

Speciella sekvenskretsar

Föreläsning 8

Digitalteknik

Mattias Krysander

Institutionen för systemteknik

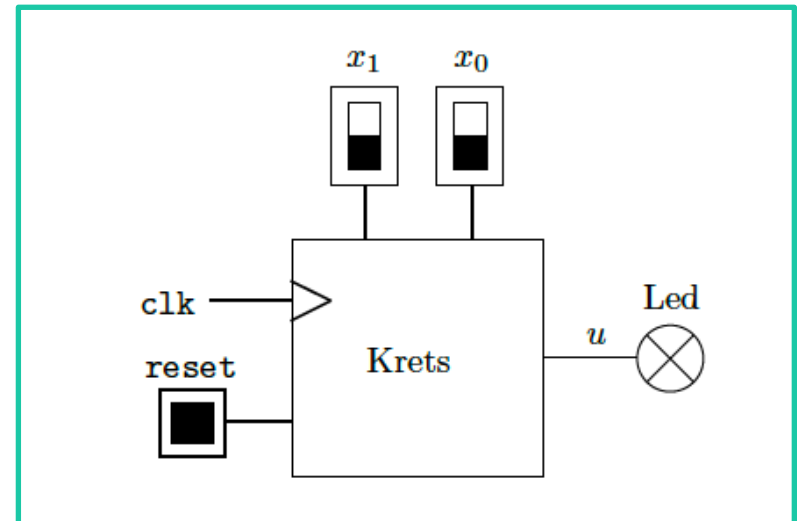
Laborationsinformation

- Lab 2 börjar på tisdag.
 - Mycket förberedelser, klurigare, börja i god tid innan.
- Kompletteringstillfälle för lab 1 och 2 torsdag 6/10
 - 17:15-19:00, 19:15-21:00 för komplettering av 1 eller 2 uppgifter
 - 17:15-21:00 främst för komplettering av hel laboration
 - Anmälan öppnar på onsdag 28/9 kl 21:00.

