

Speciella sekvenskretsar

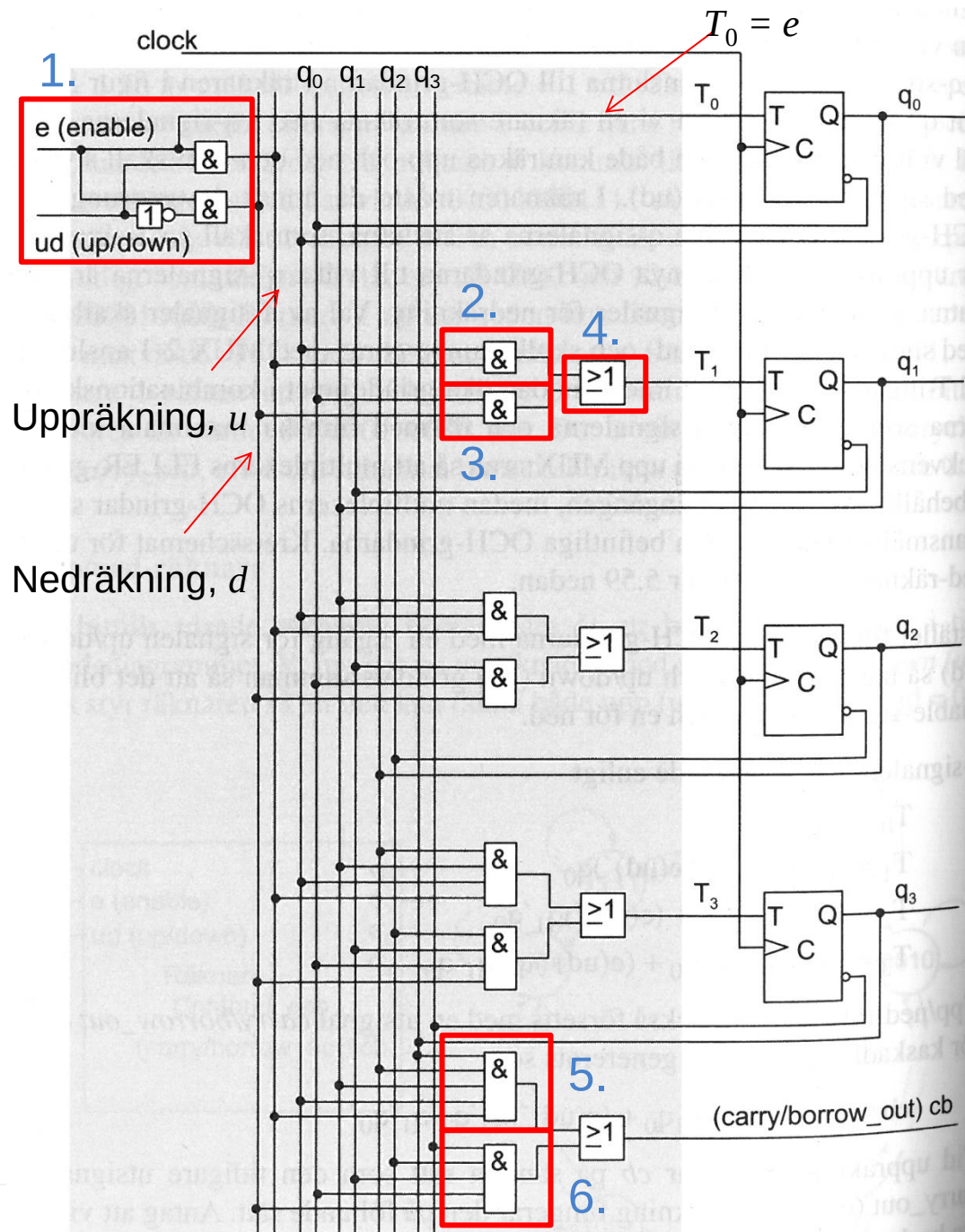
Digitalteknik – Föreläsning 9

Mattias Krysanter

Institutionen för systemteknik

Dagens föreläsning

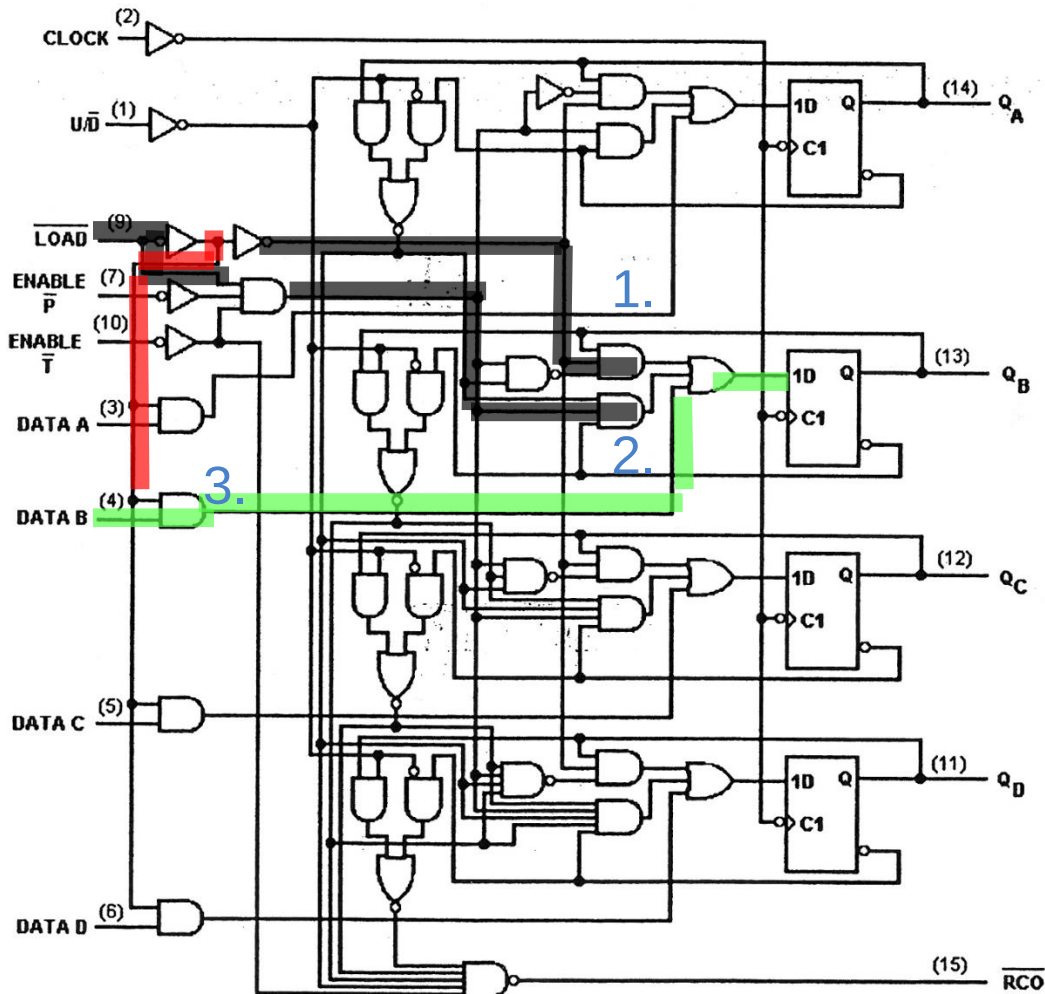
- Räknare
 - Binärräknare
 - count enable , carry out, borrow out, upp/ned-räkning, load, clear
 - kaskadkoppling
- Problemlösning med speciella sekvenskretsar
- Register och skiftregister
 - Funktion och konstruktion
 - Användningsområden



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.

