Univerza v Ljubljani
Fakulteta za elektrotehniko

Boris Vidrgar

Digitalni funkcijski generator

Seminarska naloga

pri predmetu
Elektronska vezja

V Ljubljani, maj 2002

VSEBINA

1 UVOD

Izdelal sem digitalni funkcijski generator z dvema neodvisnima/odvisnima analognima kanaloma in s priključitvijo na PC, kateri vsebuje programski vmesnik za nastavlanje vseh parametrov izhodnih signalov.

Vsak kanal ima možnost nastavitve:

- amplitude,
- offseta,
- frekvence,
- avtomatske spreminjanje frekvence (sweep),
- fazne zakasniteve med signaloma,
- poljubne oblike izhodnega signala.

1.1 Motivacija

Tržišče ponuja veliko funkcijskih generatorjih. Nekatere lahko izdelamo iz že obstoječih IC vezij, ki potrebujejo samo nekaj zunanjih elmentov in napajanje. Lahko si sestavimo relaksacijski oscilator, ki nam do dajal pravokotne impulze. Sevada se lahko zatečemo k generatoru trikotnih signalov, kateri že vsebuje tudi pravokotne impulze in z malo nadgradnje dobimo rahlo popačen sinus. Seveda si lahko kupimo funkcijsku generator, ti se že večinom digitalni, kateri nam ponujajo poleg standardnih signalov (sinus, trikot, žaga, pravokot, šum....) tudi poljubne signale, katere mu preko posebne kartice ponudimo na izbor. Seveda imajo na voljo še morje dodatnih funkcij(amplituda, offset, sweep, AM in FM modulacija....).

Že večkrat sem si hotel narediti funkcijski generator, pa niso bili uspehi kaj prida, saj nikoli nisem bil zadovoljen samo z osnovnimi signali....hotel sem nekaj več, kak exponentni signal naprimer.

Tako sem zagrabil priložnost, ki se je ponudila pri Tekmovanju v načrtovanju digitalnih vezij. Na voljo sem dobil močno programabilno digitalno vezje Spartan. Sedaj lahko sam naredim in se seznanim s problemi, ki nastanejo med snovanjem digitalnega funkcijskega generatorja, kje so omejitve in kako rešiti morebitne težave. Seveda pa je tu tudi analogen del vezja, ki pa je spet problem zase.

1.2 Funkcionalni opis vezja

Digitalni funkcijski generator je priklopljen na PC preko LPT porta in komunicira po protokolu EPP1.9. S programskim vmesnikom lahko nastavljamo naslednje parametre izhodnima kanaloma:

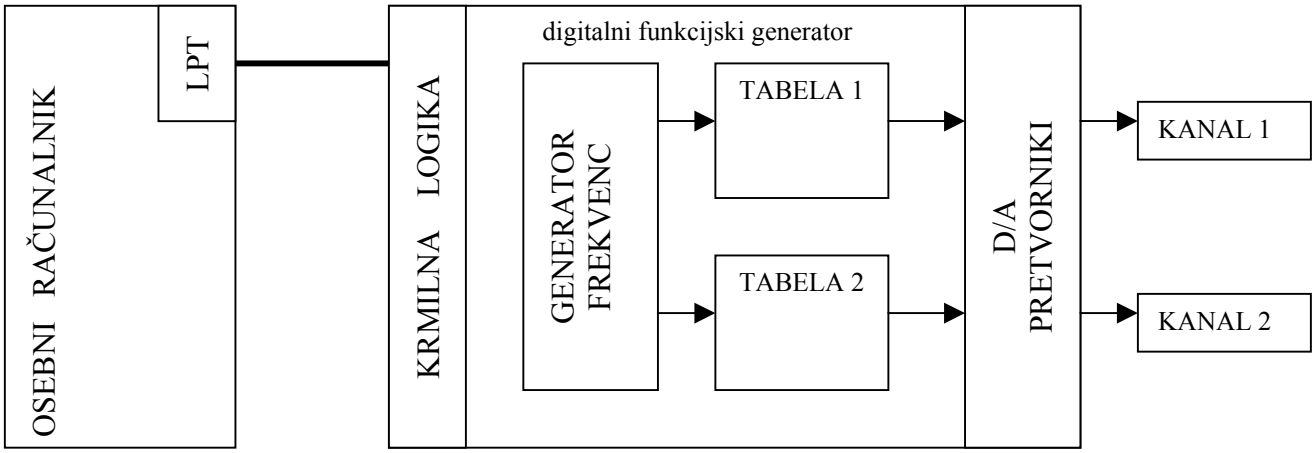
- amplitudo od 0V po 0.05V do 12.75V,
- offset od –12.7V po 0.1V do 12.7V,
- frekvenco od 2.384Hz do 156.250KHz po 2.384Hz korakih,
- avtomatsko spreminjanje frekvence (sweep) s hitrostjo od 2.384Hz/1.638ms do 2.35Hz/25ns po 2.384Hz/1.638ms korakih,
- fazno zakasnitev med signaloma od 0° po 1.412° do 360° (brez obračanja*) in 0° po 5.647° do 360° (z obračanjem*),
- poljubno obliko signala z max. 256 vzorci/periodo (brez obračanja*) in 1024 vzorci/periodo (z obračanjem*)

