

Föreläsning 7

A/D-omvandling och tillståndsmaskiner

Erik Agrell 2017-09-12

Innehåll:

1. Tre sorters A/D-omvandlare
2. Tillståndsmaskiner
3. Digital implementation av SAR

Bilder av E. Agrell och A. Linde

Veckosammanfattning

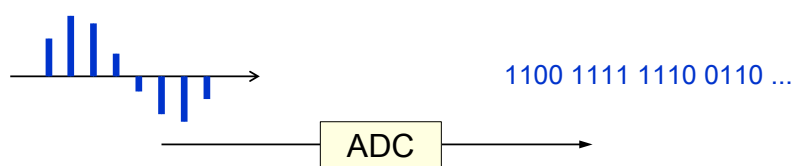
Efter kursvecka 2 behärskar ni...

- VHDL-simulering med ModelSim
 - med kommandon (FORCE, RUN)
 - med do-fil
- VHDL-syntes i Quartus
 - Kompilering
 - Pinn-tilldelning
 - Syntes
 - Test och felsökning

- D/A-omvandling
 - Teoretiska principer
 - Inkoppling av DAC0808
 - Viktade resistanser (inte bra)
 - R-2R-stege (bra)
 - Unipolär och bipolär
- Kretssimulering med LTSpice
 - Rita kopplingsschema
 - Ange komponentvärden och signalkällor
 - Simulera DC_nivåer (Op pnt), tidssignaler (Transient) och frekvensgång (AC analysis)
 - Använda mätkorset

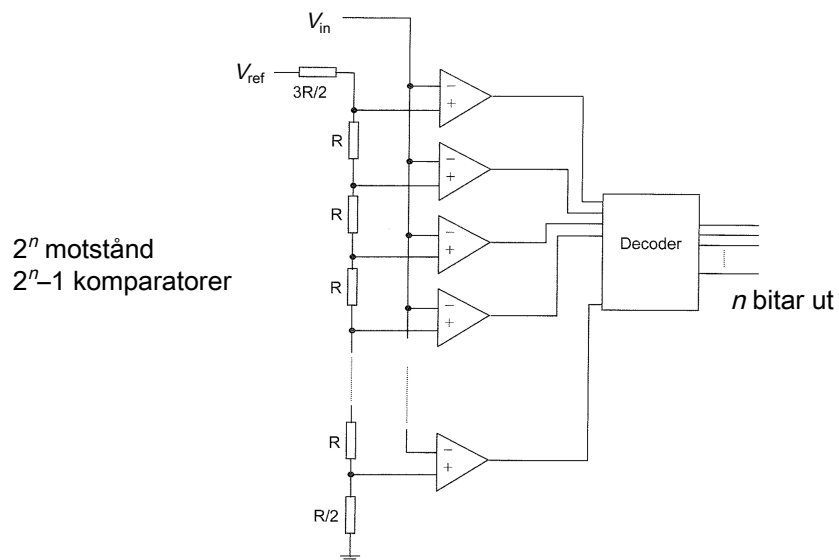
D → A	
111	7Δ
110	6Δ
101	5Δ
100	4Δ
011	3Δ
010	2Δ
001	Δ
000	0

