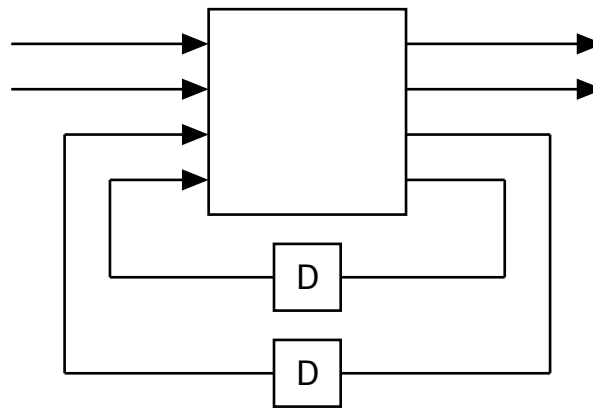
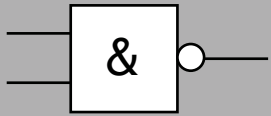


TSEA22 Digitalteknik 2019

Mattias Krylander
Ingemar Ragnemalm





Föreläsning 7. Sekv3.

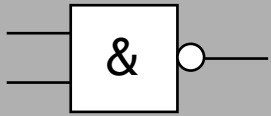
Denna föreläsning:

Lösningar närmare verkligheten

Synkronisering

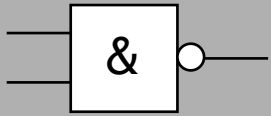
Enpulsare

Problem till design



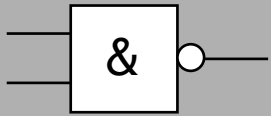
Förra föreläsningen:

- Konstruktion av sekvensnät
- Tillståndskodning
- Kodning med:
 - Mealy
 - Mealy med NOR
 - Mealy med PROM
 - Moore



Konstruktion av sekvenskretsar

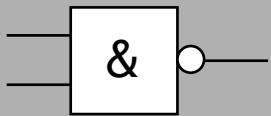
1. Specifikation
2. Tillståndsdigram
3. Tillståndskodning
4. Tillståndstabell
5. Booleska uttryck för utsignal och nästa tillstånd
6. Kretsschema



Exempel på kodningsmodeller:

Tillstånd	Binärkod	Gray-kod	One-hot-kod
S0	00	00	0001
S1	01	01	0010
S2	10	11	0100
S3	11	10	1000

Olika kodningar ger olika komplexitet.
One-hot ger fler vippor men ofta enklare nät.
Svårt att veta på förhand vad som blir enklast.

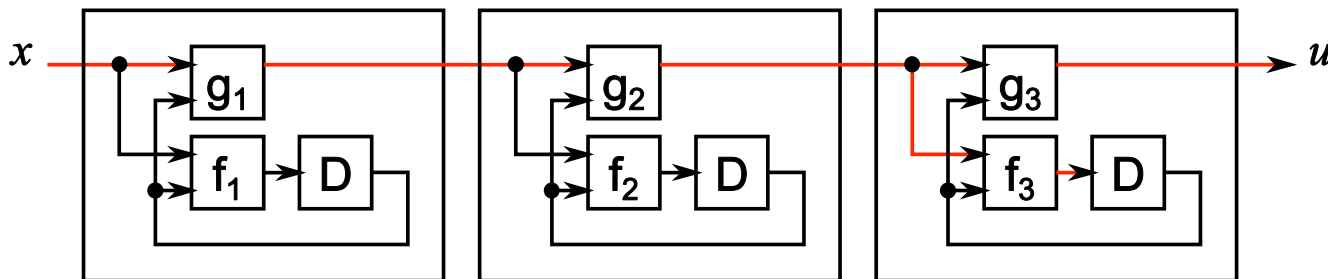


Kaskadkoppling av sekvenskretsar

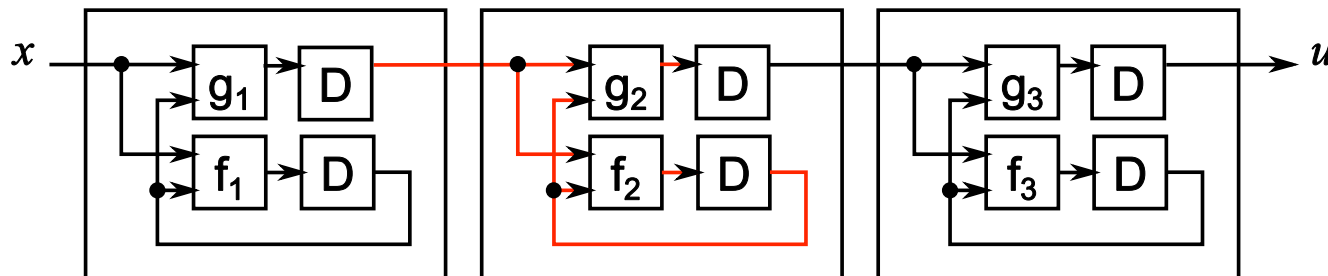
Mealy ger större grinddjup (rött) \Rightarrow tidsfördröjning \Rightarrow lägr klockfrekvens

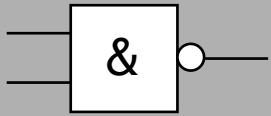
Moore tillåter snabbare klocka vid kaskadkoppling.

Kaskadkopplade Mealy-kretsar



Kaskadkopplade Moore-kretsar

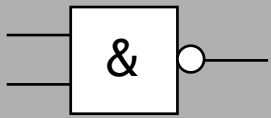




Verkligheten är alltid annorlunda

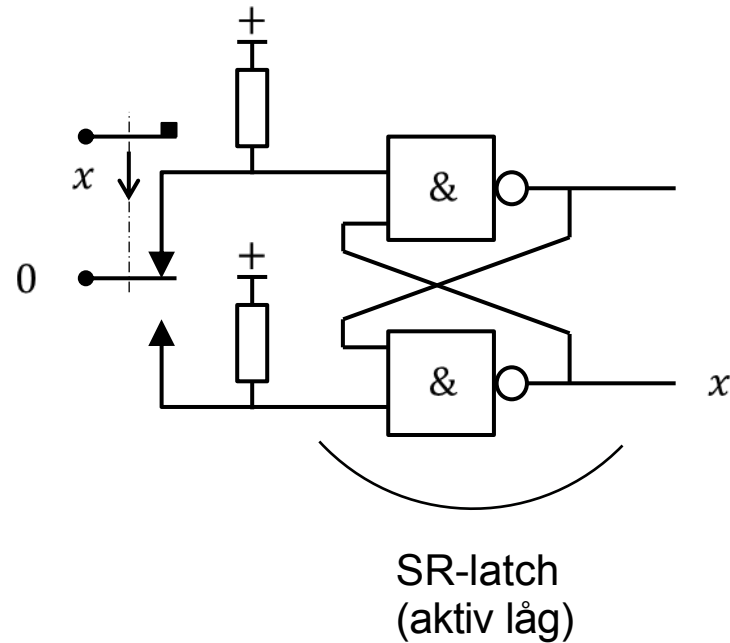
Hur kopplar vi oss till verkligheten?

- Avstudsning
- Synkronisering
- Enpulsare
- Design av lösningar till verkliga problem

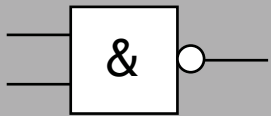


Avstudsning har vi sett förut

Enkel latch för stabilisera.



Nät för eliminering av kontaktstudsar

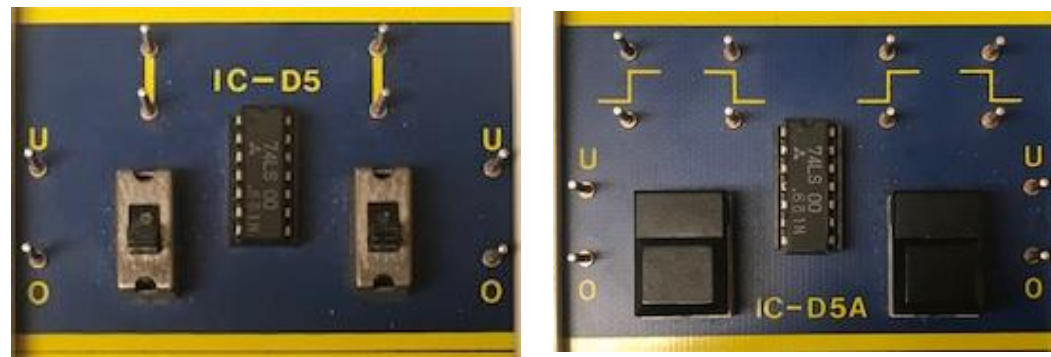


Brytare i labbet:

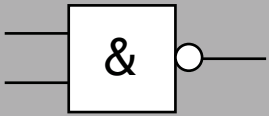
Ej avstudsade:



Avstudsade:

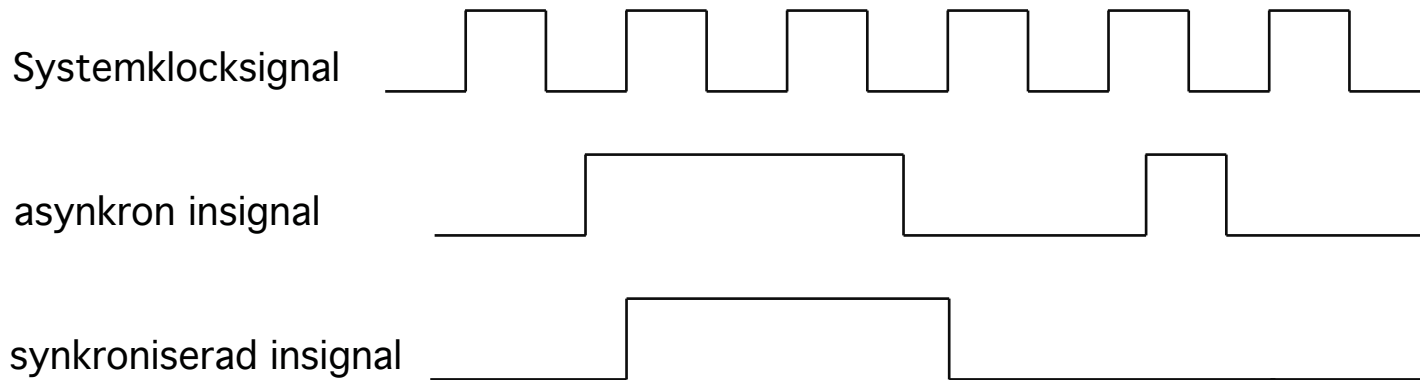
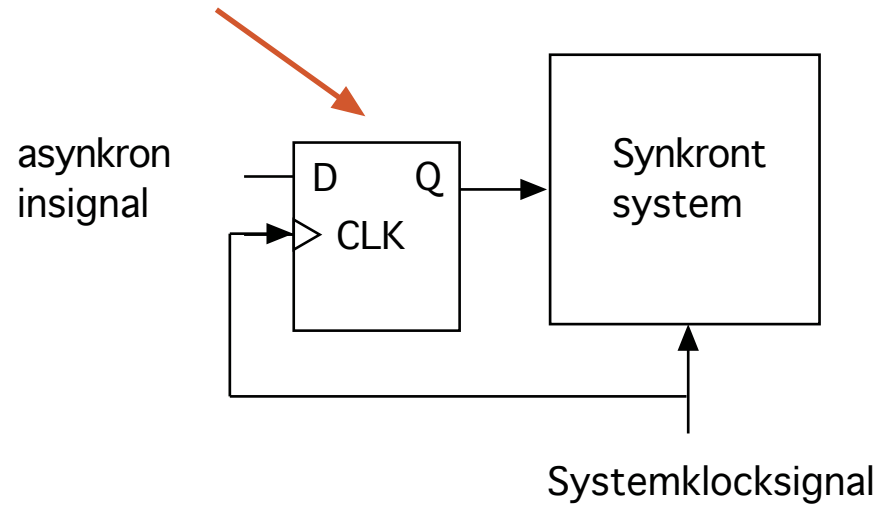


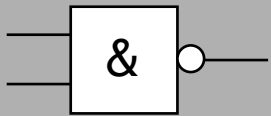
Använd dessa som insignaler till sekvenskretsar!



Synkronisering av asynkrona insignaler

Inför en *synkroniseringsvippa* på ingången



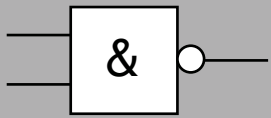


Enpulsare

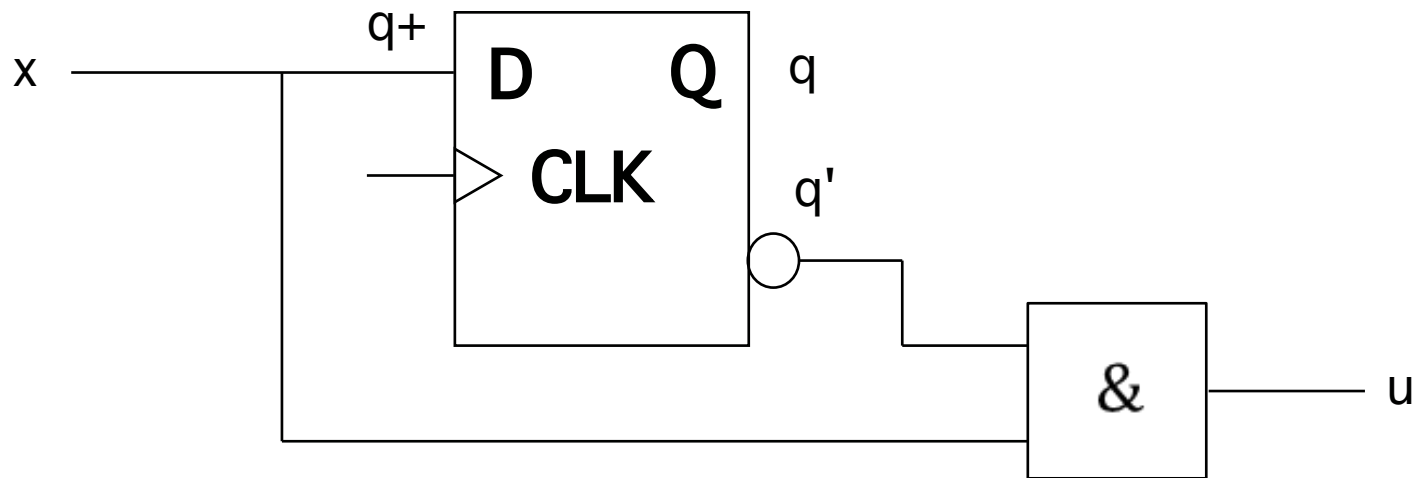
Antag att vi har en avstudsad och synkron signal.

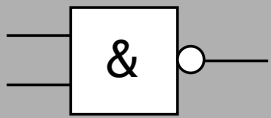
Tänk er "räkna ettor"-exemplet.

Men... hur många ettor ger ett tryck?

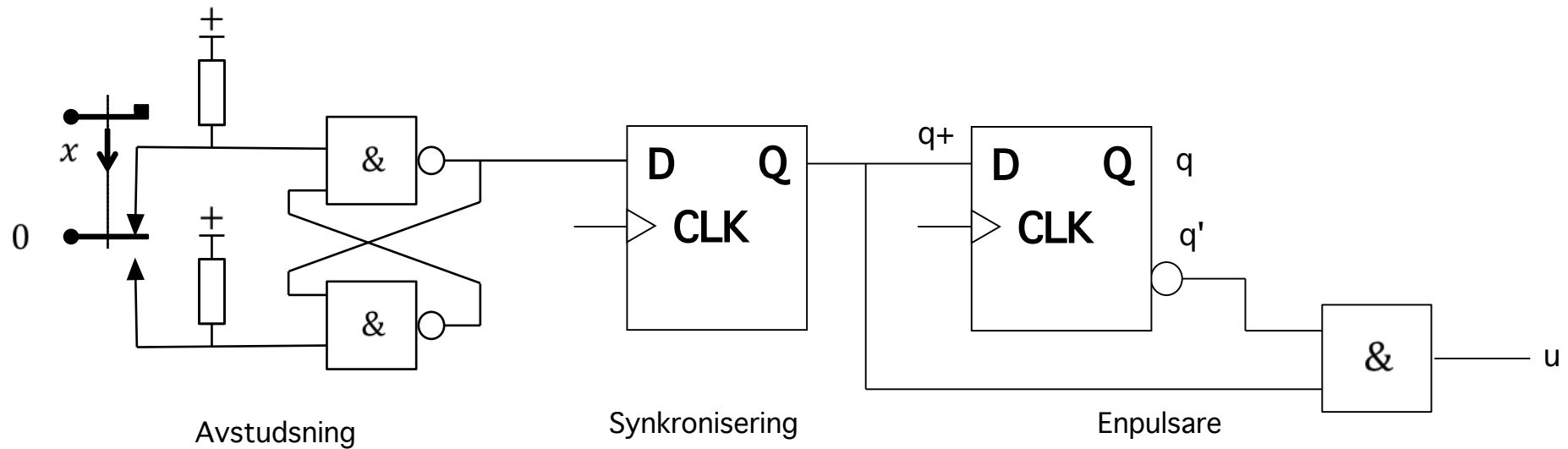


Enpulsare (Mealy)

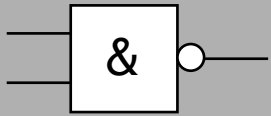




Enpulsare (Mealy) med avstudsning och synkronisering



Allt detta på 3-ettor-detektorn så börjar det likna ett verkligt system!

