Föreläsning 8
Tillståndsmaskiner och seriell transmission

Erik Agrell 2017-09-13

Innehåll:

1. SAR-omvandlaren, forts.
2. Sample-and-hold
3. Seriell transmission
4. Mer om tillståndsmaskiner
5. Synkronisering av asynkron insignal

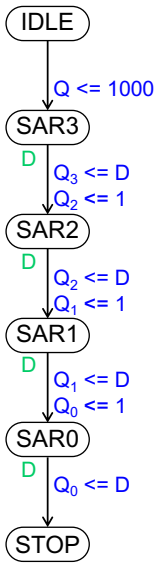
Bilder av E. Agrell och A. Linde

1. SAR-omvandlaren, forts.

```
library ieee;
use ieee.std_logic_1164.all;

entity ADC is port(
  clk50, reset: in std_logic;
  D: in std_logic;    -- input from comparator: higher or lower?
  Q: out std_logic_vector(3 downto 0));    -- parallel output
end entity;

architecture state_machine of ADC is
  type StateType is (U,IDLE,SAR3,SAR2,SAR1,SAR0,STOP); -- list of states
  signal state: StateType;

begin
  process(clk50,reset)
  begin
    ...
  end process;

end architecture;
```

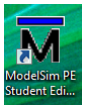
se nästa sida

```
begin
  process(clk50,reset)
  begin
    if reset='1' then
      state <= IDLE;
    elsif rising_edge(clk50) then
      case state is
        when IDLE =>
          Q <= "1000";   -- half of the maximum value
          state <= SAR3;
        when SAR3 =>
          Q(3) <= D;   -- MSB
          Q(2) <= '1';
          state <= SAR2;
        when SAR2 =>
          Q(2) <= D;
          Q(1) <= '1';
          state <= SAR1;
        when SAR1 =>
          Q(1) <= D;
          Q(0) <= '1';
          state <= SAR0;
        when SAR0 =>
          Q(0) <= D;
          state <= U;
        when others =>  -- STOP state
          -- nothing
      end case;
    end if;
  end process;
```



Simulering av SARen

Demo



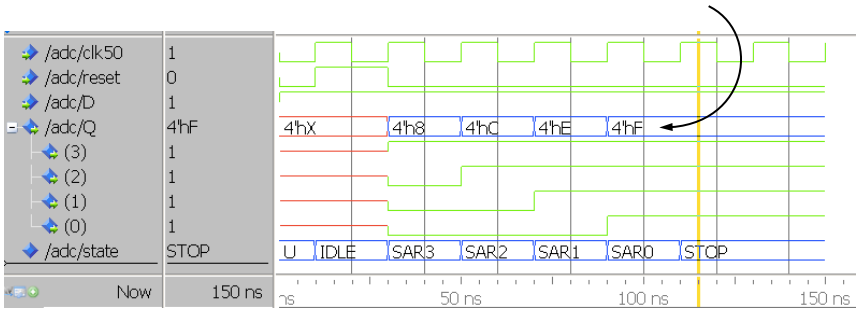
Vi kan inte simulera hela A/D-omvandlaren i ModelSim, eftersom den innehåller både digital och analog elektronik.

Antag att V_{in} är lika med maxspänningen. Då ger komparatorn D = 1 varje gång.

Exempel på do-fil:

```
# standard header:
vsim work.ADC
view wave
add wave *
#restart -force

# simulation commands:
force clk50 0 0ns, 1 10ns -repeat 20ns
force reset 0 0ns, 1 10ns, 0 30ns
# assume analog input is maximum:
force D 1

run 150ns
```