1. Tre sorters A/D-omvandlare

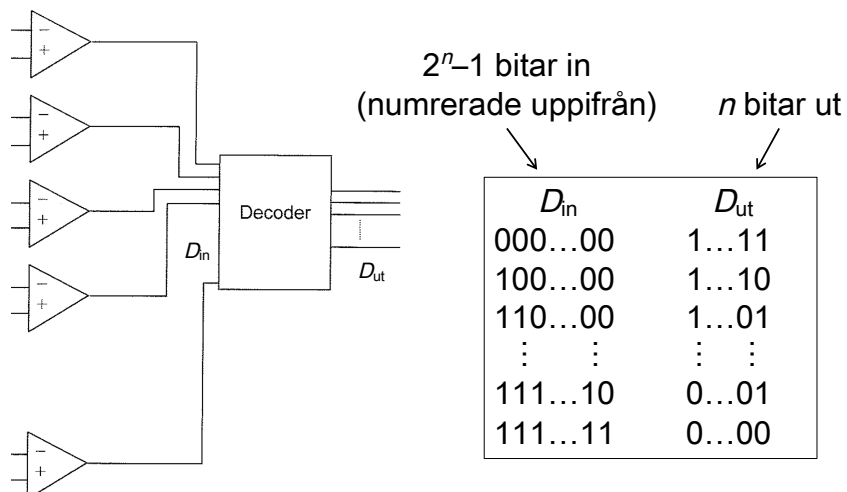


1. Flash-omvandling
2. Successive approximation, SAR
3. Dual-slope-omvandling

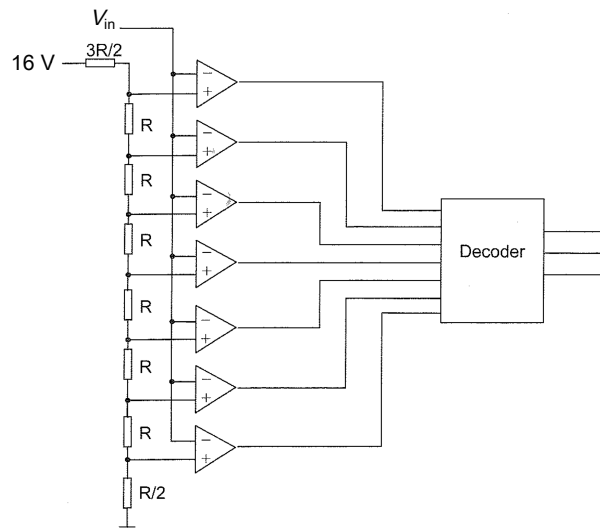
Metod 1: Flash-omvandling



Flash-omvandlarens avkodare



Exempel ($n=3$ bitar)

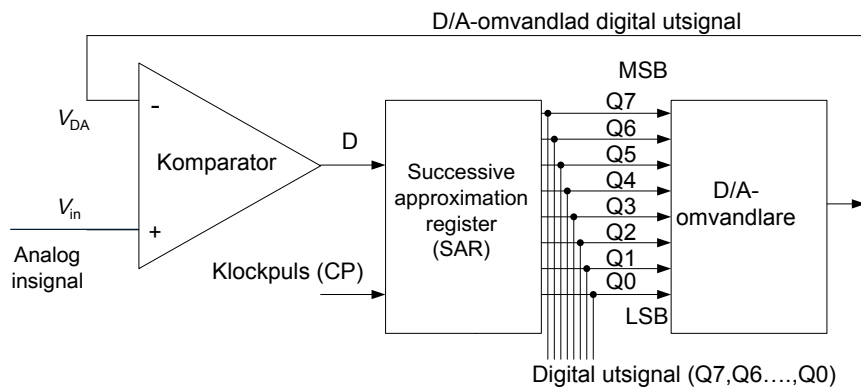


Flashomvandlarens egenskaper:

- + Mycket snabb (konstant tid oavsett n)
- Stor krets (yta $\sim 2^n$) $\Rightarrow$ stor kostnad och energiförbrukning

Används i oscilloskop (där snabbhet är viktigare än upplösning)

Metod 2: Successiv approximation



Varje fråga...

- ... ger 1 bits information
- ... halverar osäkerheten
- ... kostar 1 clockcykel

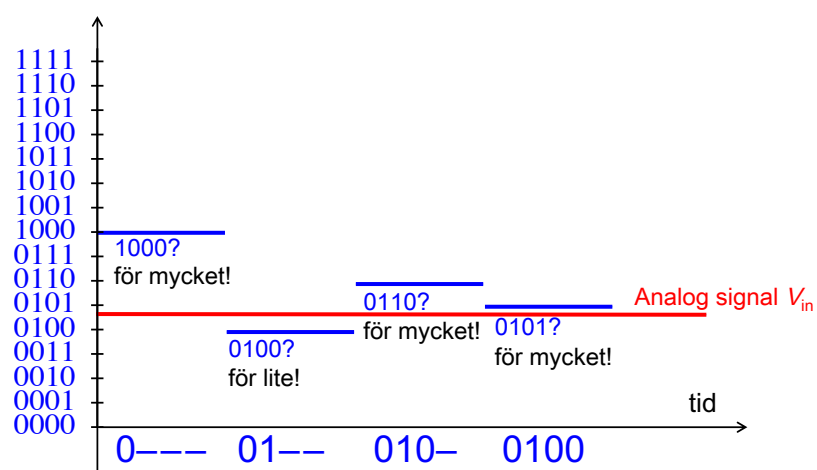
n frågor...

