

Elektrotehnièki fakultet  
Univerziteta u Beogradu

Katedra za energetske pretvaraèe i pogone

## **DIPLOMSKI RAD**

**Merenje brzine kombinovanom metodom  
pomoæu inkrementalnog optièkog enkodera**

Rukovodilac izrade rada:  
Doc. dr. Slobodan N. Vukosaviæ

student:  
Jovan S. Petroviæ IE #203/88

Beograd, oktobar 1995.

## SADRŽAJ

1.	UVOD	1
2.	KOMBINOVANA METODA ODREĐIVANJA BRZINE	2
2.1	Uvod	2
2.2	Inkrementalni optički enkoder	2
2.3	Merenje brzine	3
3.	HARDVERSKA OSNOVA	5
3.1	Programabilni Tajmer Intel 8253/8253-5	5
	3.1.1. Osnovne karakteristike	5
	3.1.2. Struktura	5
	3.1.3. Opis rada	6
3.2	8255 Programabilni Paralelni Interfejs	9
3.3	D/A Konvertor 1408 L8	12
4.	OBRADA SIGNALA DOBIJENIH SA ENKODERA	14
4.1	Prototipska kartica	14
4.2	Kolo za obradu signala od enkodera do tajmera	15
4.3	Primena tajmera za merenje brzine kombinovanom metodom	17
5.	PROGRAMSKA OSNOVA	24
6.	EKSPERIMENTALNI REZULTATI	33
7.	ZAKLJUČAK	38

LITERATURA

# 1.UVOD

Rad u oblasti razvoja elektronski kontrolisanih elektromotornih pogona iniciran je potrebom da se automatizuju proizvodni procesi u industriji, poveæa pouzdanost, smanji potreba za održavanjem i minimiziraju gubici elektriène energije. U pogonima promenljive brzine i pogonima sa servo zahtevima, uglavnom je korišæen jednosmerni (DC) motor, zbog vrlo povoljnih regulacionih karakteristika: raspregnuto upravljanje momentom i fluksom omoguæeno je ortogonalnim položajem pobudnog fluksa i magneto pobudne sile indukta i direktnom kontrolom nad strujama u rotoru i statoru, što obezbeðuje mehanièki komutator. Jedna od slabosti DC motora je upravo komutator koji zahteva redovno održavanje i smanjuje preopteretljivost motora, ograniævavajuæi maksimalnu dozvoljenu struju armature. Za razliku od DC motora, asinhroni motor je znatno složeniji za upravljanje, jer rotorske struje i fluks ne mogu direktno da se mere. Kod njega se proces komutacije obavlja van motora, u energetskom pretvaraèu. Motor je konstruktivno jednostavniji, robustniji i ne zahteva održavanje. Kratkotrajna preopteretljivost asinhronog motora odreðena je karakteristikama pretvaraèa i može da dostigne višestruku vrednost nominalnog momenta. Prednost pogona sa asinhronim motorom je i niža cena u odnosu na DC pogon. Veliki deo cene pogona sa asinhronom motorom leži u upravljaèkim kolima i energetskom pretvaraèu. Padom cene poluprovodnièkih elemenata i mikroprocesora, kao i razvojem algoritama upravljanja, dolazi do sve veæe upotrebe asinhronih motora u pogonima visokih performansi koji zahtevaju preciznu informaciju o brzini.

Potreba za bržom i taènijom informacijom o brzini i njenom kasnijom upotrebom u kontroli brzine motora, zahteva upotrebu mikroprocesora. Mikroprocesori koriste digitalne vrednosti, pa je informaciju o brzini potrebno dati u digitalnom obliku. Digitalna informacija brzine može da se dobije primenom inkrementalnog enkodera. Najèešæe korišæeni senzor za merenje brzine i pozicije, koji se ugraðuje na osovinu motora, impulsnog tipa je inkrementalni enkoder. Cilj ovog diplomskog rada jeste da prikaže jedan od naèina merenja brzine kombinovanom metodom i pokaže njene prednosti u odnosu na ostale metode merenja brzine.

U drugom poglavlju razmatrane su teorijske osnove kombinovane metode, njeno poreðenje sa metodama merenja brzine (metodom merenja širine impulsa i metodom brojanja impulsa u odreðenom vremenskom intervalu) i objašnjen naèin rada inkrementalnog enkodera.

U treæem poglavlju detaljno je opisan naèin rada programabilnog tajmera 8253, paralelnog interfejsa 8255 i D/A konvertora 1408 L8.

Kolo za obradu signala dobijenih sa enkodera, šema prototipske kartice i opis njenog rada, konkretna primena tajmera u merenju brzine, kao i listing dela programa kojim se inicijalizuje tajmer i oèitavaju njegovi brojaèi, dat je u èetvrtom poglavlju.

U petom poglavlju nabrojani su programski paketi upotrebljeni za izradu ovog rada, objašnjen je postupak prevoðenja i povezivanja asemblerskog programa, pokazano korišæenje programa za otkrivanje grešaka u izvršnim programima i dat kompletan listing programa kojim se ostvaruje merenje brzine kombinovanom metodom.

Eksperimentalni rezultati u obliku osciloskopskih snimaka prikazani su u šestom poglavlju. Talasnim oblicima napona, izmerenim na izlazu iz D/A konvertora, predstavljene su izraèunate brzine dobijene razlièitim metodama i izvršeno je njihovo poreðenje.

## 2. KOMBINOVANA METODA ODREĐIVANJA BRZINE

### 2.1 Uvod

U mikroprocesorski kontrolisanim elektromotornim pogonima potrebno je raspolagati digitalnom vrednošću brzine i pozicije, radi uspostavljanja povratne sprege i pokazivanja na digitalnom displeju. Kod vektorski kontrolisanog asinhronog motora, upravljanje fluksom i momentom ostvaruje se kontrolom amplitude vektora statorske struje i njegove orijentacije, u odnosu na vektor rotorskog fluksa. Model rotorskog kola, pri određivanju orijentacije rotorskog fluksa, oslanja se na podatak o ugaonoj brzini  $\omega_r$  ili poziciji  $\theta_r$  rotora, pa pored uticaja na tačnost u regulaciji brzine ili pozicije, tačnost merenja utiče i na regulaciju fluksa i momenta.

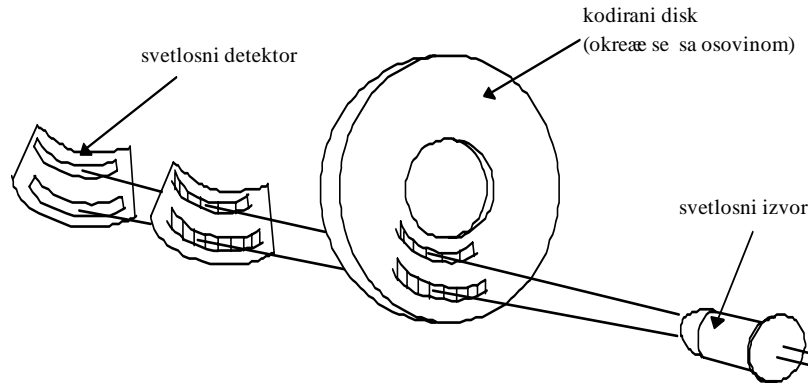
Podatak o brzini može da se dobije konverzijom analognog signala brzine, dobijenog pomoću klasičnog tahometra. Tahometar na izlaznim priključcima generiše analogni signal proporcionalan brzini. Određivanje pozicije zahtevalo bi integraciju ovog signala, što klasični tahometar čini praktično neprimenljivim za precizna poziciona merenja.

U vektorski kontrolisanim pogonima, kao pozicioni senzor, obično se koristi impulsni davač (optički enkoder) ili rezolver. Optički davači generišu digitalne signale, koji se uvode u digitalni kontroler bez dodatnih kola za prilagođavanje. Optički enkoder nije pogodan za primenu u prljavim sredinama. Rezolveri su robustniji od optičkih davača i podnose više temperature, ali im je i primena složenija. U osnovnom izvođenju rezolver generiše dva analogna signala, od kojih je jedan sinusna, a drugi kosinusna funkcija pozicije. Potrebno je da se izvrši konverzija ovih signala u digitalne, a zatim da se pomoću pozicionog dekodera izrađuna pozicija.

Laboratorijski model pogona koristi rešenje pozicionog senzora koje kombinuje dobre osobine enkoderskog i rezolverskog pristupa određivanja pozicije. Za ostvarivanje ovog rešenja izvedena su usloznavanja kola za obradu signala.

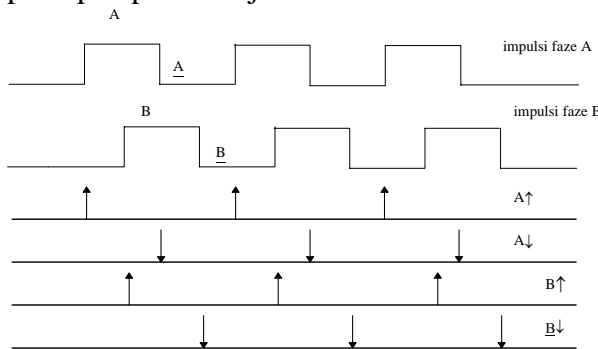
### 2.2 Inkrementalni optički enkoder

Obrtni enkoder predstavlja poseban tip elektromehaničkog pretvarača ugaonih pomeraja u povorku impulsa, koje obrazuju digitalne izlaze enkodera. Optički enkoder se sastoji od tri glavna elementa: diska, svetlosnih izvora i svetlosnih detektora. Konstruiše se u vidu prozračnog plastičnog diska, sa prstenom na periferiji, na kome su gusto naneti neprozračni ekvidistantni markeri. Čitač markera jeste svetlosni detektor. Tokom obrtanja diska čitač markera generiše povorku impulsa. U ekvidistantnim trenucima vremena (trenucima odabiranja) mikroračunar zaustavlja brojanje, očitava sadržaj brojača i ponovo omogućava brojanje.



Slika 2.1.: Princip rada optičkog enkodera

Enkoder svoju poziciju i brzinu saopštava preko impulsa faze A i faze B koji su fazno pomereni za  $90^\circ$ . Ovaj fazni pomeraj omogućava određivanje smera obrtanja, a detekcijom uzlaznih i silaznih ivica faznih impulsa i kombinacijom ovako dobijenih impulsa sa faznim impulsima, rezolucija enkodera može da se uveća četiri puta. Ovaj postupak prikazan je na slici:



$$\begin{aligned} \text{UP} &= AB\uparrow + BA\downarrow + \bar{A}B\downarrow + \bar{B}A\uparrow \\ \text{DOWN} &= AB\downarrow + \bar{B}A\downarrow + \bar{A}B\uparrow + BA\uparrow \end{aligned}$$

Zavisno od smera obrtanja generišu se UP i DOWN impulsi, i to sa četiri puta većom učestanošću od faze.

### 2.3 Merenje brzine

Postoji nekoliko načina pomoću kojih se signali, koji dolaze sa inkrementalnog enkodera, upotrebljavaju za merenje brzine. Brzina može da se odredi brojanjem impulsa proteklih u izvesnom vremenskom intervalu, merenjem širine faznog impulsa, odnosno merenjem vremena koje protekne između dva sukcesivna impulsa, ili kombinovanom metodom koja predstavlja kombinaciju dobrih osobina obe metode.

Najrasprostranjeniji način merenja brzine predstavlja merenje broja impulsa  $M$  u vremenskom intervalu  $T_s$ . Ako je broj impulsa enkodera po jednom obrtaju  $N$  brzina može da se izračuna po formuli:

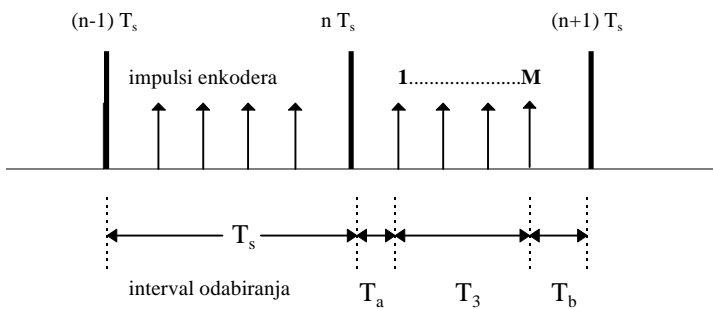
$$\eta[\%_{\min}] = \frac{60}{N \cdot T_s} \cdot M \quad (2.1)$$

Određivanje brzine merenjem vremena  $T_i$  proteklog između dva uzastopna fazna impulsa izračunava se prema:

$$\eta[\%_{\min}] = \frac{60}{N \cdot T_i} \quad (2.2)$$

Nijedna od ovih metoda ne omogućava merenje brzine u širokom opsegu. Merenje broja impulsa daje dobre rezultate pri velikim, a merenje širine impulsa pri malim brzinama.

Preciznije merenje može da se postigne kombinacijom dobrih osobina obe metode. Potrebno je izmeriti broj impulsa enkodera u intervalu odabiranja ( $M$ ) i vreme proteklo od pojave prvog do pojave poslednjeg impulsa u intervalu odabiranja ( $T_3$ ).