Gissningarna är i tur och ordning 8, 12, 14,15



Efter A/D-omvandling är *Q* = 1111

Tips för praktiskt användbar A/D

① Återvänd till IDLE efter STOP
- annars blir det bara en enda konvertering

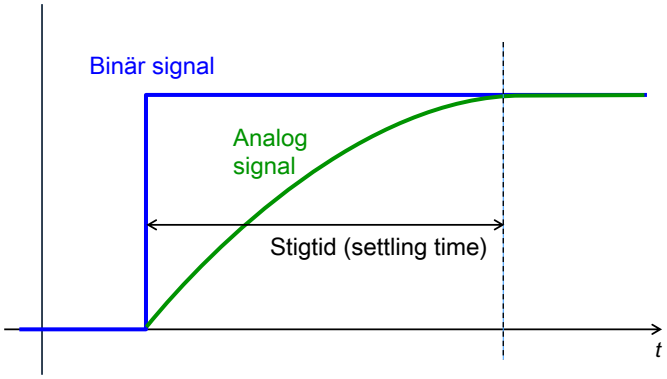
② SARen behöver två periodiska triggsignaler
- en långsam som sätter igång A/D-konverteringen av ett sampel och en snabbare som skiftar tillståndsmaskinen ett steg

③ Trigga SARen långsammare än i exemplet
- annars hinner inte D/A-omvandlaren konvergera, se nästa bild

④ Använd fler bitar än i exemplet
- annars blir signalkvaliteten dålig

Har detta med lab 3 att göra?

D/A-omvandlarens snabbhet



Typiskt stigtid är ca 150 ns

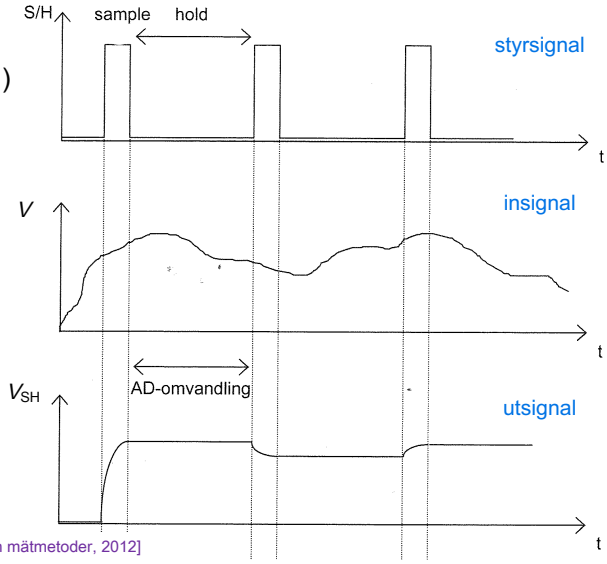
Vad innebär detta för SAR-omvandlaren?

A/D-omvandlarens snabbhet

- D/A-omvandlaren behöver ca 150 ns
- En *n*-bitars A/D-omvandlare enligt SAR-principen behöver därför ca $n \cdot 150$ ns
- Detta ger en övre gräns för frekvensen hos SAR-kretsens triggsignaler
- En 50 MHz-oscillator (20 ns mellan pulserna) är alldeles för snabb

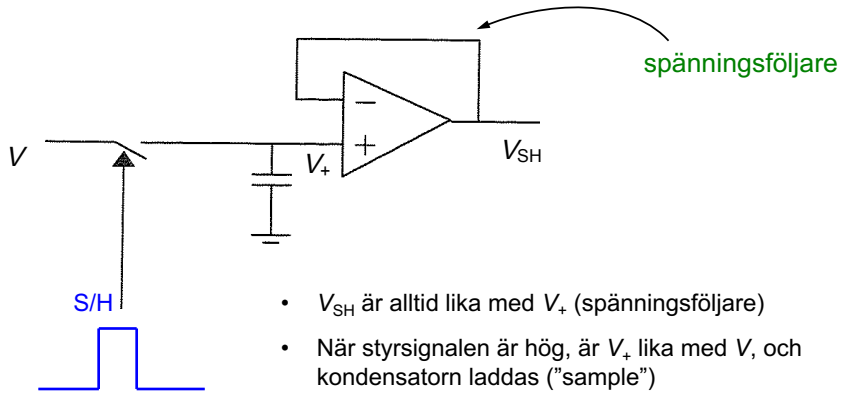
2. Sample-and-hold

En sample-and-hold (S/H) krets ”fryser” en analog spänning en stund.