Uppgift 2.1: Tips

Låset öppnas om skjutomkopplarna manövreras i sekvens enligt 1–3

1. Båda i nedre läget
2. Vänster i nedre läget, höger i övre läget
3. Vänster i övre läget, höger i övre läget

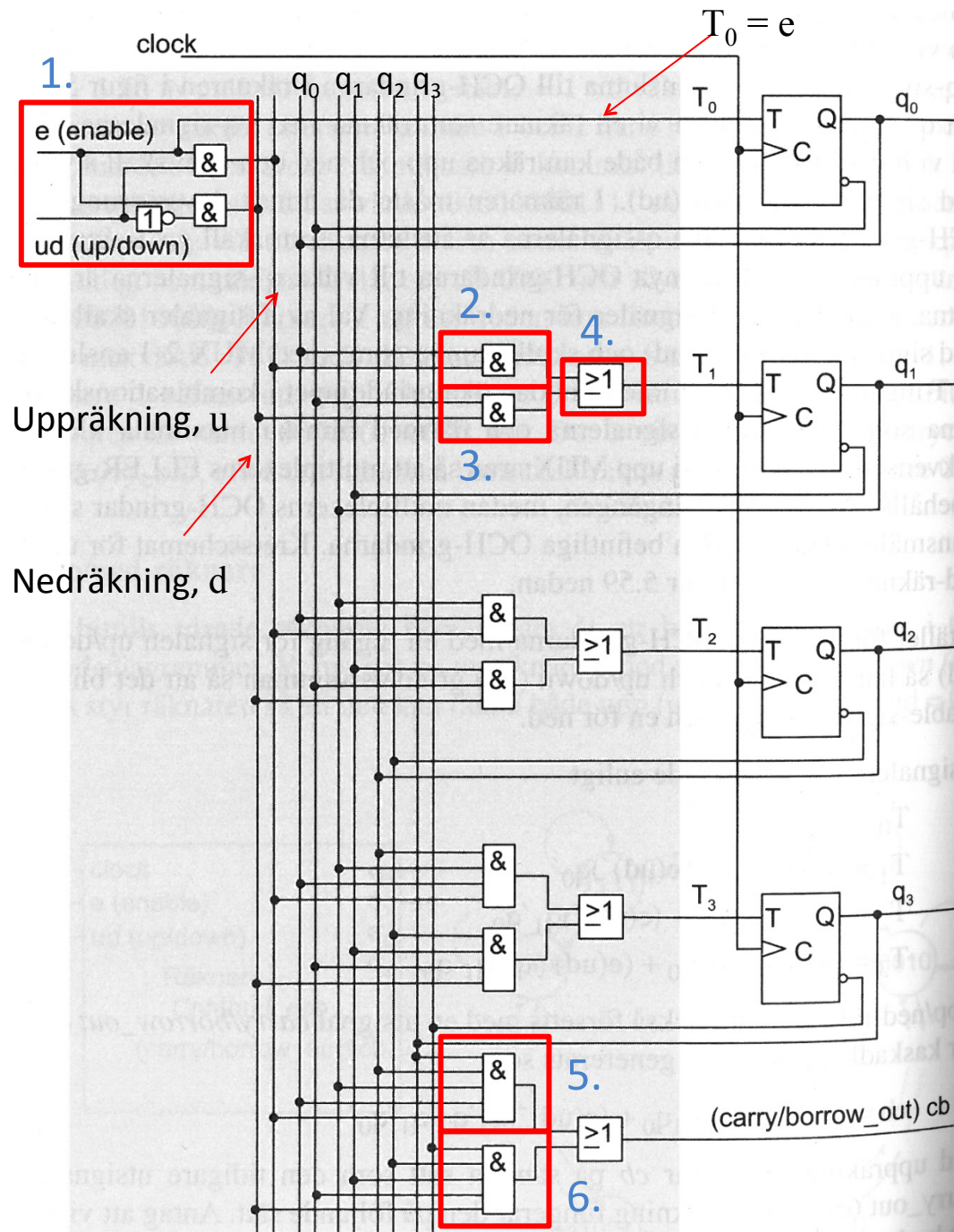


Det ska även finnas en asynkron resetknapp som återställer systemet till startläge, dvs låset är stängt och hela sekvensen 1–3 måste genomlöpas för att öppna låset igen. Detta ska ske oberoende av hur skjutomkopplarna står när resetknappen trycks ned.

- Observera att D-vipporna i labbet bara kan resetas till noll.
=> **Starttillståndet måste kodas 00**
=> **Kontrollera att hela sekvensen 1-3 krävs för att öppna låset efter nollställning oavsett hur skjutomkopplarna står vid nollställning!**