Na ovaj način moguće je odrediti srednju širinu impulsa sa znatno većom tačnošću, nego u slučaju kada se meri širina samo jednog impulsa. Ovaj način merenja ilustrovan je slikom i jednačinom:

$$\frac{T_3}{M-1} = \frac{T_s - T_a - T_b}{M-1} \quad (2.3)$$

što predstavlja srednju širinu impulsa. Sa stanovišta upravljanja, izmerena brzina nije trenutna, već predstavlja srednju brzinu u periodu odabiranja. Određivanje brzine kombinovanom metodom izračunava se prema:

$$n_{[\%_{\min}]} = \frac{60 \cdot (M-1)}{N \cdot T_3} \quad (2.4)$$

### 3. HARDVERSKA OSNOVA

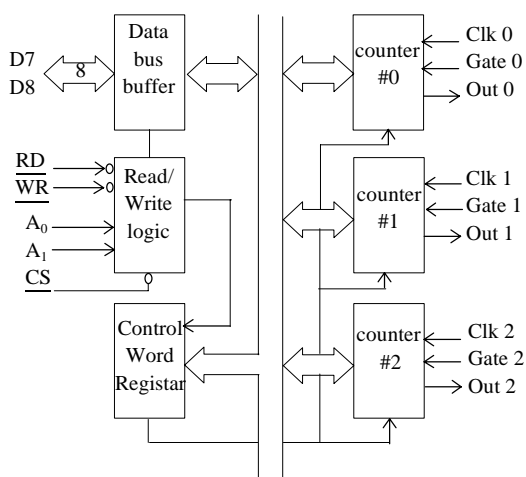
#### 3.1. Programabilni Tajmer Intel 8253/8253-5

##### 3.1.1. Osnovne karakteristike

Programabilni tajmer Intel 8253/8253-5 smešten u 24-pinski ÷ip, sastoji se od tri nezavisna šesnaestobitna brojaèa, uèestanost takta mu je do 2 MHz, napaja se sa +5V, ima mogućnost brojanja binarnih i BCD brojeva kao i mogućnost programiranja modova brojaèa.

Programabilni tajmer Intel 8253 rešava jedan od najteših problema u bilo kom mikrokompjuterskom sistemu, generiše podjednaka vremenska kašnjenja pod softverskom kontrolom. Umesto da sistemskim softverom podešava vremenske petlje, programer može da konfigurira 8253 tako da odgovara njegovim zahtevima, inicijalizuje neki od brojaèa programabilnog tajmera sa željenom vrednošću, koji kada izvrši zadatak, uputi prekid CPU-u. Ostale funkcije, koje po prirodi ne predstavljaju kašnjenje, ali su uobičajene u većini mikrokompjutera, a mogu biti predstavljene posredstvom 8253 jesu: programabilni generator takta, brojaè događaja, sat realnog vremena, delitelj uèestanosti.

##### 3.1.2. Struktura



Slika 3.1. Blok šema rada Tajmera 8253

Data Bus Buffer jeste 8-bitni bafer koji poseduje tri stanja (ulaz, izlaz i stanje visoke impedanse), bidirekcion je (podaci se prenose u oba smera) i koristi se kao interfejs između 8253 i sistemske magistrale podataka. Podaci se prenose kroz bafer u oba smera u zavisnosti od ulazno/izlaznih instrukcija procesora. Data Bus Buffer povezuje se sa magistralom podataka i ostvaruje tri osnovne funkcije:

1. programiranje modova 8253 tajmera,
2. punjenje brojaèkih registara i
3. èitanje izbrojanih vrednosti.

Logika za èitanje i pisanje (Read/Write) prihvata ulaze sa sistemske magistrale i pretvara ih u signale za kontrolisanje opštih operacija. Pristup tajmeru omogućava se, ili

onemogućava preko signala  $\overline{CS}$  (chip select), tako da ni èitanje ni pisanje ne može da se dogodi ukoliko uređaj prethodno nije bio izabran. Promena signala  $\overline{CS}$  nema nikakvog uticaja na već inicijalizovane brojaèe. Signal  $\overline{RD}$  (Read) obavestava Tajmer da procesor "èita" podatke, a  $\overline{WR}$  (Write) da procesor upisuje podatke. Signali  $A_0$  i  $A_1$  obièno su

povezani sa adresnom magistralom i njihova je funkcija da odrede koji æe od tri brojaæa da bude izabran i koja æe vrednost da se upiše u kontrolni registar.

Kontrolni registar (Control Word Register) služi za izbor brojaæa, kontrolu moda, izbor binarne, ili BCD aritmetike i naèin upisivanja ili èitanja iz brojaæa. U kontrolni registar vrednosti mogu samo da se upisuju. Vrednosti se upisuju u registar kada su  $A_0$  i  $A_1$  postavljeni na jedinice.

Brojaèi #0, #1, #2 identièni su. Svaki brojaè sastoji se od šesnaestobitnog *DOWN* brojaæa koji, pri svakom signalu na *clock* ulazu, smanjuje sadržaj brojaæa za jedan. Brojaèi posebno mogu da rade u binarnoj ili BCD aritmetici, a funkcije njihovih ulaza (*input*), izlaza (*output*) i gejtja (*gate*) određene su izborom moda koji je definisan u kontrolnom registru. Rad ovih brojaæa, izbor njihovih modova i aritmetike potpuno je nezavisan. Kontrolni registar, takođe kontroliše naèin punjenja brojaæa. Èitanje sadržaja brojaæa omogućeno je sa *READ* operacijom, a postoji i mogućnost èitanja bez prekidanja brojanja.

**Tabela 3.1.** Naèin adresiranja Tajmera 8253

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	$A_1$	$A_0$	
0	1	0	0	0	Puni brojaè #0
0	1	0	0	1	Puni brojaè #1
0	1	0	1	0	Puni brojaè #2
0	1	0	1	1	Upisivanje kontrolne reèi
0	0	1	0	0	Èita sadržaj brojaæa #0
0	0	1	0	1	Èita sadržaj brojaæa #1
0	0	1	1	0	Èita sadržaj brojaæa #2
0	0	1	1	1	Bez operacija
1	X	X	X	X	Pristup onemogućen
0	1	1	X	X	Bez operacija

### 3.1.3. Opis rada

Na poèetku rada, potrebno je da procesor u kontrolni registar upiše kontrolnu reè, za svaki brojaè posebno, kojom je određen naèin rada svakog od brojaæa i da inicijalizuje stanje brojaæa. Jednom programiran, brojaè nastavlja da izvršava zadatu funkciju sve do ponovnog upisa kontrolne reèi.

Modovi rada postavljaju se jednostavnim ulazno/izlaznim operacijama. Svaki brojaè se nezavisno programira upisivanjem reèi u kontrolni registar.

- Format kontrolne reèi:

**Tabela 3.2.** Format kontrolne reèi (nalazi se na adresi 030F Hex)

$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
SC1	SC2	RL1	RL0	M2	M1	M0	BCD

- Izbor brojaæa određuje se bitovima SC1 i SC0 (*Select Counter*):



**Tabela 3.3.** Izbor brojaèa

SC1	SC0	
0	0	Brojaè #0
0	1	Brojaè #1
1	0	Brojaè #2
1	1	Nedozvoljeno

Naèin upisivanja i èitanja stanja brojaèa, odreðuje se bitovima RL1 i RL0 (*Read/Load*):

**Tabela 3.4.** Naèin upisivanja ili èitanja

RL1	RL0	
0	0	Zadržavanje sadržaja brojaèa
0	1	Èitanje/Upisivanje samo višeg bajta
1	0	Èitanje/Upisivanje samo nižeg bajta
1	1	Èitanje/Upisivanje prvo nižeg pa višeg bajta

- Izbor moda određen je bitovima M2, M1 i M0 (*Mode*):

**Tabela 3.5.** Izbor moda

M2	M1	M0	
0	0	0	Mod 0
0	0	1	Mod 1
X	1	0	Mod 2
X	1	1	Mod 3
1	0	0	Mod 4
1	0	1	Mod 5

Izbor aritmetike određen je bitom BCD (*Binary Coded Decimal*):

**Tabela 3.6.** Izbor aritmetike

BCD	
0	16-bitna binarna aritmetika
1	BCD aritmetika

### Definicije modova:

**MOD 0:** Prekid po završenom brojanju (*Interrupt on Terminal Count*). Izlaz (*Out*) biæe nula za vreme postavljanja (setovanja) moda. Pošto je broj upisan u određen brojaèki registar, izlaz ostaje nula i poèinje brojanje. Kada je brojaè odbrojao upisanu vrednost, izlaz postaje jedinica i ostae na toj vrednosti sve dok se brojaèki registar ponovo ne napuni novim brojem, ili dok se ne promeni mod. Ako se, tokom brojanja, pojavi zahtev za promenom sadržaja brojaèa, pri upisu prvog bajta novog broja zaustaviæe se tekuæe brojanje, a upisivanjem drugog bajta poèeæe novi ciklus odbrojavanja. Silazna ivica, ili nula, na ulazu u gejt, onemoguæavaju brojanje dok ga jedinica omoguæava.

**MOD 1:** Programirani okidaè (*Programmable One-Shot*). Kada se na ulaz gejta dovede uzlazna ivica, izlaz æe da postane nula. Po odbrojavanju broja upisanog u brojaè, izlaz postaje jedinica. Ako je novi broj upisan dok je izlaz nula, brojanje se ne prekida sve do

dolaska uzlazne ivice signala na gejt. Posle svakog dolaska uzlazne ivice signala na gejt, izlaz æe postati nula i ostaæe takav, sve dok brojaè ne odbroji upisani broj.

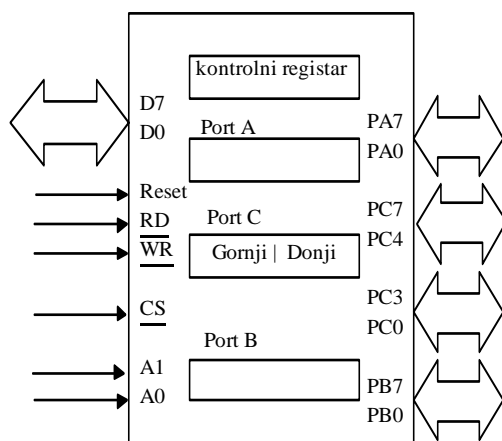
MOD 2: Generator uèestanosti (Rate Generator). Deli uèestanost brojem upisanim u brojaè. Uèestanost predstavljaju signali na ulazu u *clock*. Izlaz æe da bude nula samo u jednom, poslednjem periodu ulaznog kloka. Period između dve promene izlazne velièine jednak je broju upisanom u registar. Ukoliko se brojaèki registar u toku brojanja ponovo napuni brojem koji je razlièit od prethodno upisanog, tekuæe brojanje neæe da se prekine, ali to æe da se odrazi na naredni period. Dok god je signal na ulazu u gejt postavljen na nulu, brojanje æe biti zaustavljeno i izlaz æe da bude postavljen na jedan. Svaki put, kada na gejt doðe uzlazni signal, brojanje poèinje od broja upisanog u registar. Ulaz gejta moæe da se koristi za sinhronizaciju brojaèa. Kada je ovaj mod programiran izlaz æe biti jedan sve dok se brojaèki registar ne napuni.

MOD 3: Generator èetvrtki (Square Wave Rate Generator). Ovaj mod slièan je modu 2, samo što, za razliku od njega, izlaz ostaje jedan, sve dok se broj upisan u registar ne smanji na pola (ovo vaæi za parne brojeve), dok za vreme druge polovine izlaz postaje nula. Ako je upisani broj neparan, izlaz æe biti jedan za vreme dok brojaè odbroji  $(N+1)/2$ , a nula dok odbroji ostalih  $(N-1)/2$ .

MOD 4: (Software Triggered Strobe). Kada se postavi ovaj mod izlaz postaje jedan. Odmah po upisivanju broja, brojaè poèinje da odbrojava. Po završenom brojanju izlaz æe da padne na nulu za jedan period klok ulaza, i onda æe ponovo da postane jedan. Ako se između dve promene izlaznih signala registar napuni, tekuæi period neæe da bude ugroæen, ali ova promena æe da se odrazi veæ na sledeæi period. Brojanje æe da bude onemoguæeno sve dok je ulazni signal na gejtu nula. Posle punjenja brojaèkog registra novim brojem, poèeæe brojanje od tog, upisanog broja.

MOD 5: (Hardware Triggered Strobe). Brojaè poèinje da odbrojava odmah po dolasku uzlazne ivice signala na gejtu i postaeæe nula za jedan period kloka, èim odbroji broj koji je upisan u registar. Izlaz neæe da postane nula dok god brojaè ne odbroji zadati broj.