[Bengtsson: Elektriska mätsystem och mätmetoder, 2012]

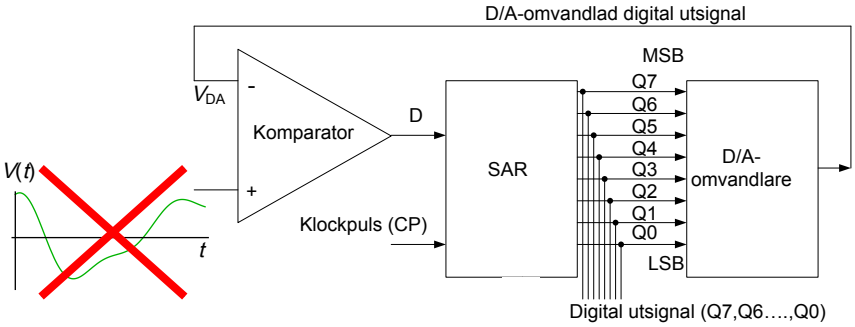
Principen för sample-and-hold



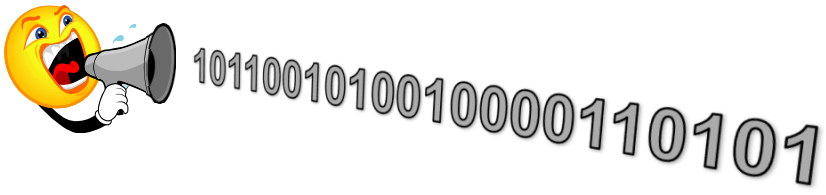
- V_{SH} är alltid lika med V_+ (spänningsföljare)
- När styrsignalen är hög, är V_+ lika med V, och kondensatorn laddas (”sample”)
- När styrsignalen är låg, behåller kondensatorn spänningen V_+ (”hold”)

När har man nytta av en S/H-krets?

Svar: Exempelvis vid A/D-omvandling.
A/Dn behöver en stabil insignal.

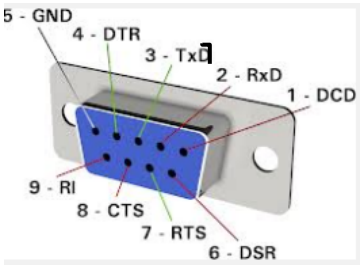


3. Seriell transmission



RS-232-standarden

- Standardiserat protokoll för seriell datakommunikation
- Definierades 1962
- Föregångaren till USB

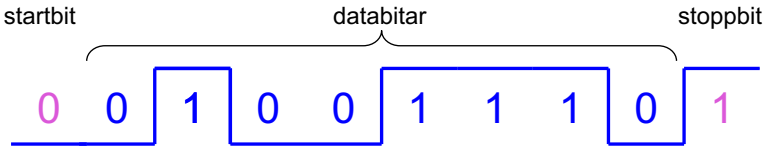


- 9-polig standardkontakt
- Logisk 0 motsvarar hög spänning och 1 är låg
- För varje byte (8 databitar) skickas 1 start- och 1 stoppbit
- LSB skickas först

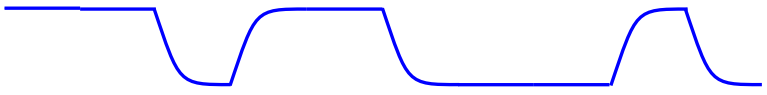
- Flexibla nivåer: från 3 till 15 V för 0 och från –15 till –3 V för 1
- Flexibel datahastighet

Logiska och elektriska nivåer

- Logisk nivå (0 eller 1), syns i ModelSim:

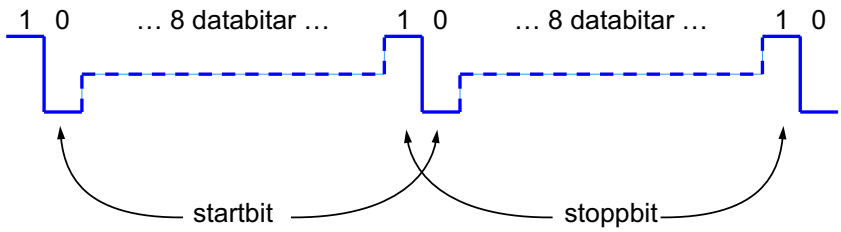


- Elektrisk nivå (spänning), syns på oscilloskop:



Blanda inte ihop dem!