Dagens föreläsning

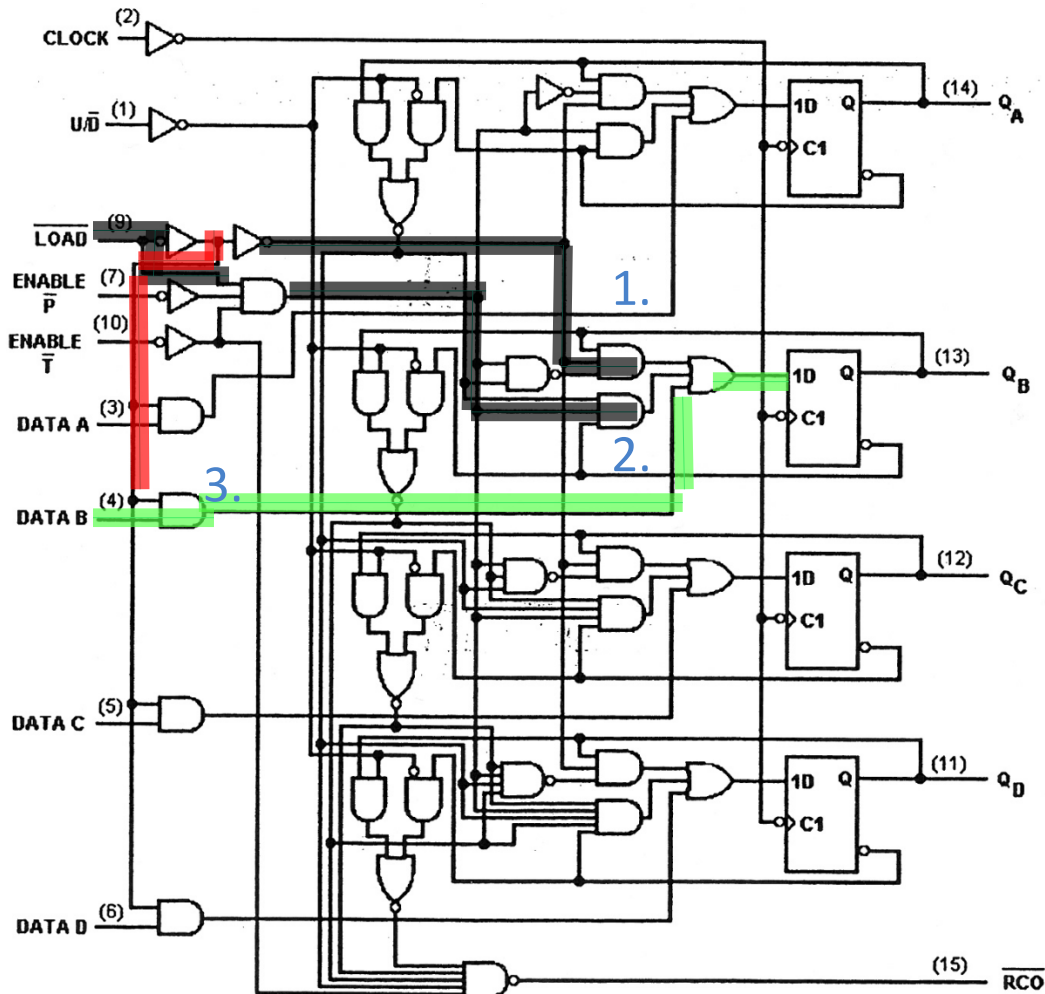
- Räknare
 - count enable , carry out, borrow out, upp/ned-räkning, load, clear
 - kaskadkoppling
- Problemlösning med speciella sekvenskretsar
- Illustrera 2 designprinciper:
 - Funktionsorienterad utveckling
 - Top-down design
- Register



Upp/ned-räknare

1. Skapar upp- och nedräkningssignal u och d
2. Skapar T_i vid uppräkning
 $T_{1u} = uq_0$
3. Skapar T_i vid nedräkning
 $T_{1d} = dq'_0$
4. Skapar T_i :
 $T_1 = T_{1u} + T_{1d}$
5. Carry out:
 $c = uq_3q_2q_1q_0$
6. Borrow out:
 $b = dq'_3q'_2q'_1q'_0$

Reversibel räknare med load



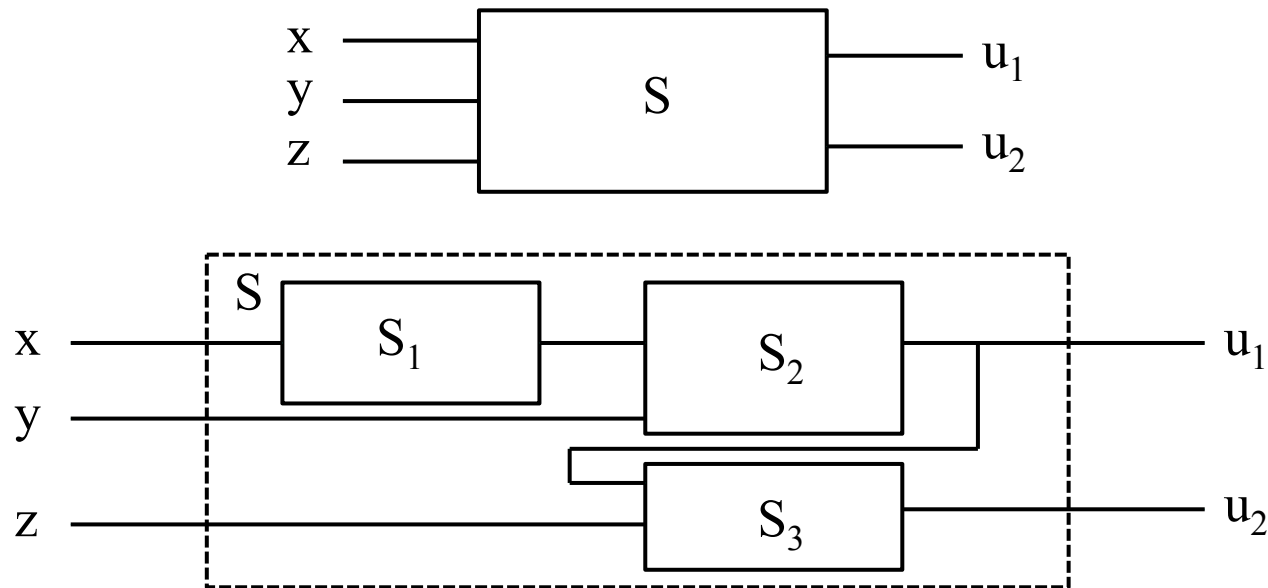
Load aktiveras med 0:a.
Då kopplas
dataingången (DATA X)
direkt till vippans
ingång.

