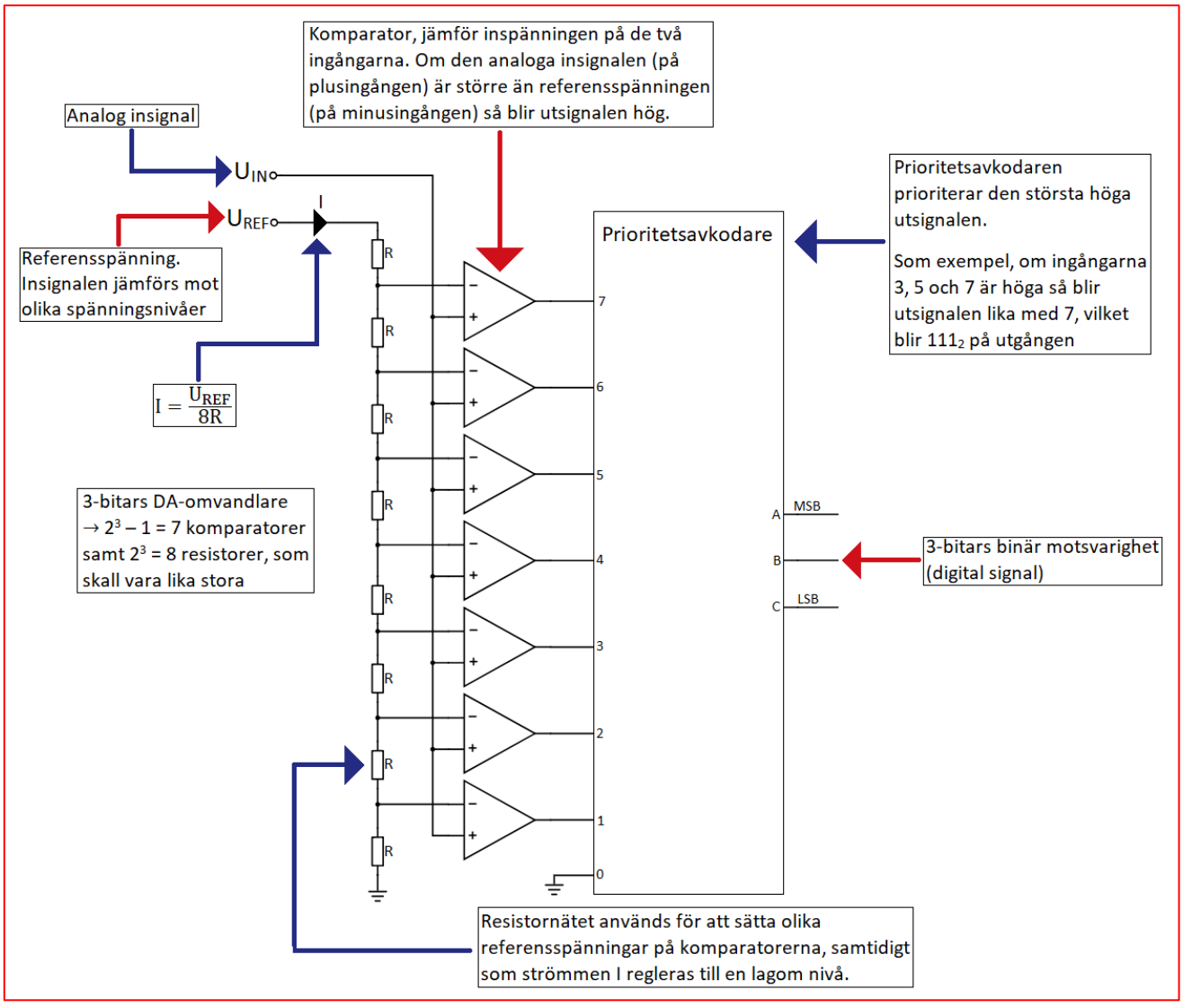
1.5 - Konstruktion av AD-omvandlare

1.5.1 - Introduktion

- AD-omvandlare (analog till digital omvandlare) är integrerade kretsar som omvandlar analoga signaler till motsvarande digitala signaler. Som exempel så kan signaler från en mikrofon AD-omvandlas till motsvarande digitalt värde för att kunna genomgå digital signalbehandling, exempelvis för att filtrera ut brus ur den analoga signalen eller för komprimering

- Som ett mer praktiskt exempel så kan en sinusformad växelspänning, som varierar mellan 0–5 V, omvandlas till digitala värden mellan 0–255 om omvandlingen sker med en 8-bitars omvandlare.