### 3.2. 8255 Programabilni Paralelni Interfejs



Slika 3.2. Blok šema Programabilnog paralelnog interfejsa

Dijagram 8255 prikazan je na slici. Ovaj interfejs ima kontrolni registar i tri *data buffer* registra. Ova tri *data* registra nisu razdvojena na *data in* i *data out*. Svaki od njih moæe imati obe funkcije. Za ovakva tri registra vezane su 24 linije koje mogu da budu ulazi i izlazi. Mora da postoje zasebne adrese za PORT A, PORT B i PORT C. Zbog njihovog karaktera, svaki od njih moæe da ima obe funkcije, mora da postoje dve adresne linije za odreðivanje sa kojim registrima se komunicira. To odreðuju A0 i A1. Da bi uopšte komunicirali sa portovima (*read* ili *write*) signal *chip select* mora da

bude aktivan.

Portovi su fizički razdvojeni ali se vrši njihovo logičko grupisanje u dve grupe (A i B). Grupišu se: port A i gornja polovina porta C, i port B i donja polovina porta C. Ovakvo logičko grupisanje izvršeno je da bi se deo porta C iskoristio za programiranje logike rukovanja. Ako nema rukovanja, deo porta C služi kao običan ulazno izlazni četvorobitni registar.

**Tabela 3.7.** Način adresiranja Programabilnog paralelnog interfejsa

A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	0	1	0	sa porta A na <i>data bus</i>
0	1	0	1	0	sa porta B na <i>data bus</i>
1	0	0	1	0	sa porta C na <i>data bus</i>
0	0	1	0	0	sa <i>data bus</i> na port A
0	1	1	0	0	sa <i>data bus</i> na port B
1	0	1	0	0	sa <i>data bus</i> na port C
1	1	1	0	0	upis u kontrolni registar ako je D7=1; ako je D7=0 onda je reset
X	X	X	X	1	D7-D0 su u stanju visoke impedanse
1	1	0	1	0	nedozvoljena kombinacija
X	X	1	1	0	D7-D0 su u stanju visoke impedanse

U tabeli je dat prikaz adresiranja 8255A. Bitovi A1, A0,  $\overline{RD}$ ,  $\overline{WR}$ ,  $\overline{CS}$  određuju smer prenosa. Kada je  $\overline{RD}$  jednako nuli, odnosno kada je  $\overline{RD}$  aktivan, radi se o čitanju sa odgovarajućeg porta, tj. podaci idu sa porta na magistralu podataka. Kada je  $\overline{WR}$  jednako nuli, odnosno kada je  $\overline{WR}$  aktivan, radi se o pisanju na odgovarajući port, tj. podaci idu od magistrale podataka na port. Sedma kombinacija bitova govori da se radi o upisu u kontrolni registar (ovo važi ako je D7 jednako jedan. Ako je jednako nuli upis sa magistrale podataka tretira se kao reset instrukcija). U osmoj kombinaciji  $\overline{CS}$  je jedinica, tj.  $\overline{CS}$  je neaktivan. Tada ne može da se vrši nikakva komunikacija sa registrima interfejsa: D7-D0 idu u stanje visoke impedanse.

Za svaku od grupa u kontrolnom registru definišemo određenu grupu bita koji govori šta će da radi grupa A a šta grupa B. Na ovaj način se programiraju portovi odnosno određuje se šta će koji da radi.

Format kontrolnog registra 8255A dat je na slici:

**Tabela 3.8.** Format kontrolnog registra (nalazi se na adresi 0303 Hex)

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

Sadržajem ovog registra definiše se način rada (*mod*) grupa A i B. Kada upisujemo sadržaj u kontrolni registar, bit D7 mora da bude jedan. Kada je nula, to predstavlja reset interfejsa. Za definisanje načina rada grupe A koristimo četiri bita, a za grupu B tri. Bitovi D6 i D5 govore o modu rada grupe A (00-mod 0, 01-mod 1 i 1X - mod 2). Bitovi D4 i D3 pripadaju grupi A, ali zavise od moda (D4 se odnosi na port A a D3 na gornji deo porta C i 1 znači ulaz a 0 izlaz). Bit D2 određuje mod rada grupe B (0-mod 0 a 1-mod 1), a bitovi D1 i D0 zavise od definisanog moda rada grupe B (D1 se odnosi na port B, a D0 na donji deo porta C i 1 znači ulaz, a 0 izlaz).

Postoje tri različita moda rada:

MOD 0 - Ako su grupe u modu nula, dele se na dva skupa. Za grupu A ovi skupovi su port A i gornja četiri bita porta C, a za grupu B port B i donja četiri bita porta C. Svaki ulaz može da se koristi za ulaz (prijem) ili izlaz (slanje), ali ne istovremeno za ulaz i izlaz. Bitovi D4, D3, D1, D0 kontrolnog registra određuju koji je skup za ulaz, a koji za izlaz:

D4 se odnosi na port A

D3 se odnosi na gornju polovinu porta C

D1 se odnosi na port B

D0 se odnosi na donju polovinu porta C

Ako je bit nula odgovarajućei deo se koristi za izlaz, a ako je jedan za ulaz.

MOD 1 - Kada je grupa A u ovom modu port A se koristi za ulaz ili izlaz u skladu sa bitom D4 (kada je D4 jedinica onda se koristi za ulaz), a gornja polovina porta C koristi se za rukovanje i kontrolne signale. Kod ulaza (prijema) četiri bita porta C imaju sledeće značenje:

- PC4 ( $\overline{STB}_A$ ) (*Strobe Input*) nula na ovom ulazu učitava podatke PA7-PA0 u ulazni latch.

- PC5 ( $IBF_A$ ) (*Input Buffer Full*), govori da je ulazni buffer pun. On je jedan kada port A sadrži podatak koji još nije prenesen u procesor. Kada je nula na ovom pinu uređaj može upisati novi bajt u interfejs.

- PC6, PC7 mogu da budu korišćeni kao izlazni kontrolni signali ka uređaju (ako je D3 nula), ili za stanje ulaza sa uređaja (ako je jedan).

Pinovi PC4 i PC5 služe za rukovanje, a PC6 i PC7 kao kontrolni signali i signali za stanje. Značenje bitova porta C drukčija su kod izlaza (slanja):

- PC4 i PC5 imaju istu ulogu kao PC6 i PC7 kod ulaza,

- PC6 ( $\overline{ACK}_A$ ) uređaj postavlja ovaj pin na nulu kada prihvati podatke sa porta A

- PC7 ( $\overline{OBF}_A$ ) govori da li je izlazni buffer pun. Postaje nula kada je port A poslao novi podatak koji uređaj treba da prihvati.

Bitovi za rukovanje kod prijema i slanja razdvojeni su, a to je zbog istovremenosti tih operacija. Kad D4 odredi smer, automatski su određeni signali za rukovanje. Rukovanje može da se objasni na sledeći način:

*Rukovanje pri ulazu (prijemu):* Ulazni/Izlazni uređaj unosi podatke u interfejs. Sve dok je podatak u interfejsu, a procesor ga nije očitao, interfejs signalizira da je pun. Kad je podatak očitao interfejs šalje signal da je prazan.

*Rukovanje pri izlazu (slanju):* CPU upiše vrednost u interfejs i interfejs preko PC7 javlja da ima podatak. Ulazno/Izlazni uređaj učitava podatak i javlja da je to uradio.

*Značenje pojedinih bita u grupi B je sledeće:* PC3 je označen kao  $INTR_B$  i pridružen je grupi A. On se koristi kao linija za zahtev za prekid. Kod ulaza (prijema) u port A, ovaj pin postaje jedan kada je novi podatak stavljen u port A, a briše se, kada procesor uzme podatak. Kod izlaza ovaj se pin postavlja na jedan, kada se sadržaj porta A preuzme od strane uređaja, a briše se, kada je novi podatak poslat sa procesora. Kada je grupa B u modu 1 koristi se port B za ulaz ili izlaz, a šta je u pitanju govori nam bit D1 u kontrolnom registru (jedan za ulaz, a nula za izlaz). Za ulaz PC2 i PC1 označeni su kao  $\overline{STB}_B$  i  $IBF_B$  respektivno i imaju isto značenje kao  $\overline{STB}_A$  i  $IBF_A$  kod grupe A. Slično, za izlaz, PC1 i PC2 su označeni  $\overline{OBF}_B$  i  $\overline{ACK}_B$ . PC0 postaje  $INTR_B$ . Omogućavanje prekida za grupu A kontroliše se postavljanjem ili brisanjem internih flegova. Postavljanje ili brisanje ovih flegova simulirano je postavljanjem ili brisanjem PC4 za ulaz i PC6 za izlaz, koristeći

set/reset naredbu. Slično omogućavanje prekida za grupu B kontroliše se postavljenjem ili brisanjem PC2 i za ulaz i za izlaz.

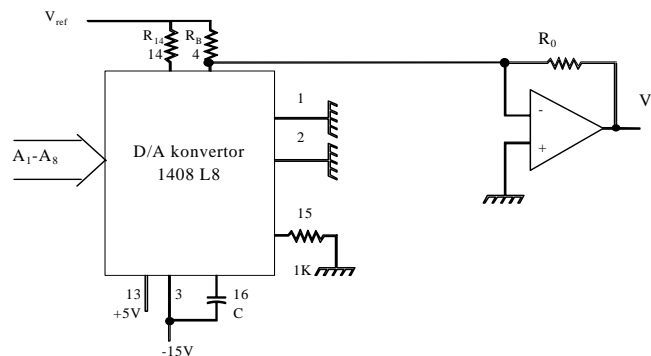
**MOD 2** - Ovaj mod primenjuje se samo za grupu A, mada se takođe koristi PC3 za zahtev za prekid. U modu 2 port A je dvosmerni, tj. sa A mogu da se šalju i primaju podaci.

- PC4 ( $\overline{STB}_A$ ) (*Strobe Input*) ima isto značenje kao kod moda 1 pri ulazu;
- PC5 ( $IBF_A$ ) (*Input Buffer Full*), postaje jedan, kada je port A napunjen novim podatkom sa linija PA7-PA0, a briše se kada ove podatke primi procesor;
- PC6 ( $\overline{ACK}_A$ ) (*Acknowledge*) govori da je uređaj spreman da primi podatke sa PA7-PA0;
- PC7 ( $\overline{OBF}_A$ ) (*Output Buffer Full*), postaje nula kada je port A napunjen novim podatkom iz procesora, a postavlja se na jedan, kada je uređaj prihvatio podatak.

Dok grupa A radi u modu 2, grupa B može da radi u modu 1 ili 0. Ako npr. grupa B radi u modu 0, tada samo PC2-PC0 mogu da se koriste za ulaz ili izlaz jer je grupa A "pozajmila" PC3 za liniju za zahtevanje prekida.

### 3.3. D/A Konvertor 1408 L8

D/A konvertor služi za konverziju digitalne veličine u analognu. Ulaz u D/A konvertor je digitalna reč, a izlaz analogni napon.



Slika 3.3. Šema D/A konvertora

Rezoluciju D/A konvertora predstavlja ukupan broj stanja izlaznog analognog napona. Ovaj D/A konvertor ima osmobaritnu rezoluciju, što znači da preko njega može da se prikaže maksimalno  $2^8 = 256$  različitih vrednosti. Iz procesora se šalje osmobaritna reč na njegov ulaz ( $A_1-A_8$ ), a na izlaz (pin 4) vezuje se operacioni pojačavač. Napon  $V_0$  na izlazu iz operacionog pojačavača predstavlja informaciju o poslatom digitalnom broju prema formuli:

$$V_0 = \frac{V_{ref}}{R_{14}} \cdot R_0 \cdot \left[ \frac{A_1}{2} + \frac{A_2}{4} + \frac{A_3}{8} + \frac{A_4}{16} + \frac{A_5}{32} + \frac{A_6}{64} + \frac{A_7}{128} + \frac{A_8}{256} \right] - \frac{V_{ref}}{R_B} \cdot R_0$$

$A_1$  do  $A_8$  predstavljaju osmobaritni broj koji treba pretvoriti u analognu veličinu. Referentni napon koji figuriše u jednačini ima vrednost od +5V, dok su vrednosti otpornika  $R_{14}$ ,  $R_0$  i  $R_B$ : 1k $\Omega$ , 2k $\Omega$  i 4k $\Omega$  respektivno. Vrednost izlaznog napona  $V_0$  kreće se u opsegu od -10V do +10V za vrednosti digitalne reči od 0 do 255. Kako je prihvatni registar DAC-a dovoljne dužine, tako da prihvata, bez odsecanja, sve digitalne vrednosti

ulaznog signala, analogni napon na izlazu ostaje sa zateèenom konstantnom vrednošæu sve dok se ulazni digitalni signal ne promeni.

Digitalna reè na ulaze DAC-a dolazi sa portova A i B PPI #2, preko dva invertora i optokaplera, što je prikazano na šemi D/A konverzije datoj na kraju ovog poglavlja. Galvansko odvajanje mikroprocesora od ostalog dela upravljaèkog kola izvršeno je optokaplerom. Na kartici postoje dva D/A konvertora 1408 L8 koji se koriste za prikazivanje i snimanje izmerenih velièina na osciloskopu. Informacije o brzinama izraèunatim po metodi merenja brzine brojanjem impulsa u toku perioda odabiranja, odnosno o brzini dobijenoj kombinovanom metodom, izbacuju se na portove A i B druge PPI-e, prikazane na šemi PC kartice.