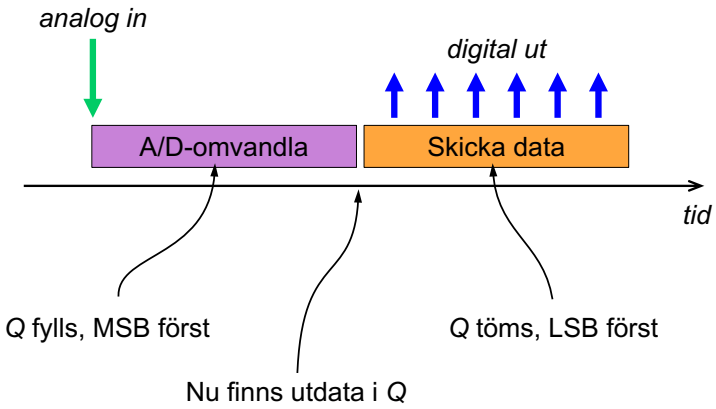
Start- och stoppbitar enligt RS-232



Alltid positiv flank mellan stopp- och startbit

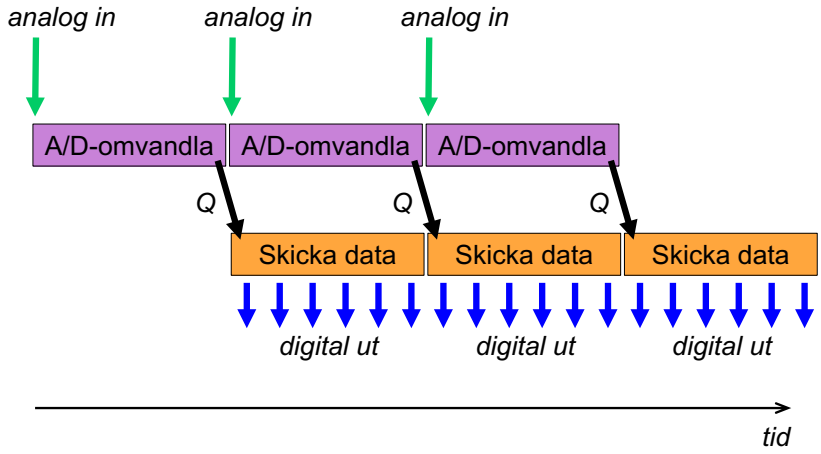
⇒ underlättar synkronisering

Kombination av A/D och seriell transmission



Lab 3 Lab 1

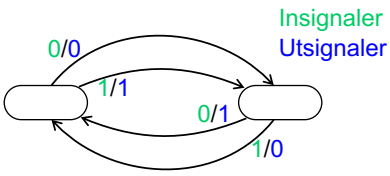
Upprepad A/D-omvandling



4. Mer om tillståndsmaskiner

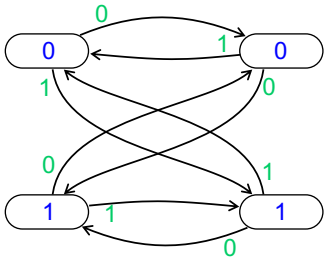
Det finns två typer av tillståndsmaskiner. Exempel (från F7 bild 23):