Figuren visar $LOAD = 0$
Svart = 0, Rött = 1, Grönt
= signalväg

1. Behåll avstängd
2. Byt avstängd
3. Läsning aktiverad

Problemlösning med komplexa komponenter

- Den formella metodiken med tillståndsdigram, tillståndstabell, osv blir för omständlig för större system.
- För syntes med komplexa komponenter finns inga generella metoder utan intuition och logiskt tänkande får leda designen.
- Ett tillvägagångssätt: top-down design

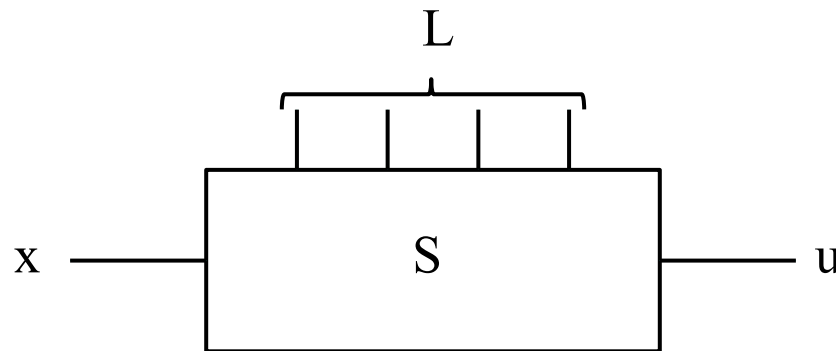


Exempel

En sekvenskrets, S , skall detektera synkroniserade pulser av en bestämd längd, L . L kan väljas mellan 1 och 15 räknat i klockintervall.

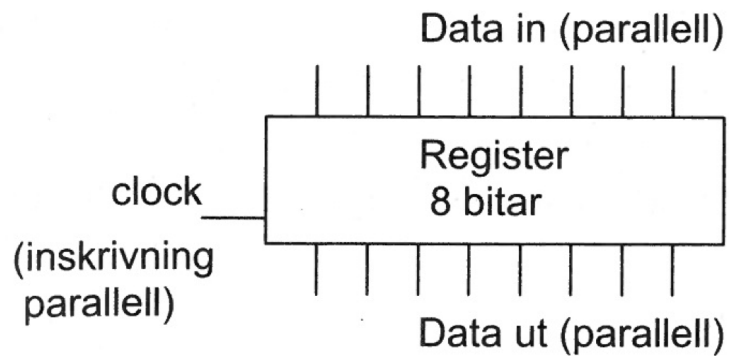
Utsignalen skall vara $u = 1$ ett klockintervall efter detekterad puls. (Ingen puls på x är längre än 15 klockintervall.)

Konstruera S med en 4-bits binärräknare med ingångar för “clear” (synkron) och “count enable” samt valfria grindar och vippor.

