

Föreläsning 4 ***D/A-omvandling***

Erik Agrell 2017-09-05

Innehåll:

1. Tre sätt att simulera i ModelSim
2. Principer för D/A- och A/D-omvandling
3. Konstruktion av D/A-omvandlare
4. Bipolär D/A-omvandling

Bilder av E. Agrell och A. Linde

Veckosammanfattning

Efter förra veckan behärskar ni...

- VHDLs grunder
 - bibliotek, entitet och arkitektur
 - in- och ut-signaler
 - signaltilldelning med **<=**
 - vektorer
 - heltalsaddition
- Parallell VHDL (utanför process)
 - grindnivå (**and**, **or**, **not**)
 - interna signaler
 - **with**

- Sekventiell VHDL (innanför process)
 - sensitivitetslista
 - variabler
 - **if** och **case**
 - asynkron VHDL (utanför **if rising_edge()**)
 - synkron VHDL (innanför **if rising_edge()**)
 - reset-signaler (asynkron och synkron)
- Simulering med ModelSim

1. Tre sätt att simulera i ModelSim

1. Klicka och skriv kommandon
 - som vi har gjort hittills (Add Wave, force, run,...)
-  2. Exekvera en do-fil (script)
 - praktiskt om man vill köra samma simulering flera gånger, t.ex. vid felsökning
3. Kör ett VHDL-program för att testa ett annat (testbänk)
 - inte i den här kursen, men viktigt i stora projekt

Do-filer

Arbetsgång:

1. Starta ModelSim och öppna ett projekt med en VHDL-fil, t.ex. and-grinden från föreläsning 1 och 2:

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

entity and_gate is
port (
  x,y: in std_logic;
  f: out std_logic);
end entity;

architecture arch of and_gate is
begin
  f <= (x and y);
end architecture;
```

2. Kompilera

3. Välj *File > New > Source > Do*
4. Skriv eller klistra in lämpliga ModelSim-kommandon, t.ex.