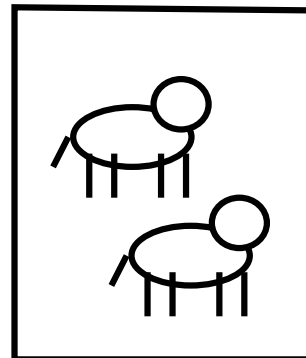


Lejonburen

Säkerhetssystem för djurskötare

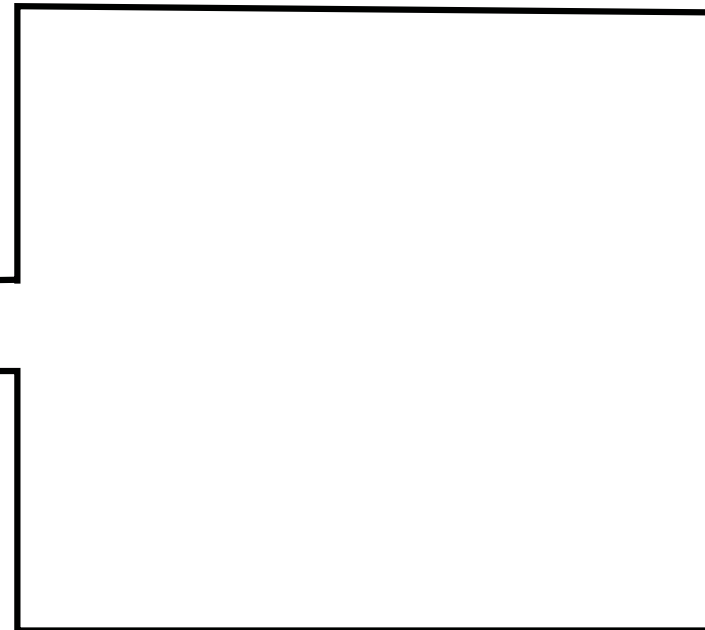
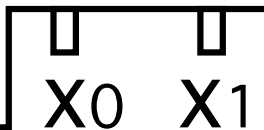
Hägn

Bur

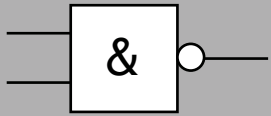


2 lejon

Port

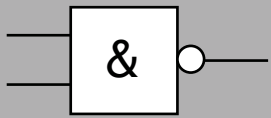


2 lejon. Lampa vid grinden skall lysa om båda lejonerna är i buren.

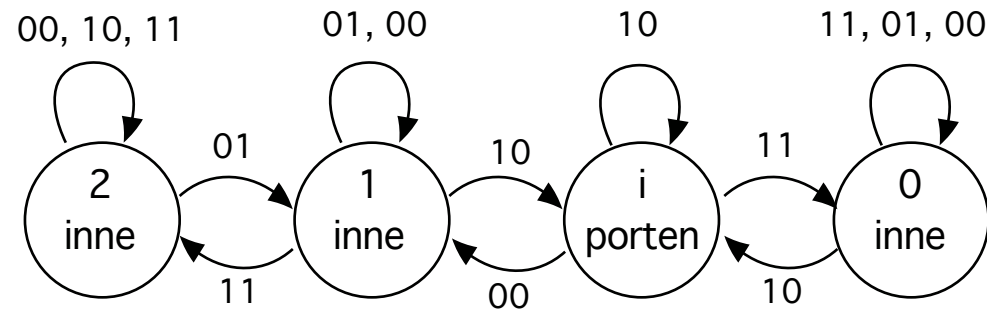


Några villkor om Lejonburen:

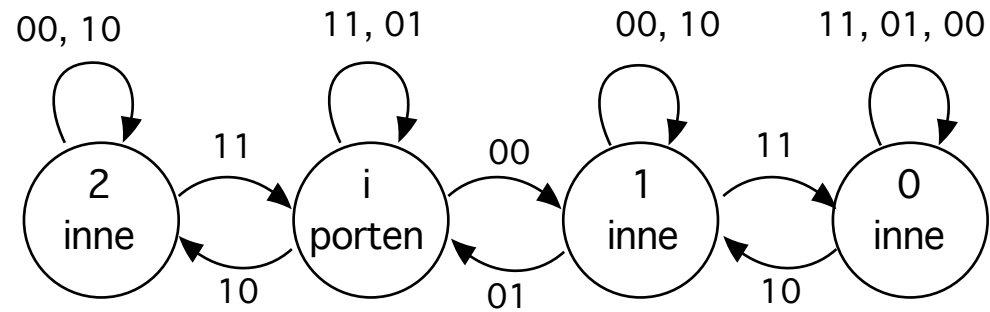
- x_0 och x_1 fotoceller. Ger 1 om fotocellen är skymd.
- Lampan: $u = 0$: Släckt, $u=1$: Tänd
- Start med båda lejonerna i buren.
- Max ett lejon i porten.
- Lejon vänder eller backar aldrig.
- Lejonerna är längre än avståndet x_0 till x_1 .
- Lejonerna rör sig långsamt relativt klockan.

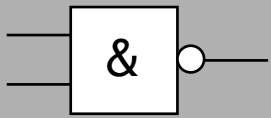


Tillståndsdigram för lejonburen



Alternativ lösning:



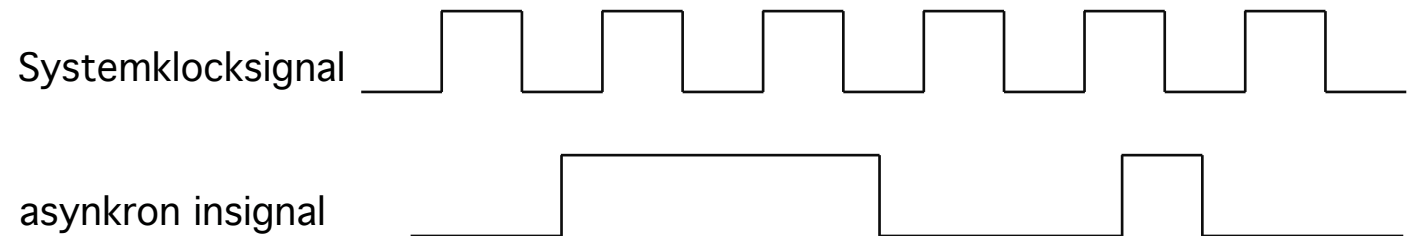
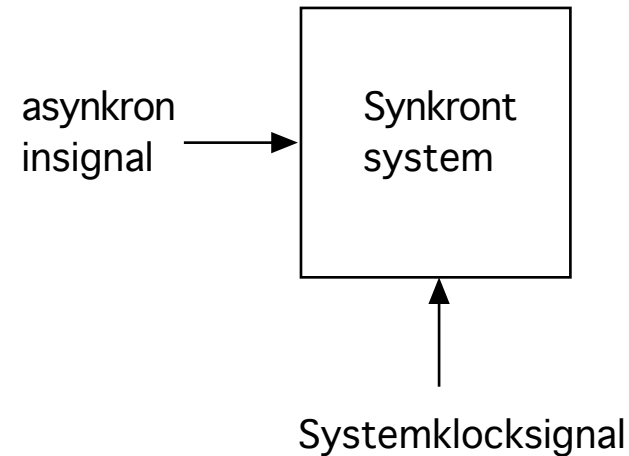


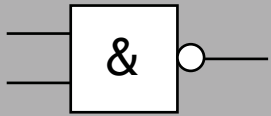
Asynkrona insignaler

Kan ge problem med i övrigt synkrona system.