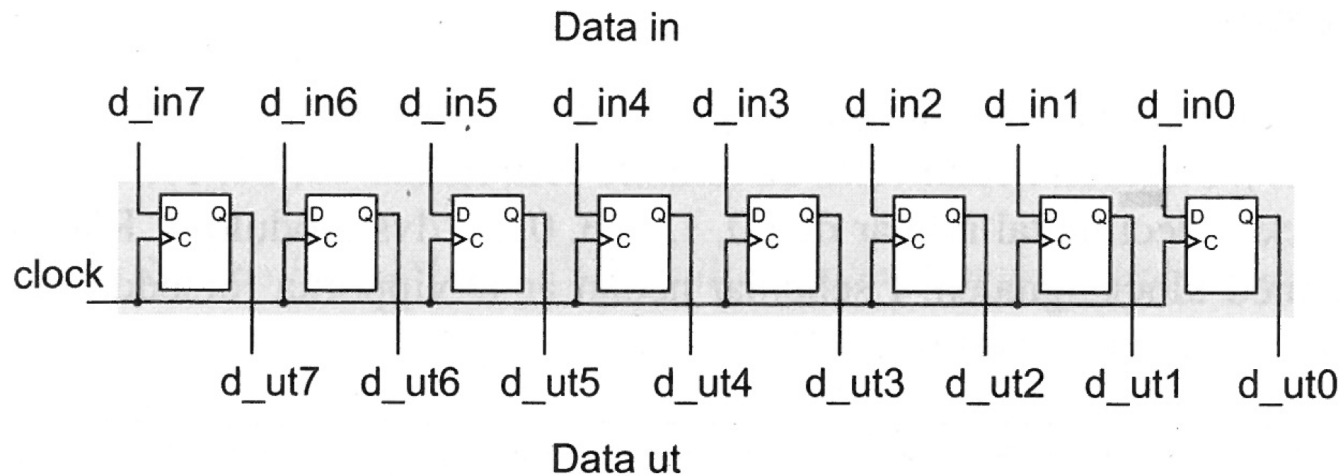


Register

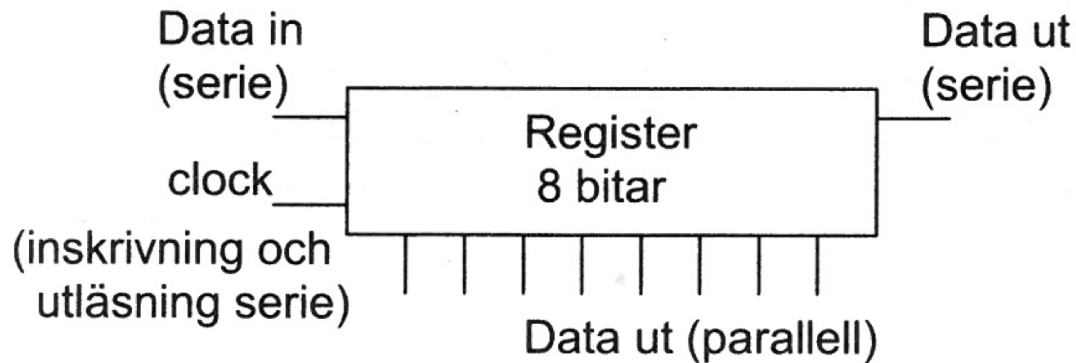
Register (lagringsutrymme)



