

# Appendix 2: Grundläggande A/D- och D/A-omvandling.

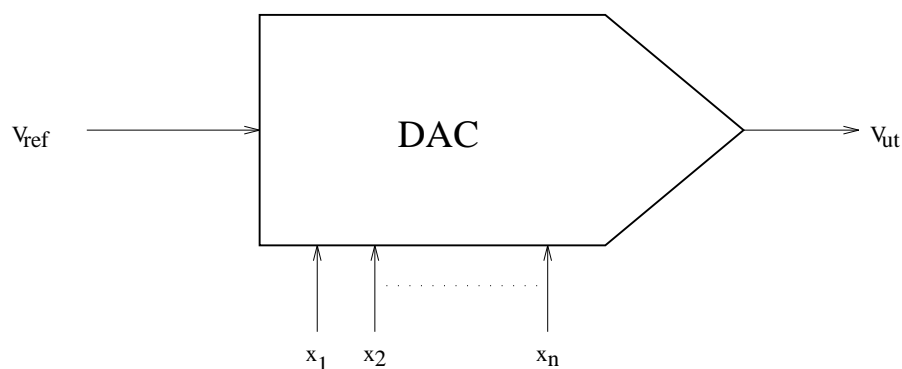
## Inledning

Användningen av datorer ökar alltmer i allt fler sammanhang. Men eftersom vår värld är analog, behöver vi något sätt att omvandla t.ex. mätvärden till digital form, för att datorn ska kunna behandla dem. Efter avslutad beräkning behöver vi kanske styra en process med hjälp av de nya värden, som datorn tagit fram. Då kan vi behöva något sätt att omvandla de digitala värdena till analog form.

Syftet med denna text är att ge grundläggande definitioner för analog/digital-omvandling (avsnitt 2) och digital/analog-omvandling (avsnitt 3). I avsnitt 4 berörs tre olika A/D-omvandlare: trappstegsomvandlaren, den följande omvandlaren och successiv approximationsomvandlaren.

## Digital/analog(D/A)-omvandling

Vid omvandling från digitalt till analogt låter man värdet av ett digitalt ord representeras av t.ex. en analog spänning. Detta sker i en D/A-omvandlare (eng. digital-analog converter, DAC), se figur 42.



Figur 42: Symbol för en D/A-omvandlare

Insignal är det digitala ordet  $(x_1, x_2, \dots, x_n)$  där  $x_1$  anger största värde (most significant bit, MSB) och  $x_n$  anger minsta värde (least significant bit, LSB).  $V_{ut}$  ges av följande överföringsfunktion:

$$V_{ut} = K \cdot V_{ref} \cdot D \quad (1)$$

där:  $V_{ref}$ : referensspänning

$K$ : skalfaktor

$D$ : "värdet" av det digitala ordet

Det digitala ordet skrivs i en form som kallas *fractional binary* och som då ger ett värde enligt följande:

$$D = \frac{x_1}{2^1} + \frac{x_2}{2^2} + \dots + \frac{x_n}{2^n} \quad (2)$$

där  $x$  kan anta värdena 0 eller 1.

Full Scale Voltage ( $V_{FS}$ ) definieras av:

$$V_{FS} = K \cdot V_{ref} \quad (3)$$

Observera att  $V_{ut}$  aldrig kan nå detta värde, eftersom  $D$  alltid är mindre än ett. Alltså:

$$V_{ut} = V_{FS} \cdot D. \quad (4)$$

I litteraturen kan förekomma olika definitioner av  $V_{FS}$ , men den här givna definitionen är den som är vanligast förekommande.

Utsignalen kan bara anta vissa bestämda värden. Den minsta förändring som kan ske, kallas för omvandlarens *upplösning*. Antalet nivåer i utsignalen är  $2^n$  där  $n$  är antal bitar i omvandlaren. Upplösningen ges då av

$$V_{LSB} = \frac{V_{FS}}{2^n}, \quad (5)$$

eftersom en förändring i den minst signifikanta biten motsvarar den minsta möjliga förändringen. Största möjliga utspänning fås när samtliga bitar i det digitala ordet är 1:

$$V_{max} = K \cdot V_{ref} \cdot \left( \frac{1}{2^1} + \frac{1}{2^2} + \dots + \frac{1}{2^n} \right) = V_{FS} \cdot \frac{2^n - 1}{2^n} \quad (6)$$

Detta värde är alltså alltid mindre än  $V_{FS}$ .

