområde for en latch begrenset. Men når vi skal kople en mekanisk bryter til et digitalt system må vi være sikker på at bryteren gir en – og bare en – puls inn til systemet. Dette kan være problematisk da mekaniske brytere kan "prelle" – dvs. det kan oppstå en rekke små korte pulser før bryteren faller til ro.

Med en latch vil utgangen Q låses til riktig verdi ved første kontakt. Se figuren under . Vi slår bryteren fra posisjon 2 til 1.



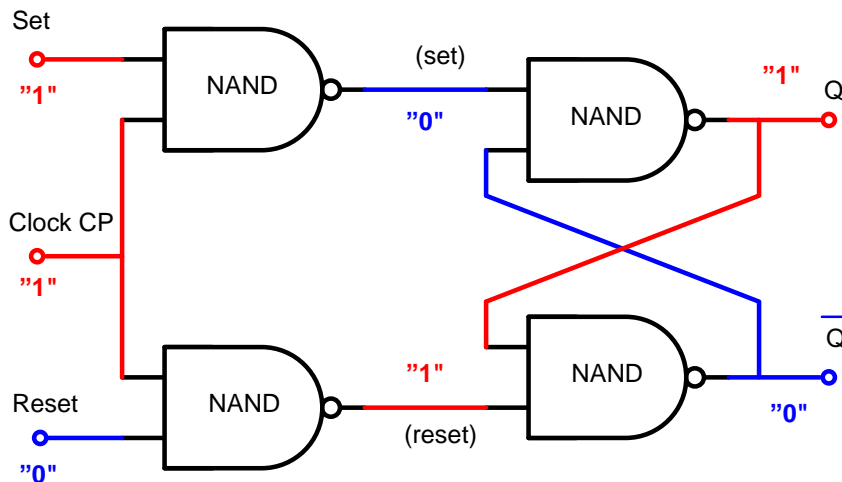
DIGITALE kretser og systemer

Sekvensielle systemer

Klokkestyrt LATCH

Skal vi bruke LATCH som et aktivt element i digitale "klokkestyrte" systemer må vi tilføre kretsen ekstra styrelogikk. Vi legger på 2 NAND-porter foran latchesen. Klokkesignalet (CP) bestemmer om latchesen skal settes eller resettes. Det er først når CP går til "1" at signalet på "Set" og "Reset" vil komme igjennom NAND-portene slik at de kan endre tilstand på utgangen Q.

Men vi kan få et problem hvis både "Set" og "Reset" blir "1" samtidig. Dette kan vi løse med en J K LATCH (flip flop)



Input		Output	
<i>R</i>	<i>S</i>	<i>Q</i>	\bar{Q}
0	0	<i>Q</i>	\bar{Q}
1	0	0	1
0	1	1	0
1	1	Ubestemt	