* razloženo kasneje

2 GLAVNI DEL

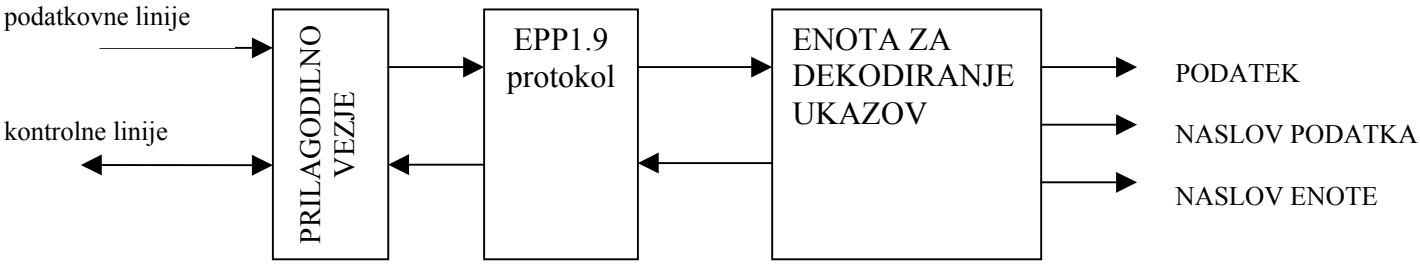
2.1 Opis delovanja podsklopov vezja s shematskim prikazom podsklopov

Digitalni funkcijski generator ni samostojno vezje. Ko sprejme vse potrebne podatke od PC-ja pa postane samostojno vezje.
Grobo predstavo prikazuje naslednja slika:

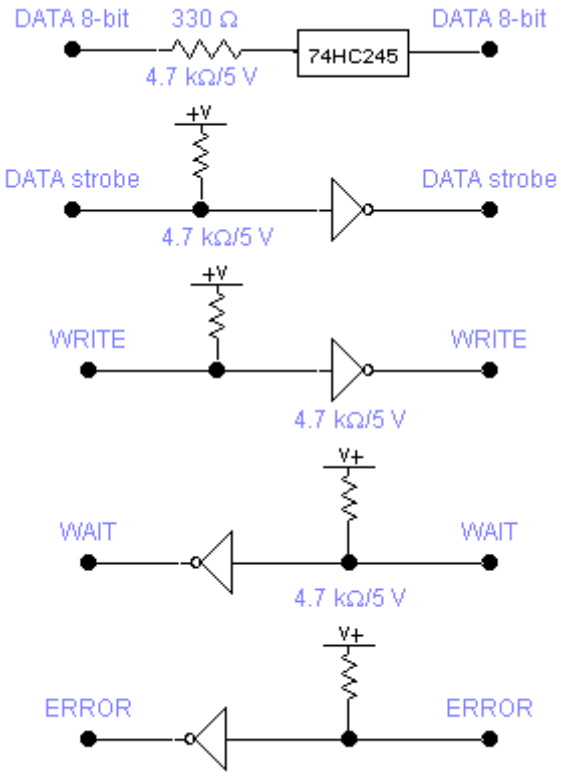


Osebni računalnik skrbi za prenos podatkov od uporabnika do digitalnega funkcijskega generatorja-DFG na čimbolj preprost in enostaven način za uporabnika.

Krmilna logika je razdeljena na več podsklopov, kot prikazuje spodnja slika:



Prilagodilno vezje skrbi za prilagoditev nivojev in zaščito. Sestavljen je iz line driverja 74HC245, 4-ih negatorjev, ter uporov.
Shemo prikazuje spodnja slika:



EPP1.9 protokol je programsko implementiran v FPGA vezje. Ta protokol omogoča vmesniku na PC-ju da kontrolira, če je DFG uspešno prejel podatek. Po protokolu EPP1.9 mora periferija v manj kot 10µs odgovoriti PC-ju (po protokolu to stori s signalom WAIT), da je podatek sprejela, drugače vmesnik signalizira Time Out bit v statusnem registru paralelnega vmesnika (EPP 1.7 te opcije nima). Program, ki teče na PC-ju lahko s tem ugotovi, če je komunikacija z napravo dobra.

Enota za dekodiranje ukazov je prav tako implementirana v FPGA vezje. Izhodi, ki so povezani na notranje vodilo, so: podatkovni (16-bitni), podatkovno naslovni (8-bitni) s katerim naslavljamo lokacije tabele (katera vsebuje potek signala) in še eno naslovno, s katerim pa naslovimo željeno enoto.
Enota za dekodiranje ukazov predstavla sekvenčno vezje, kateremu stanje se spremeni ob uspešno sprejetim podatkom. Katero stanje bo naslednje pa odloča sprejeti podatek. Prehode prikazuje diagram stanj na sliki, ki se nahaja na zadnjih straneh. En cikel predstavlja uspešno sprejet ukaz, podatk ter naslov enote.
Za prehod iz stanja READY v ERROR je potrebn napačen ukaz, za prehod nazaj pa kateri koli podatek.
Za vpisovanje poteka signala v tabelo pričnemo cikel z ukazom $90, nadaljujemo z addresso nato podatkom in z naslovom enote. Če je naslov napačen (če ni 01 ali 02), sledi stanje ERROR, drugače READY.
Za naslavanje parametrov generatorja frekvenc, ki so 16-bitni, uporabimo ukaz $C0, kateremu sledi zgornji del podatka nato spodnji del in naslov enote. Vsi naslovi so prikazani v tabeli 2.1. Če je naslov napačen, sledi stanje ERROR.

Zadnji ukaz je za vpisovanje amplitude, offseta in kontrolnega registra, kateri so vsi 8-bitni. Ukazu $A0 sledi podatak in naslov enote. Stanje ERROR sledi ob nepravilno poslanem naslovu.

Spodnja tabela prikazuje zaporedje pošiljanja podatkov za posamezen ukaz:

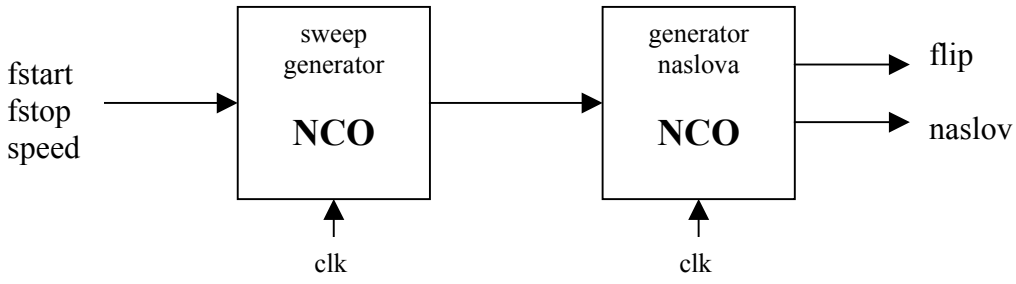
ukaz_1 = $90	addressa 8-bit	data 8-bit	naslov
ukaz_2 = $A0	data 8-bit	naslov	
uakz_3 = $C0	data high	data low	naslov