- Flashomvandlaren är en AD-omvandlare med mycket hög precision, som omvandlar analoga spänningar till en digital motsvarighet, som består av ett visst antal bitar. Figuren nedan visar en 3-bitars flashomvandlare, så den analoga signalen omvandlas till ett 3-bitars binärt tal mellan 000_2 – 111_2, vilket motsvarar 0_{10} – 15_{10} i det decimala talsystemet.



3-bitars AD-omvandlare.

- Den analoga insignalen jämförs med olika spänningsnivåer på de olika OP-förstärkarna, som används som komparatorer. Komparatorerna jämför den analoga signalen mot olika spänningsnivåer från spänningsdelaren.

- För en 3-bitars AD-omvandlare så behöver vi $2^3 - 1 = 7$ komparatorer, en för varje ingång på prioritetsavkodaren förutom den nedersta ingången, som kan kopplas direkt till jord, då denna alltid skall vara låg. Om den analoga insignalen är lika med 0 V så kommer därmed samtliga ingångar bli låga och utsignalen blir också låg.

- Komparatorerna är ordnade i en så kallad *ladder* (trappa). Den analoga signalen kommer in på varje komparators plusingång och jämförs sedan med spänningsvärdet på minusingången, som blir lägre ju längre ned i trappan man kommer.

- Om insignalen V_+ på plusingången är större än inspänningen V_- på minusingången så blir komparatorns utsignal U_{UT} hög, alltså ungefär lika med matningsspänningen V_{CC}, som vanligtvis ligger mellan 1–5 V. Komparatorns utsignal medför en hög insignal på en av prioritetsavkodarens ingångar, som används för att indikera en digital motsvarighet till den analoga insignalen.

- Däremot om insignalen V_+ på plusingången är mindre än inspänningen V_- på minusingången så blir komparatorns utsignal U_{UT} låg, alltså 0 V, vilket medför en låg insignal in på en av prioritetsavkodarens ingångar. Ju högre den analoga signalen är, desto fler av prioritetsavkodarens insignaler blir höga, men det är endast insignalen längst upp i trappan som prioriteras.

- Därefter så kommer prioritetsavkodaren, som prioriterar den mest signifikanta insignal som är hög, medan övriga bitar inte har någon betydelse. Därefter så skapas ett 3-bitars digitalt värde utifrån den största insignalen. Som exempel, om insignal 7 är hög, så blir utsignalen lika med 111_2. Även insignaler 0 – 6 är höga, men har därmed ingen påverkan på utsignalen. Däremot om insignal 7 är låg, samtidigt som bit 6 är hög, så blir utsignalen lika med 110_2. I detta fall är också insignaler 0 – 5 höga, vilket dock inte har någon betydelse.

- För att få fram olika spänningsnivåer på komparatorernas minusingångar så används en spänningsdelare samt ett flertal resistorer, se figuren till höger. Dessa resistorer skall vara lika stora.

- Värdet på R bestämmer vilken ström som flödar genom spänningsdelaren. Eftersom vi har åtta resistorer som är lika stora så kan ett lämpligt värde på R beräknas med formeln

$$R = \frac{U_{REF}}{8I},$$

  där U_{REF} är referensspänningen och I är strömmen genom spänningsdelaren. Ett bra riktvärde på strömmen I är ca 1 mA.

- För en referensspänning på 5 V så kan resistor R sättas till 0,68 kΩ, så att strömmen I blir 1 mA:

$$R = \frac{U_{REF}}{8 * I} = \frac{5}{8 * 1m} = 0{,}625\ k\Omega$$

- Närmaste värde i E12-serien är 0,68 kΩ, som vi använder. Därmed blir strömmen något lägre än 1 mA (ca 0,92 mA), men detta går bra; 1 mA är ett bra riktvärde, men absolut precision behövs inte. Ett värde nära 1 mA går bra.

- OP-förstärkarna används som komparatorer, dvs. de jämför två signaler mot varandra. Den inkommande signalen jämförs mot olika spänningsnivåer via en komparator, där insignalen kommer in på komparatorns minusingång och referensvärdet på plusingången. Om insignalens spänning är större än referensspänningen på komparatorns plusingång så blir insignalen till prioritetsavkodaren hög, annars blir den låg. Ju större den analoga inspänningen är, desto fler av prioritetsavkodarens insignaler blir höga.