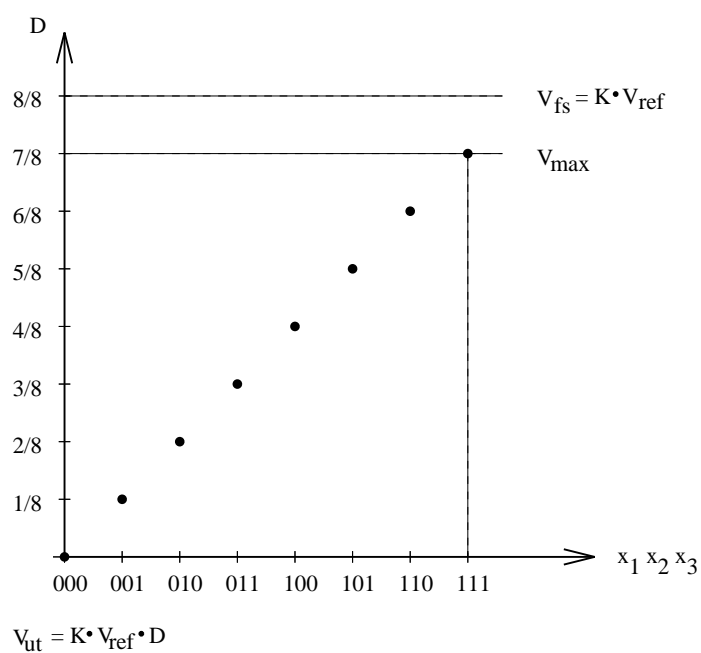
**Exempel, 3-bitars D/A-omvandlare.**

Antag en 3-bitars D/A-omvandlare med referensspänningen 1.0V och  $K = 2$ . Denna får överföringsfunktionen

$$V_{ut} = 2.0 \cdot \left\{ \frac{x_1}{2} + \frac{x_2}{4} + \frac{x_3}{8} \right\}$$

Om vi går igenom samtliga kombinationer av nollor och ettor i det digitala ordet får vi följande resultat (se även figur 43):

digitalt ord	utspänning
0 0 0	0.0 0
0 0 1	0.250 2/8
0 1 0	0.500 4/8
0 1 1	0.750 6/8
1 0 0	1.000 8/8
1 0 1	1.250 10/8
1 1 0	1.500 12/8
1 1 1	1.750 14/8



Figur 43: Överföringsfunktion för en 3-bitars D/A-omvandlare.

Med hjälp av tabellen och överföringsfunktionen kan vi bestämma värdena på Full Scale Voltage, upplösning och största möjliga utspänning:

$$V_{FS} = K \cdot V_{ref} = 2.0V$$

$$V_{LSB} = \frac{V_{FS}}{2^n} = 0.250V$$

$$V_{max} = V_{FS} \cdot \frac{2^n - 1}{2^n} = 1.750V$$

### Exempel, Ökad upplösning.

Observera att den analoga utsignalen i föregående exempel antar vissa bestämda diskreta värden. Spänningarna mellan dessa nivåer kan inte representeras med denna D/A-omvandlare. Vill man minska skillnaderna mellan nivåerna måste man öka upplösningen, dvs öka antalet bitar i det digitala ordet och i D/A-omvandlaren.