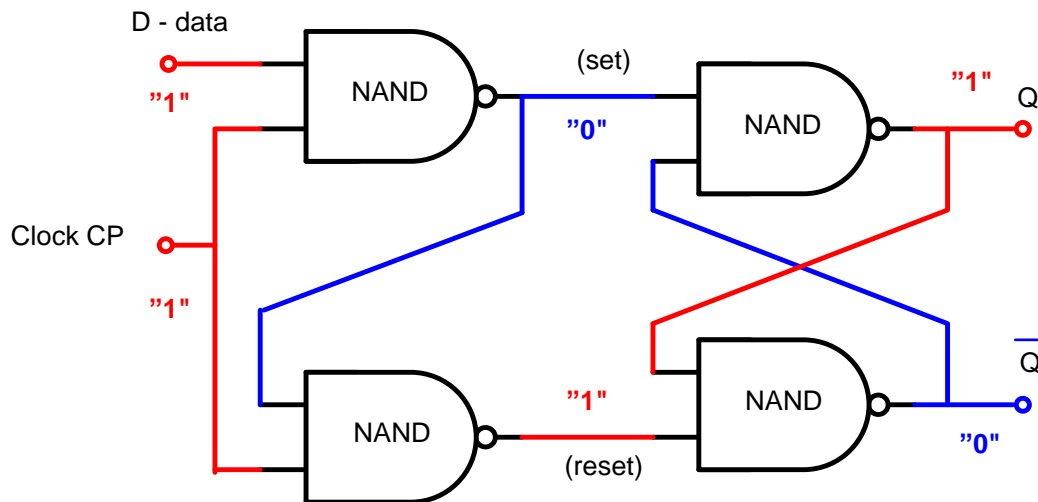
DIGITALE kretser og systemer

Sekvensielle systemer

D-LATCH (Data LATCH)

Utgangen Q vil bli satt til verdien på D-inngangen når klokkesignalet går til "1". Vær oppmerksom på at så lenge klokka er høy ("1") kan D endre verdi – og denne verdien vil legges ut til Q.

Når klokka går fra "1" til "0" vil D-verdien bli "låst" til Q-utgangen.



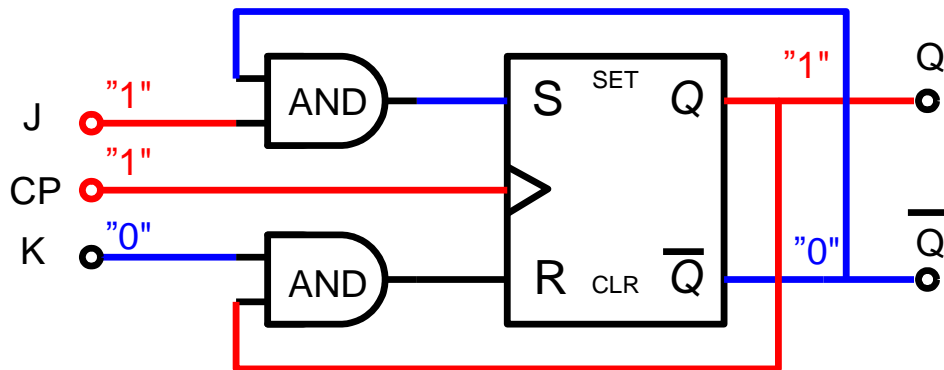
Input		Output	
<i>D</i>	<i>CP</i>	<i>Q</i>	\bar{Q}
0	0	<i>Q</i>	\bar{Q}
1	0	<i>Q</i>	\bar{Q}
0	1	0	1
1	1	1	0

DIGITALE kretser og systemer

Sekvensielle systemer

J K LATCH (flip flop)

I den klokkestyrte "latchen" har vi et problem hvis "Set" og "Reset" begge er "1". Vi får da en ubestemt tilstand på Q. Dette kan vi unngå hvis vi lar "S" og "R" legges inn til en AND- port som styres av "latchens" Q-utganger. Se figuren under.



J	K	Q
0	0	Q
1	0	1
0	1	0
1	1	Q'

Ulempen med denne kretsen er at - så lenge CP er høy kan utgangene variere.