Asynkrona insignaler:

- Brytare
- Sensorer
- System med annan klocka inkl processorer





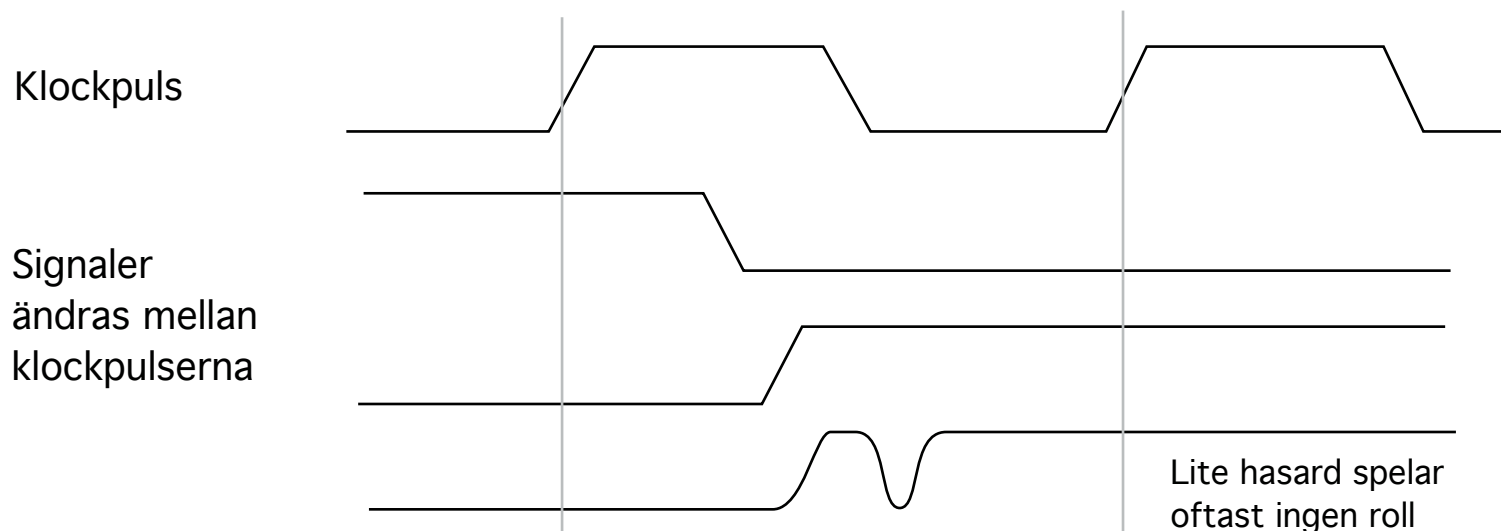
Problemet med asynkrona insignaler

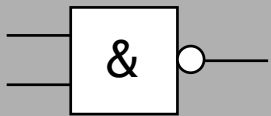
Analoga beteenden under grindomslag kan påverka beteendet.

Vi befinner oss oftare i otillåtna nivåer.

Hasard kan lättare påverka systemet.

Synkront system: Alla insignaler är stabila innan klockpuls. Sedan ändras de men är stabila innan nästa klockpuls.





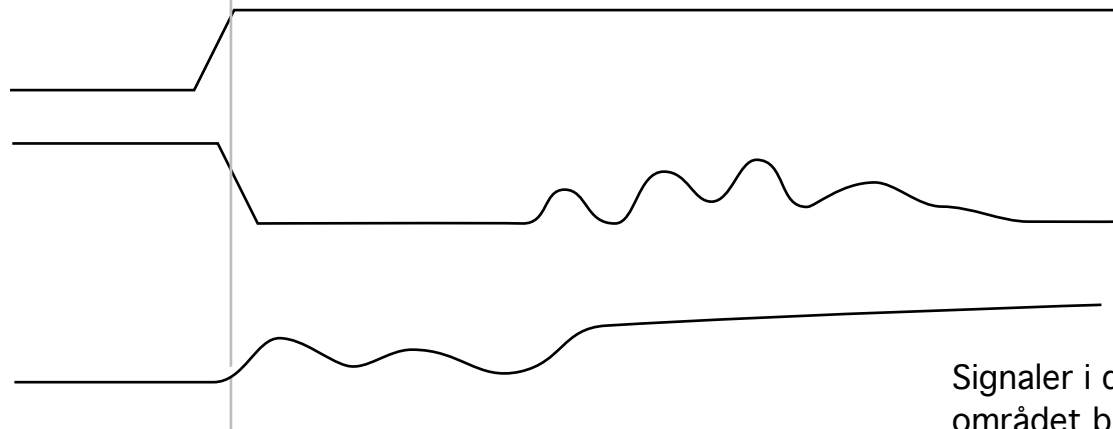
Asynkrona insignaler

Den asynkrona signalen kan läsas av när den har ett osäkert värde. Det osäkra värdet påverkar efterföljande logik på ett odefinierat sätt.

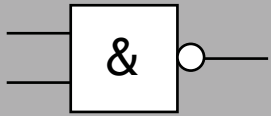
Klockpuls



Signaler
ändras mitt i
klockpulserna



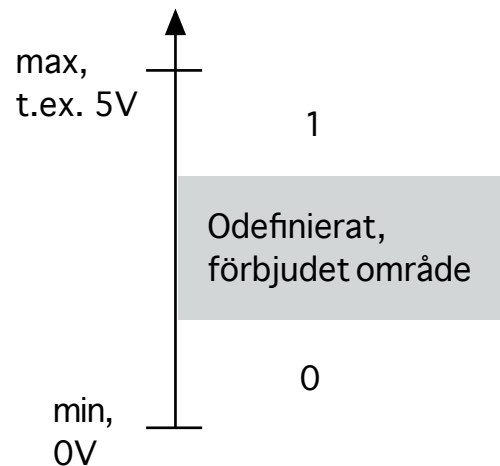
Signaler i det odefinierade området blir vanligare och kan vara längre

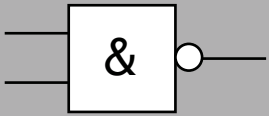


Metastabila tillstånd

Omslagstiden gör att ett grindnät kan ha signaler i det förbjudna området, som inte garanterat tolkas som 1 eller 0.

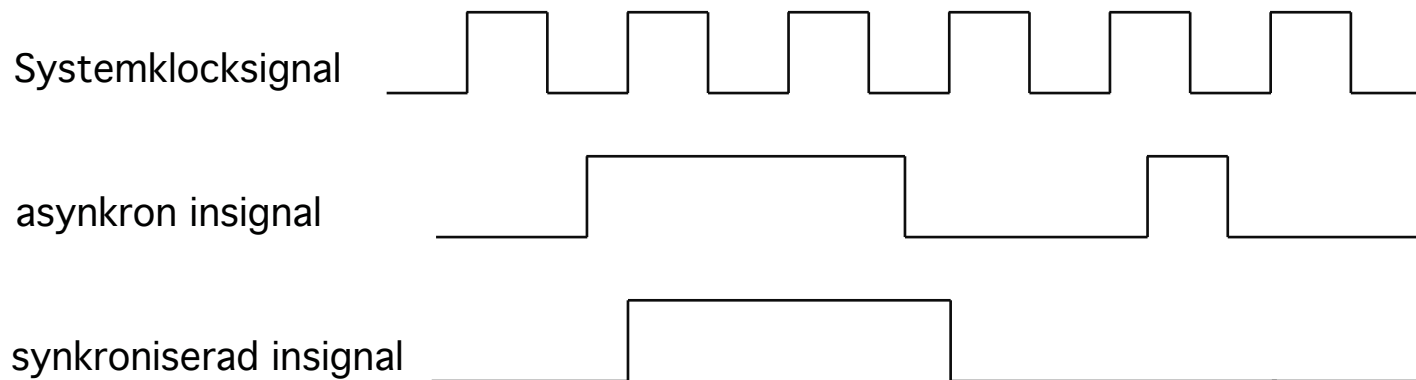
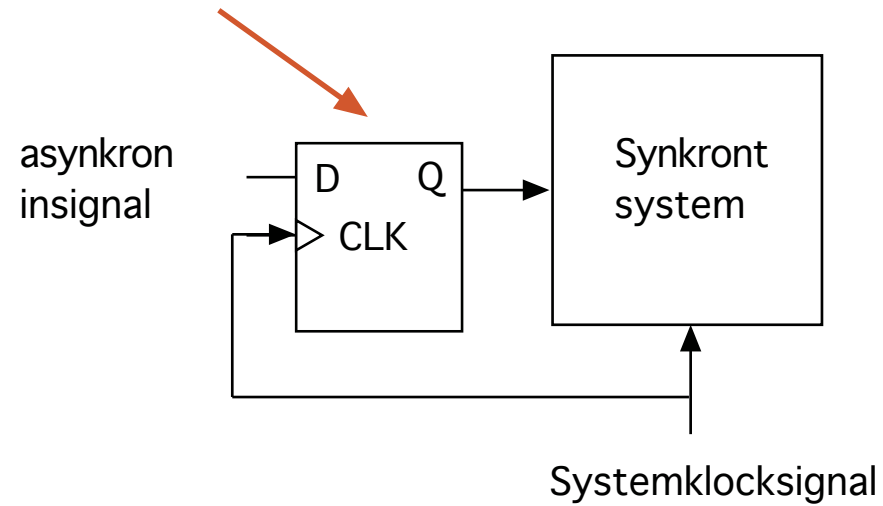
Asynkrona insignaler kan öka denna tid, vilket skapar oförutsägbara beteenden.