- ... ger n bits information
- ... ger upplösningen $\Delta = V_{\max}/2^n$
- ... kostar n klockcykler

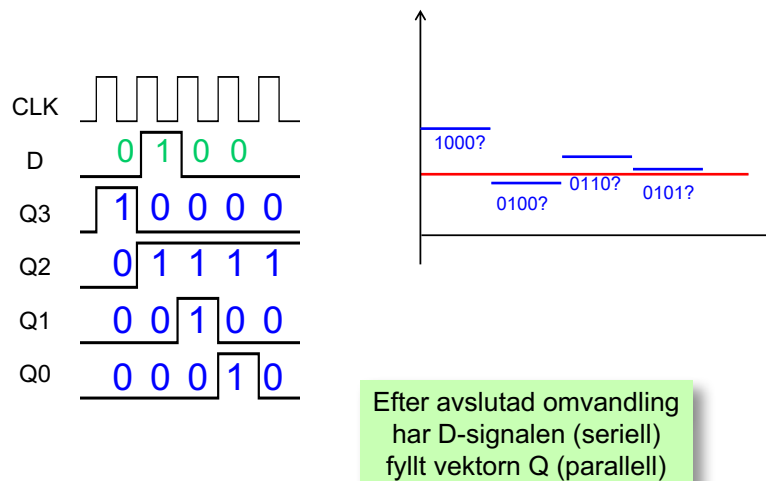
SAR-algoritmen

1. Gissa på $V_{\max}/2$ 10000...
2. För lite $\Rightarrow$ öka med $V_{\max}/4$ 11000...
För mycket $\Rightarrow$ minska med $V_{\max}/4$ 01000...
3. För lite $\Rightarrow$ öka med $V_{\max}/8$
För mycket $\Rightarrow$ minska med $V_{\max}/8$
- ...

Exempel



Tidsdiagram

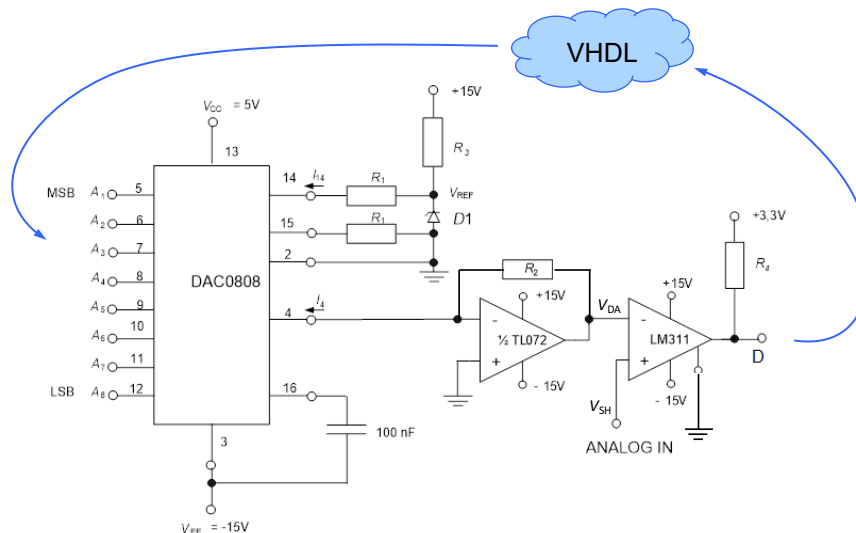


SAR-omvandlarens egenskaper:

- + Mycket mindre komplex än flash (yta $\sim n$)
- Långsammare än flash (tid $\sim n$)
- Kräver DAC

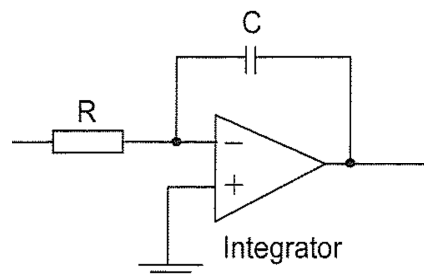
Används i digital kommunikation, t.ex. lab 3

SAR-omvandlaren i lab 3



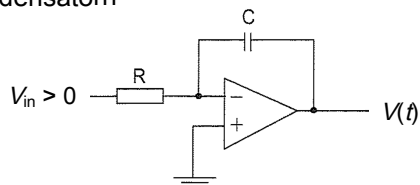
Metod 3: Dual-slope-omvandling

Dual-slope-omvandlingen utnyttjar en integrator, som först laddas upp och sedan laddas ur



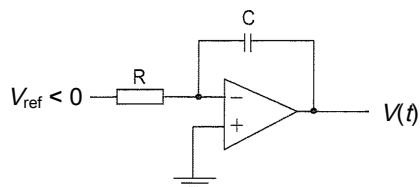
Steg 1: Uppladdning av kondensatorn

$$0 \leq t < t_1$$



Steg 2: Urladdning av kondensatorn

$$t_1 \leq t < t_1 + t_2$$



t_2 väljs så att $V(t_1 + t_2) = 0$, d.v.s. helt urladdad kondensator

Dual-slope-algoritmen

1. Uppladdning av kondensatorn

- Okänd spänning V_{in}
- Känd tid t_1

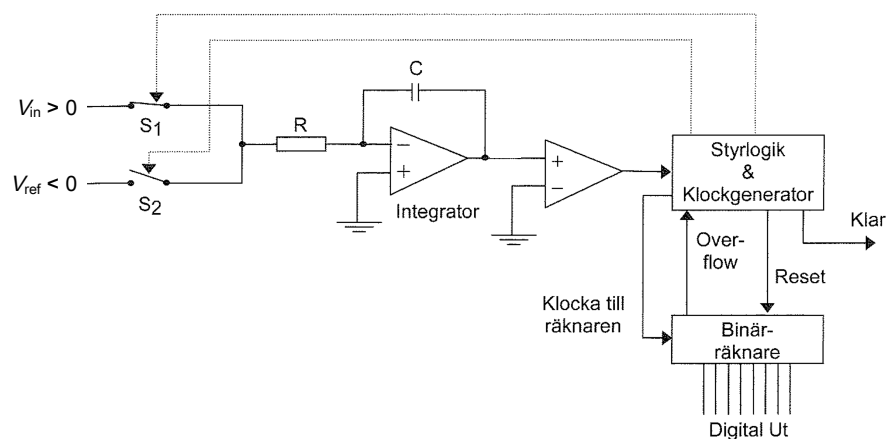
2. Urladdning av kondensatorn

- Känd spänning V_{ref}
- Uppmätt tid t_2

3. Beräkning

- $V_{in} = -\frac{t_2}{t_1} V_{ref}$

Dual-slope-omvandlarens implementering



Dual-slope-omvandlarens egenskaper:

- + Mycket noggrann
 - Tidmätning kan göras mycket noggrann (kristallklocka)
 - Resultatet är oberoende av RC
- + Kräver ingen DAC
- Mycket långsam

Används i voltmetrar (där upplösning är viktigare än snabbhet)

2. Tillståndsmaskiner

- Ett **tillståndsmaskin** består av ett antal tillstånd (lägen, moder) med övergångar mellan
- Används för att beskriva system vars beteende ändras som en följd av tidigare åtgärder
- Abstrakt? Nej, vi använder tillståndsmaskiner varje dag!
- *Exempel:* Funktionen hos en vanlig dörr