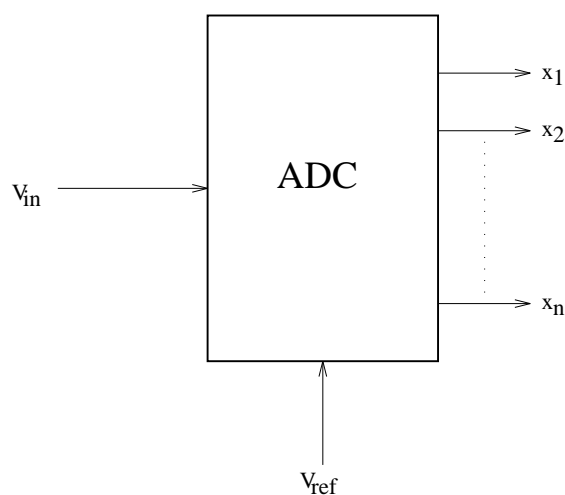
Antag att vi vill kunna stega med 1mV eller mindre. Hur många bitar skulle detta kräva i en D/A-omvandlare med  $V_{FS} = 2V$ ?

$$V_{LSB} = \frac{2}{2^n} < 1mV \Rightarrow 2^n > 2000$$

Närmaste 2-potens som är större än 2000 är  $2^{11} = 2048$ . Alltså behövs det en 11-bitars D/A-omvandlare om vi vill ha en upplösning som är högre än 1mV när  $V_{FS} = 2.0V$ .

**Analog/digital(A/D)-omvandling.**

En A/D-omvandlare (eng. analog-digital converter, ADC) utför motsatsen till vad en D/A-omvandlare utför. Den omvandlar en analog insignal till ett digitalt ord, se figur 44. Detta digitala ord kan sedan användas till att utföra beräkningar, styra processer etc.



Figur 44: Symbol för en A/D-omvandlare.

Överföringsfunktionen ges av

$$D = \frac{V_{in}}{V_{FS}} \quad (7)$$

De definitioner som tidigare givits för upplösning och  $V_{FS}$  gäller även här:

$$V_{LSB} = \frac{V_{FS}}{2^n}, \quad (8)$$

$$V_{FS} = K \cdot V_{ref} \quad (9)$$

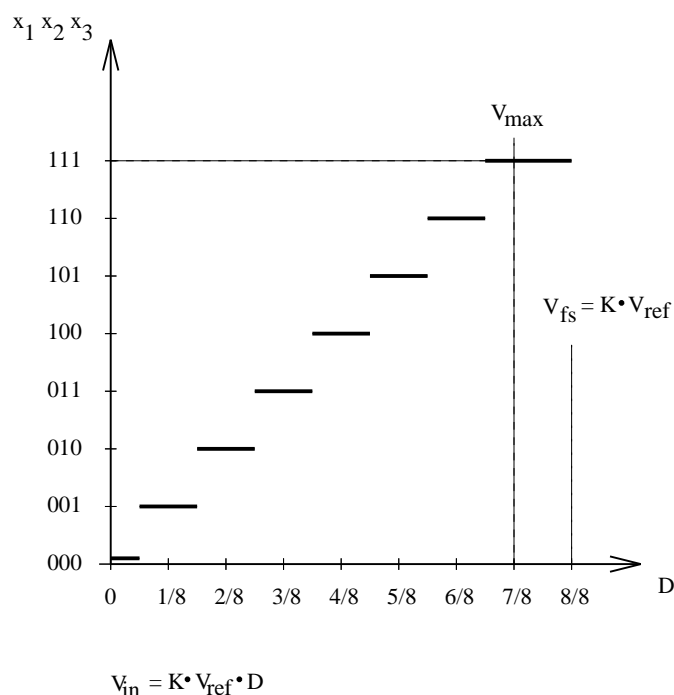
$V_{LSB}$  är den upplösning vi har i omvandlaren och  $D$  är det decimala värdet av det digitala ordet. I och med att vi bara kan representera diskreta värden digitalt, får vi här automatiskt en kvantisering av insignalens nivå. Dessutom får vi en kvantisering i tid, eftersom varje omvandling tar en viss tid att fullfölja och vi därigenom inte kan detektera insignalens värde vid varje tidpunkt.

**Exempel, 3-bitars ADC.**

En 3-bitars ADC med  $V_{FS} = 2V$  får följande överföringsfunktion:

$$D = \frac{V_{in}}{V_{LSB}} = \frac{V_{in}}{\frac{V_{FS}}{2^n}} = \frac{V_{in}}{0.25V}$$

För beräkning av upplösningen se exempel 2.1. Grafen för funktionen visas i figur 45.