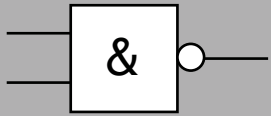




Synkronisering av asynkrona insignaler

Inför en *synkroniseringsvippa* på ingången

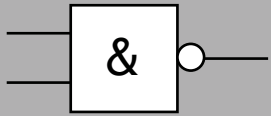




Varianter på synkronisering av asynkrona insignaler

- Dubbla synkroniseringsvippor.
- Tillåt bara att *en* asynkron insignal påverkar tillståndsövergång.
- Tillåt bara tillståndsövergångar som ändrar *en* bit.

Svårare design eller fler vippor med mer fördröjning.



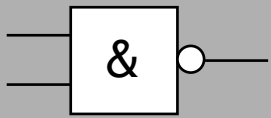
Lite mer om asynkrona system och klockning

Vippor, asynkrona latchar och klockade latchar

Vippor = klockade

Latchar = inte klockade?

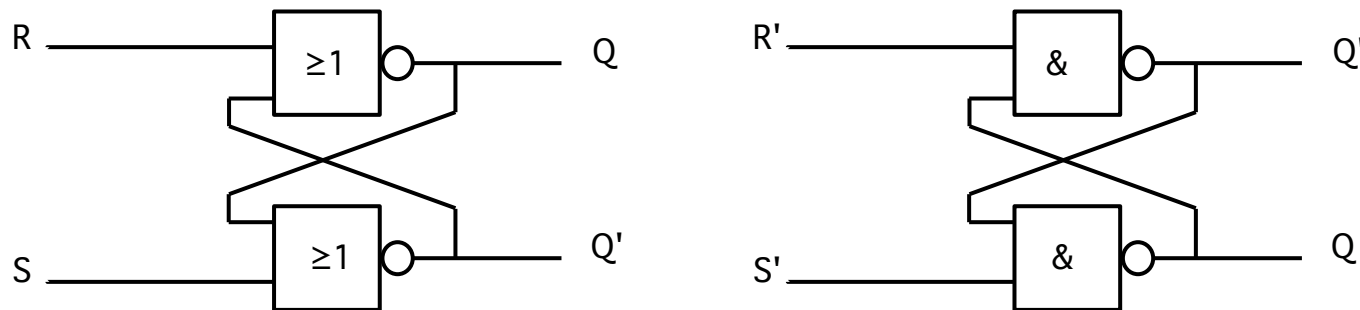
Men det finns också *klockade latchar!*



SR-latchen

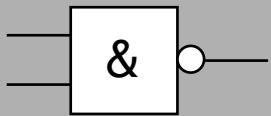
Fantastiskt enkel: Två vippor och jag hade en minnesfunktion!

Helt asynkron. Det finns ingen klocka.



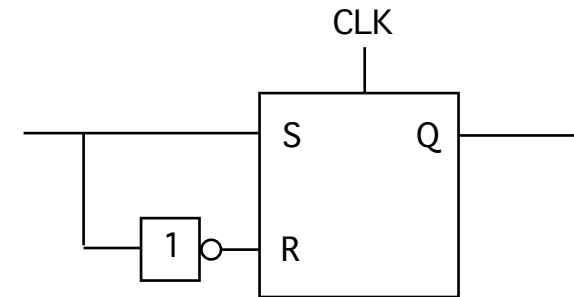
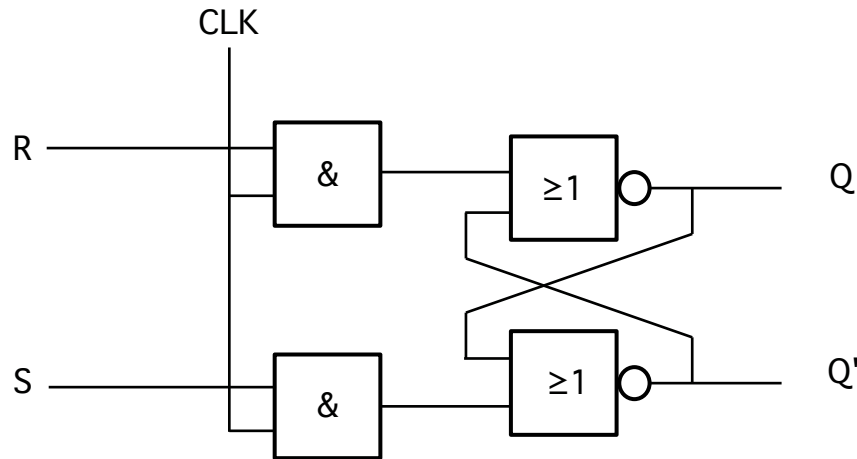
SR-latch

Används bl.a. i statiska datorminnen med miljontals latchar!



Klockad SR-latch

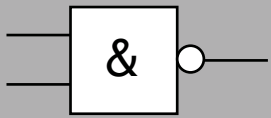
Skillnad mot vippa?



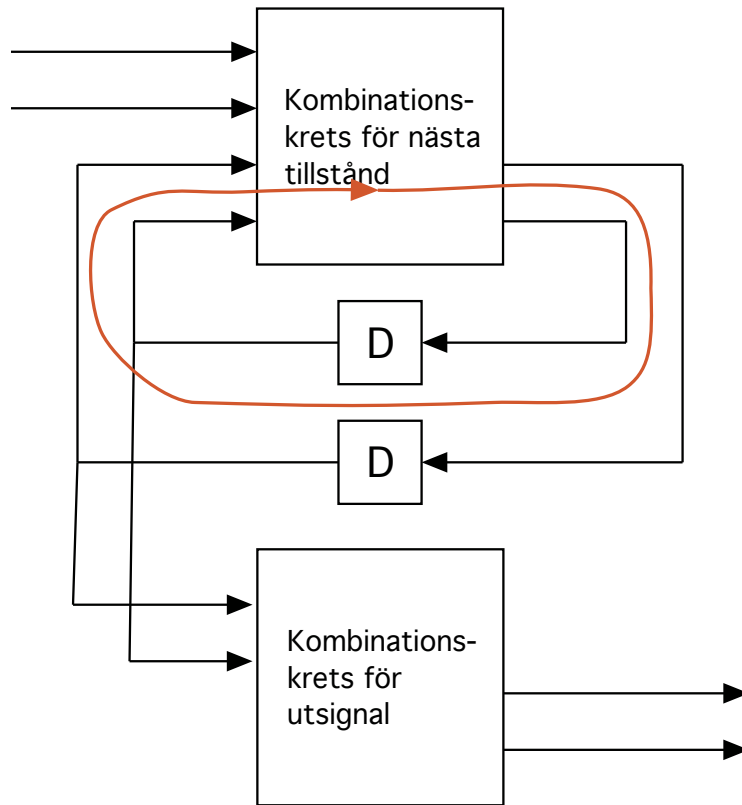
D-latch baserad på klockad SR-latch.
Inte samma sak som D-vippa!

Slår inte om vid *flank* utan när som helst medan klockan är hög! Den är *nivåtriggad*. Enklare, känsligare.

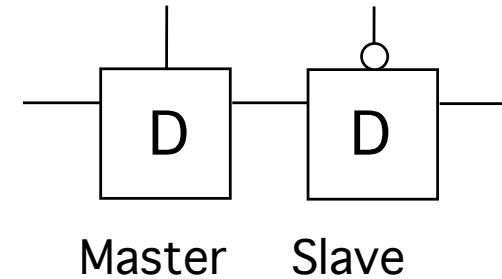
Fungerar dåligt i sekvenskretsar. Ändrat tillstånd slår igenom till nästa. Systemet är *transparent*.



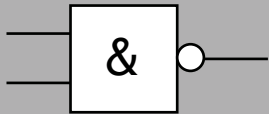
Nivåtrigging ger transparens genom sekvenskretsar



Master-slave-koppling med tvåfasklocka



Slutsats: Flanktrigging kanske är en bra lösning?



Nollställning

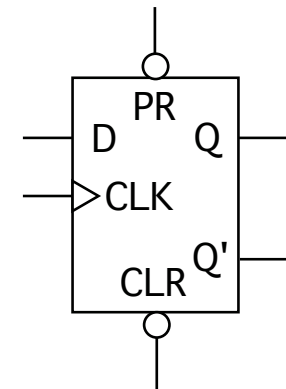
Många sekvenskretsar måste kunna nollställas, startas i känt läge. Två sätt:

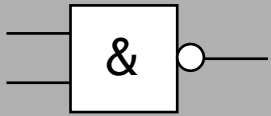
- Synkron
- Asynkron

Grundregel: Blanda inte asynkrona och synkrona signaler!