```
# standard header:
vsim work.and_gate
view wave
add wave *

# simulation commands:
force x 0 0ns, 1 200ns
force y 0 0ns, 1 100ns, 0 400ns
run 500ns
```

Starta simulering och välj entitet **and_gate**

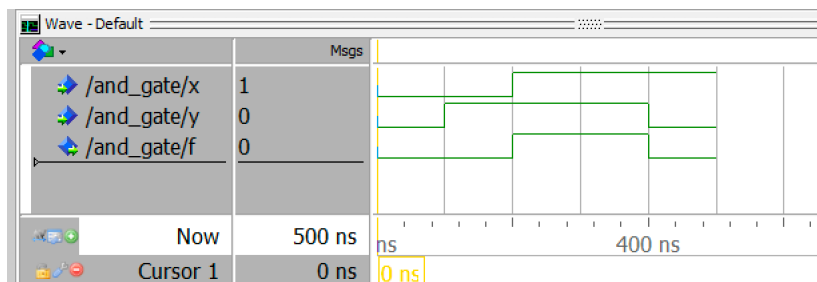
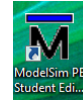
Lägg till alla signaler till **wave-fönstret**

ModelSims **simuleringskommandon**

- De tre första raderna är ett standardhuvud som man kan använda i alla sina do-filer (och ändra entitetsnamnet på första raden)
- Kommentarer markeras med #
- Simuleringskommandon kan kombineras med semikolon

5. Spara filen och kalla den t.ex. **and_gate.do**
6. Ekekvra do-filen genom att skriva
do and_gate.do
i kommandofönstret

Demo



Samma resultat som när vi simulerade manuellt

Textfiler

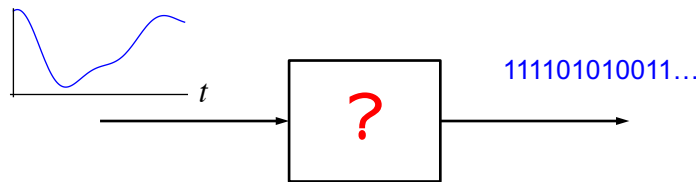
VHDL-filer (.vhd) och DO-filer (.do)
sparas som ren text



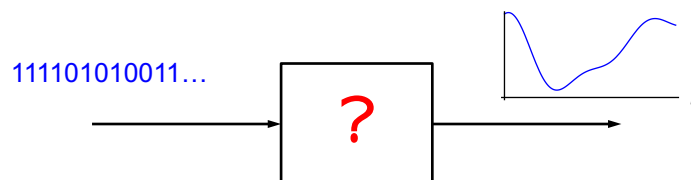
Kan redigeras i vilken editor som helst,
även utanför ModelSim och Quartus

2. Principer för D/A- och A/D-omvandling

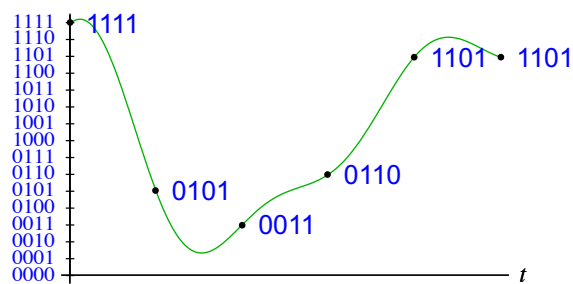
Ofta har man en vågform (ljud, video, mätdata,...) som man vill representera som bitar:



Eller tvärt om:



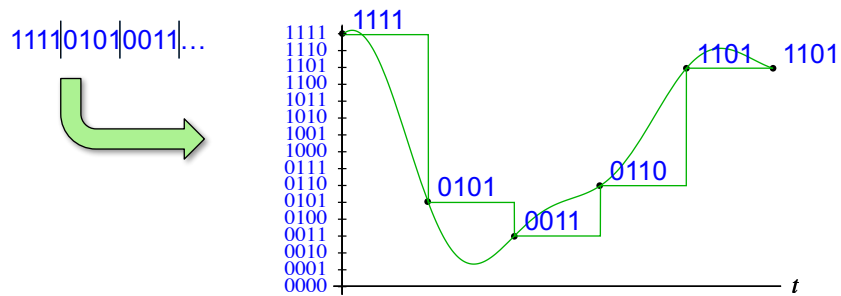
A/D: Från vågform till bitar



1. Sampling:
Hacka signalen i tiden
2. A/D-omvandling (kvantisering):
Representera varje tidssampel som en sträng bitar
3. Sätt ihop strängarna

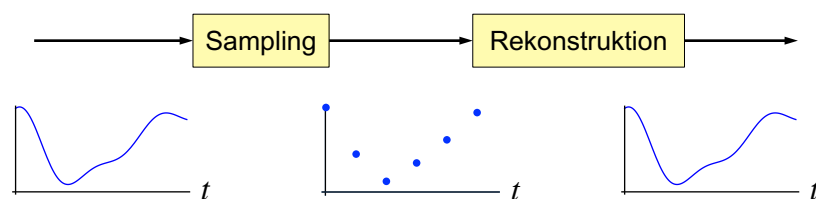
1111|0101|0011|...

D/A: Från bitar till vågform