Figur 45: Överföringsfunktion för en ideal 3-bitars ADC.

När spänningen ändras slår utsignalen om, när man når halvvägs mellan de kvantiserade nivåerna. Därigenom kommer varje digitalt ord i praktiken att motsvara spänningsnivåer som kan vara  $\pm \frac{1}{2} \cdot V_{LSB}$  det nominella värdet. Detta kallas för *kvantiseringsfelet*.

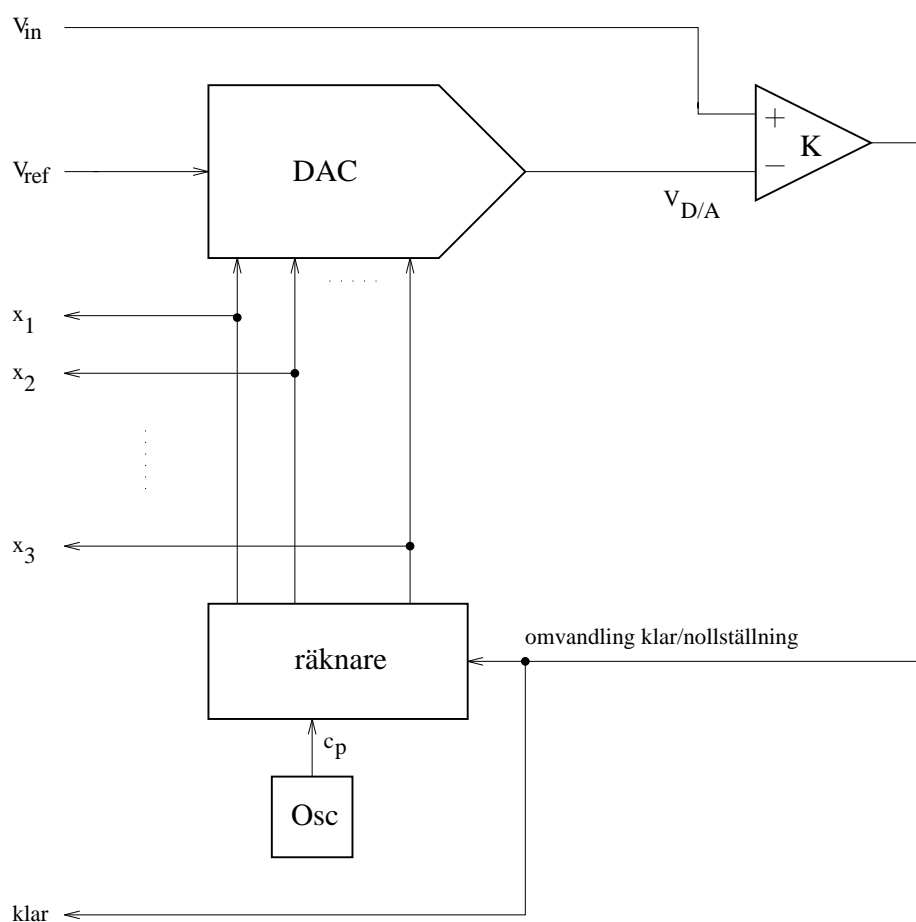
**Sample-and-hold (S&H)**

Som tidigare angivits, tar det en viss tid för omvandlaren att presentera det digitala ordet utifrån det analoga värdet. Under denna tid, omvandlingstiden, bör insignalen inte ändras. Alltså behöver man en koppling som läser av (samlar) det analoga värdet vid en viss tidpunkt, och sedan håller detta värde tills omvandlingen är klar. Hur en sådan S&H-koppling kan utföras finns beskrivet i litteraturen.

## A/D-omvandlare

I det här avsnittet behandlas dels trappstegsomvandlaren, dels den följande omvandlaren.