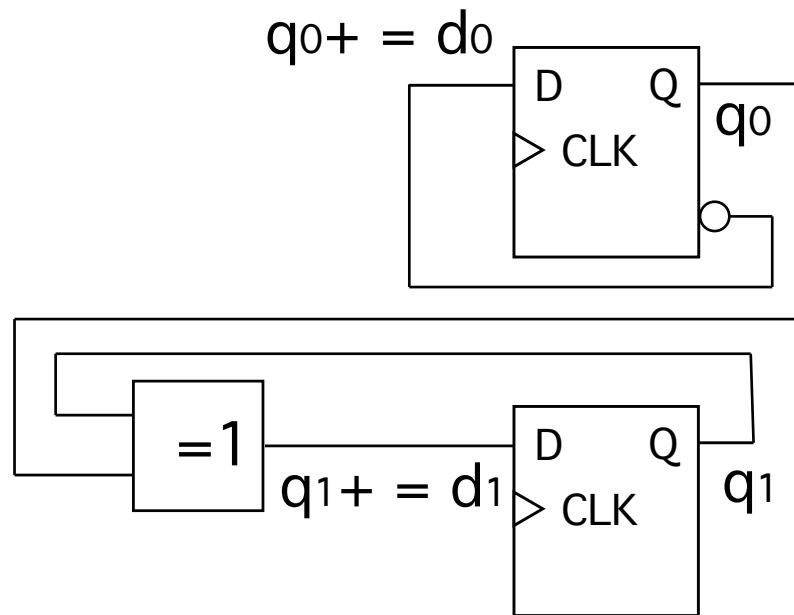
D-vippa med asynkron preset och clear.

- Synkron
- Asynkron



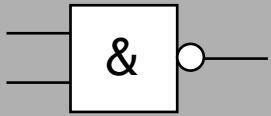


Exempel: Autonom (ej styrbar) 2-bitsräknare

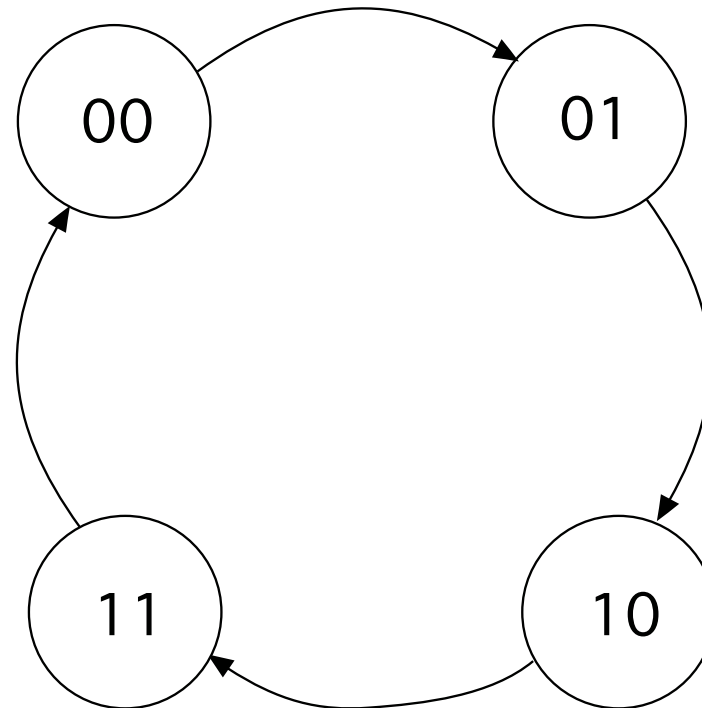


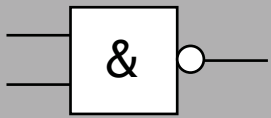
q1	q0	q1+	q1+
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Exempel: 2-bitarsräknare