1. Dela upp bitsekvensen i ord av samma längd
2. **D/A**-omvandling:
Representera varje digitalt ord som en spänningsnivå (eller ström)
3. Låt utspänningen ligga konstant tills nästa sampel kommer
4. Lågpasfiltrera

Samplingsteoremet



Villkor för att signalen skall kunna rekonstrueras utan distortion:

$$f_s \geq 2 f_{\max}$$

f_s : samplingsfrekvensen
 $f_{\max}$: signalens högsta frekvens

3. Konstruktion av D/A-omvandlare

Digital/analog (D/A)-omvandling:
att omvandla bitar till spänning
eller ström

D/A-omvandlare finns som
integrerade kretsar, t.ex. DAC0808

1111	+	1.5 V
1110	+	1.4 V
1101	+	1.3 V
1100	+	1.2 V
1011	+	1.1 V
1010	+	1.0 V
1001	+	0.9 V
1000	+	0.8 V
0111	+	0.7 V
0110	+	0.6 V
0101	+	0.5 V
0100	+	0.4 V
0011	+	0.3 V
0010	+	0.2 V
0001	+	0.1 V
0000	+	0.0 V



Datablad

Att läsa datablad är ofta en förutsättning för att använda komponenten

 **National Semiconductor**

May 1999

DAC0808
8-Bit D/A Converter

General Description

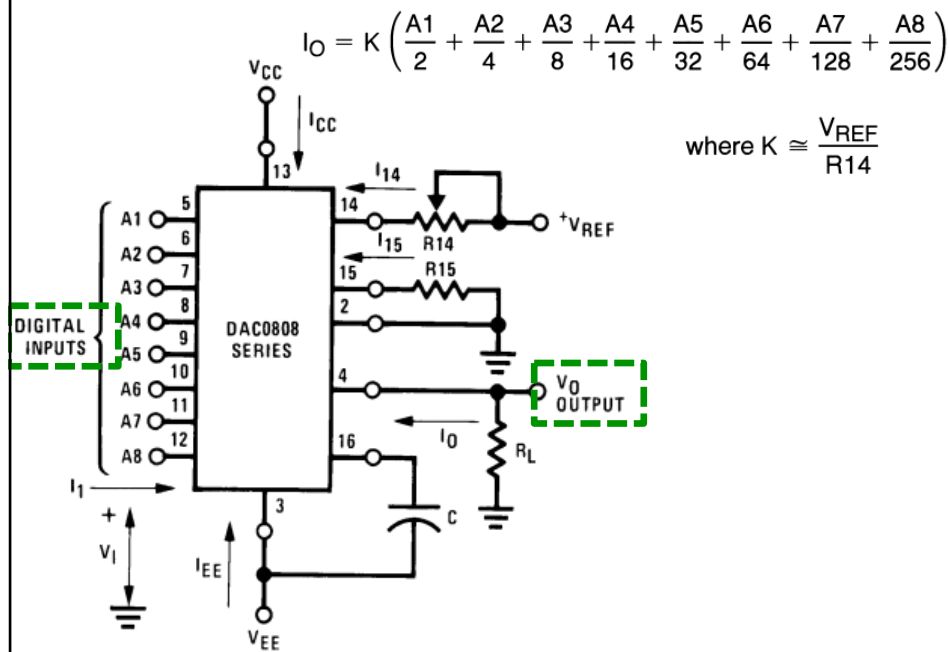
The DAC0808 is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with $\pm 5V$ supplies. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ± 1 LSB of $255 I_{REF}/256$. Relative accuracies of better than $\pm 0.19\%$ assure 8-bit monotonicity and linearity while zero level output current of less than $4 \mu A$ provides 8-bit zero accuracy for $I_{REF} \geq 2$ mA. The power supply currents of the DAC0808 is independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