- Prioritetsavkodaren skapar det binära värdet som den analoga signalen översätts till. Prioritetsavkodaren bryr sig endast om den insignal som är störst, så om exempelvis insignal 1, 4 och 6 är höga och resten låga så kommer den endast bry sig om värdet 6. Därför blir utsignalen ur prioritetsavkodaren 6_{10}, vilket motsvarar det binära talet 110_2, därav namnet prioritetsavkodare. Prioritetsavkodaren, som i detta avsnitt ritas ut dess blockschema, kan realiseras med ett flertal logiska grindar, vilket dock inte visas här.

1.5.2 - Dimensionering av en 3-bitars flashomvandlare

- Vi skall konstruera en 3-bitars flashomvandlare, vars referensspänning är 5 V.

$$U_{REF} = 5\ V$$

- Vi sätter strömmen I genom spänningsdelaren till ca 1 mA.

- Som vi såg tidigare så kan resistansen på varje resistor i spänningsdelaren sättas till 0,68 kΩ för att strömmen I skall hamnar nära 1 mA vid en referensspänning på 5 V, eftersom

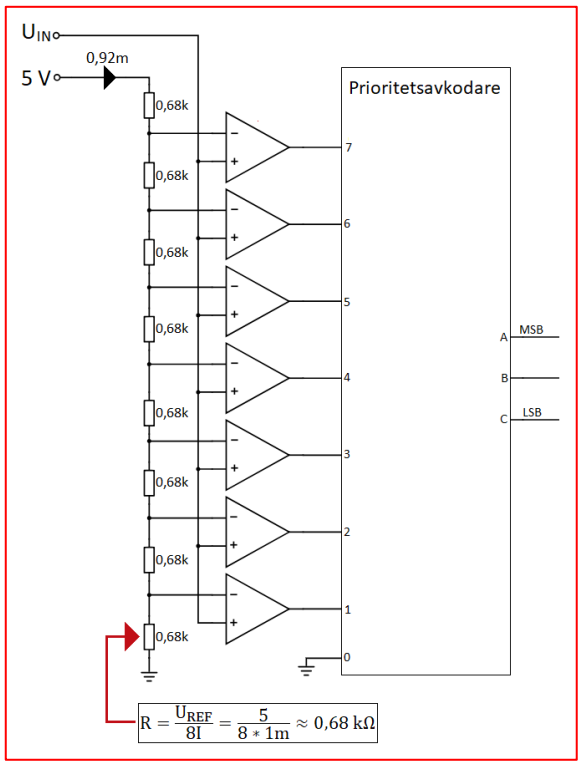
$$R = \frac{U_{REF}}{8 * I} = \frac{5}{8 * 1m} = 0{,}625\ k\Omega$$

- Närmaste värde i E12-serien är 0,68 kΩ, som vi därmed använder:

$$R = 0{,}68\ k\Omega$$

- Därmed blir strömmen I som flödar genom spänningsdelaren något lägre än 1 mA (ca 0,92 mA), men denna skillnad är obetydlig.

- Digitala enheter såsom prioritetsavkodaren konstrueras normalt i ett hårdvarubeskrivande språk, såsom VHDL eller SystemVerilog. Se bilaga A för exempel på konstruktion av en 3-bitars prioritetsavkodare i VHDL.



3-bitars flashomvandlare med dimensionerad spänningsdelare.

1.5.3 - Konstruktion av en 8-bitars flashomvandlare

- För en 8-bitars flashomvandlare så krävs $2^8 - 1 = 255$ komparatorer samt $2^8 = 256$ resistorer. Eftersom denna konstruktion blir otroligt stor så ritas den inte ut här. Spänningsdelaren kommer alltså bestå utav 256 lika stora resistorer. Anta att vi använder referensspänningen 5 V och siktar på att strömmen genom spänningsdelaren ligger omkring 1 mA. Ett lämpligt värde på varje resistor är då

$$R = \frac{U_{REF}}{256 * I} = \frac{5}{256 * 1m} \approx 19{,}5\ \Omega$$

- Närmaste värde i E12-serien är 18 Ω:

$$R = 18\ \Omega$$

- Strömmen I som flödar genom spänningsdelaren blir då ca 1,1 mA, då

$$I = \frac{U_{REF}}{256 * R} = \frac{5}{256 * 18} \approx 1{,}1\ mA,$$

vilket bör vara tillräckligt nära 1 mA. Vi hade också kunnat använda det näst närmaste resistorvärdet i E12-serien, vilket är 22 Ω. Då hade strömmen genom spänningsdelaren blivit något lägre än 1 mA:

$$I = \frac{U_{REF}}{256 * R} = \frac{5}{256 * 22} \approx 0{,}9\ mA,$$

vilket också bör vara tillräckligt nära 1 mA. Det hade inte spelat någon större roll vilket resistorvärde vi använder i detta fall, men 22 Ω:s resistorer kan föredras, för att effektförbrukningen blir något lägre (eftersom strömmen I som flödar genom spänningsdelaren är lägre). Dock är denna skillnad mycket liten i praktiken.