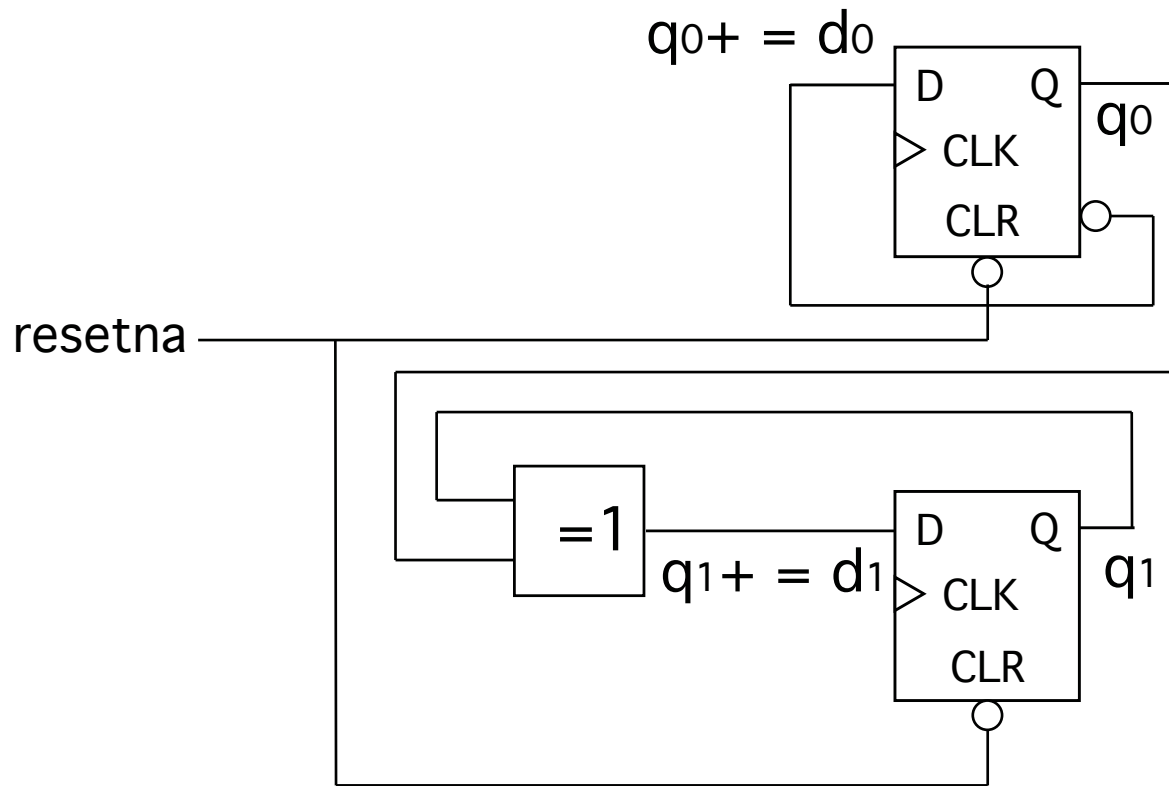
Tillståndsdigram för räknaren



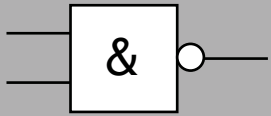


Asynkron reset

reset-negativ-asynkron = resetna

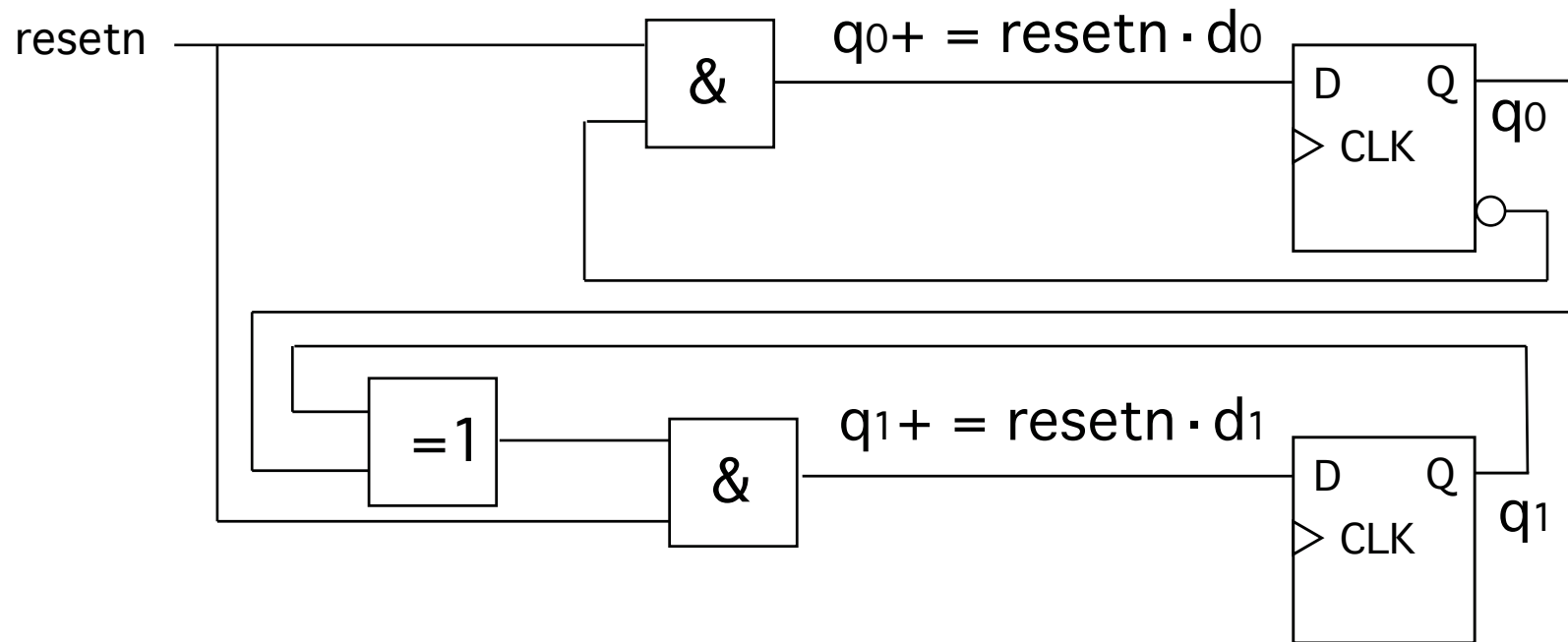


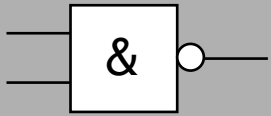
Nollställs asynkront med resetna = 0



Synkron reset

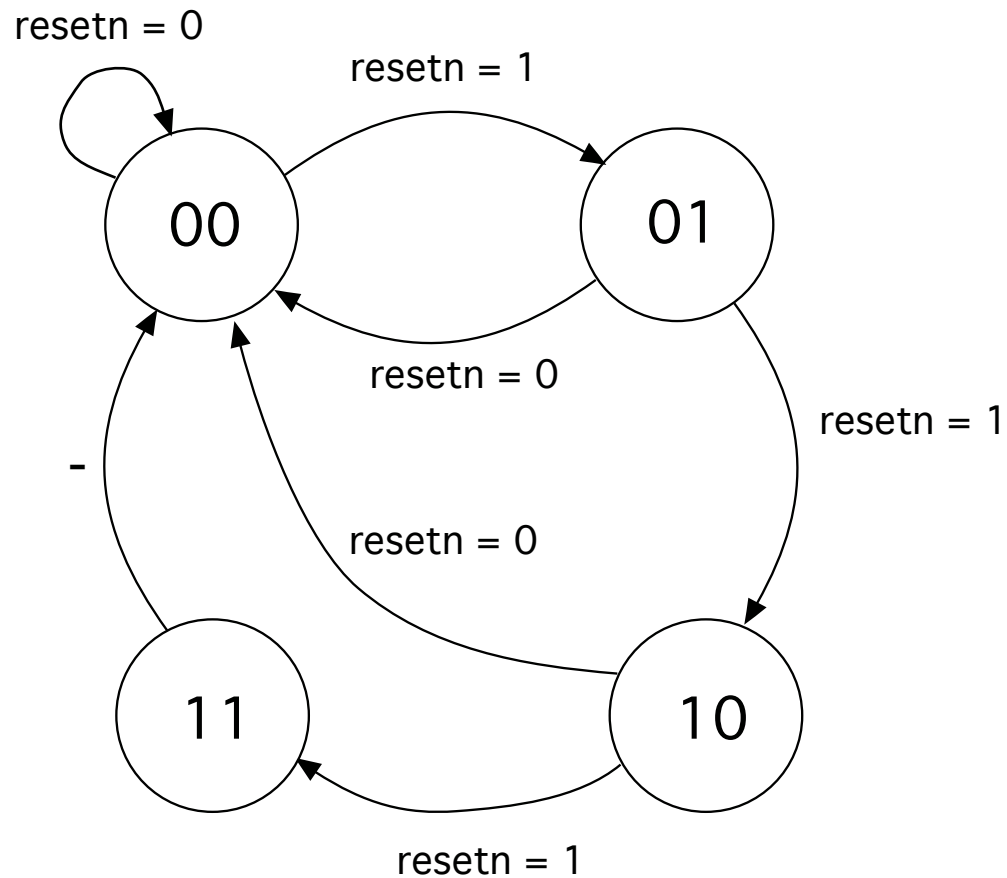
reset-negativ = resetn

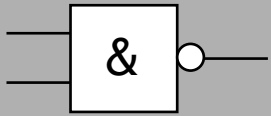




Synkron nollställning

Hanteras som en del av sekvenkretsen som vilken insignal som helst!





Kopplingsregler

Synkron funktion

- Insignaler måste synkas
- Påverkar bara synkrona signaler

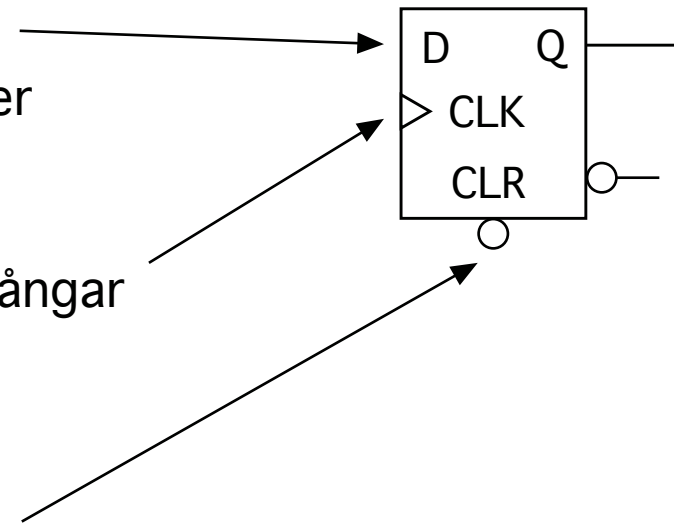
Klocka

- Kopplas direkt till alla klockingångar
- Grinda aldrig!

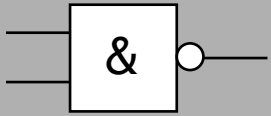
Asynkron reset:

- Nollställer alla register/vippor
- Kopplas till asynkrona ingångar

Och så lämnar vi inga ingångar oanslutna!



Blanda aldrig
signaltyperna!



Snabbintroduktion till speciella sekvenskretsar och programmerbara kretsar

Räknare

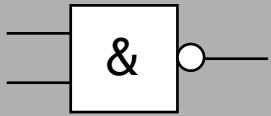
Register

Skiftregiter

PLD

FPGA

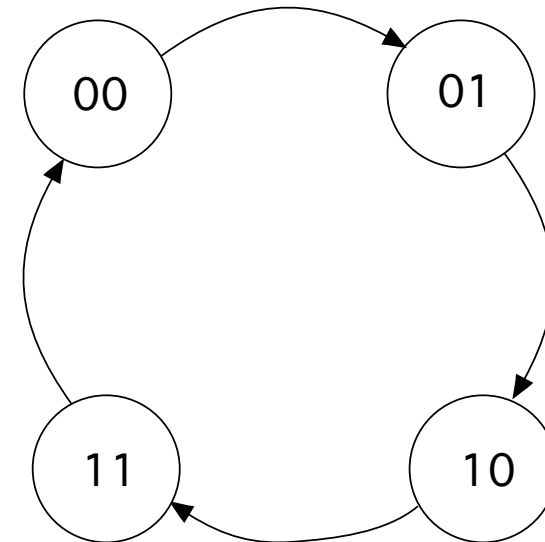
Mikrocontrollers

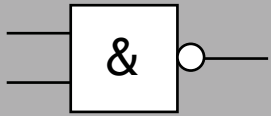


Räknare

- Binärräknare
- Dekadräknare
- Styrning, count enable, carry, load, clear
- Upp/ner-räknare
- Kakadkoppling av räknare

Tillståndsdigram för räknaren

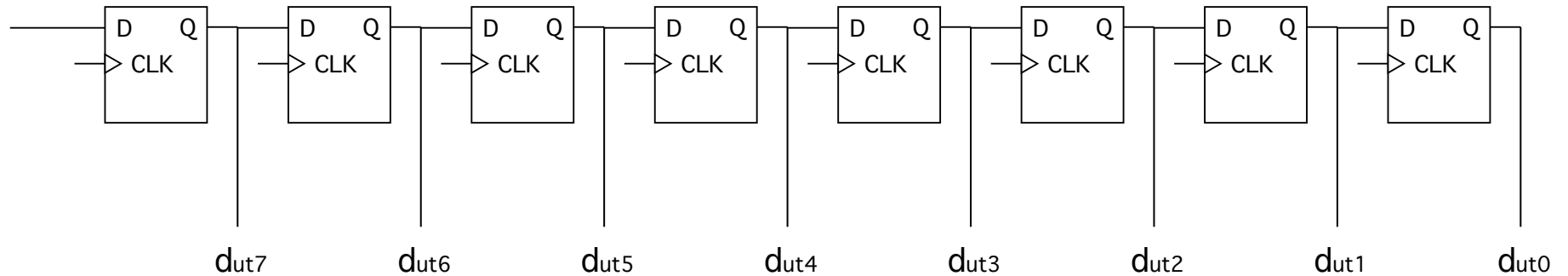


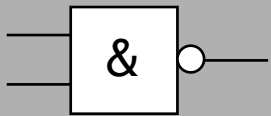


Skiftregister

Lagrar ett antal bitar som skiftas ett steg per klockpuls.