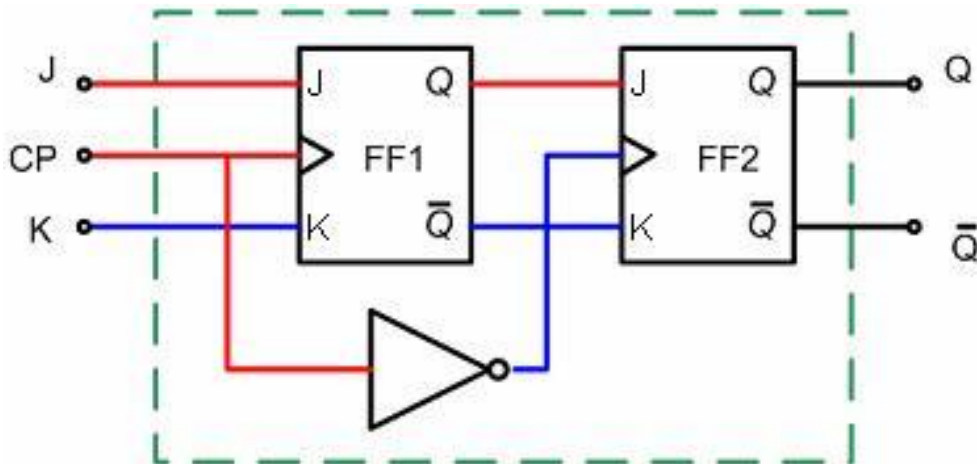
DIGITALE kretser og systemer

Sekvensielle systemer

Master – Slave Flip- Flop

Utgangen på FF1 settes når klokkepulsen (CP) er høy, ”1”. Verdiene bestemmes av nivåene på J og K. Utgangene fra FF1 koples ikke frem til utgangene på FF2. Dette skyldes inverteren som kopler klokkepulsens utgang til FF2. Så lenge CP på FF1 er høy ”1” – vil klokkeinngangen på FF2 være lav, ”0” - og følgelig koples ikke nivåene på J og K til utgangen.

I det øyeblikk klokkepulsens (CP) på FF1 skifter fra høy til lav vil inverteren sikre at FF2 får ”1” på sin klokkeinngang. Nivåene som i dette øyeblikk befinner seg på J og K låses nå til utgangen på FF2.



J	K	Q
0	0	Q
1	0	1
0	1	0
1	1	Q'

Transisjonstabell for master-slave Flip Flop. Settes på negativ flanke av klokkepulsens

DIGITALE kretser og systemer

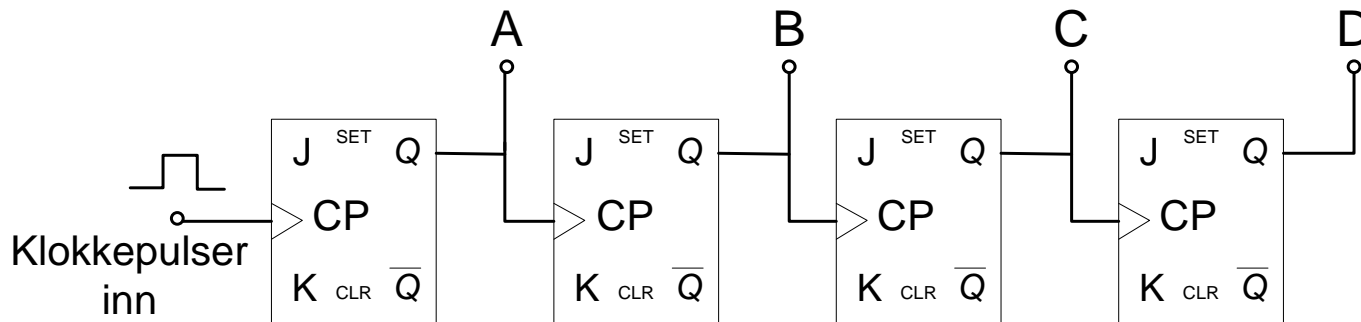
Sekvensielle kretser – tellere og registre

Binærteller

Tellere finnes i mange utgaver – men er i sin grunnkonstruksjon svært like. Alle er bygget opp rundt flere "flip-flop" kretser av "master slave" type. Dette kan vi demonstrere med den enkleste av alle tellere – Binærtelleren.

Alle J og K er satt til "1" – det betyr at Q-utgangen endrer tilstand for hver fallende flanke på klokkeinngangen - (klokkesignalet transisjon fra "1" til "0"). Utgangen fra FF-A legges som klokkesignal inn på FF-B osv. Tilstandsendingene "ripler" nedover rekken. Vi kaller dette en rippelteller - en **asynkron krets**.

Tallet som leses ut fra denne telleren er organisert i "reversert" rekkefølge – dvs. D holder "høyeste bit" – mens A holder "laveste bit".



