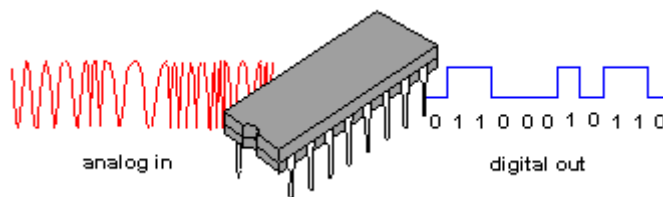


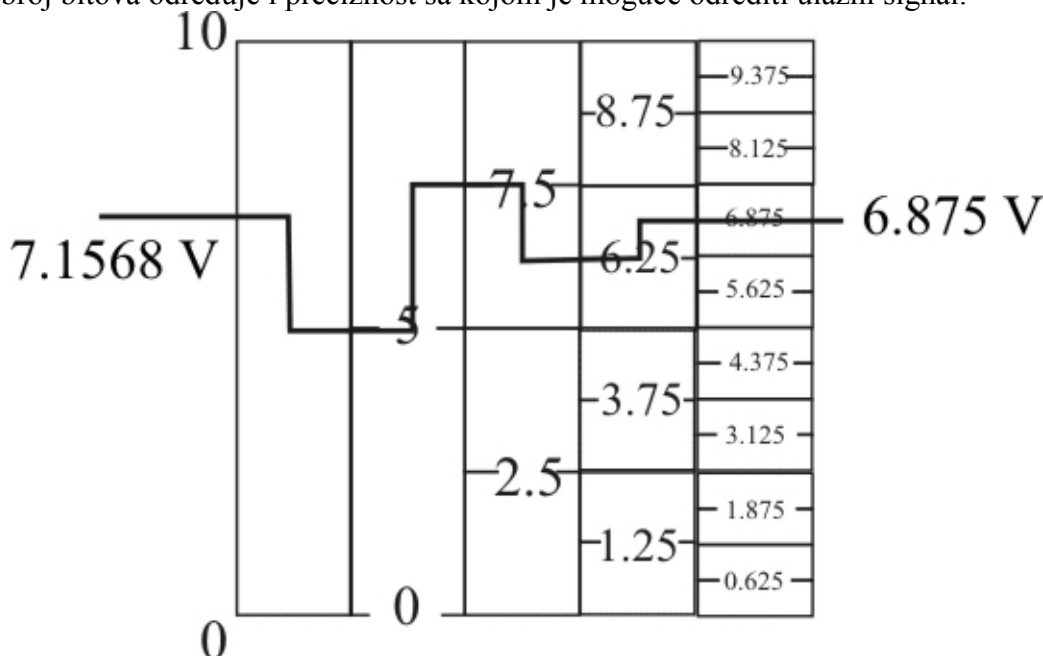
### VEŽBA 5

Naziv vežbe	A/D KONVERZIJA PODATAKA
Trajanje vežbe	30'
Potrebno predznanje	Programiranje u MatLab-u
Broj studenata	1
Cilj vežbe	U ovoj vežbi studenti imaju za cilj da, uz pomoć funkcije u MatLab-u koja oponaša A/D konvertor, uporede originalni signal i digitalizovan signal.

#### TEORIJSKE OSNOVE



Analogni signal u vidu jačine struje ili napona pretvara se u digitalni pomoću uređaja koji se naziva Analogno/Digitalni konvertor, ili skraćeno *A/D* konvertor. *A/D* konvertor radi po principu upoređenja dolaznog signala komparatorima koji u određenom broju koraka prepoznaju interval u kom se nalazi vrednost dolaznog signala. Svaki *A/D* konvertor ima više nivoa komparacije koji se zovu bitovi. Na primer dvadesetčetvorobitni konvertor ima 24 nivoa komparacije i u svakom nivou određuje se približna vrednost signala. Samim tim broj bitova određuje i preciznost sa kojom je moguće odrediti ulazni signal.