## 4. OBRADA SIGNALA DOBIJENIH SA ENKODERA

Da bi se omogućilo upravljanje pogonom pomoću PC računara, u laboratorijski model ugrađeni su dodatni elektronski sklopovi. U računar je ugrađen sklop za prilagođavanje, izvršena je optička izolacija računara i invertora, ugrađeni su brojački sklopovi za merenje širine i broja impulsa davača i ostvarena veza računara sa A/D i D/A konvertorima.

### 4.1. Prototipska kartica

U računar je ugrađena kartica za prilagođavanje po šemi priloženoj u uputstvu za PC - "IBM Technical Reference". Šema ove kartice data je na kraju ovog poglavlja (šema: SHEMA PC KARTICE). Uloga ove kartice jeste da omogućí adresiranje elemenata na spoljnim karticama i da prenese podatke u oba smera (od elementa ka procesoru i obratno).

Mikroprocesor kompjutera komunicira sa karticama preko takozvanih ulazno-izlaznih portova. To praktično znači da mikroprocesor, ako hoće nešto da saopšti hardveru na kartici, na adresnu magistralu postavlja neku adresu i istovremeno aktivira signal koji se zove IOWR (*input/output write*), a ako hoće da uzima podatak sa kartice, onda uz određenu adresu aktivira signal IORD (*input/output read*). Na taj način, svaka od periferija dobija svoje "ime" (zapravo adresu) po kome će se odazivati. Za potrebe prototipskih kartica IBM je odvojio mesto na adresama od 300 Hex do 31F Hex.

Na šemi prototipske kartice, vidi se da za prenos podataka između spoljnih uređaja i procesora služi 74LS245 "Octal Bus Transceiver" čija je uloga osmo linijska asinhrona dvosmerna komunikacija između magistrala podataka. Pinom 1, označenim na šemi sa DIR ("*DIR*ection *I*nput"), kontroliše se smer prenosa podataka u zavisnosti od logičkog nivoa, dok se pinom 19, na slici označen sa  $\overline{G}$ , vrši izolovanje magistrale podataka. Komunikacija Transivera sa magistralama podataka obavlja se preko pinova označenih sa A1 do A8 i sa B1 do B8, koji mogu da budu ulazni, odnosno izlazni. Smer prenosa podataka određuje se pinovima  $\overline{G}$  i DIR prema tablici:

**Tabela 4.1.** Određivanje smera podatak transivera

$\overline{G}$	DIR	
0	0	podaci idu prema procesoru
0	1	podaci idu od procesora
1	X	magistrale podataka su izolovane

Za adresiranje spoljnih uređaja koriste se dva 74LS244 "Octal Buffer/Line Driver". Bafer omogućava ostvarivanje tri stanja na izlazu: nula, jedan i stanje visoke impedanse, što znači da spoljni uređaj može da se adresira slanjem nule ili jedinice ili da adresna magistrala bude neaktivna. Na pinove 2 i 4 prvog Bafera dovode se signali za čitanje odnosno za upisivanje u ili iz spoljnog uređaja ( $\overline{RD}$  i  $\overline{WR}$ ). Na pin 11 dovodi se RESET signal, a na pinove 13, 15, 17 kao i pinove 2, 4, 6, 8, 11, 13 i 15 drugog Bafera dovode se signali sa adresne magistrale od A0 do A9. Način rada je prikazan tablicom:

**Tabela 4.2.** Određivanje stanja izlaza Bafera

$\overline{1G}, \overline{2G}$	D	Izlaz
0	0	0
0	1	1
1	X	(Z)

Procesor komunicira sa spoljnim uređajima tako što im se prvo obrati, a zatim šalje ili uzima podatke sa njih. Ovo obračunavanje ostvaruje se signalom  $\overline{CS}$  *Chip Select*. Pomoću kola 74LS138 *Decoder/Demultiplexer* moguće je sa tri adresne linije (A2, A3 i A4 na šemi označene sa A, B i C) da se ostvari osam  $\overline{CS}$  signala. Na ovaj način omogućeno je adresiranje osam spoljnih uređaja. Adrese koje se formiraju na ovaj način date su u tabeli:

**Tabela 4.3.** Način formiranja adresa prototipske kartice

	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
(HEX 300)	1	1	0	0	0	0	0	0	0	0
(HEX 31F)	1	1	0	0	0	1	1	1	1	1
		1	0	0	0	X	X	X	X	X

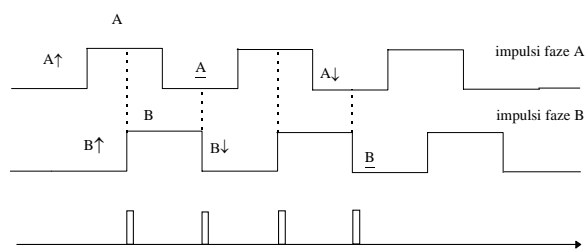
Na šemi kartice prikazano je povezivanje tri PPI 8255 i jednog 8253 tajmera i način, na koji su formirane njihove adrese. Adrese za PPI #1 su od 300H do 303H; PPI #2 su od 304H do 307H; PPI #3 su od 308H do 30BH i adrese tajmera su od 30CH do 30FH.

#### 4.2. Kolo za obradu signala od enkodera do tajmera

U laboratorijski model pogona kao pozicioni senzor ugrađen je inkrementalni optički enkoder. Za obradu i usložnjavanje signala, dobijenih sa enkodera, koristi se kartica čiji je deo šeme prikazan na kraju poglavlja (HEMA ENCODERA - BROJACI). Na ovoj kartici vrši se detekcija pozitivnih i negativnih ivica faznih impulsa. Kombinacijom ovako dobijenih impulsa sa faznim impulsima faze A i faze B enkodera, koji su fazno pomereni za  $90^\circ$ , rezolucija enkodera dva puta je veća. Ovaj postupak ilustrovan je slikom 4.1. Ugrađenom enkoderu sa 1250 impulsa po obrtaju, posle ovog usložnjavanja rezolucija se povećava na  $\frac{2 \cdot \pi}{2500}$ . Pri brzini od 1500 o/min, impulsi se pojavljuju svakih 16  $\mu$ s.

Sa enkodera prema kartici idu četiri žice. Dve služe da se njima dovedu signali faza A i B, jedna je 0V, a druga je +5V. Impulsi faza dovode se na integralno kolo obeleženo na slici kao ICZ. Izlazi ovog kola dovode se na optokapler koji služi za galvansko odvajanje električnih kola. Kombinacijom signala sa pinova 6 i 10 kola IC1 (na izlazu se pojavljuju signali A i  $\overline{A}$ , sa signalima pinova 6 i 8 kola IC4 (služe za detekciju uzlaznih i silaznih ivica i na izlazu pojavljuju signali se  $B\uparrow$  i  $B\downarrow$ ), na pinu 6 kola IC6 i IC7 formiraju se signali koji detektuju smer obrtanja:





Slika 4.1. Obrada faznih impulsa enkodera

$$UP = AB\uparrow + \bar{A}B\downarrow$$

$$DOWN = AB\downarrow + \bar{A}B\uparrow$$

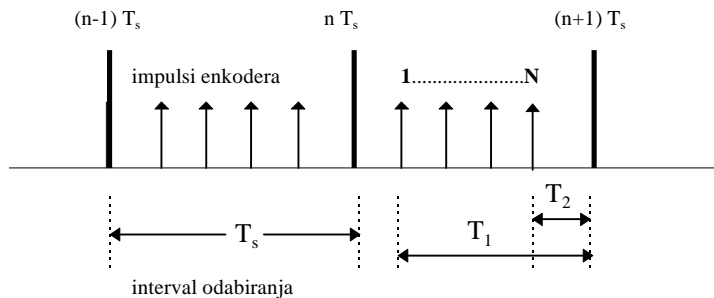
Kombinacijom ovih signala, odnosno njihovim dovođenjem na NI kolo (IC22 nalazi se na šemi na kraju poglavlja: SHEMA ENCODERA-FLIP FLOPOVI), dobija se signal koji predstavlja udvostručenu informaciju o brzini, u odnosu na onu, datu optičkim enkoderom.

Ovaj signal dovodi se na klock ulaz jednog od brojača tajmera 8253, koji ima zadatak da broji pojavu ovih impulsa u toku perioda odabiranja. Po dolasku impulsa na klock ulaz, broj upisan u brojač smanjuje se za jedan. Očitavanjem brojača na kraju svakog perioda odabiranja i njegovim poređenjem sa inicijalnom vrednošću, dobija se broj impulsa N potreban za izračunavanje brzine.

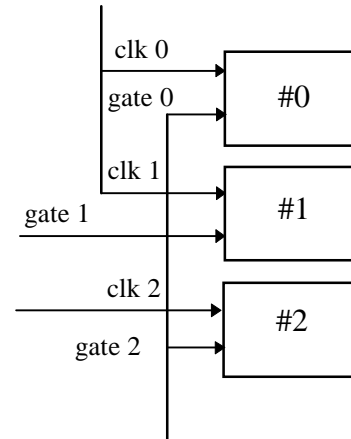
Ostala dva brojača služe za merenje vremena, pa se na njihov klock ulaz dovodi signal sa kristala učestanosti 1.8 MHz. Upravljanje brojačima vrši se kontrolom signala na gejtu. Šema dela kola kojim se upravlja ovim signalima data je na šemi enkodera (flip-flopovi) koja se nalazi na kraju poglavlja. Gejt signali brojača #0 i #2, (brojač #0 meri vreme od pojave prvog impulsa u periodu odabiranja do kraja ovog perioda, a brojač #2 meri prethodno opisan broj impulsa), dolaze sa pina 9 integrisanog kola sa "D" flip-flopovima (IC21). D flip-flop, kada su mu signali *set* i *clear* postavljeni na jedinicu, na izlazu postavlja signal sinhronizovano sa taktnim impulsom (izlaz se postavlja samo kada se na signal kloka dovede uzlazna ivica). Dovođenjem signala PC0 i PC1 određuje se početak brojanja. Dolaskom nule na PC0 i PC1 *clear* ulaz postaje 1. Kada se signali *set* i *clear* postave na jedinicu, pojavom prvog impulsa na pinu 11 (klock flip-flopa), pin 9 (izlaz flip flopa na slici označen sa Q) postaje jedan. Dolaskom jedinice na "gate" brojača omogućen je početak brojanja. Na kraju svakog vremena odabiranja izvršava se reset brojača tako što će se signali PC0 i PC1 postaviti na jedinicu, a samim tim će i izlaz Q postati nula. Sa šeme enkodera (flip-flopovi) takođe se vidi da se izlaz iz flip-flopa (Q) i izlaz sa pina 3 kola IC2, koji se dovodi i na klock ulaz flip-flopa, dovodi na jedno NI kolo (IC20). Ovim se kontroliše signal gejt brojača #1, čija je uloga da meri vreme od pojave poslednjeg impulsa do kraja intervala brojanja. Dolaskom novog impulsa sa enkodera, impuls pina 3 kola IC3 postaje nula pa i signal na gejtu brojača #1 postaje nula. Signal na gejtu biće jedan samo kada su i pin 3 kola IC3 i izlaz iz flip-flopa (Q) jedinice. Ovim je ostvareno resetovanje brojača #1 na početnu vrednost, svaki put kada se pojavi impuls sa enkodera i time omogućeno da se izvrši merenje vremena od pojave poslednjeg impulsa do kraja intervala odabiranja.

### 4.3. Primena tajmera za merenje brzine kombinovanom metodom

Merenje brzine, primenjeno na eksperimentalnom prototipu vektorski kontrolisanog pogona, vrši se određivanjem srednje širine impulsa u vremenu odabiranja  $T_s = 3ms$ . Za ovo merenje koristi se programabilni tajmer Intel 8253/8253-5 koji se sastoji od tri brojača događaja. Na prvi brojač dovode se impulsi koji predstavljaju vreme od pojave prvog impulsa u intervalu odabiranja  $T_s$  do kraja ovog intervala ( $T_1$ ). Drugi brojač meri impulse koji predstavljaju vreme od poslednjeg impulsa u intervalu odabiranja do kraja ovog intervala ( $T_2$ ). Treći brojač meri broj impulsa u periodu odabiranja ( $N$ ).



Slika 4.2. Merenje srednje širine impulsa u okviru intervala  $T_s$



Slika 4.3. blok šema rada tajmera

Brojač #0 nalazi se na adresi 30Ch,  
 Brojač #1 nalazi se na adresi 30Dh,  
 Brojač #2 nalazi se na adresi 30Eh,  
 Adresa registra u kome se nalazi kontrolna reč je 30Fh.