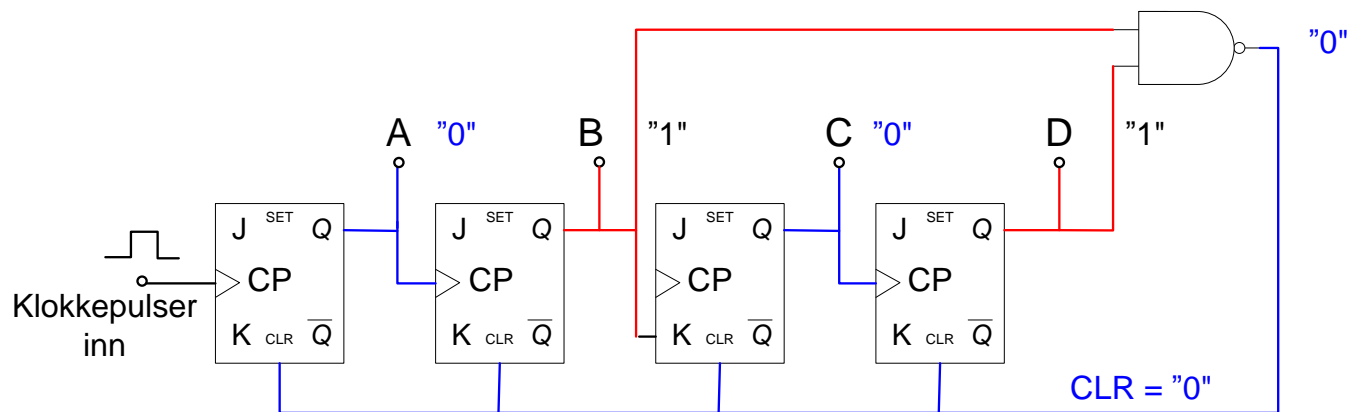
DIGITALE kretser og systemer

Sekvensielle kretser – tellere og registre

Dekadeteller

En krets som teller til 10 - etter puls #10 vil vi at kretsen returnerer tilbake til 0. Tallet 10 vil på binært format leses som 0101 på vår teller. (reversert format – mot normalt 1010 med LSB minst sign, bit først)

Vi må lage en krets som returnerer til "0" når denne bitkombinasjonen oppstår. Dette lar seg lett realisere med en kopling av fire flip-flop som vist under:



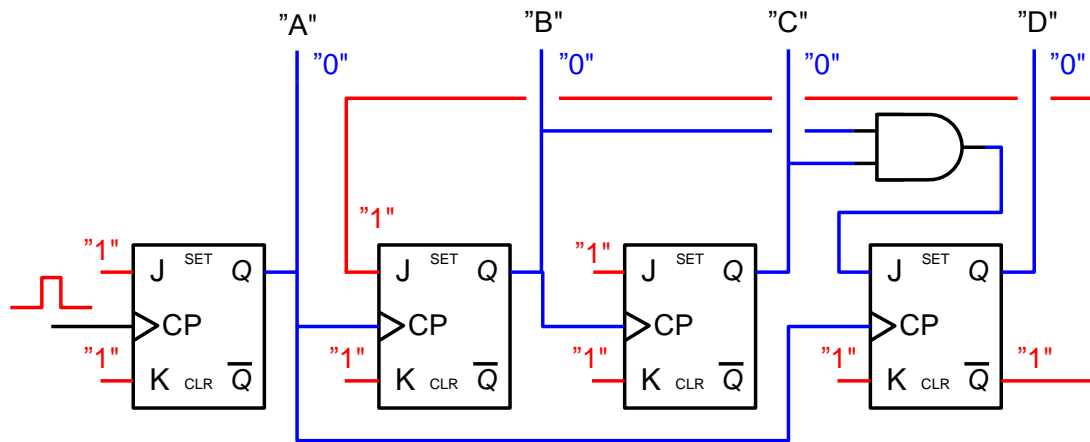
Når både B og D er "1" sendes et CLR-signal til alle FF- elementene. Denne koplingen er ikke trygg. Vi ser at når B eller D går fra "1" til "0" vil CLR-signalet fra NAND forsvinne – men vi kan ikke vite om alle FF er satt til "0".