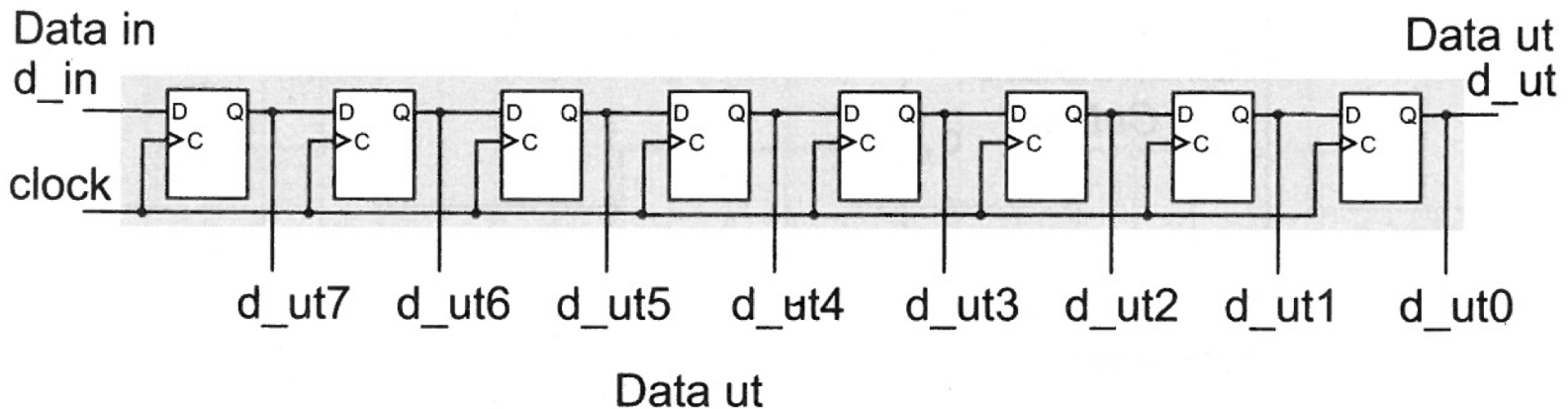
Parallell in parallell ut



Skiftregister

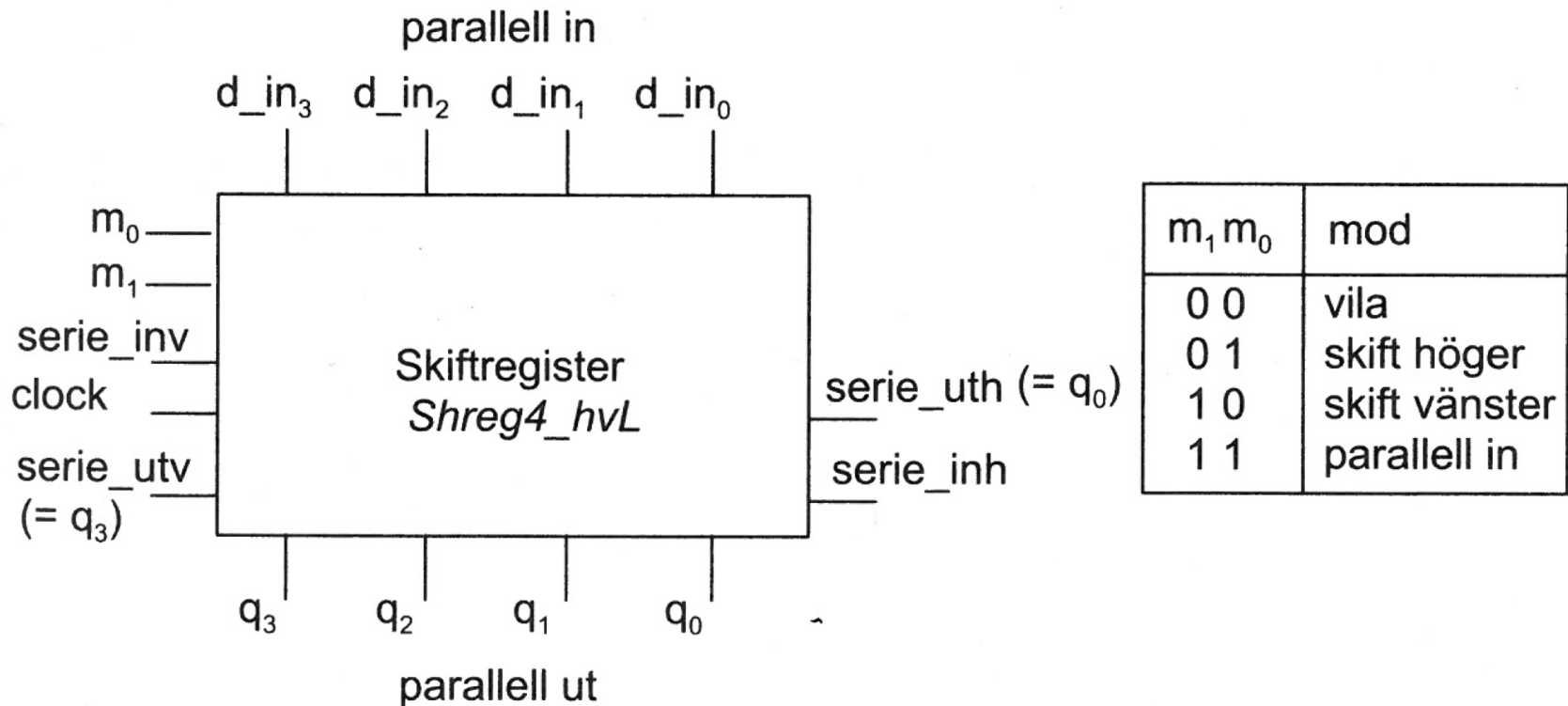


- Serie in – parallell ut/serie ut
- högerskift

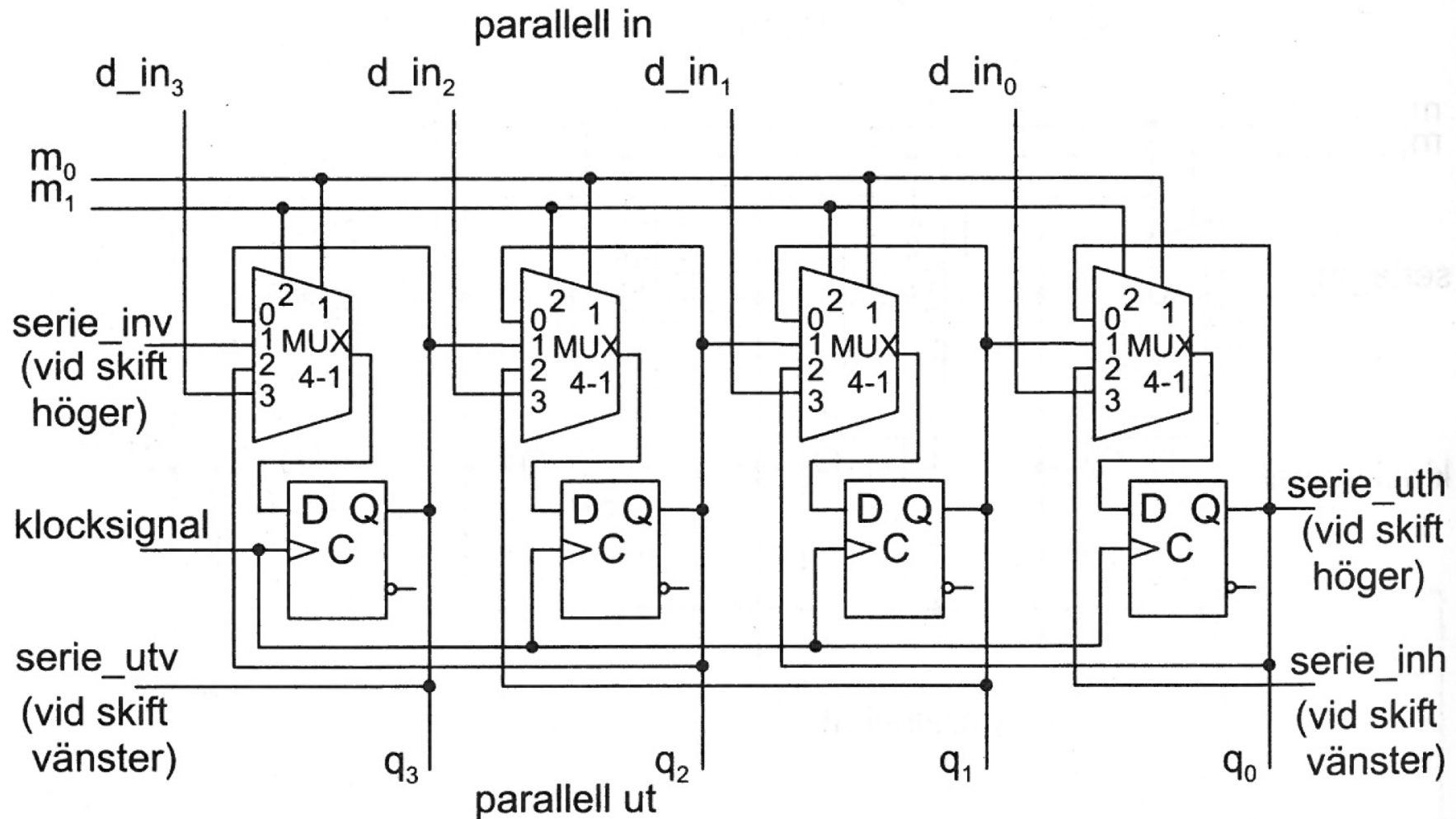


Generellt (skift)register