Pre početka brojanja potrebno je definisati režim rada (mod), i upisati početnu vrednost u brojač. Na donjem listingu prikazano je upisivanje kontrolne reči za svaki brojač. Dva bita najveće važnosti služe za određivanje brojača (00-brojač #0, 01- brojač #1 i 10- brojač #2), sledeća dva govore o načinu upisivanja ili čitanja stanja brojača (11 određuje da se prvo Upisuje/Čita bajt manje važnosti, pa onda bajt veće važnosti), režim rada (mod) predstavljen je sa sledećim tri bita (000 označava mod 0, a 001 mod 1) i poslednji bit govori da se radi o binarnim brojevima.

```

;*****
;upisivanje kontrolnih reci (30Fh je adresa kontrolnog registra)
MOV AL,30h ;00 11 000 0b => counter #0, Lsb Msb, mod 0,binarno
MOV DX,30Fh ;00110000b = 30h
OUT DX,AL
MOV DX,30Fh

MOV AL,72h ;01 11 001 0b => counter #1, Lsb Msb, mod 1,binarno
OUT DX,AL ;01110010b = 72h
MOV AL,0B0h ;10 11 000 0b => counter #2, Lsb Msb, mod 0,binarno
MOV DX,30Fh ;10110000b = 0B0h
OUT DX,AL
;*****
    
```

Upisivanje početne vrednosti u brojač prikazano je na sledećem listingu:

```

;*****
;punjenje pocetnih vrednosti brojaca na FFFFh

MOV DX,30Ch ;30Ch je adresa counter #0
    
```

```

MOV AL,0FFh ;upisivanje novog broja u counter #0
OUT DX,AL ;prvo Lsb pa Msb (Odredjeno sadrzajem
OUT DX,AL ;kontrolnom reci)

MOV DX,30Dh ;30Dh je adresa counter #1
MOV AL,0FFh ;upisivanje novog broja u counter #1
OUT DX,AL ;prvo Lsb pa Msb
OUT DX,AL

MOV DX,30Eh ;30Eh je adresa counter #2
MOV AL,0FFh ;upisivanje novog broja u counter #2
OUT DX,AL ;prvo Lsb pa Msb
OUT DX,AL
;*****

```

Na početku rada, u brojaèe je upisana vrednost FFFFh (65535d) što predstavlja maksimalni broj koji može da se predstavi sa šesnaest bita. Na klok ulaz brojaèa #0, koji meri vreme T1, dovodi se signal sa kristala uèestanosti 1.8 MHz. Ovaj brojaè radi u modu 0, što znaèi da æe smanjivati broj koji je upisan u njegov registar po taktu koji mu dolazi na klok ulaz, kada je signal na gejtju jedan. Kada je signal na gejtju nula brojanje je zaustavljeno. Brojaè #2 takođe radi u modu 0, ali se na njegov klok ulaz dovode impulsi sa kartice koji predstavljaju impulse enkodera obrađene na naèin objašnjen u drugom delu ovog poglavlja. Kada mu je signal na gejtju jedan, brojaè æe da smanjuje broj koji je upisan u njegov registar posle svakog signala sa kartice. Da bi se započeo novi ciklus brojanja, potrebno je dovesti nulu na gejt, znaèi zaustaviti brojanje, oèitati vrednost i zatim upisati novi broj. Upisom drugog bajta novog broja u registar brojaèa poèeæe novi ciklus brojanja. Ovo se radi periodièno u svakom intervalu odabiranja, pa se poređenjem oèitanog broja i broja upisanog na početku dobija informacija o vremenu koje je proteklo, u prvom sluèaju, ili o broju impulsa, koji su se pojavili u drugom. Uèestanost kristala je 1.8 MHz što znaèi da æe brojaè svake 0.55µs da odbroji po jedan broj. Množeæi oèitani broj sa 0.55µs dobija se potrebno vreme. Brojaè #1 radi u modu 1, a to znaèi da æe posle svake uzlazne ivice na gejtju poèeti da odbrojava od početka. Uzlazna ivica gejta pojavljavaæe se svakim dolaskom novog impulsa enkodera. Ovaj brojaè služi da izmeri vreme od poslednjeg impulsa do kraja intervala odabiranja. Poređenjem oèitanog broja sa upisanim i množeæi rezultat sa 0.55µs, dobija se potrebno vreme.

Na kraju intervala odabiranja potrebno je oèitati vrednosti brojaèa i upotrebiti ih za izraèunavanje brzine. Brojanje brojaèa koji rade u modu 0 (#0 i #2) zaustavlja se dovođenjem signala gejta na nulu. Za to je potrebno postaviti bitove PC0 i PC1 Programabilnog Paralelnog Interfejsa 8255 na jedan. Ovo je prikazano na listingu:

```

;*****
;ovim se gejtovi postavljaju na Low (0) i
;potrebno je za nov ciklus brojanja postaviti gejtove na High (1)

MOV AL,0FFh ;sa ovim postavljam PC0 i PC1 na 1
MOV DX,302h ;302h predstavlja adresu porta C PPI#1
OUT DX,AL

;postavljanjem gejtova na nulu u modu 0 se brojanje zaustavlja a u modu
;jedan pocinje se ponovo od upisanog broja
;*****

```

Kada je brojanje zaustavljeno vrši se oèitavanje vrednosti:

```

;*****
MOV DX,30Dh ;ocitavanje vrednosti tajmera 1 i smestanje u T1
IN AL,DX ;(vreme od poslednjeg impulsa do kraja intervala)
MOV BL,AL ;prvo LSB pa onda MSB
IN AL,DX ;privremeni smestaj u BL
MOV AH,AL
MOV AL,BL ;T0,T1 i N2 predstavljaju promenljive veæ
MOV T1,AX ;rezervisane u RAM-u. Ovo je objasnjeno u
;glavnom listingu

MOV DX,30Ch ;ocitavanje vrednosti tajmera 0 i smestanje u T0
IN AL,DX ;(vreme od prvog impulsa do kraja intervala)
MOV BL,AL ;prvo LSB pa onda MSB
IN AL,DX ;privremeni smestaj u BL
MOV AH,AL
MOV AL,BL
MOV T0,AX

MOV DX,30Eh ;ocitavanje vrednosti tajmera 2 i smestanje
IN AL,DX ;u N2 (broj impulsa u toku intervala)
MOV BL,AL ;prvo LSB pa onda MSB
IN AL,DX ;privremeni smestaj u BL
MOV AH,AL
MOV AL,BL
MOV N2,AX
;*****

```

Da bi se zapoèeo novi ciklus brojanja potrebno je ponovo upisati broj u registar brojaèa #0 i #2. Brojaè #1 radi u modu 1 pa se on resetuje svakim dolaskom uzlazne ivice na "gate". Posle upisivanja novih vrednosti potrebno je postaviti bitove PC0 i PC1 na nulu odnosno postaviti signale gejtova na jedan.

```

;*****
;punjenje pocetnih vrednosti brojaca #0 i #2 na FFFFh
;i pocetak novog intervala brojanja
MOV DX,30Ch
MOV AL,0FFh ;upisivanje novog broja u counter 0
OUT DX,AL ;prvo Lsb pa Msb
OUT DX,AL

MOV DX,30Eh
MOV AL,0FFh ;upisivanje novog broja u counter 2
OUT DX,AL ;prvo Lsb pa Msb
OUT DX,AL
;*****
;postavljanje gejtova na 1 i pocetak intervala brojanja

MOV DX,302h ;adresa porta C u kojoj dajem vrednosti PC0 i PC1=0
MOV AL,00h ;302h predstavlja adresu porta C PPI#1
OUT DX,AL
;*****

```

Ovim je objašnjen deo programa koji služi da inicijalizuje brojaèe tajmera (odredi im mod i upiše poèetnu vrednost), kao i deo programa koji oèitava vrednosti brojaèkih registara, smešta te vrednosti u promenljive ranije definisane u RAM-u i zapoèinje novi ciklus brojanja. Na poèetku brojanja u brojaèke registre upisivan je broj FFFF Hex zbog opštosti primera. U zavisnosti od vremena odabiranja ovaj broj može da bude manji pa tako mogu da se dobiju manje vrednosti i da se olakša kasniji raèun.

## 5. PROGRAMSKA OSNOVA

U ovom poglavlju biæe reæi o programskim paketima pomoæu kojih je ostvarena izrada ovog rada. Na kraju poglavlja dat je i listing programa za izraèunavanje brzine kombinovanom metodom.

Za obradu teksta korišæen je program "*MicroSoft Word for Windows*" verzija 6.1, za obradu slika program "*CorelDRAW*" verzija 4.00, za brzu Furijeovu transformaciju i obradu grafikona program "*MicroCal Origin*" verzija 3.01, za crtanje elektronskih šema program "*OrCad*" verzija 4.10 i za prenos podataka sa osciloskopa "*Hitachi VC-6202*" u raèunar korišæenjem *GPIB* interfejsa program "*TVOSC*". Program za izraèunavanje brzine kombinovanom metodom napisan je u "*MicroSoft MACRO Assembler*" verzija 1.25, a povezan je korišæenjem "*MicroSoft Linker*" verzija 2.40. Otkrivanje grešaka u programu vršeno je "*IBM Fullscreen Debug*" verzija 1.4. Pored literature date na kraju rada korišæena je i literatura u elektronskoj formi: Uputstvo za *OrCad* autora Berislava Todoroviæa kao i "*Norton Guide - The Assembly Language database*".

Program napisan u editoru teksta, potrebno je prevesti na mašinski jezik, jezik razumljiv mikroprocesoru. Ovo prevoðenje izvršeno je programom "*MACRO Assembler*", koji konvertuje instrukcije asemblerskog jezika u binarni oblik (object code). Ovako formirana datoteka ima ekstenziju \*.OBJ i pored izvršnog mašinskog koda sadrži informacije o strukturi izvršnog programa. Posle prevoðenja potrebno je izvršiti povezivanje odvojenih delova objektnih fajlova, koji mogu da potiæu i od viših programskih jezika, u jedan i njegovo konvertovanje u izvršni program sa ekstenzijom \*.EXE. Ovo se vrši programom "*Link*". Otkrivanje grešaka i provera, u izvršnom programu obavlja se programom "*Fullscreen Debug*", koji omoguæava da se program unese u sistemsku memoriju i izvršava naredbu po naredbu. Ovaj program omoguæava potpuni uvid u stanje sadržaja svih registara i memorijskih lokacija.

Na slici je prikazan proces prevoðenja i povezivanja programa èiji je listing dat na kraju poglavlja:

```
C:\PROGRAMI>masm 3ms1
Object filename [3ms1.OBJ]:
Source listing [NUL.LST]:
Cross reference [NUL.CRF]:
Warning Severe
Errors Errors
0          0

C:\PROGRAMI>link 3ms1
Run file [3ms1.EXE]:
List file [NUL.MAP]:
Libraries [.LIB]:

C:\PROGRAM>fsd_
```

**Slika 5.1.** Prikaz prevoðenja, povezivanja i provere programa pisanog u Asemblerskom jeziku

Prevoðenjem i povezivanjem dobija se izvršni fajl 3ms1.exe. koji je potrebno proveriti. Ova provera vršena je u programu *Fullscreen Debug* koji se startuje naredbom *fsd*. Radni ekran ovog programa dat je na slici:

AX 0000	SI 0000	CS 1F27	IP 0100	Stack +0 0000	FLAGS 0200
BX 0000	DI 0000	DS 1F27		+2 01A6	
CX 0000	BP 0000	ES 1F27		+4 4689	OF DF IF SF ZF AF PF CF
DX 0000	SP 2C3E	SS 1F27	FS 1F27	+6 83F8	0 0 1 0 0 0 0 0
-----					
CMD > 1 3ms1				<b>1</b>	0 1 2 3 4 5 6 7
				DS:0000	CD 20 FF 9F 00 9A 3E 2B
				DS:0008	58 FD 67 02 FB 11 58 33
<b>0100 55</b>	<b>PUSH</b>	<b>BP</b>		DS:0010	FB 11 80 26 FB 11 1D 11
0101 8BEC	MOV	BP, SP		DS:0018	01 01 01 00 02 FF FF FF
0103 83EC42	SUB	SP, 0042		DS:0020	FF FF FF FF FF FF FF FF
0106 8A4606	MOV	AL, [BP+06]		DS:0028	FF FF FF FF DE 11 80 D2
0109 2AE4	SUB	AH, AH		DS:0030	FB 11 14 00 18 00 27 1F
010B 50	PUSH	AX		DS:0038	FF FF FF FF 00 00 00 00
010C 90	NOP			DS:0040	06 14 00 00 00 00 00 00
010D 0E	PUSH	CS		DS:0048	00 00 00 00 00 00 00 00
-----					
<b>2</b>	0 1 2 3 4 5 6 7	8 9 A B C D E F			
DS:0000	CD 20 FF 9F 00 9A 3E 2B	58 FD 67 02 FB 11 58 33		. . . . .>+ X.g...X3	
DS:0010	FB 11 80 26 FB 11 1D 11	01 01 01 00 02 FF FF FF		...&....	
DS:0020	FF FF FF FF FF FF FF FF	FF FF FF FF DE 11 80 D2		.....	
DS:0030	FB 11 14 00 18 00 27 1F	FF FF FF FF 00 00 00 00		.....	
DS:0040	06 14 00 00 00 00 00 00	00 00 00 00 00 00 00 00		.....	
-----					
<b>1</b>	<b>Step</b>	<b>2</b>	<b>StepProc</b>	<b>3</b>	<b>Retrieve</b>
<b>4</b>	<b>Help</b>	<b>5</b>	<b>Set BRK</b>	<b>6</b>	<b>7</b>
<b>up</b>	<b>8</b>	<b>dn</b>	<b>9</b>	<b>le</b>	<b>0</b>
<b>ri</b>					

Slika 5.2. Radni ekran programa za otkrivanje grešaka *Fulscreen Debug*

*Fulscreen Debug* omogućava da se program unese u sistemsku memoriju i izvršava naredbu po naredbu. Tokom izvršavanja naredbi, na ekranu može da se prate sadržaji registara, memorijskih lokacija i *PSW*. Na taj način vrši se provera predloženog rešenja i otklanjanje eventualno napravljenih grešaka. Program se učitava komandom: "**1 imeprograma**" (**1** - *load*). Kada je provera završena izlazak se ostvaruje naredbom: "**quit**". U gornjem levom delu ekrana prikazani su sadržaji svih registara, dok je desno od njih prikazan sadržaj Procesorske Statusne Reči (*PSW*). U levom delu ekrana nalazi se komandna linija, ispod koje se nalazi deo listinga programa čiji se sadržaj proverava. Sadržaj memorijskih lokacija na navedenim adresama prikazan je u donjem i desnom delu ekrana. U poslednjem redu ekrana nalazi se meni sa često korišćenim komandama.