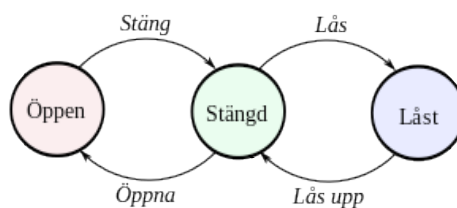
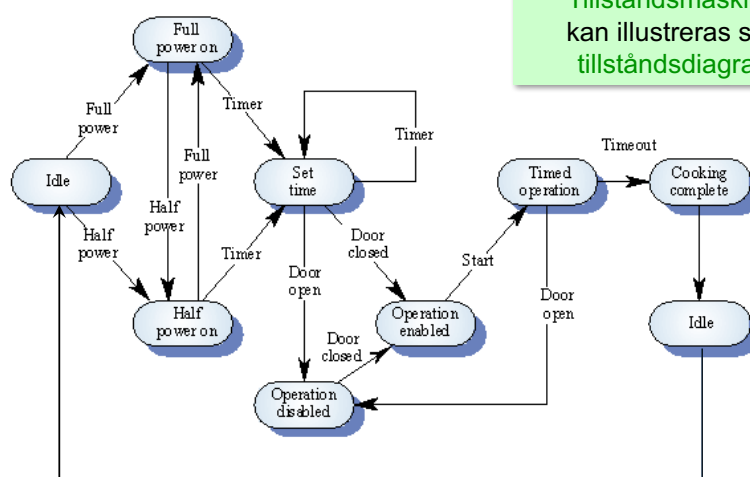


Illustration från Wikipedia

Exempel (mikrovågsugn)

Tillståndsmaskiner
kan illustreras som
tillståndsdigram



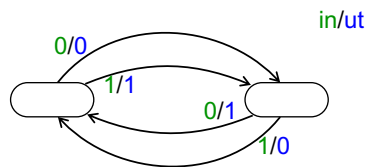
Källa: http://nas.uhcl.edu/helm/swen5231/Ch_16_1/sld017.htm

Digitalt exempel

Invertera varannan bit i en sekvens:

in: 00001111

ut: 01011010



VHDL-kod

```

library ieee;
use ieee.std_logic_1164.all;

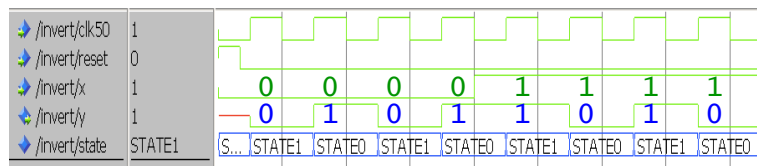
entity invert is port(
  clk50, reset: in std_logic;
  x: in std_logic;
  y: out std_logic);
end entity;

architecture state_machine of invert is
  type StateType is (U, STATE0, STATE1);
  signal state: StateType;

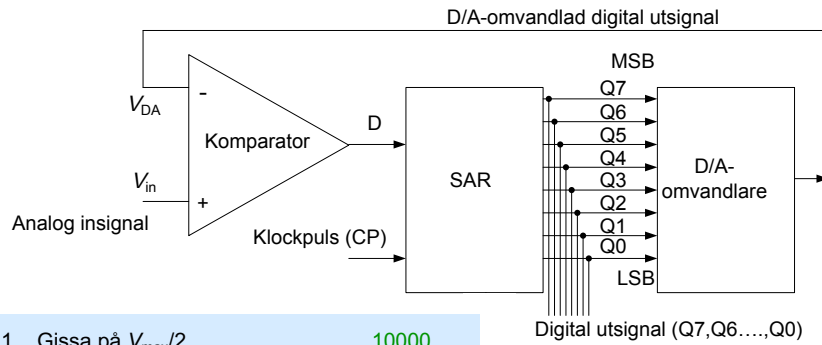
begin
  process(clk50, reset)
    begin
      if reset='1' then
        state <= STATE0;
      elsif rising_edge(clk50) then
        case state is
          when STATE0 =>
            y <= x;
            state <= STATE1;
          when others => -- STATE1
            y <= not x;
            state <= STATE0;
        end case;
      end if;
    end process;
  end architecture;