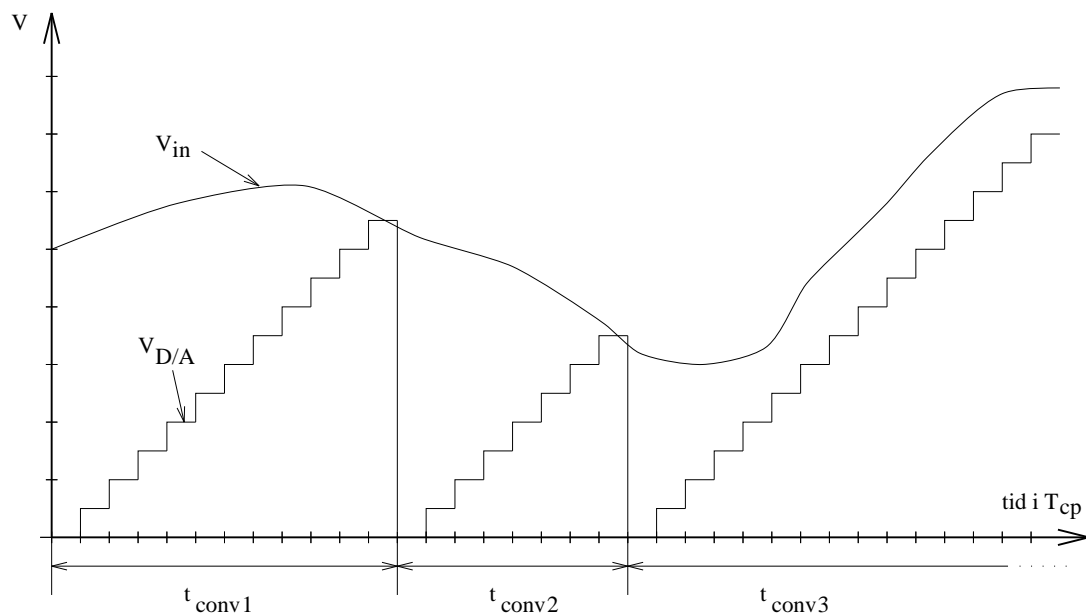
### Trappstegsomvandlaren



Figur 46: Blockschemata för trappstegsomvandlaren.

Trappstegsomvandlaren innehåller en D/A-omvandlare, en digital räknare, en oscillator, en komparator och eventuellt ytterligare logiska kretsar. När räknaren räknar upp bildas en trappstegsformad signal på D/A-omvandlarens utgång. Denna spänning jämförs med insignalen. När D/A-omvandlarens utspänning,  $V_{D/A}$ , blir större än inspänningen, byter komparatorn värde. På räknarens utgångar finns nu ett digitalt värde, som motsvarar den analoga signalen. Med hjälp av klarsignalen kan detta värde lagras i t.ex. något register. Klarsignalen används också till att nollställa räknaren och därefter kan en ny omvandlingscykel påbörjas.

En stor nackdel med den följande omvandlaren är att omvandlingstiden blir olika beroende på hur stor insignalen är, se figur 47.