The DAC0808 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the MC1508/MC1408. For higher speed applications, see DAC0800 data sheet.

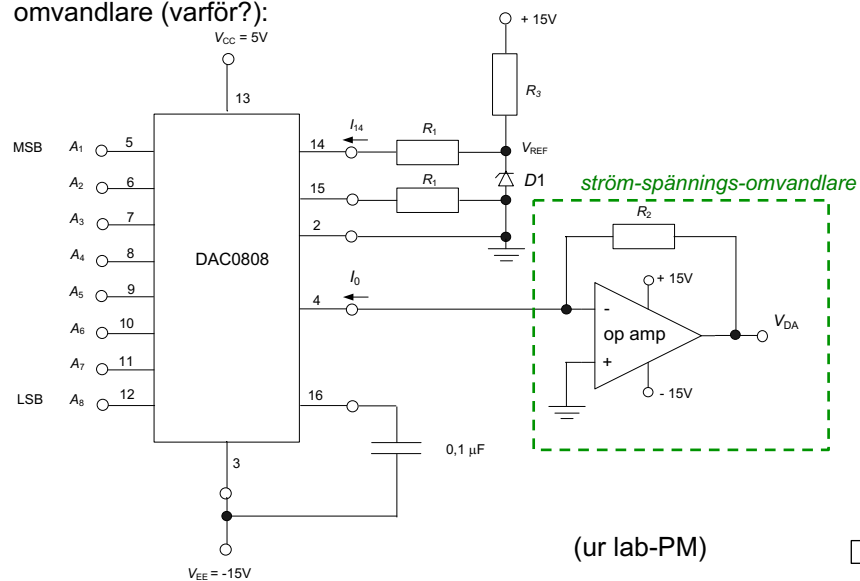
Features

- Relative accuracy: $\pm 0.19\%$ error maximum
- Full scale current match: ± 1 LSB typ
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/ μs
- Power supply voltage range: $\pm 4.5V$ to $\pm 18V$
- Low power consumption: 33 mW @ $\pm 5V$

DAC0808 8-Bit D/A Converter

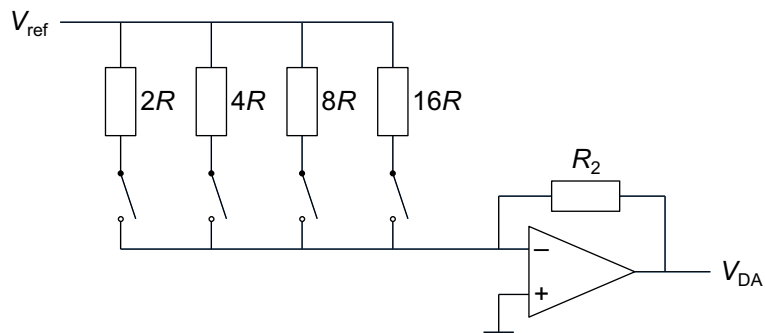


Vi kombinerar D/A-omvandlaren med en ström-spännings-omvandlare (varför?):



Två konstruktionsmetoder av D/A

1. Viktade resistanser

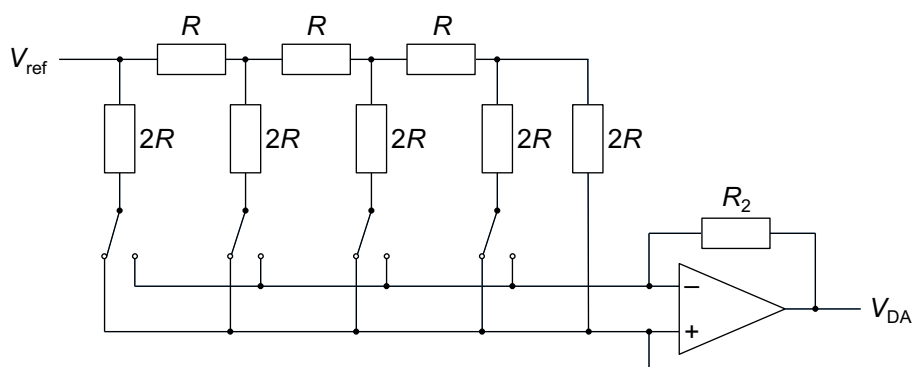


Enkel men onoggrann konstruktion

(varför?)



2. R-2R-stege



Stabil och populär konstruktion

(varför?)



Varför R-2R-stege?

- Alltid samma ström genom motstånden, oavsett omkopplarnas lägen
- I kretsen behöver endast två storlekar på motstånd implementeras (eller en storlek, om $2R$ realiseras som $R+R$ i serie)
- I integrerad kretsteknologi kan motstånd av *samma storlek* konstrueras mycket exakt

⇒ Hög noggrannhet, även vid hög ordlängd (många bitar)

Ur databladet för DAC0808

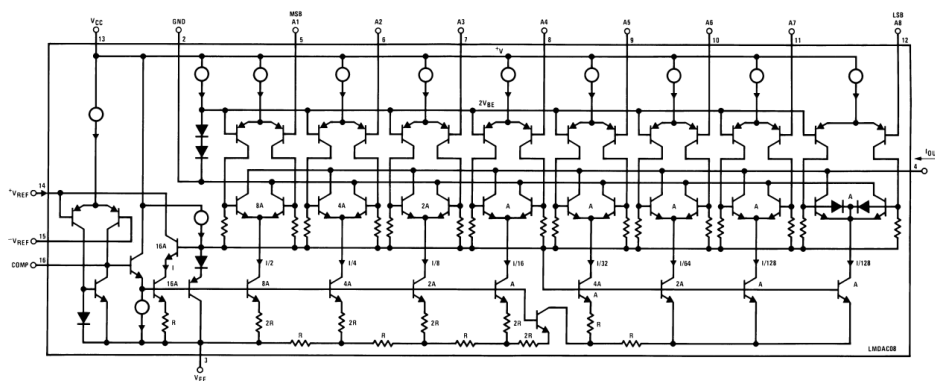
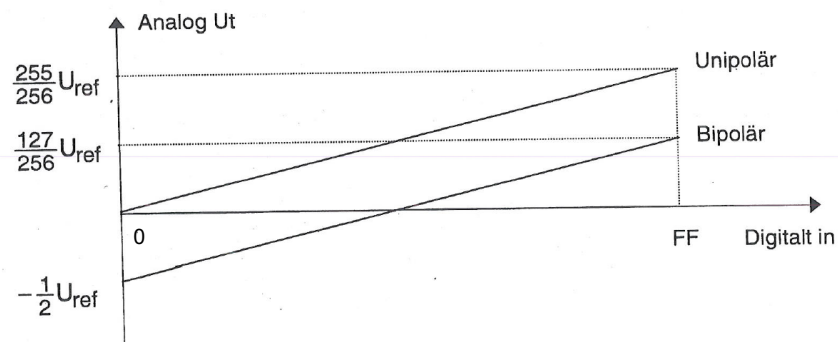


FIGURE 2. Equivalent Circuit of the DAC0808 Series (Note 8)

DS00008F-4

4. Bipolär D/A-omvandling

Spänningssomfånget för en D/A-omvandlare kan vara mellan 0 och U_{ref} (unipolär) eller mellan $\pm U_{\text{ref}}/2$



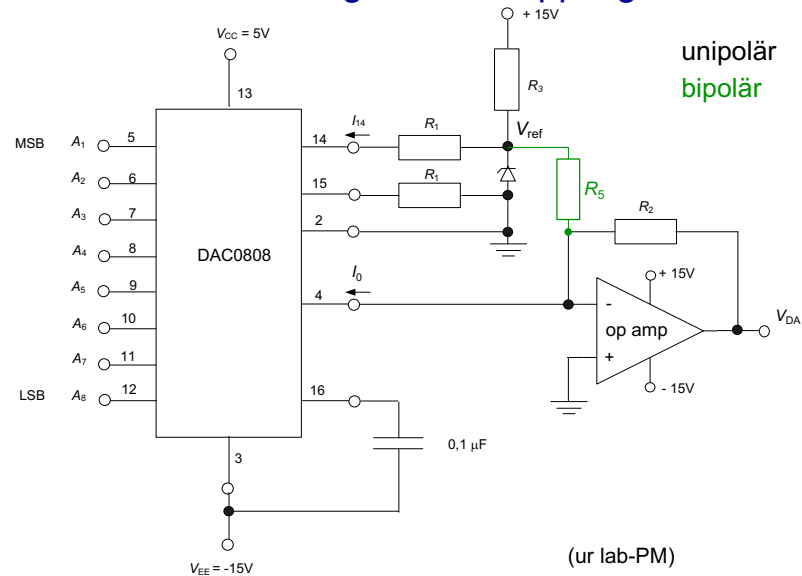
Unipolär (från bild 13):

1111	+	1.5 V
1110	+	1.4 V