```

Nytt: **type**
definierar en
egen datatyp

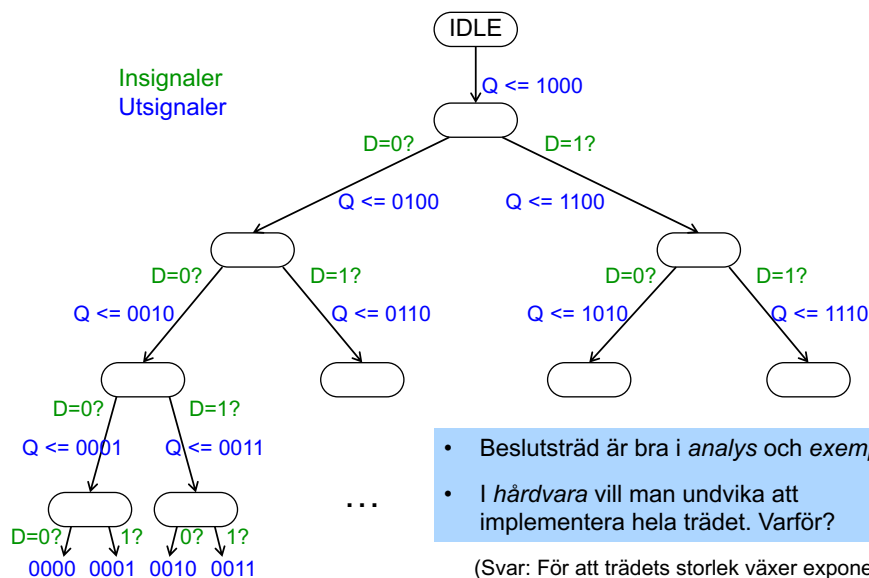


3. Digital implementation av SAR

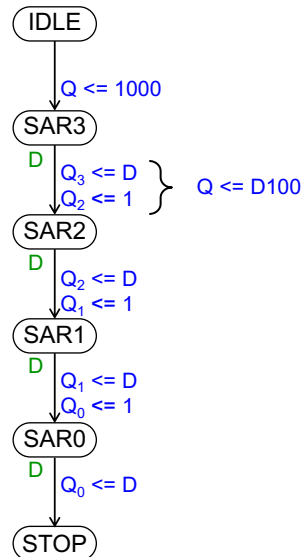


1. Gissa på $V_{max}/2$ 10000...
2. För lite $\Rightarrow$ öka med $V_{max}/4$ 11000...
För mycket $\Rightarrow$ minska med $V_{max}/4$ 01000...
3. För lite $\Rightarrow$ öka med $V_{max}/8$
För mycket $\Rightarrow$ minska med $V_{max}/8$
- ...

Beslutsträd



Tillståndsdigram – bättre



VHDL-kod

```

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;

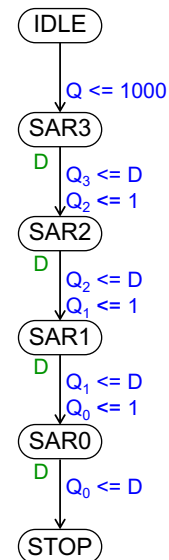
```