Kompletan listing programa kojim se određuju režimi rada (modovi) tajmera i PPI-a, inicijalizuju vrednosti brojača tajmera, ispisuje dobijena vrednost brzine na ekranu i određuje brzina kombinovanom metodom i metodom brojanja impulsa u intervalu odabiranja, uz njihovo prikazivanje na D/A konvertorima radi poređenja, prikazan je uz odgovarajuće komentare. Deo programa, u kome se vrši očitavanje vrednosti brojača, izračunavanje brzine i ispis na D/A konvertorima, ostvaruje se svake *3ms* u posebnoj proceduri INHAR (INterrupt HANdleR). Ova procedura prekida petlju u kojoj se ispisuju brojevi na ekranu zato što je njen zahtev za prekidom veće prioriteta. Nakon izvršenja procedure INHAR, petlja nastavlja da se izvršava od prekinutog mesta do pojave sledećeg zahteva za prekidom.

```

;*****
;** Program za očitavanje brzine asinhronog motora **
;** Jovan Petrovic **
;** avgust, 95 (verzija za Ts = 3ms) **
;*****

```

```
NAME DIPL3ms1 ;Ime programa
```

```
STEK SEGMENT STACK ;Na ovaj način rezervisem prostor na
```

```

        DB 64 DUP(?)          ;steku gde smestam tekuce vrednosti
        STEK ENDS            ;registara prilikom interapta.

;*****
DATA SEGMENT                ;Rezervisanje prostora u RAM-u za promenljive

        T0      dw 1 ;vreme od prvog impulsa do kraja intervala
        T1      dw 1 ;vreme od poslednjeg impulsa do kraja intervala
        N2      dw 1 ;broj impulsa u toku intervala odabiranja (sa
                    ;enkodera)

        OST     dw 1 ;sluzi za smestanje ostatka pri delenju
        xb      db 1 ;promenljive koje koristim za ispis vrednosti
        yb      db 1 ;trenutne brzine na ekranu
        zb      db 1
        dhi     db 1
        hil     db 1
        sto     db 1
        des     db 1
        jed     db 1
        xw      dw 1
        yw      dw 1
        zw      dw 1

        OMEGA   dw 1 ;vrednost brzine koja se ispisuje na ekranu
        OMEGA_1 dw 1 ;brzina dobijena kombinovanom metodom (za DAC)
        OMEGA_2 dw 1 ;brzina dobijena metodom brojanja impulsa (za DAC)
        stari_seg dw 1 ;ovde ùvavam segment i ofset originalne
        stari_off dw 1 ;prekidne rutine koji menjam u programu

        DATA ENDS

;*****
PROG SEGMENT                ;izvrnsni deo programa
ASSUME CS:PROG,SS:STEK,DS:DATA ;dodela segmentnim registrima
                                ;adrese segmenata
PRIM PROC                    ;pocetak procedure PRIM

        MOV AX,DATA            ;punjenje DATA segmenta
        MOV DS,AX              ;vrsi se na ovaj nacin
;*****
;vraca adresu interrupt-handling rutine (ES:BX)
        MOV AH,35H             ;na ovaj nacin ce preko interapta 21h i
        MOV AL,1CH             ;njegove funkcije 35h da bude sacuvana
        INT 21H                ;originalna prekidna rutina interapta 1CH
        MOV DX,ES
        MOV STARI_SEG,DX       ;pamtim je da bih posle mogao da izadjem
        MOV STARI_OFF,BX       ;iz programa
;*****
;*****
;upisivanje kontrolnih reci 30Fh je adresa kontrolnog registra
        MOV AL,30h             ;00 11 000 0b => tajmer 0, Lsb Msb, mod 0,binarno
        MOV DX,30Fh           ;00110000b = 30h
        OUT DX,AL

        MOV DX,30Fh           ;01 11 001 0b => tajmer 1, Lsb Msb, mod 1,binarno
        MOV AL,72h            ;00110010b = 72h
        OUT DX,AL

        MOV AL,0B0h           ;10 11 000 0b => tajmer 2, Lsb Msb, mod 0,binarno
        MOV DX,30Fh           ;10110000b = 0B0h
        OUT DX,AL
;*****
;punjenje pocetnih vrednosti brojaca na FFFFh
        MOV DX,30Ch           ;adresa countera 0
        MOV AL,0FFh           ;upisivanje novog broja u counter 0
        OUT DX,AL             ;prvo Lsb pa Msb,

```

```

OUT DX,AL ;(odredjeno sadrzajem kontrolnog registra)

MOV DX,30Dh ;adresa countera 1
MOV AL,0FFh ;upisivanje novog broja u counter 1
OUT DX,AL ;prvo Lsb pa Msb
OUT DX,AL ;(odredjeno sadrzajem kontrolnog registra)

MOV DX,30Eh ;adresa countera 2
MOV AL,0FFh ;upisivanje novog broja u counter 2
OUT DX,AL ;prvo Lsb pa Msb
OUT DX,AL ;(odredjeno sadrzajem kontrolnog registra)
;*****
;postavljanje gejtova na 1 i pocetak intervala brojanja
MOV DX,303h ;postavljamo CR PPI#1 gde su svi portovi ulazni
MOV AL,9Ah ;osim 4 niza bita porta C (PC0&PC1 su izlazni)
OUT DX,AL

MOV DX,302h ;adresa porta C u kojoj dajem vrednosti PC0 i PC1=0
MOV AL,00h
OUT DX,AL

MOV DX,307H ;postavljam CR PPI#2 80h = 1 0000 000
MOV AX,80H ;grupa A (mod 0 port A out port C upper out)
OUT DX,AX ;grupa B (mod 0 port B out port C lower out)
;*****
;promena ucestanosti interapta i promena adrese prekidne rutine
CLI ;onemogucavanje dogadjanja interapta
MOV AL,36H ;00 11 011 0b tajmer 0, Lsb Msb, mod 3,binarno
OUT 43H,AL ;prekidi svakih 3ms
MOV AL,0FCH ;Lsb = FCh
OUT 40H,AL
MOV AL,0DH ;Msb = 0Dh
OUT 40H,AL ;0DFCh = 3580d 1,19318MHz/3580 = 333,29Hz (3ms)
STI ;omogucavanje dogadjanja interapta

MOV DX,SEG INSER ;promena sadrzaja prekidne rutine
MOV DS,DX ;interapta lCH pomocu funkcije 25h
MOV DX,OFFSET INSER ;INT 21h. Interaptu lCH dodeljuje se
MOV AL,lCH ;nova prekidna rutina INSER
MOV AH,25H
INT 21H
;*****
MOV AX,DATA
MOV DS,AX

PETLJA:
MOV DL,0DH ;priprema za ispis podataka na ekranu. 02H je Dos f-ja
MOV AH,02H ;za ispisivanje sadrzaja DL na ekran. 0DH je Carriage
INT 21H ;Return

MOV AX,OMEGA ;Ispisivanje izmerene brzine na ekranu.
MOV ZW,AX

MOV DX,0 ;dx:ax/ source = ax, ostatak u dx
MOV AX,10000 ;cifra na mestu deset hiljada
MOV XW,AX ;xw imenilac
MOV AX,ZW ;dx:ax brojilac

DIV XW ;ax rezultat (desetine hiljada) dx ostatak
MOV DHI,AL ;dhi desetine hiljada
MOV ZW,DX ;zw cuva ostatak
;*****

```



```

MOV DX,0           ;dx:ax/ source = ax, ostatak u dx
MOV AX,1000        ;cifra na mestu hiljada
MOV XW,AX          ;xw imenilac
MOV AX,ZW          ;dx:ax brojilac

DIV XW             ;ax rezultat (hiljade) dx ostatak
MOV HIL,AL         ;hil hiljade
MOV ZW,DX          ;zw cuva ostatak.....
;*****
MOV DX,0           ;dx:ax/ source = ax, ostatak u dx
MOV AX,100         ;cifra na mestu stotina
MOV XW,AX          ;xw imenilac
MOV AX,ZW          ;dx:ax brojilac

DIV XW             ;ax rezultat stotine
MOV STO,AL         ;ostatak su jedinice
MOV ZW,DX          ;zw cuva ostatak.....

;*****
MOV DX,0           ;dx:ax/ source = ax, ostatak u dx
MOV AX,10          ;cifra na mestu desetice
MOV XW,AX          ;xw imenilac
MOV AX,ZW          ;dx:ax brojilac

DIV XW             ;ax rezultat desetice
MOV DES,AL         ;ostatak su jedinice
MOV ZW,DX          ;ostatak su jedinice

;*****
;ispis podataka na ekranu
MOV AX,ZW
MOV JED,AL

MOV DL,DHI         ;cifra desetina hiljada ubacujem u registar
ADD DL,30H         ;DL, dodajem toj vrednosti broj 30H i dobijam
MOV AH,02H         ;ASCII vrednost te cifre
INT 21H           ;karakter od 0 do 9 imaju ASCII vrednosti od
                  ;30h do 39h, koji se sada nalaze u registru DL
MOV DL,HIL         ;i na taj nacin se pomocu f-je 02h interapta 21h
ADD DL,30H         ;ispisuju na ekranu
MOV AH,02H         ;isto primenjujem i za ispis cifara: hiljada,
INT 21H           ;stotina, desetina i jedinca

MOV DL,STO
ADD DL,30H
MOV AH,02H
INT 21H

MOV DL,DES
ADD DL,30H
MOV AH,02H
INT 21H

MOV DL,JED
ADD DL,30H
MOV AH,02H
INT 21H

;*****
;potrebno je za izlazak iz programa
MOV AH,06H         ;funkcija 06h interapta 21h ispituje da li
MOV DL,0FFH        ;pritisnut neki taster i ako jeste prelazi
INT 21H           ;na podprogram EXIT
JNZ EXIT

```

```

JMP PETLJA

;*****
;izlaz iz programa
EXIT: CLI
      MOV AL,36H           ;izlazak iz programa tako sto se
      OUT 43H,AL          ;sistemski tajmer vraca na generisanje
      MOV AL,0FFH         ;interapta 08h na svakih 55ms punjenjem
      OUT 40H,AL          ;njegovog counera #0 na adresi 40h
      OUT 40H,AL          ;jedincama
      STI

      MOV DX,STARI_OFF    ;vracane originalne rutine interaptu
      MOV BX,STARI_SEG    ;1Ch pomocu f-je 25h ciji su segment i
      MOV DS,BX           ;ofset bili sacuvani u stari_seg i
      MOV AH,25H          ;stari_off
      MOV AL,1CH
      INT 21H

      MOV AH,4CH          ;izlazak iz programa i povratak u DOS pomocu
      INT 21H             ;f-je 4Ch interapta 21h

      PRIM ENDP          ;kraj procedure PRIM
;*****
      INHAR PROC ;prekidna rutina koja se dodeljuje interaptu 1Ch
                  ;umesto originalne
INSER: PUSH AX        ;smestam sadrzaje registara koji se koriste u
      PUSH BX         ;ovoj prekidnoj rutini na stek
      PUSH CX
      PUSH DX
      PUSH DS
      PUSH SI
      PUSH DI

      MOV AX,DATA
      MOV DS,AX
;*****
;ovim se gejtovi postavljaju na L (0) i potrebno je za nov ciklus
;brojanja postaviti gejtove na H (1)
      MOV AL,0FFh      ;sa ovim postavljaj PC0 i PC1 na 1
      MOV DX,302h     ;odnosi se na PPI#1
      OUT DX,AL
;postavljanjem gejtova na nulu u modu 0 se brojanje zaustavlja a u modu
;jedan pocinje se ponovo od upisanog broja
;*****
      MOV DX,30Dh     ;ocitavanje vrednosti tajmera 1 i smestanje u T1
      IN AL,DX        ;(vreme od poslednjeg impulsa do kraja intervala)
      MOV BL,AL       ;prvo LSB pa onda MSB
      IN AL,DX        ;privremeni smestaj u BL
      MOV AH,AL
      MOV AL,BL       ;T0,T1 i N2 predstavljaju promenljive vec
      MOV T1,AX       ;rezervisane u RAM-u. Na pocetku programa

      MOV DX,30Ch     ;ocitavanje vrednosti tajmera 0 i smestanje u T0
      IN AL,DX        ;(vreme od prvog impulsa do kraja intervala)
      MOV BL,AL       ;prvo LSB pa onda MSB
      IN AL,DX        ;privremeni smestaj u BL
      MOV AH,AL
      MOV AL,BL
      MOV T0,AX

      MOV DX,30Eh     ;ocitavanje vrednosti tajmera 2 i smestanje u
      IN AL,DX        ;u N2 (broj impulsa u toku intervala)
      MOV BL,AL       ;prvo LSB pa onda MSB