1101	+	1.3 V
1100	+	1.2 V
1011	+	1.1 V
1010	+	1.0 V
1001	+	0.9 V
1000	+	0.8 V
0111	+	0.7 V
0110	+	0.6 V
0101	+	0.5 V
0100	+	0.4 V
0011	+	0.3 V
0010	+	0.2 V
0001	+	0.1 V
0000	+	0.0 V

Bipolär:

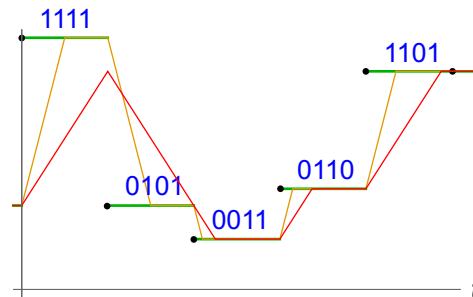
1111	+	0.7 V
1110	+	0.6 V
1101	+	0.5 V
1100	+	0.4 V
1011	+	0.3 V
1010	+	0.2 V
1001	+	0.1 V
1000	+	0.0 V
0111	+	-0.1 V
0110	+	-0.2 V
0101	+	-0.3 V
0100	+	-0.4 V
0011	+	-0.5 V
0010	+	-0.6 V
0001	+	-0.7 V
0000	+	-0.8 V

Modifiering av D/A-kopplingen



- Samma integrerade krets kan användas för unipolär som bipolär D/A
- Nollpunkten kan förskjutas med en ström-bias på kretsens utgång
- Kopplingen på förra sidan blir bipolär om strömmen genom R_5 är $I_{0\max}/2$ (förklara varför!)

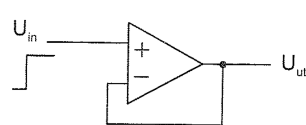
D/A-omvandlarens tidsegenskaper



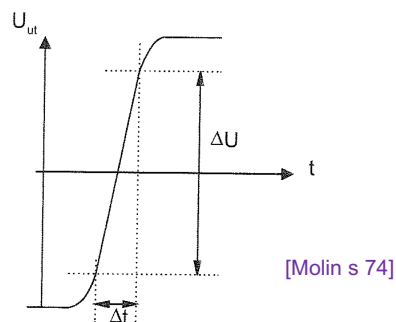
- **Grönt:** Ideal utsignal, stegfunktion
- **Gult:** Mer realistisk utsignal, OK efter filtrering
- **Rött:** För långsam utsignal, ger allvarlig distortion

Slew rate

Slew rate = maximal spänningsförändring per tid



$$\text{Slew rate} = \frac{\Delta U}{\Delta t}$$



Slew rate påverkar hur snabbt varierande signaler elektroniken kan hantera, t.ex. hur snabbt D/A-omvandlaren kan klockas

Läs & lös

Läs:

- Läs Molin 1.3: Analogt kontra digitalt
- Läs Bengtsson 4.2: D/A-omvandling
- Läs (översiktligt) databladet DAC0808
- (Vid behov, repetera Molin 2.3: Op-förstärkarkopplingar)

Lös:

- I inlämningsuppgift 1 lämnas displayen blank om alla switcharna står på 1