DIGITALE kretser og systemer

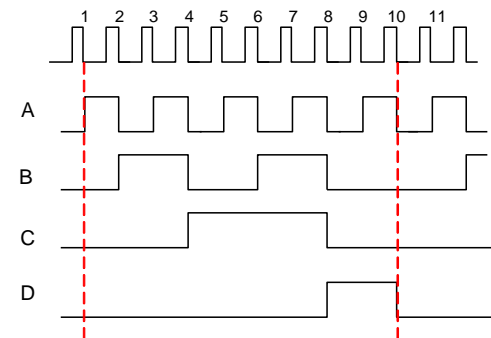
Sekvensielle kretser – tellere og registre

Dekadeteller

Vi bruker noen ekstra forbindelser og får en "trygg" dekadeteller (BCD)



Denne koplingen sikrer en trygg overgang etter 10 pulser. J og K som ikke er i bruk legges til "1". Puls # 10 (neg. flanke) setter telleren tilbake til 0000. Se transisjonsdiagrammet til høyre.



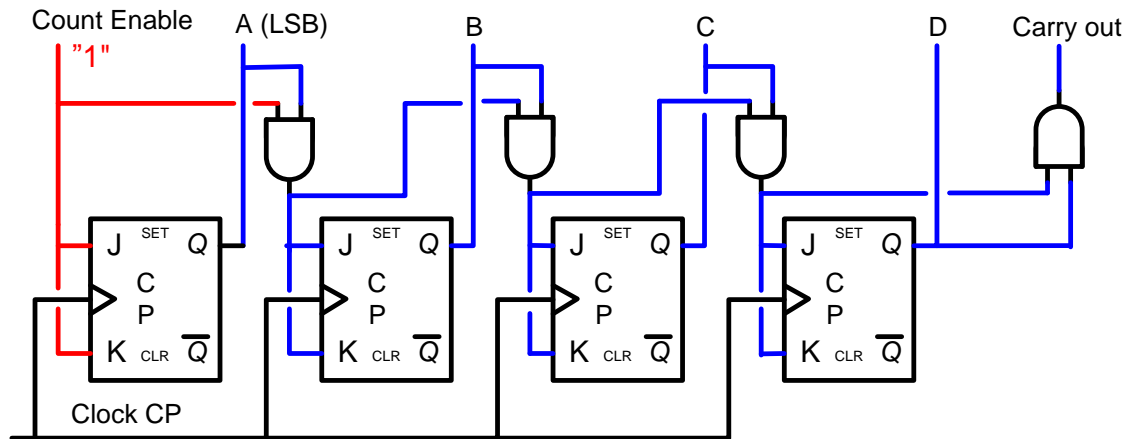
DIGITALE kretser og systemer

Sekvensielle kretser – tellere og registre

Synkronteller

På figuren viser en 4 bit binær **synkronteller** - en teller hvor alle FF klokkes samtidig – de har felles klokkesignal. (Husk rippelteller - asynkron)

J	K	Q
0	0	Q
1	0	1
0	1	0
1	1	Q'



States				Count
D	C	B	A	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Kretsen kan kort forklares slik :

A forandrer tilstand ved hver klokkepulss – fordi $J = K = 1$

B forandrer tilstand bare når $A = 1$ ($J = K = A$)