Bilaga A

Konstruktion av en synkron 3-bitars prioritetsavkodare i VHDL

- Nedan demonstreras hur en 3-bitars prioritetsavkodare, såsom beskrevs i avsnitt 1.5.1 - 1.5.2 hade kunnat realiseras i praktiken. Denna prioritetsavkodare är synkron och arbetar därför efter en klocka, där uppdatering av utsignalen sker vid klockpuls från en systemklocka. En asynkron reset-signal används för att omedelbart nollställa utsignalen vid reset.

- Konstruktionen, som i VHDL-koden kallas modul, består av en entitet samt arkitekturen. I entiteten så deklareras konstruktionens in- och utportar, såsom *data_in* samt *data_out.* Entiteten kan därmed ses som konstruktionens skal eller utsida.

- I arkitekturen så beskrivs konstruktionens beteende, alltså hur konstruktionen skall fungera. Arkitekturen kan därmed ses som konstruktionens insida. Här uppdateras utsignal *data_out* vid klockpuls (stigande flank) eller vid reset. Utsignalen utgör i övrigt en funktion av mest signifikanta höga insignal *data_in,* såsom beskrivet i föregående avsnitt.

- Nedanstående modul är döpt till *PriorityEncoder.* I VHDL har modulens entitet samma namn som modulen, medan arkitekturen kan ges valfritt annat namn, såsom *Behaviour* eller RTL *(Register Transfer Level),* vilket indikerar att konstruktionens dataflöde definieras. Här realiseras arkitekturen på en relativt hög abstraktionsnivå, där avkodarens beteende definieras via beskrivningar med if-else satser.

```
--------------------------------------------------------------------------------
-- Modulen PriorityEncoder utgör en 3-bitars synkron prioritetsavkodare, där mest
-- signifikant insignal data_in[7:0] avgör utsignal data_out[3:0] enligt nedan:
--
-- data_in[7:0]   data_out[2:0]
--  1xxxxxxx          111
--  01xxxxxx          110
--  001xxxxx          101
--  0001xxxx          100
--  00001xxx          011
--  000101xx          010
--  0000001x          001
--  0000000x          000

--
-- Nätet är synkront med en asynkron reset-signal, på så sätt att utsignal
-- data_out enbart uppdateras vid klockpuls (stigande flank av insignal clock)
-- eller vid nedtryckning av reset-knappen (när insignal reset_n blir låg).
--
-- Insignaler:
--   data_in
--   clock
--   reset_n
--
-- Utsignaler:
--   data_out
--------------------------------------------------------------------------------
library IEEE;
use IEEE.std_logic_1164.all;

--------------------------------------------------------------------------------
-- I entiteten deklareras modulens portar (in- och utsignaler). Entiteten kan ses
-- som utsidan / skalet på ett digitalt byggblock.
--------------------------------------------------------------------------------
entity PriorityEncoder is
    port
    (
        data_in  : in std_logic_vector(7 downto 0);
        clock    : in std_logic;
        reset_n  : in std_logic;
        data_out : out std_logic_vector(2 downto 0)
    );
end entity;
```

```
-------------------------------------------------------------------------------
-- I arkitekturen så definieras modulens beteende via beskrivning av dess
-- interna uppbyggnad. Arkitekturen kan ses som insidan av ett digitalt byggblock.
-------------------------------------------------------------------------------
architecture Behaviour of PriorityEncoder is
begin

    process (clock, reset_n) is
    begin
        if (reset_n = '0') then
            data_out <= "000";
        elsif (rising_edge(clock)) then
            if (data_in(7) = '1') then
                data_out <= "111";
            elsif (data_in(6) = '1') then
                data_out <= "110";
            elsif (data_in(5) = '1') then
                data_out <= "101";
            elsif (data_in(4) = '1') then
                data_out <= "100";
            elsif (data_in(3) = '1') then
                data_out <= "011";
            elsif (data_in(2) = '1') then
                data_out <= "010";
            elsif (data_in(1) = '1') then
                data_out <= "001";
            else
                data_out <= "000";
            end if;
        end if;
    end process;

end architecture;
```