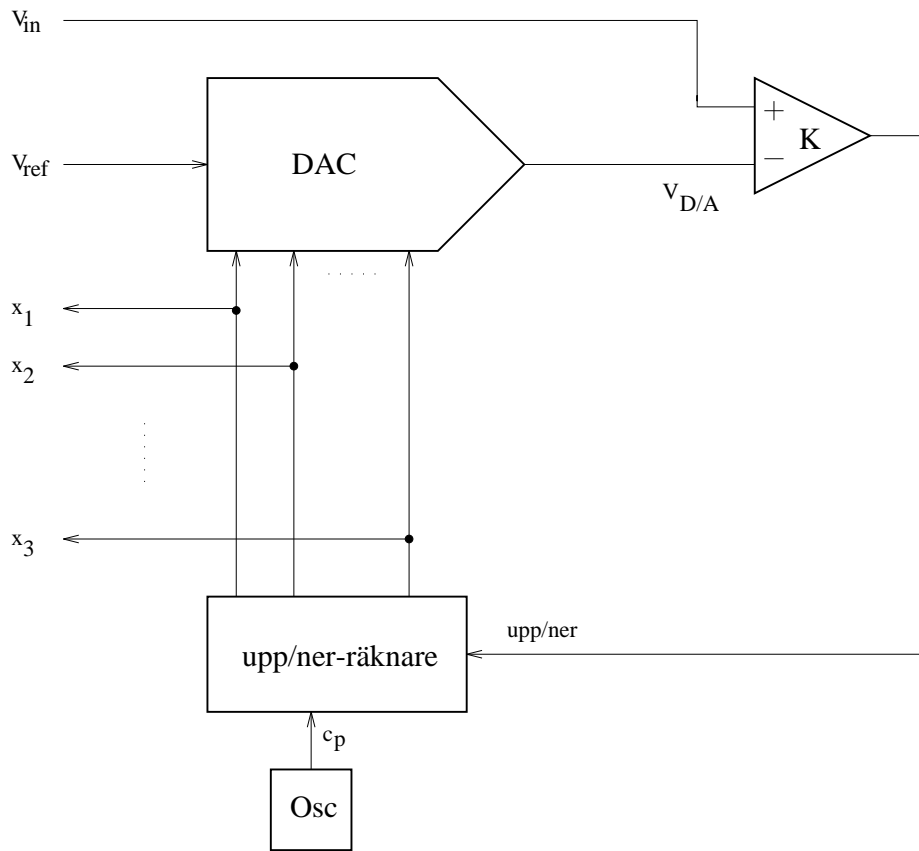


Figur 47: Tidsdiagram för en trappstegsomvandlare ( $t_{conv}$  = conversion time, omvandlingstid).

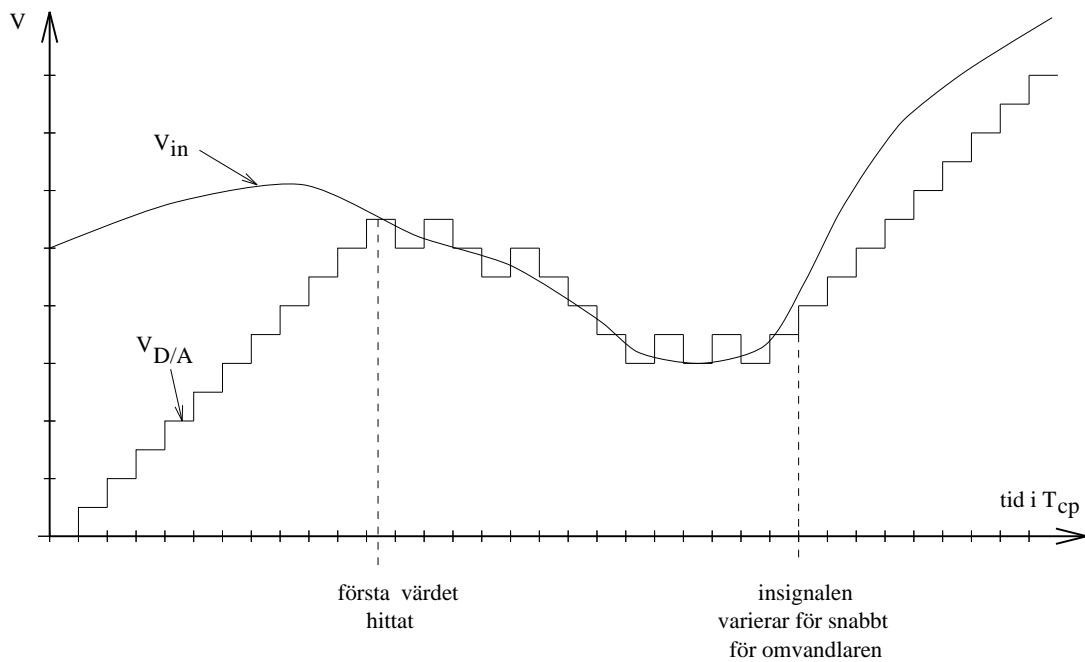
### Den följande omvandlaren

Den följande omvandlaren är till stora delar lik trappstegsomvandlaren. Skillnaden ligger i den digitala räknaren, som här är en upp/ner-räknare. Utsignalen från komparatorn styr huruvida räknaren ska räkna upp eller ner. När insignalens nivå en gång hittats och om den inte varierar alltför kraftigt, kan insignalen följas med små variationer i omvandlingstiden. Denna tid blir även oberoende av om insignalen ligger på en låg eller hög nivå, se tidsdiagram i figur 49!