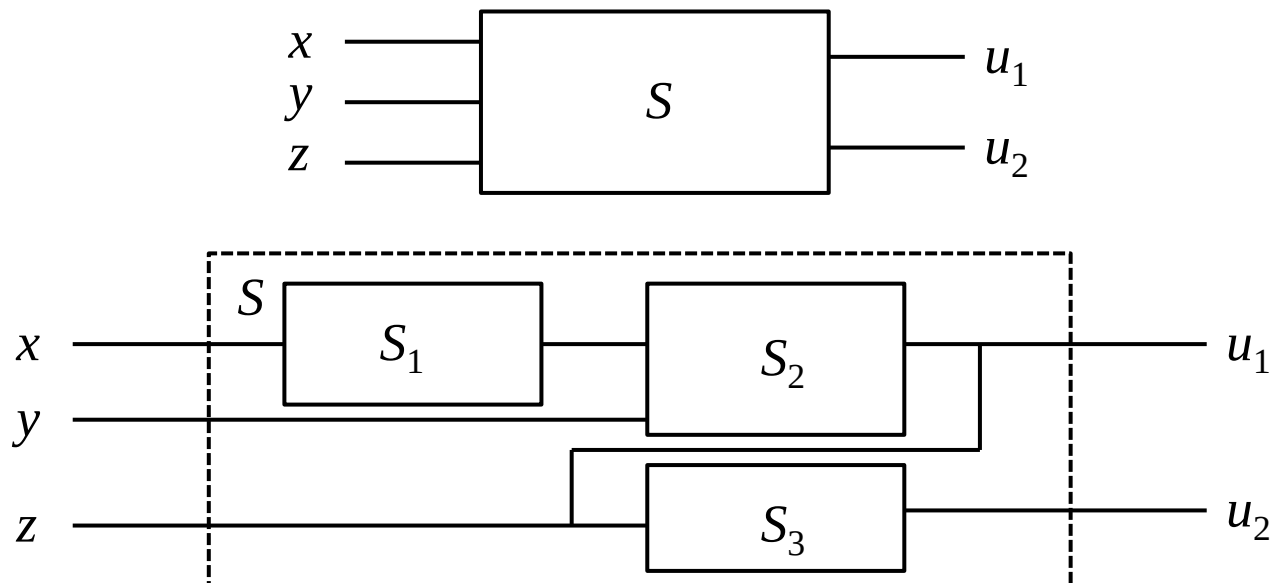
Då kopplas dataingången (DATA X) direkt till vippans ingång.

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

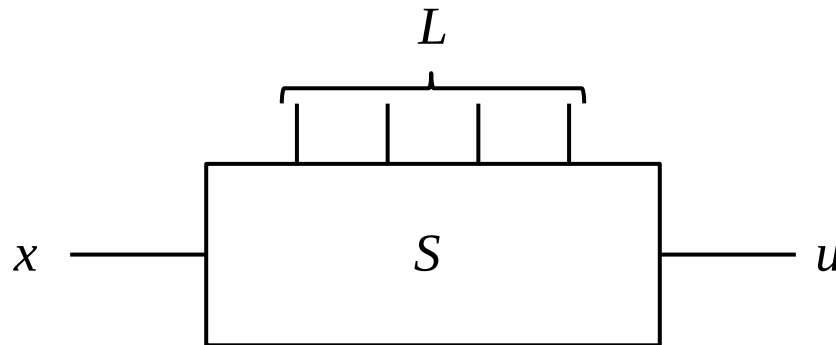


Exempel

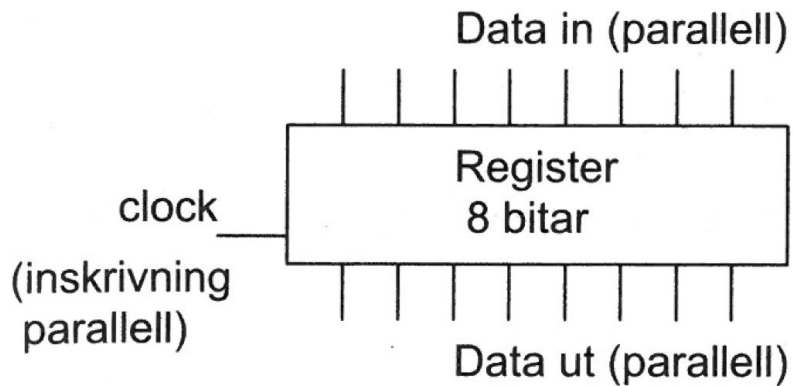
En sekvenskrets, S , skall detektera synkroniserade pulser av en bestämd längd, L . L kan väljas mellan 1 och 15 räknat i klockintervall.