C forandrer tilstand bare når $A = B = 1$

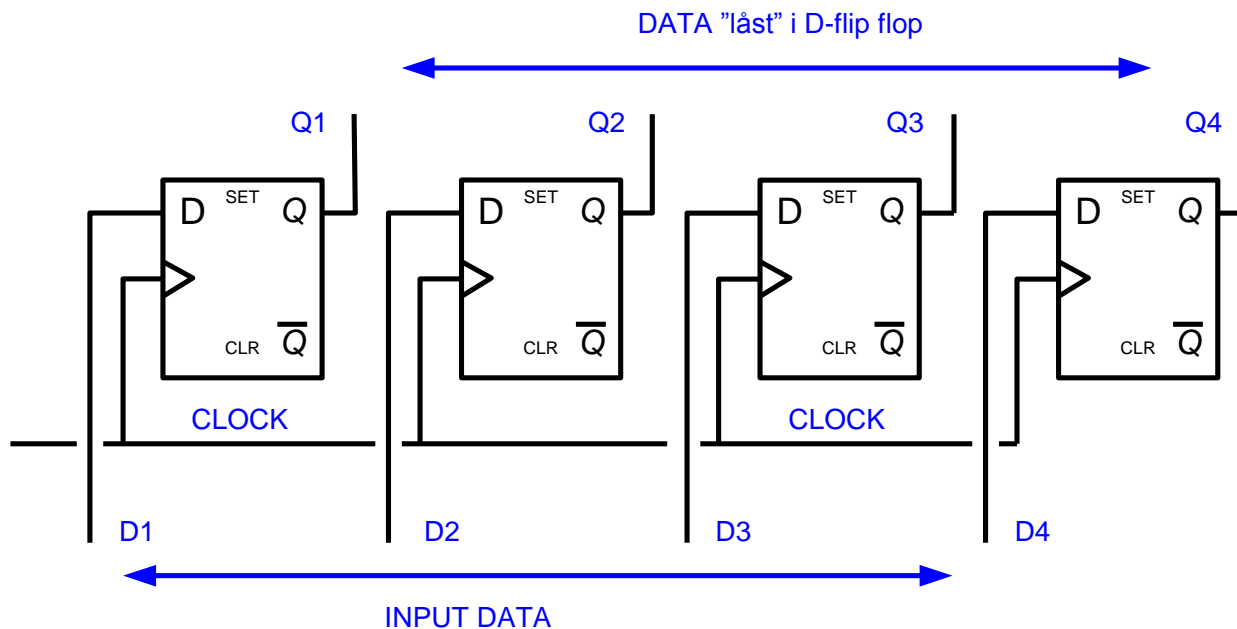
D forandrer tilstand bare når $A = B = C = 1$

DIGITALE kretser og systemer

Sekvensielle kretser – registre

Memory register

Skal et datasystemet huske binære tallverdier må vi ha en form for datahukommelse. Ved hjelp av "flip flop" av D-type (master slave) kan vi konstruere en slik hukommelse. De databit som ligger på D-inngangene når klokkesignalet går fra "1" til "0". (negativ flanke) vil bli låst til Q-utgangene. Så lenge klokkesignalet ligger statisk høy eller lav vil endringer på D-inngangene ikke forandre tilstanden på Q.