```

```

        IN  AL,DX      ;privremeni smestaj u BL
        MOV AH,AL
        MOV AL,BL
        MOV N2,AX
;*****
;punjenje pocetnih vrednosti brojaca #0 i #2 na FFFFh
;i pocetak novog intervala brojanja
        MOV DX,30Ch      ;adresa countera #0
        MOV AL,0FFh     ;upisivanje novog broja u counter 0
        OUT DX,AL       ;prvo Lsb pa Msb
        OUT DX,AL

                MOV DX,30Eh      ;adresa countera #2
                MOV AL,0FFh     ;upisivanje novog broja u counter 2
                OUT DX,AL       ;prvo Lsb pa Msb
                OUT DX,AL
;*****
;postavljanje gejtova na 1 i pocetak intervala brojanja
        MOV DX,302h     ;302h je adresa porta C u kojim saljem vrednosti
        MOV AL,00h     ;PC0 i PC1=0
        OUT DX,AL
;*****
;izracunavanje brzine u [ob/min], po formuli
;omega=(60/2500)*1 800 000*[(65535-N2)-1]/[(65535-T0)-(65535-T1)]
;sto je ekvivalentno omega = 43200*(65534-N2)/(T1-T0)
;60*1800 000/2500 = 43200 (216*200=43200)
        MOV AX,T1
        CMP AX,T0
        JBE DELNUL     ;provera da ne dodje do deljenja nulom
        JMP SKOK
DELNUL:
        MOV AX,0       ;ako je deljenje nulom ispisi nula na ekranu
        JMP SKOK_1
SKOK:
        SUB AX,T0

        MOV BX,AX

        MOV AX,0FFFEh ;65534d = 0FFFEh potrebno je da se odredi
        SUB AX,N2     ;koliko je brojeva counter #2 odbrojao
        MOV CX,216   ;mnozim ga prvo sa 216 da bi ostao u opsegu
        MUL CX       ;216d=D8h
        SUB DX,DX     ;praznim registar DX
        DIV BX       ;rezultat u AX a ostatak u DX

        MOV OST,DX   ;sklanjam ostatak u OST zbog donjeg deljenja

        MOV CX,200   ;mnozim ga sa 200 jer je 216*200=43200
        MUL CX       ;200d=C8h
        MOV OMEGA,AX ;ovaj deo sluzi za dobijanje jos
        MOV AX,OST   ;jednog decimalnog mesta
        MOV CX,10    ;mnozim ostatak sa 10 pa cu posle sa 20
        MUL CX       ;da bi se dobilo decimalno mesto

        SUB DX,DX     ;praznim registar DX
        DIV BX       ;rezultat u AX ostatak u DX

        MOV OST,DX   ;sklanjam ostatak u OST zbog donjeg deljenja
        MOV CX,20    ;ovo je prvo decimalno mesto pa je 20 umesto
        MUL CX       ;200 20d=14h
        ADD AX,OMEGA

        MOV OMEGA,AX ;ovaj deo sluzi za dobijanje jos
        MOV AX,OST   ;jednog decimalnog mesta (drugog)

```

```

MOV CX,10      ;mnozim sa 10 pa cu posle da 2
MUL CX        ;da bi se dobilo decimalno mesto

SUB DX,DX     ;praznim registar DX
DIV BX       ;rezultat u AX a ostatak u DX
MOV CX,2     ;2d=2h
MUL CX
ADD AX,OMEGA

SKOK_1:      MOV OMEGA,AX
;*****
;priprema brzine dobijene motodom brojanja impulsa u intervalu
;odabiranja, za proveru dobijenih rezultata. Brzina se izracunava po
;formuli: (60/2500)*(65535-N2)/Ts   Ts=3ms vreme odabiranja
;OMEGA_2 = 8*(65535-N2)
MOV AX,65535  ;realizacija gore navedene formule
SUB AX,N2
MOV CX,8
MUL CX
MOV CX,1244  ;prilagodjavanje ispisu na DAC-u
SUB AX,CX
MOV OMEGA_2,AX ;OMEGA_2 je brzina izracunata metodom
MOV AX,OMEGA  ;brojanja impulsa u intervalu odabiranja
MOV CX,1244
SUB AX,CX    ;OMEGA_1 je brzina izracunata kombinovanom
MOV OMEGA_1,AX ;metodom
;*****
;ispis na D/A konvertor (adrese DAC_1 = 304H DAC_2 = 305H)
;na D/A bice ispisane vrednosti od 1244 do 1500 zbog rezolucije
;(8bita => 256)
MOV DX,305H
OUT DX,AL    ;u AL se vec nalazi OMEGA_1
MOV DX,304H
MOV AX,OMEGA_2
OUT DX,AL    ;ispis OMEGA_2 n D/A
;*****
POP DI      ;vracanje sadrzaja registara koji su
POP SI     ;korisceni u interaptu na vrednosti
POP DS     ;pre nastanka interapta
POP DX
POP CX
POP BX
POP AX

IRET      ;povratak u glavnu petlju

INHAR ENDP ;kraj procedure INHAR
;*****
PROG ENDS ;kraj programskog segmenta
END PRIM  ;kraj programa.
;*****

```

## 6. EKSPERIMENTALNI REZULTATI

Da bi se prikazale prednosti merenja brzine kombinovanom metodom bilo je potrebno da se izmerena i izraunata vrednost brzine, uporedi sa brzinom dobijenom na neki drugi naèin. Rezultate dobijene u digitalnom obilku trebalo je prikazati na osciloskopu. Za ovo je bio upotrebljen D/A konvertor 1408 L8 prikazan u poglavlju 3.3.

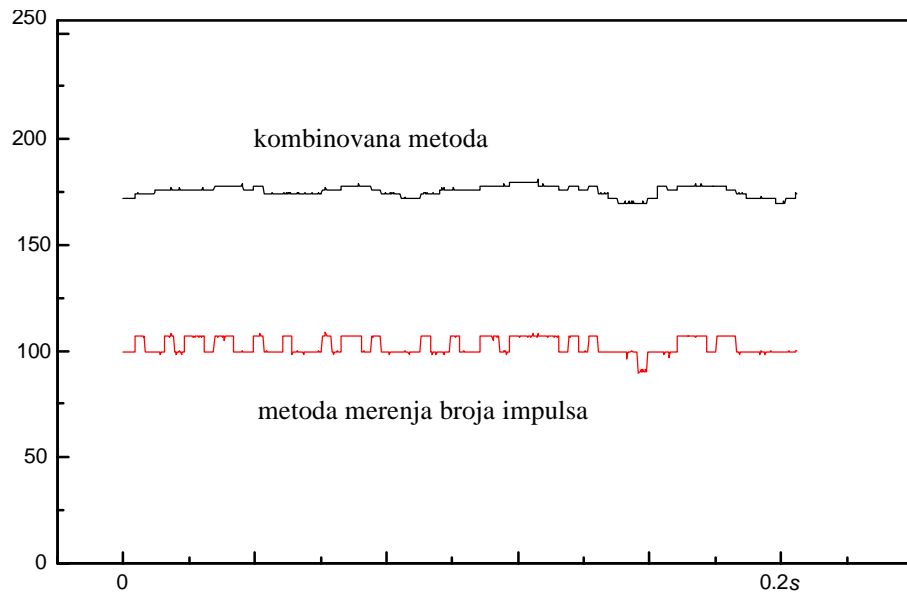
Ovaj D/A konvertor ima osmorbitnu rezoluciju, što znaèi da se preko njega moæe prikazati maksimalno  $2^8 = 256$  razlièitih vrednosti. Iz procesora se preko PPI 8255 #2, šalju vrednosti na D/A konvertor na èijim se izlazima mere dobijene analogne vrednosti. Na port A PPI 8255, odnosno na prvi D/A konvertor poslata je digitalna reè koja predstavlja informaciju o brzini izraunatoj po metodi merenja brzine brojanjem impulsa u toku perioda odabiranja, a na port B, odnosno drugi D/A konvertor, poslata je informacija o brzini dobijena kombinovanom metodom.

Brzina je predstavljena naponom od -10V do +10V, za 256 obrtaja, što predstavlja 0.08V po obrtaju. Merenja su vršena obradom impulsa dobijenih sa enkodera koji je ugrađen na vratilo asinhronog motora i obradom impulsa dobijenih sa ton generatora. Fazni pomeraj impulsa sa ton generatora ostvaren je primenom RC-filtra i Šmitovog kola. Otpornik i kondenzator izabrani su tako da se ostvari kašnjenje jednog signala u odnosu na drugi za  $90^\circ$  ( $\tau = R \cdot C = \frac{1}{4 \cdot f}$ ).

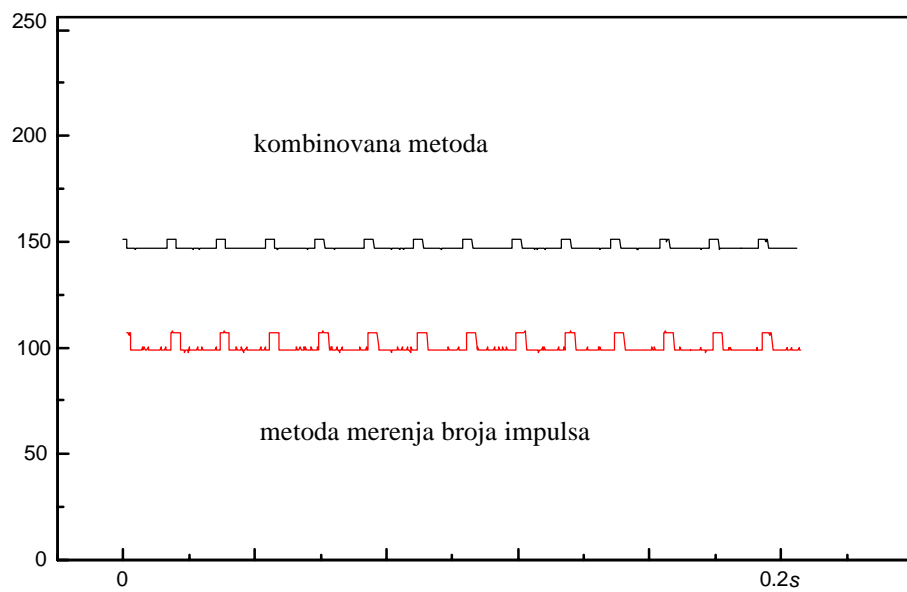
Talasnici snimani su pomoæu digitalnog memorijskog osciloskopa *HITACHI VC-6020*, a zatim preko *GPIB* interfejsa preneti u raèunar. Pored *GPIB* interfejsa, korišæen je program *TVOSC* koji vrši transfer podataka sa osciloskopa *VC-6020* u raèunar. Obrada podataka izvršena je u programu "*MicroCal Origin*", verzija 3.01.

Snimljene i obraðene vrednosti, za velike i male brzine dobijene merenjima impulsa sa enkodera i ton generatora, prikazane su na gornjim slikama. Na svakoj od slika prikazane su istovremeno dobijene vrednosti kombinovanom metodom i metodom merenja impulsa u intervalu odabiranja, u cilju njihovog poreðenja. Razmera za Y - osu uzeta je od 0 do 256 što predstavlja jedan obrtaj po minutu zasvaki podeok.