} 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;

```



Läs & lös

Läs:

- Bengtsson s 154–162 och 165–168

(U_{ref} skall vara $-U_{ref}$ i ekvationerna på s 158–159, eftersom $U_{ref} < 0$)

Lös:

- Övningsuppgift SAR (nästa sida)
- Bengtsson uppg 4.10, 4.11, 4.13 och 4.16

Övningsuppgift SAR

En 4-bitars A/D-omvandlare accepterar inspänningar i intervallet 0–7.5 V utan att det blir overflow. Den fungerar enligt SAR-principen.

- (a) Vad blir den digitala utsignalen om insignalen är 5.8 V?
- (b) Vilken sekvens av digitala gissningar Q kommer SAR-kretsen att generera för en insignal på 5.8 V, och vilka utsignaler D kommer komparatorn att ge för vart och ett av dessa Q-värden? Komplettera beslutsträdet på bild 26 och rita in den aktuella vägen.
- (c) Följ tillståndsdigrammet på bild 27 för samma insignal. Ange Q-värdet i varje steg och verifiera att utsignalen till slut blir samma som i (a) och (b).

4.10 En dual slope-AD liknande den i figur 4.63, har en binärräknare bestående av 10 bitar och en referensspänning på -5 V. Klockan till binärräknaren går med 1 MHz.

- a) Inom vilket spänningsområde får A_{in} ligga? (Här är $A_{in} = V_{in}$)
- b) Hur lång tid skulle AD-omvandlingen ta om $A_{in} = 3$ V?
- c) Rita upp- och urladdningskurvan om $RC = 1$ ms (och $A_{in} = 3$ V).
- d) Anta att resistorns och kondensatorns värden ändras (på grund av temperaturvariationer eller åldrande) så att $RC = 0,9$ ms. Rita upp- och urladdningskurvan även i detta fall. Hur påverkas resultatet av AD-omvandlingen av detta?

4.11 Spänningen 2,84 V ska AD-omvandlas i en 6-bitars successiv approximationsomvandlare med referensspänningen 5 V (vilket innebär att DA-omvandlaren har en referensspänning på -5 V).

- Hur många jämförelser görs innan AD-omvandlingen är klar?
- Rita ett tidsdiagram som visar DA-omvandlarens utsignal under omvandlingen.
- Vad blir resultatet av omvandlingen och hur stort blir felet?

4.13 Figur 4.65 visar de möjliga spänningsnivåerna vid de olika jämförelserna i en successiv approximations-AD.

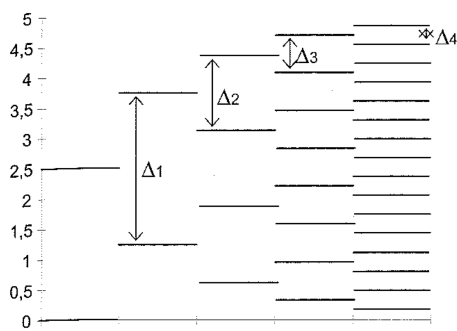


Fig 4.65 Alla möjliga jämförelsenivåer

- Hur många bitar består AD-omvandlaren av?
- Vilken referensspänning har den?
- Beräkna $\Delta_1 - \Delta_4$.
- Hur stor är AD-omvandlarens upplösning?
- Vilken utsignal skulle AD-omvandlaren ge för insignalen 2,1 V? Gör inga beräkningar utan använd en linjal och mät i figur 4.65

4.16a) Hur många komparatorer består AD-omvandlaren i figur 4.67 av?

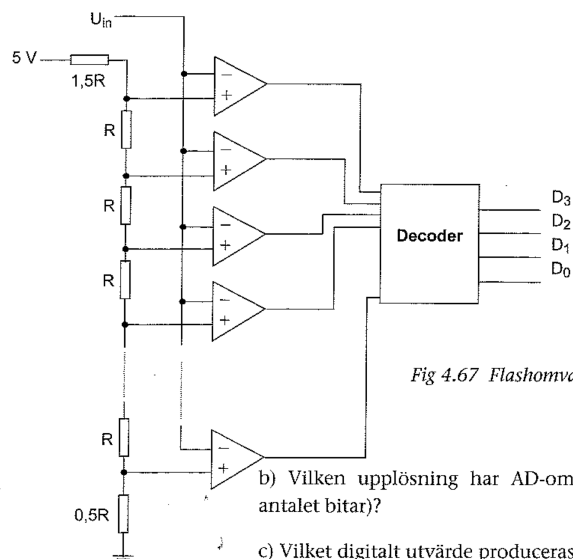


Fig 4.67 Flashomvandlare

b) Vilken upplösning har AD-omvandlaren ovan (uttryckt i volt och antalet bitar)?

c) Vilket digitalt utvärde produceras om $U_{in} = 3,79$ volt?

d) Inom vilket intervall ligger U_{in} om $D_{ut} = 12$?