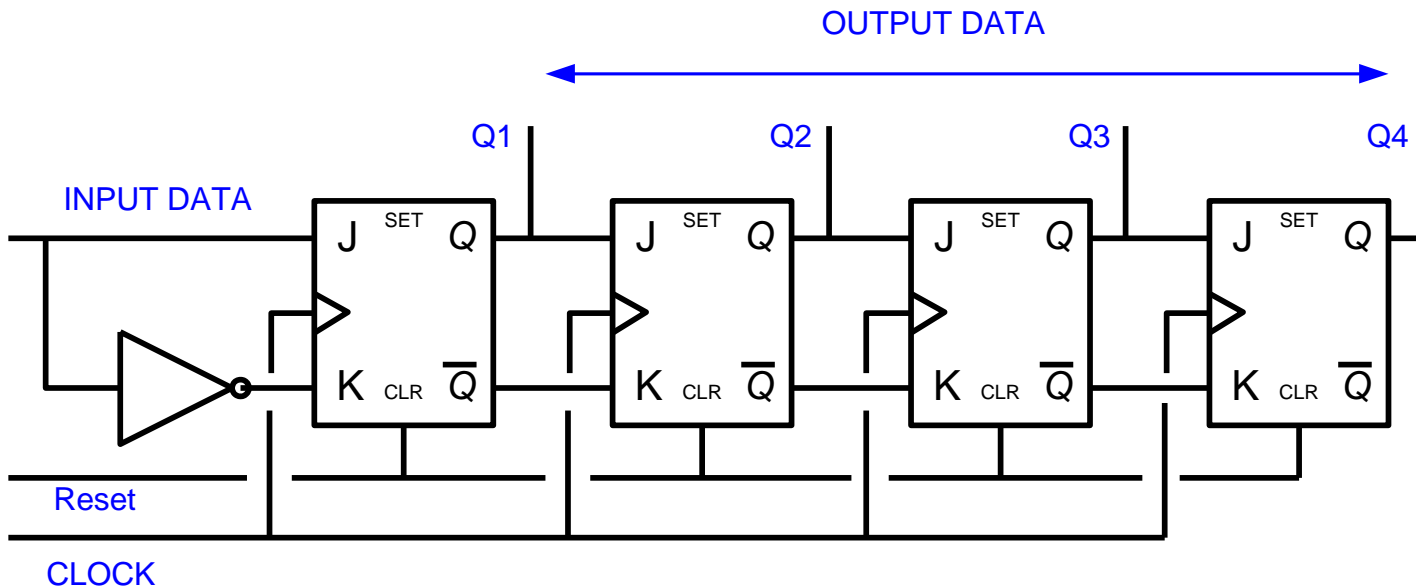


DIGITALE kretser og systemer

Sekvensielle kretser – registre

Skiftregister

Med en sammenstilling av 4 stk. JK- "flip flop" som vist på figuren kan vi klokke en seriell datastrøm (på 4 bit) inn i et register - hvoretter vi kan ta dataene ut i parallell form etter 4 klokkepulser.



Historien bak Informasjons- og Kommunikasjons Teknologien

- noen "milepæler" de første 20 år

- 1958 Første integrerte krets konstruert av Jack Kilby (Texas Instruments)
12 komponenter på samme brikke. (Kilby får Nobel-prisen i 2000)
- 1958 Programmeringsspråket ALGOL (Algebraic Language) introdusert
- 1958 Seymour Cray bygger den første transistoriserte "super computer", CDC1604
- 1959 IBM kommer med sine første transistoriserte datamaskiner; 1620 og 1790
- 1960 Den første "mini computer" utviklet av DEC. – PDP-1
- 1961 Første TTL produsert av Fairchild og Texas Instruments ,
- 1963 Første CMOS IC produsert av RCA
- 1964 Texas Instruments introduserer 74xxx serien for standardisert TTL
- 1966 Første bipolare Emitter Coupled Logic (ECL) produsert av Motorola
- 1966 Første "Single transistor DRAM" produsert av IBM
- 1967 Norsk Data stiftes. Produserer sin første NORD-1 datamaskin basert på TTL.
- 1965-67 SIMULA utvikles av Ole Johan Dahl og Kristen Nygaard.
- 1970 Første kommersielle Dynamiske RAM (DRAM) – 1KBits.
- 1971 Verdens første mikroprosessor - 4 bit - Intel 4004 - 2300 transistorer -100kHz
- 1972 Utvider 4004 prosessoren til 8 bit - Intel 8008, klokke på 200kHz
- 1972 Første hovedfagsoppgave i elektronikk på fysisk institutt, UiO som benytter en mikroprosessor – Intel 4004
- 1972 Norsk Data leverer NORD-5. Verdens første 32 bit "Super mini" datamaskin.
Operativsystemet NORD-1 TSS, Multilingual Time Sharing System, blir utviklet - oppfattes den gang som verdens mest avanserte operativsystem
- 1974 Første kommersielle mikroprosessor - Intel 8080, 6000 transistorer - 2MHz NMOS-teknologi . Brukt i Altair computer (CP/M operativsystem, G.Kildall)
- 1975 Cray-1 supercomputer introdusert (Seymour Cray har opprettet eget firma)
- 1975 Microsoft etablert av Bill Gates og Paul Allen. Skriver software til MITS/Altair
- 1978 Intel 8088/8086 med 29000 transistorer, 5 -10 MHz klokke