Serie in/parallell in – serie ut/parallell ut – skift höger/
skift vänster



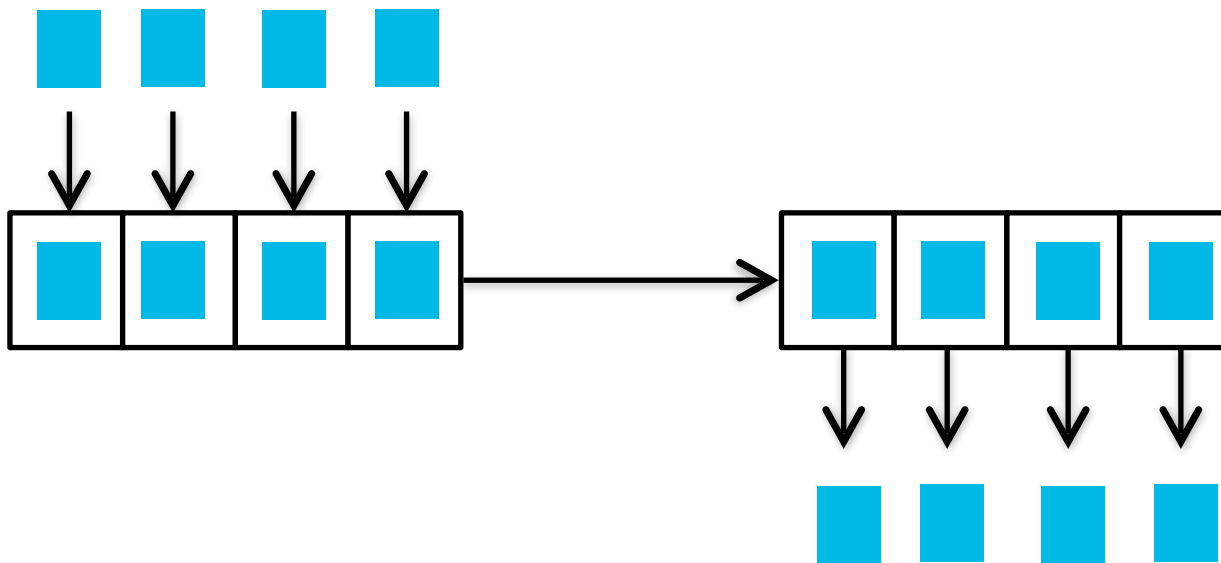
Generellt register - konstruktion



Register - användningsområden

- Parallell/seriell omvandling
- Fördröjning