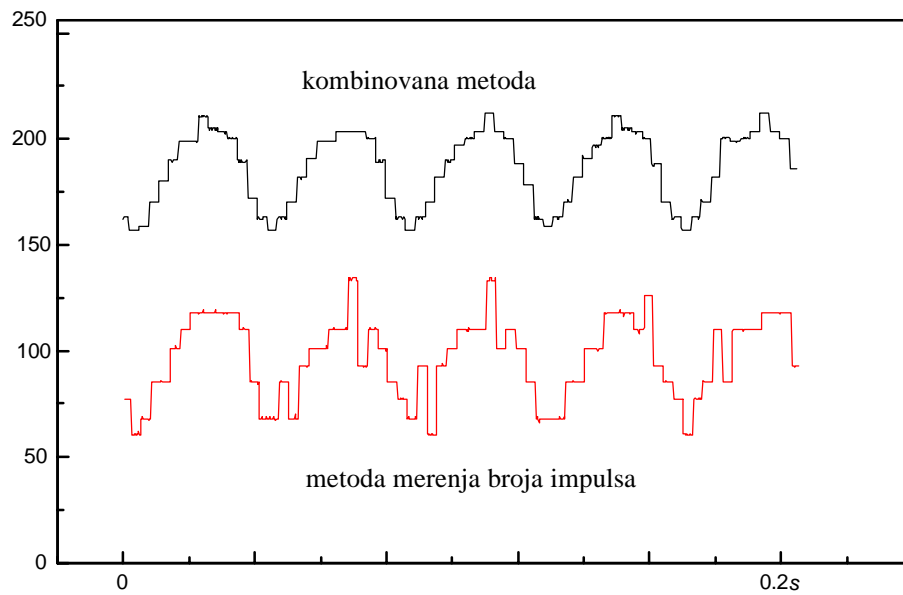
Na slikama 6.1. i 6.2. prikazani su talasni oblici brzine dobijeni pri malim brzinama, dok slike 6.3., 6.4. i 6.5. predstavljaju talasne oblike brzine snimljene pri velikim brzinama. Slike 6.1., 6.3. i 6.4. predstavlja realan sluèaj (merenje je izvršeno obradom impulsa koji dolaze sa enkodera ugrađenog na osovinu motora), a slike 6.2. i 6.5 predstavljaju idealizovan sluèaj (merenje je izvršeno obradom impulsa simuliranih ton generatorom). Sve slike, osim 6.4., koja je pri snimanju imala vremensku bazu 50ms, snimljene su pri vremenskoj bazi od 20ms. Na slikama 6.6. i 6.7. prikazana je spektralna analiza sluèaja prikazanog na slici 6.5. koja je izvršena brzom Furijeovom transformacijom. Sa ove slike, moæe se videti da je energija šuma pri merenju brzine klasiènom metodom mnogo veæa od energije šuma pri kombinovanoj metode.



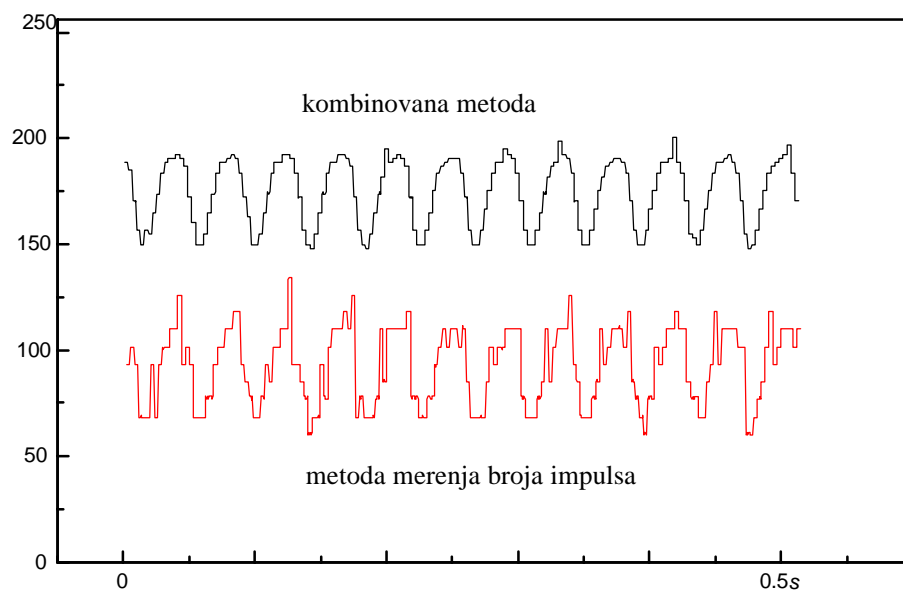
**Slika. 6.1.** Osciloskopski snimak brzina merenih enkoderom (očitana vrednost brzine sa ekrana je 50 ob/min)



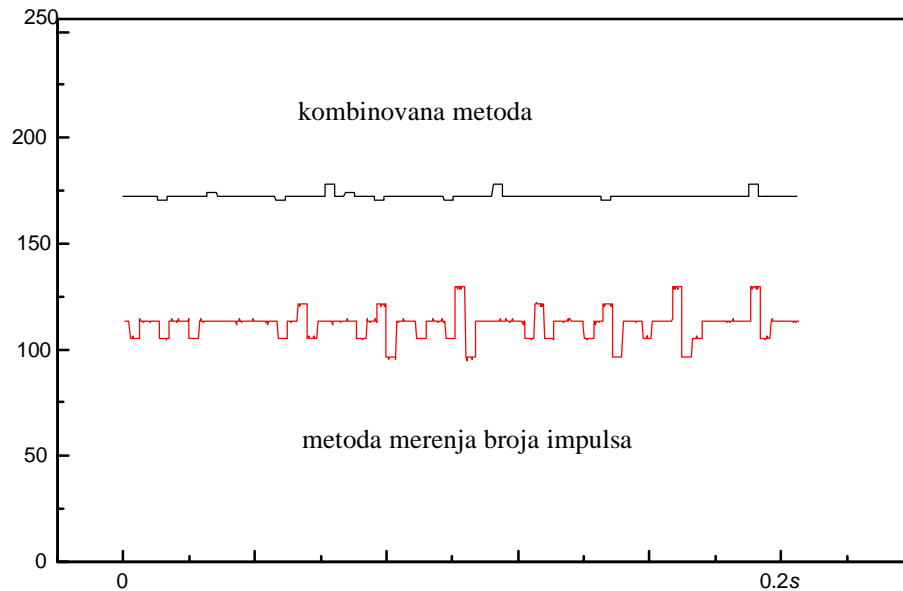
**Slika. 6.2.** Osciloskopski snimak brzina simuliranih ton generatorom (očitana vrednost brzine sa ekrana je 12 ob/min) podešena frekvencija impulsa na ton generatoru je 500 Hz



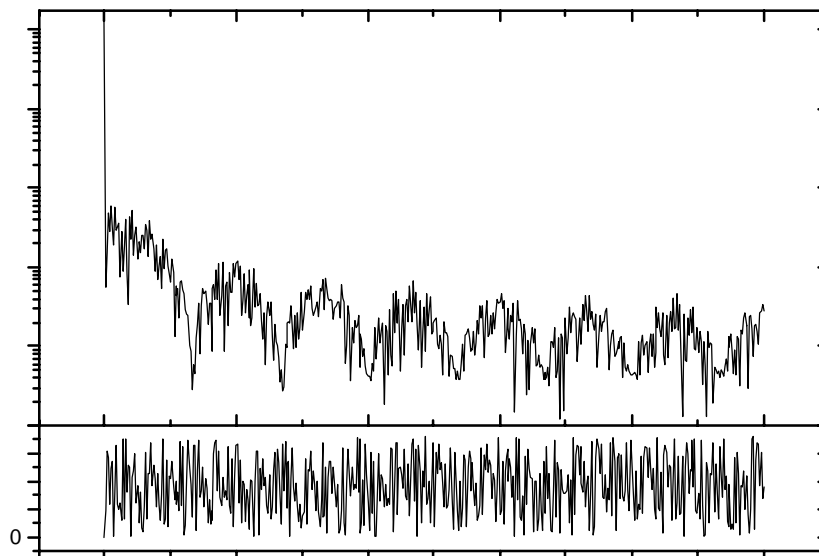
**Slika. 6.3.** Osciloskopski snimak brzina merenih enkoderom (očitana vrednost brzine sa ekrana je 1350 ob/min)



**Slika. 6.4.** Osciloskopski snimak brzina merenih enkoderom (očitana vrednost brzine sa ekrana je 1400 ob/min)



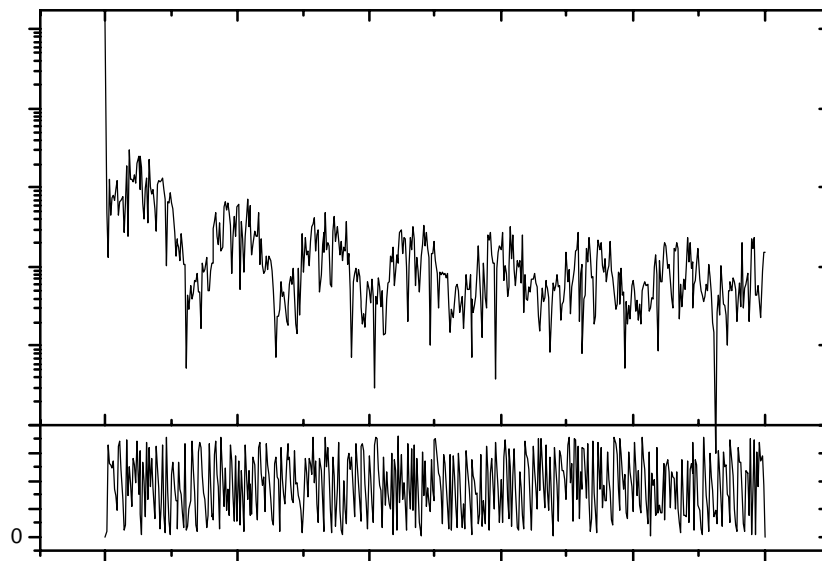
**Slika. 6.5.** Osciloskopski snimak brzina simuliranih ton generatorom (očitana vrednost brzine sa ekrana je 1412 ob/min) podešena frekvencija impulsa na ton generatoru je 27 KHz



**Slika 6.6.** Brza Furijeova transformacija izvršena za slučaj prikazan na slici 6.5.



pri izraèunavanju brzine kombinovanom metodom



**Slika 6.7.** Brza Furijeova transformacija izvršena za sluèaj prikazan na slici 6.5. pri izraèunavanju brzine metodom brojanja impulsa u intervalju odabiranja

## 7. ZAKLJUÈAK

Cilj ovog diplomskog rada, bio je da detaljno prikaže jedan od naèina za merenje brzine kombinovanom metodom i praktièno pokaže prednosti ovog postupka u odnosu na druge naèine merenja brzine. Izrada ovog rada poèela je detaljnim teoretskim razmatranjima iznetim u drugom poglavlju. Na prototipskoj kartici, koja je veæ postojala u raèunaru nalazio se Programabilni tajmer 8253, koji je sa svoja tri brojaèa predstavljao dobru hadversku osnovu za realizaciju ovakvog merenja brzine. Na spoljnim karticama laboratorijskog modela pogona, veæ je postojalo kolo za uslozljavanje signala koje je bilo potrebno da se prilagodi tajmeru. Za oèitavanje vrednosti signala koji sa enkodera, preko katicice za obradu dolaze do tajmera napisan, je asemblerski program èiji je listing dat u petom poglavlju. Korišæenjem ovako izraèunate vrednosti, preko Paralelnog Programabilnog Interfejsa, informacije o brzini poslate su na D/A konvertor. Merenjem napona na izlazu iz D/A konvertora, dobijeni su talasni osciloskopski snimci prikazani u šestom poglavlju, koji su potvrdili polazne pretpostavke.

Na ovim snimcima jasno se vide prednosti merenja brzine kombinovanom metodom, kod sluèajeva kada je u vrlo kratkom periodu odabiranja potrebno da se raspolaže preciznom informacijom o brzini. Kod brzine izraèunate kombinovanom metodom signali su ujednaèeniji, a uticaj viših harmonika i energija šuma je manja

Ovakav naèin merenja brzine ima primenu u vektorski kontrolisanim pogonima gde se pri odreðivanju orijentacije rotorskog fluksa javlja potreba za što taènijim i bržim odreðivanjem brzine, jer taènost utièe na regulaciju fluksa i momenta.

Pri izradi ovog rada korišæeno je udvostruèavanje signala koji dolaze sa enkodera. U drugom poglavlju pokazano je da ovi signali mogu da se uèetvorostruèe. Ovim bi se dobila još taènija informacija o brzini pa bi i eksperimentalni rezultati bili još bolji.

### Literatura:

1. Vukosaviæ, S., "Projektovanje adaptivnog mikroprocesorskog upravljanja brzinom i pozicijom asinhronog motora", *Doktorska disertacija, Elektrotehnièki fakultet, Beograd, 1989.*
2. Stojiæ, R. M., "Digitalni sistemi upravljanja", *Nauèna knjiga, Beograd, 1989.*
3. Tešić, L.J. S., Vasiljeviæ, M. D., "Osnovi Elektronike", *Nauèna knjiga, Beograd, 1990.*
4. Norton, P., "Inside the IBM PC", *Prentice Hall Press, New York, 1986.*
5. Stojèev, K.M., "Savremeni 16-bitni mikroprocesori I", *Nauèna knjiga, Beograd, 1990.*
6. Dinia, È., "PC/ROM BIOS - prekidne rutine sa primerima u DEBUG-u", *Tehnièka knjiga, Beograd, 1992.*

7. *Medenica, G., "Izrada hardvera i softvera laboratorijskog radnog mesta za ispitivanje vektorske regulacije asinhronog motora", Diplomski rad, Elektrotehnički fakultet, Beograd, 1995.*