Mealy



in: 00001111
ut: 01011010

Moore

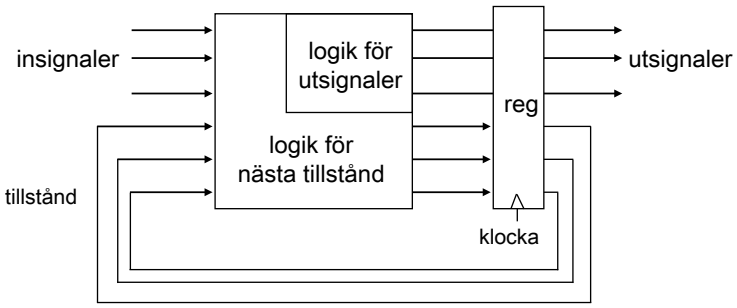


in: 00001111
ut: 01011010

De båda typerna realiserar samma funktion (invertera varannan bit)

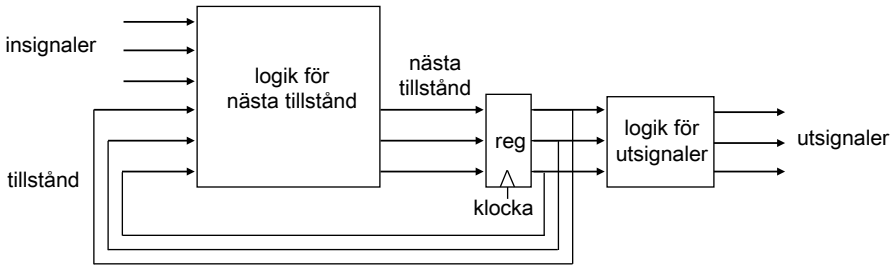
Principschema, Mealy

Mealy: utsignalerna beror på tillståndet och insignalerna



Principschema, Moore

Moore: utsignalerna beror endast på tillståndet

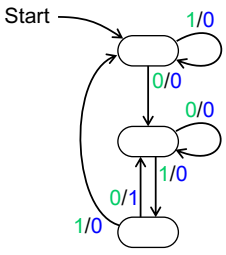


- Moore-typen kräver ofta fler tillstånd än Mealy.
- Mealy-typen finns i två varianter, synkron och asynkron. I den här kursen använder vi synkron Mealy.

Mönsterigenkänning

- Mönsterigenkänning innebär att leta efter ett visst mönster i en insignal
- *Problem:* En synkron krets skall konstrueras för att söka efter sekvensen ”010” i en insignal. Varje gång detta inträffar skall utsignalen vara 1, annars noll.