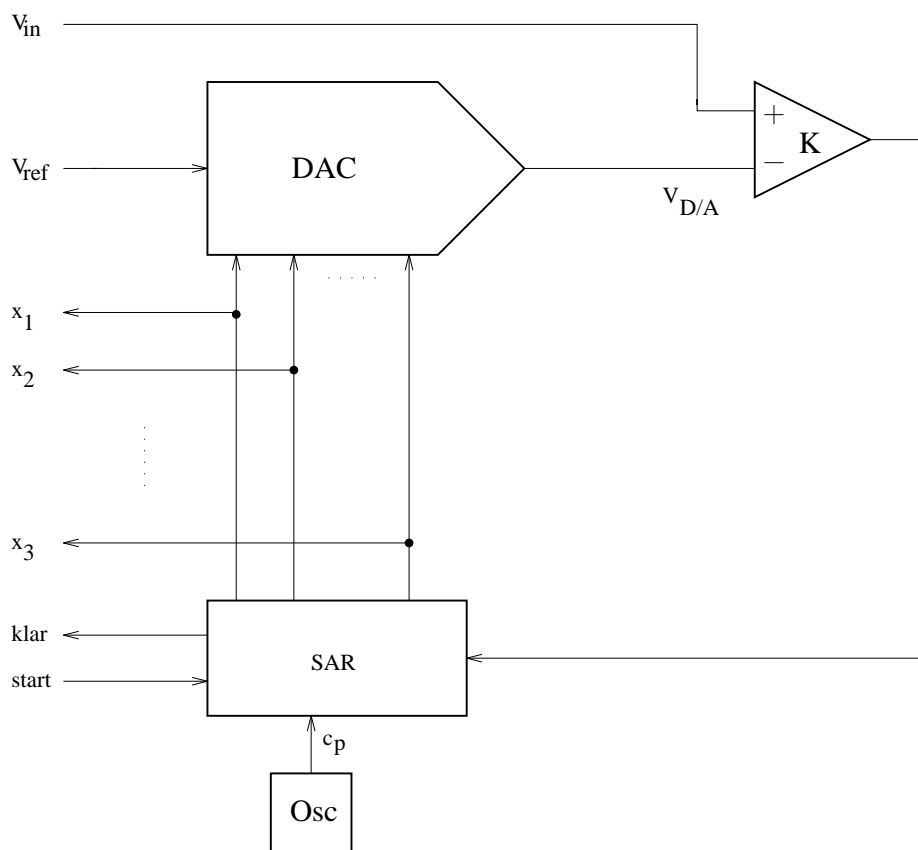
Figur 48: Blockschema för den följande omvandlaren.



Figur 49: Tidsdiagram för en följande omvandlare.

## Successiv approximation

För att slippa nackdelen med varierande omvandlingstider, kan man använda ett **SAR** (successive approximation register) istället för en räknare, se figur 50. Det är mer komplext än en räknare, men i gengäld är omvandlingstiden konstant oberoende av insignalens amplitud.