Slika 1: Šematski prikaz 4-tvorobitnog *A/D* konvertora

Moderan dizajn  $A/D$  konvertora podrazumeva veliki broj komparatora i direktno upoređivanje analognog signala sa mogućim vrednostima koje su određene sa brojem bitova  $A/D$  konvertora. Na Slici 1 je prikazan šematski prikaz kako ulazni signal napona 7.1568 mV prolazi kroz 4-voro bitni  $A/D$  konvertor. Vidi se da je prikazanim  $A/D$  konvertorom **moгуće odrediti jedino približna vrednost u kojoj se nalazi ulazni napon** i on iznosi 6.875 mV.

#### POSTUPAK IZRADE VEŽBE

Vežba je računska i izvodi se na računaru i to uz pomoć softverskog paketa MatLab. Svaki student treba da dobije jedan mereni signal visoke tačnosti (sa velikim brojem decimala) u vidu vremenskog niza i tip  $A/D$  konvertora koji treba da simulira (bitažu konvertora). Sa dobijenim merenim signalom je potrebno obaviti sledeće:

- a. odrediti ekstremne vrednosti (minimum i maksimum) i srednju vrednost dobijene serije
- b. proceniti periodičnost  $T$  vremenske serije pomoću dijagrama
- c. digitalizovati vremensku seriju pomoću (zaokružiti dve vrednosti):
  - i. 4-bitnog,
  - ii. 6-bitnog,
  - iii. 8-bitnog,
  - iv. 10-bitnog
  - v. 12-bitnog

$A/D$  konvertora (listing dat u nastavku). Nakon toga izdvojiti podatke iz digitalizovane vremenske serije sa različitim vremenskim intervalima i to:

**0.2T                  0.9T                  1.1T                  1.5T                  1.9T**  
(zaokružiti 3 vrednosti)

- d. odrediti ekstremene vrednosti tako dobijene tri vremenske serije, srednju vrednost i periodičnost

Dobijeni merni signal skalirati u granicama  $[0,10]$  mV i ponoviti korake iz tačke 4, od a. do d. Skaliranje signala u intervalu  $[0,10]$  mV podrazumeva linearnu transformaciju podataka tako da minimalna vrednost bude 0 mV, a maksimalna 10 mV.

Priložiti sledeće grafičke priloge:

- Na dijagramima priložiti originalnu (neskaliranu) i digitalizovane vremenske serije (ukupno tri dijagrama)
- Na novim dijagramima priložiti originalnu (skaliranu) i digitalizovane vremenske serije (ukupno tri dijagrama)
- Na svim dijagramima označiti ekstremene vrednosti i periodu

Neskalirana vremenska serija

Broj bitova:			
	interval 1	interval 2	interval 3
Originalni signal			
min			
max			
avg			
T			

Broj bitova:			
	interval 1	interval 2	interval 3
Originalni signal			
min			
max			
avg			
T			

Skalirana vremenska serija

Broj bitova:			
	interval 1	interval 2	interval 3
Originalni signal			
min			
max			
avg			
T			

Broj bitova:			
	interval 1	interval 2	interval 3
Originalni signal			
min			
max			
avg			
T			

MatLab program koji simulira A-D konverziju podataka u opsegu 0-10 mV:

```
function [k, DData]=AD(Data, NoBits)
%AD-funkcija koja simulira rad A/D konverzije signala
%Ulazi:
%Data-podatak koji je potrebno konvertovati
%NoBits-broj bitova u A/D konvertoru
%Izlazi:
%k-broj koraka uporedjivanja
%DData-digitalizovani signal
```

```
leftbound=0;%granice ulaza A/D konvertora
rightbound=10;
```

```
k=0;
for i=1:NoBits
    k=k+1;
    if Data>(rightbound+leftbound)/2
        leftbound=(rightbound+leftbound)/2;
    elseif Data<(rightbound+leftbound)/2
        rightbound=(rightbound+leftbound)/2;
    else
        leftbound=(rightbound+leftbound)/2;
        rightbound=(rightbound+leftbound)/2;
        break
    end
end
```

```
if abs(Data-leftbound)<abs(Data-rightbound)
    DData=leftbound;
else
    DData=rightbound;
end
```