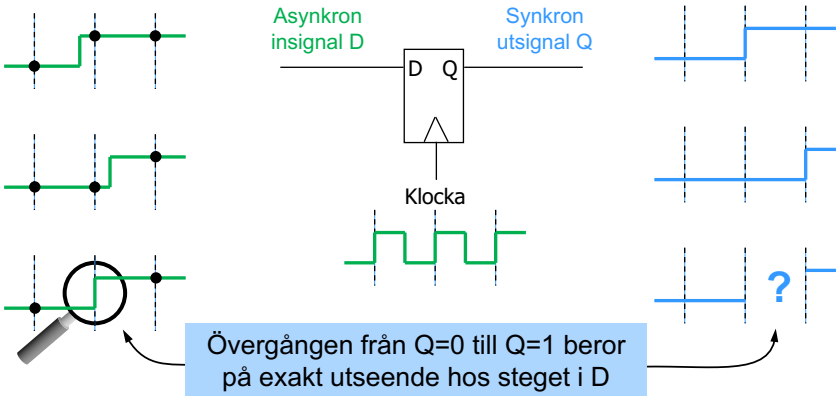
In: 00010011010101
Ut: 00001000001010



□

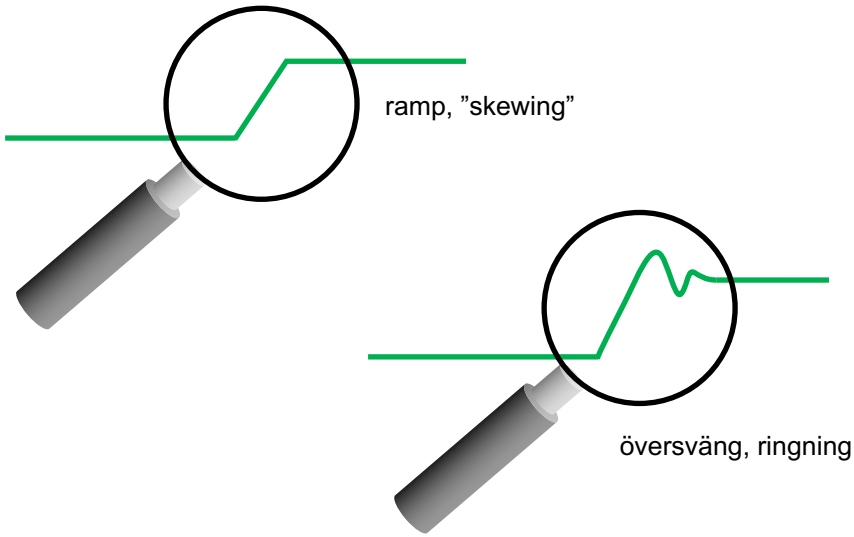
5. Synkronisering av asynkron insignal

I mötet mellan asynkront och synkront kan konstiga saker hända…



Övergången från Q=0 till Q=1 beror på exakt utseende hos steget i D

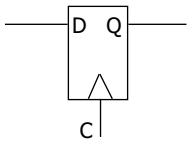
Språng i verkligheten

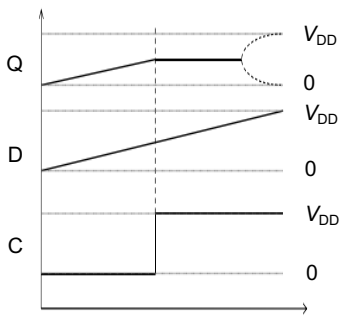


Risk 1: metastabilitet

Digitala signaler kan i hårdvara inte hoppa momentant mellan '0' och '1'.

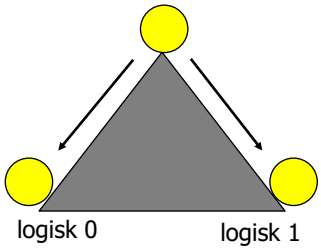
- Antag att insignalen D till vippan slår om långsamt i förhållande till klockan C, och att C slår om precis när D är vid $V_{DD}/2$.
- Då låser sig vippan vid ett mellanläge. Efter en tid slår vippan om till antingen '0' eller '1'.
- Fenomenet kallas metastabilitet

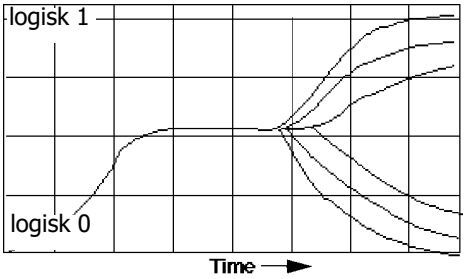




Metastabilitet uppstår när vippans ingång ändras nära klockans flank

- vippan hamnar i ett nästan stabilt (d.v.s ”metastabilt”) tillstånd – varken 0 eller 1
- den kan stanna i detta tillstånd obegränsat länge
- osannolikt men inte omöjligt