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

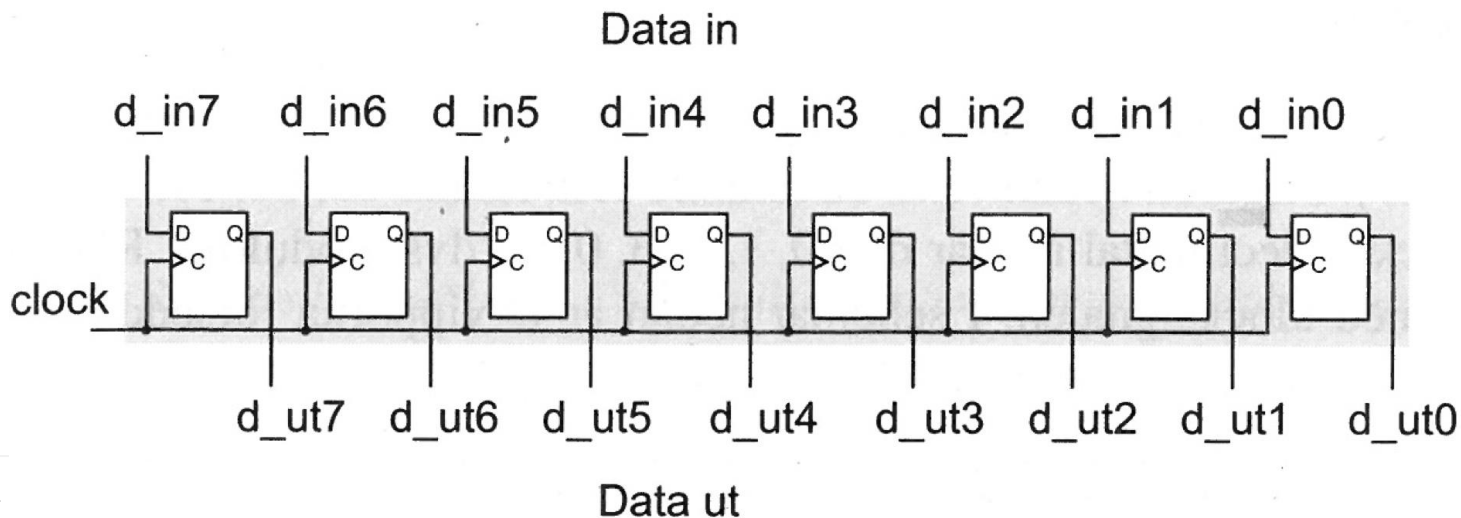
Konstruera S med en 4-bits binärräknare med ingångar för “clear” (synkron) och “count enable” samt valfria grindar och vippor.



Register (lagringsutrymme)