Exempel: Skiftregister med D-vippor, seriellt in, parallellt ut:



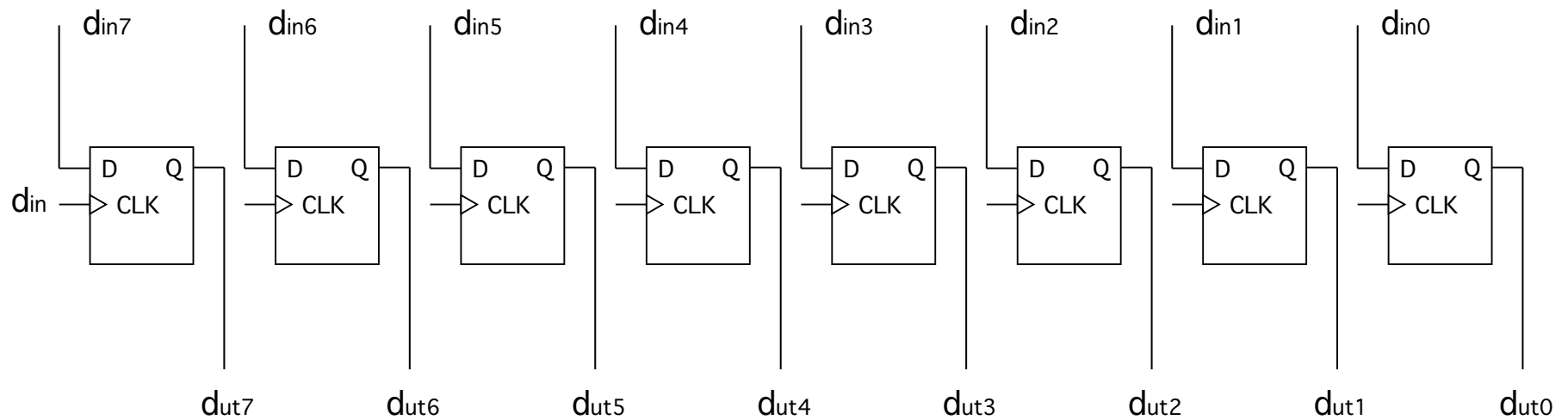


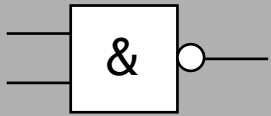
Register

Minne för lagring av ett ord.

Vi såg dessa i datorskissen tidigare.

Kan utföras med D-vippor.





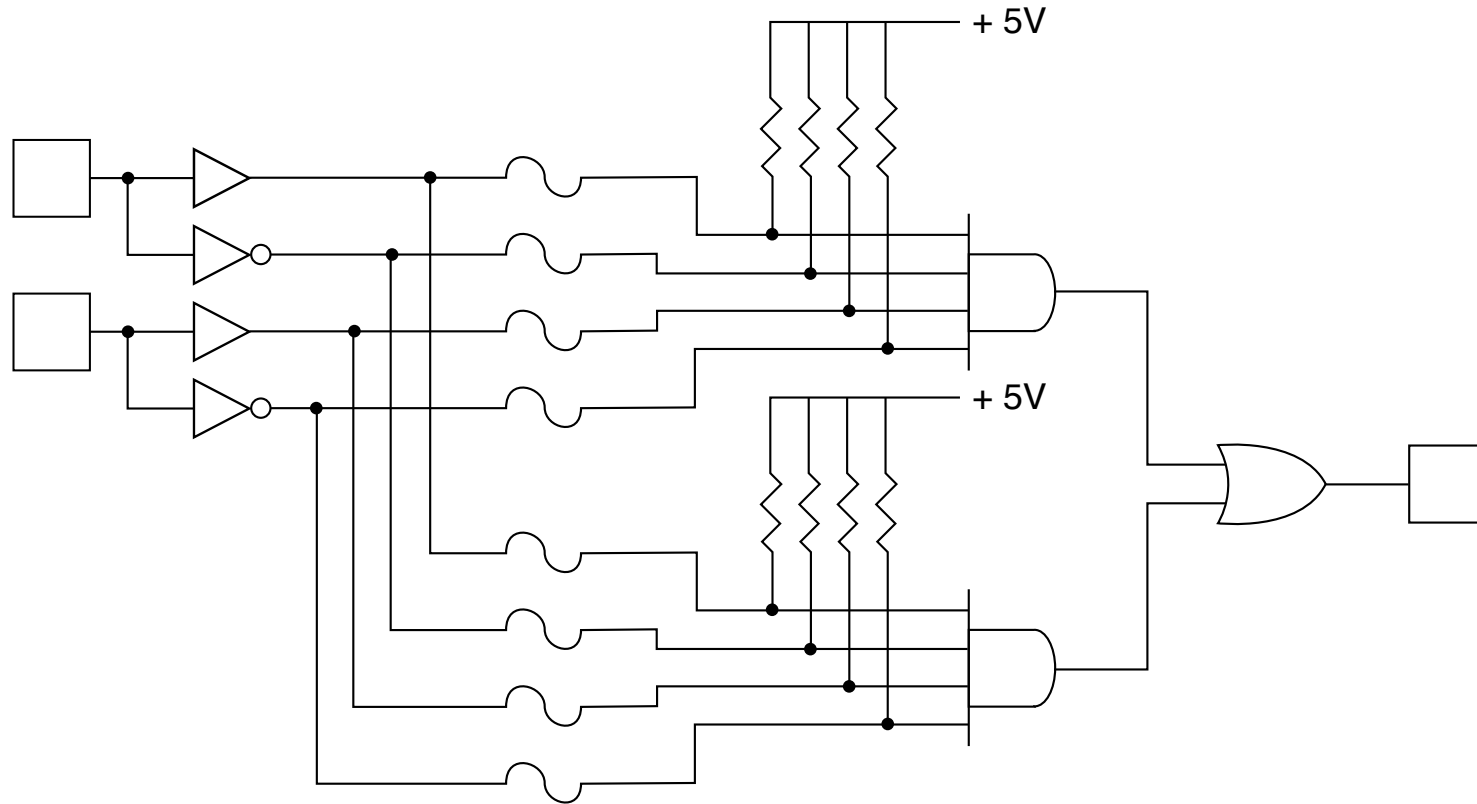
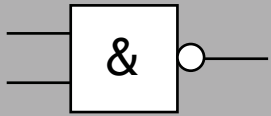
Programmerbara kretsar

ROM

PLD

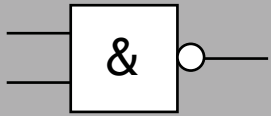
FPGA

Stora kombinations- och sekvenskretsar med programmerbara (öppningsbara/stängningsbara) punkter



Simplified programmable logic device

Source: Wikipedia



Mikrocontrollers

Små CPUer med tillgängliga portar (kontakter) för enkel inkoppling av annan logik.

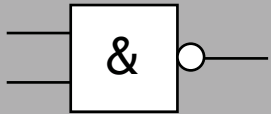
PIC

Atmega

Attiny

Enkelt tillgängliga på färdiga kort: Arduino, NodeMCU, Digispark

Mellanting: Raspberry Pie, mer lik en vanlig dator men med generella kontakter.



Mikrocontrollers + logik

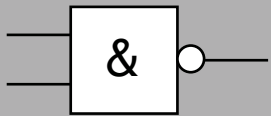
Många små lösningar för inbyggnad kan göras med en mikrocontroller plus externa kretsar.

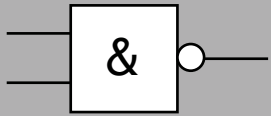
Typfall: En mikrocontroller + skiftregister. Skiftregistret ökar antalet tillgängliga utgångar genom seriell utmatning av data!

Exempel: Arduino + skiftregister + display.

Exempel 2: Sensorer + lysdioder + inverterare + skiftregister + display + Arduino

TSEA22 Digitaltechnik





NÄSTA FÖRELÄSNING

Mattias tur!

Speciella sekvenskretsar.

Räknare och register.