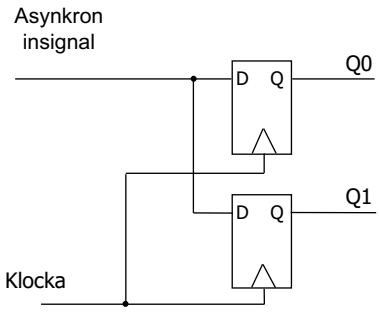




Hur det kan se ut på ett oscilloskop när metastabilitet uppträder

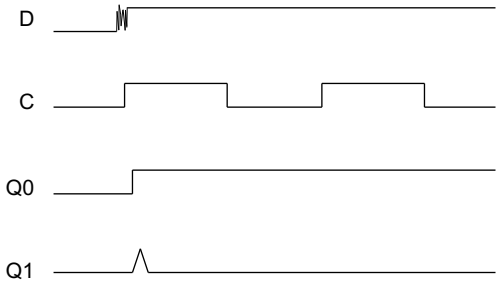
Asynkron insignal till två vippor

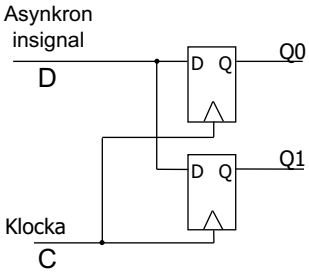
Ännu värre kan det bli om samma asynkrona insignal går till två vippor



Här förväntar man sig att Q0=Q1 alltid, eller hur?

Risk 2: Otillåtet tillstånd





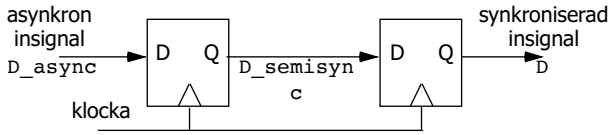
En vippa tolkar D som 0, den andra som 1

⇒ Q0 ≠ Q1, otillåtet tillstånd!

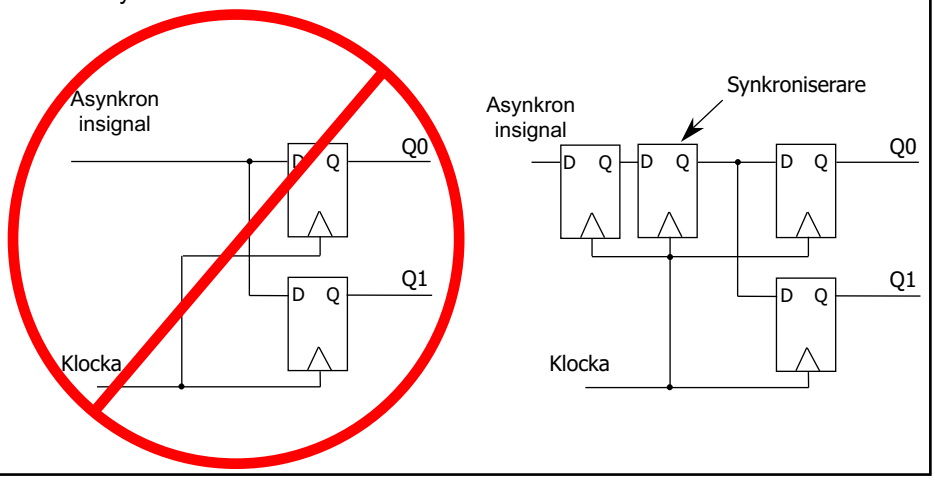
Synkronisering av asynkron ingång

- Risken för metastabilitet och förbjudna tillstånd kan aldrig elimineras helt, men den kan minskas radikalt
  - Använd långsammare systemklocka, vilket ger vippan mer tid för att nå ett stabilt tillstånd
  - Använd snabbast möjliga logikteknologi
  - Klocka den asynkrona signalen, gärna två gånger

```
process(clk50)
  begin
    if rising_edge(clk50) then
      D_semisync <= D_async;
      D <= D_semisync;
    end if;
  end process;
```



- Aldrig låta asynkrona insignaler att vara kopplade till mer än en vippa
  - synkronisera så snart som möjligt och behandla därefter signalen som synkron



Läs & lös

Nästa föreläsning: 22 sept i J121

Läs:

- Bengtsson avsnitt 4.4 (sample-and-hold)
- ”Kretskonstruktion med VHDL”, avsnitt 8
- Lab-PM avsnitt 3

Lös:

- Använd en slumpmässig sekvens med 10 bitar som insignal till tillståndsmaskinerna på bild 25 och verifiera att de ger samma utsignal
- Förberedelseuppgifter till Lab 3 (jippi, SAR!)
- Inlämningsuppgift 2
- Förbeberedelse till Lab 4: uppg 4.1–4.3, 4.6