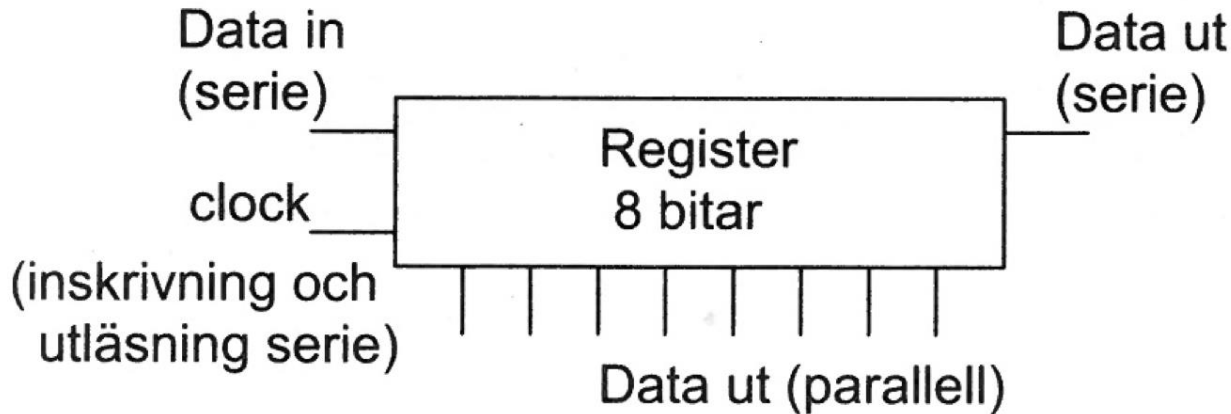


Parallell in parallell ut

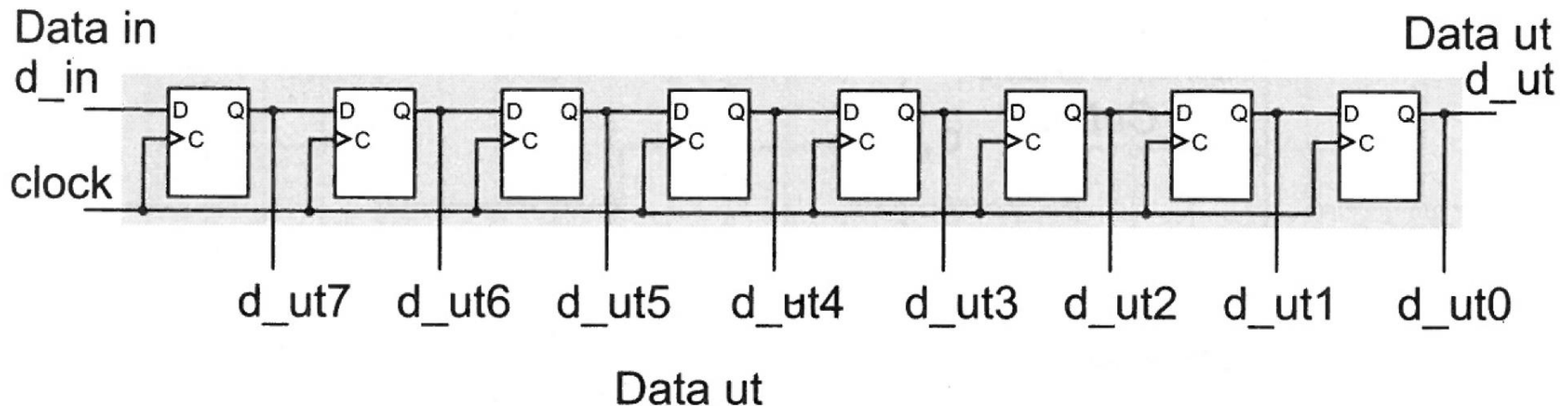


EXPANDING REALITY

Skiftregister

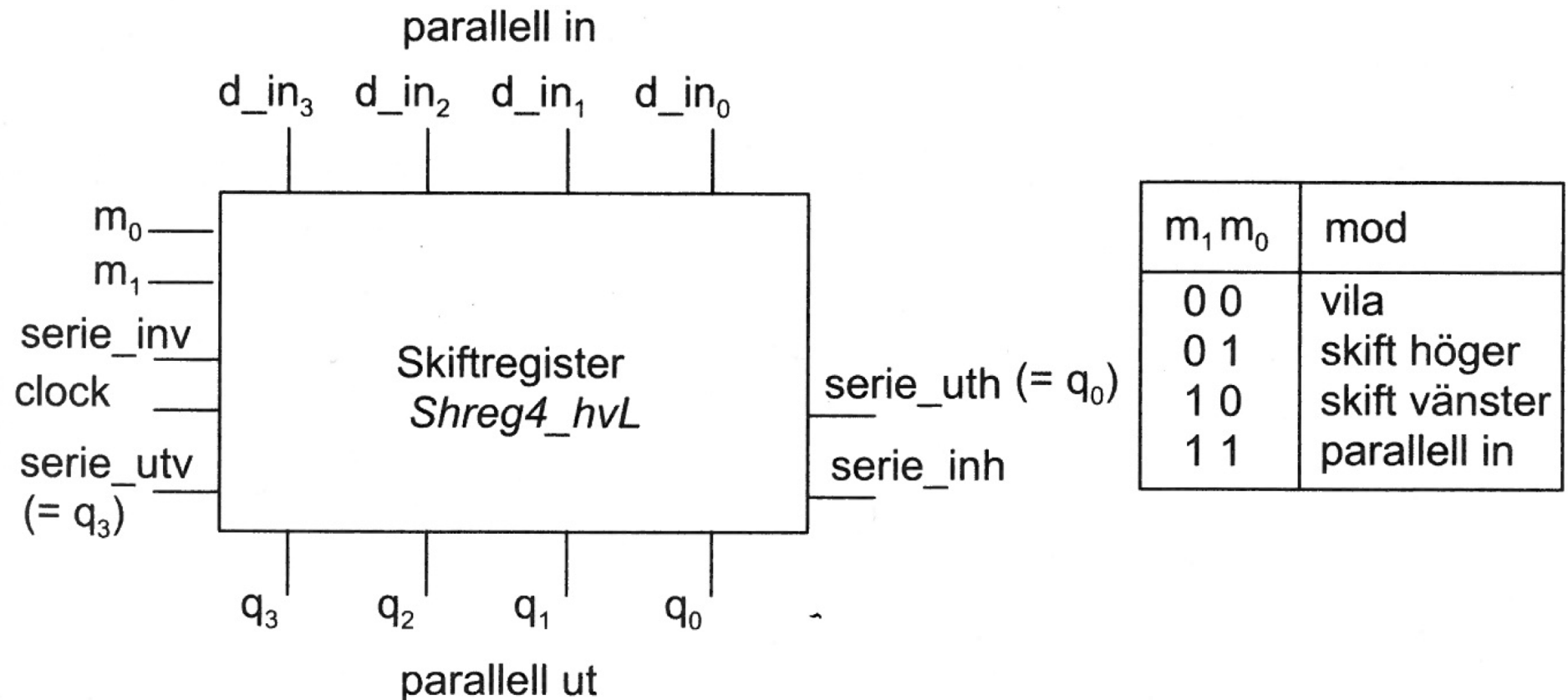


- Serie in – parallell ut/serie ut
- högerskift

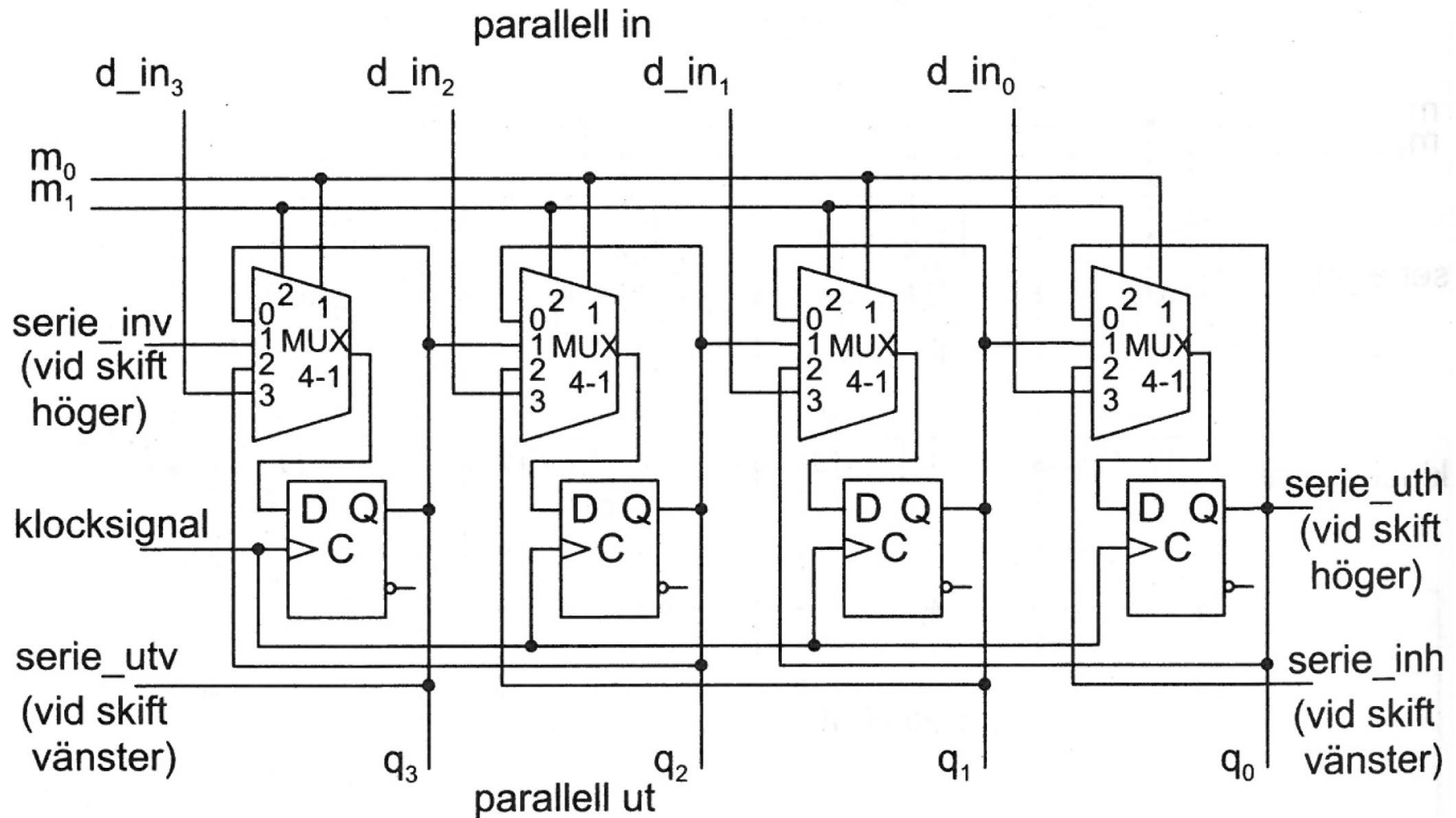


Generellt (skift)register

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



Generellt register - konstruktion



Register - användningsområden

- Parallell/seriell omvandling
- Fördröjning
- Implementering av stackar



Linköpings universitet

expanding reality

www.liu.se