- Implementering av stackar

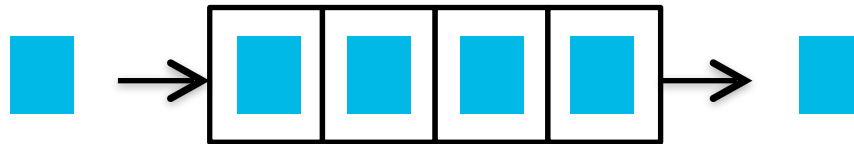
Parallell/seriell omvandling



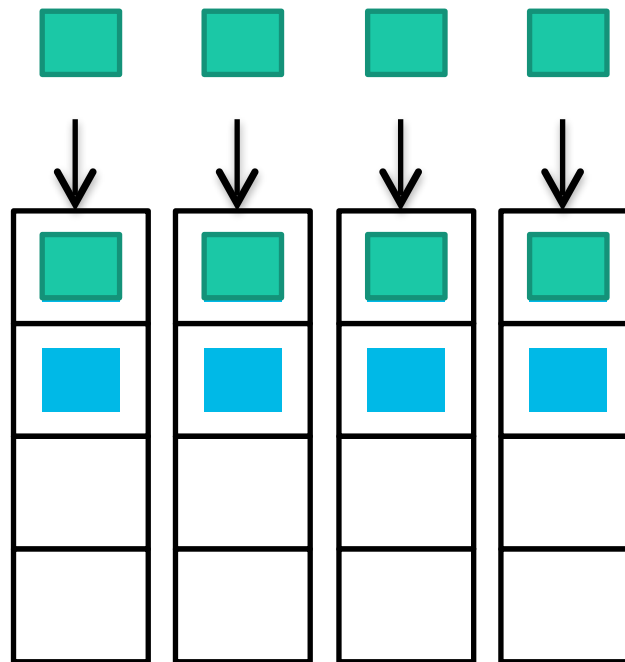
Shift right

Fördröjning

Shift right



Stackar (First in last out)



Digitalteknik

Mattias Krysender

www.liu.se