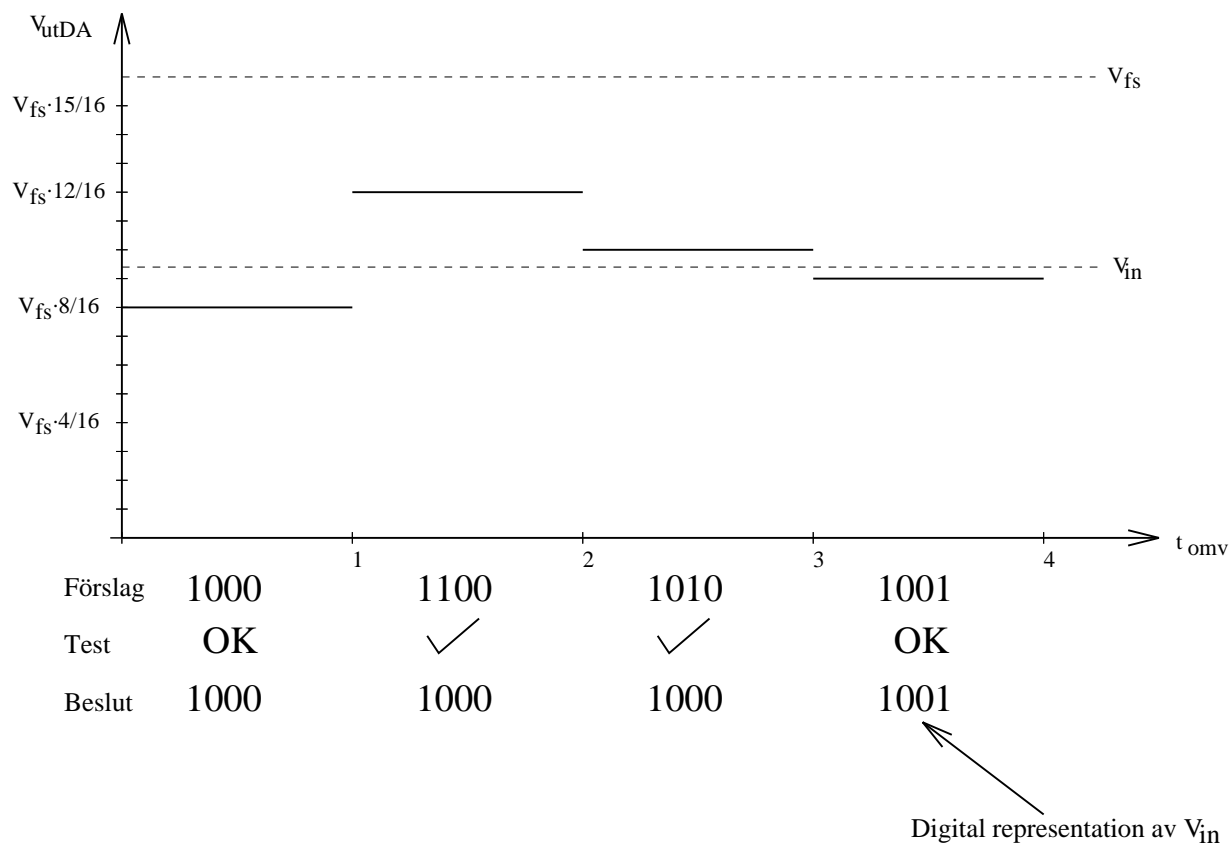


Figur 50: Blockschema för omvandlare med SAR (successive approximation register).

SAR är ett logiskt nät, som genererar ett antal digitala ord som insignal till DA-omvandlaren enligt följande (se figur 51):

1. Samtliga bitar nollställs. Startsignal ges till SAR.
2. Den mest signifikanta biten (MSB) sätts hög. Utsignalen från DA-omvandlaren blir då  $V_{FS}/2$ .
3.  $V_{utDA}$  jämförs med insignalen. Om insignalen är störst, behålls ett högt värde på MSB, annars nollställs denna.
4. Den näst mest signifikanta biten sätts hög.  $V_{utDA}$  blir antingen  $V_{FS} \cdot \frac{3}{4}$  eller  $\frac{V_{FS}}{4}$  beroende på om den mest signifikanta biten var hög eller låg.
5.  $V_{utDA}$  jämförs med insignalen. Om insignalen är störst, behålls ett högt värde på biten, annars nollställs denna.

6. Nästa bit sätts hög etc.
7. När den minst signifikanta biten (LSB) fått sitt värde är omvandlingen klar. Utsignalen finns nu på registrets utgång (DA-omvandlarens ingång). En klarsignal genereras.



Figur 51: Tidsdiagram för omvandling med SAR.

Omvandlingstiden blir alltså proportionell mot antalet bitar i det digitala ord som ska omvandlas. För varje bit i det digitala ordet kan en eller två klockpulser behövas, beroende på hur registret är konstruerat.