Spodnja tabela prikazuje s katerimi ukazi naslavljamo enote in kakšen je naslov enot:

UKAZ	ENOTA	NASLOV
ukaz_1 = $90	block 1	$01
	block 2	$02
ukaz_2 = $A0	ampl 1	$03
	ampl 2	$04
	offset 1	$05
	offset 2	$06
	faza	$07
	kontrol	$08
ukaz_3 = $C0	fstart1	$09
	fstop1	$0A
	speed 1	$0B
	fstart 2	$0C
	fstop 2	$0D
	speed 2	$0E

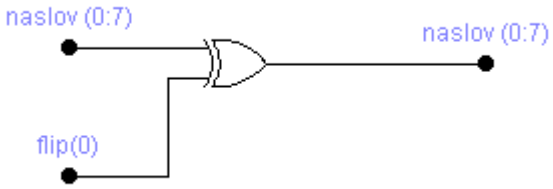
tabela 2.1

Generator frekvenc predstavlja dva ločena dela. Dva generatorja frekvenc vsak za svoj kanal. Oba sta zgrajena na istem principu, s to razliko, da ima drugi dva načina delovanja: “free running”delovanje, deluje kot prvi in fazno sklenjeno delovanje, ki je pa sledenje prvemu, vendar z določeno fazno zakasnitvijo. V osnovi sta dva NCO-ja (numerical controlled oscilator). Eden opravlja funkcijo sweep generatorja, drugi pa generator naslova za podatkovno tabelo. Blok shemo prikazuje spodnja slika:



Sweep generator je 32-bitni NCO, kateri poveča svojo vrednost za speed (če je to omogočeno v kontrolnem registru) ob vsakem clk impulzu. Ko preseže vrednost fstop na zgornjih 16-bitih se vrne na vrednost fstart, katera se vpiše na zgornjih 16-bitov in tako nadaljuje naprej. Izhod je zgornjih 16 bitov, kateri predstavlajo "konstanto" za generiranje frekvenc. Če je zahteva za sweep onemogočena je 16-bitni izhod preprosto fstart.

Generator naslova predstavlja 26-bitni NCO, kateremu se vrednost ravno tako poveča ob vsakem clk impulzu za vhodno 16-bitno besedo. Zgornja dva bita sta izhod flip (katera hranita podatek za obračanje) naslednjih 8-bitov pa je naslovni izhod.
Generatorju naslova je dodana še rutina za obračanje signala. Namreč, če želimo sestaviti sinus in samo ¼ periode, kar nam omogoči 1024-bitno resolucijo, moramo signal, ki je podan v tabeli, enkrat brati iz začetka drugič iz konca tabele. Princip prikazuje spodnja simbolična slika:



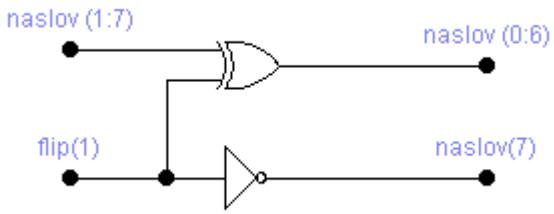
Seveda pa je potrebno tudi sam signal po amplitudi obračati, kar je pa obrazloženo v nadaljevanju.
Enačba za izhodno frekvenco signala:

$$frek = \frac{clk}{256} \times \frac{beseda}{65536}$$

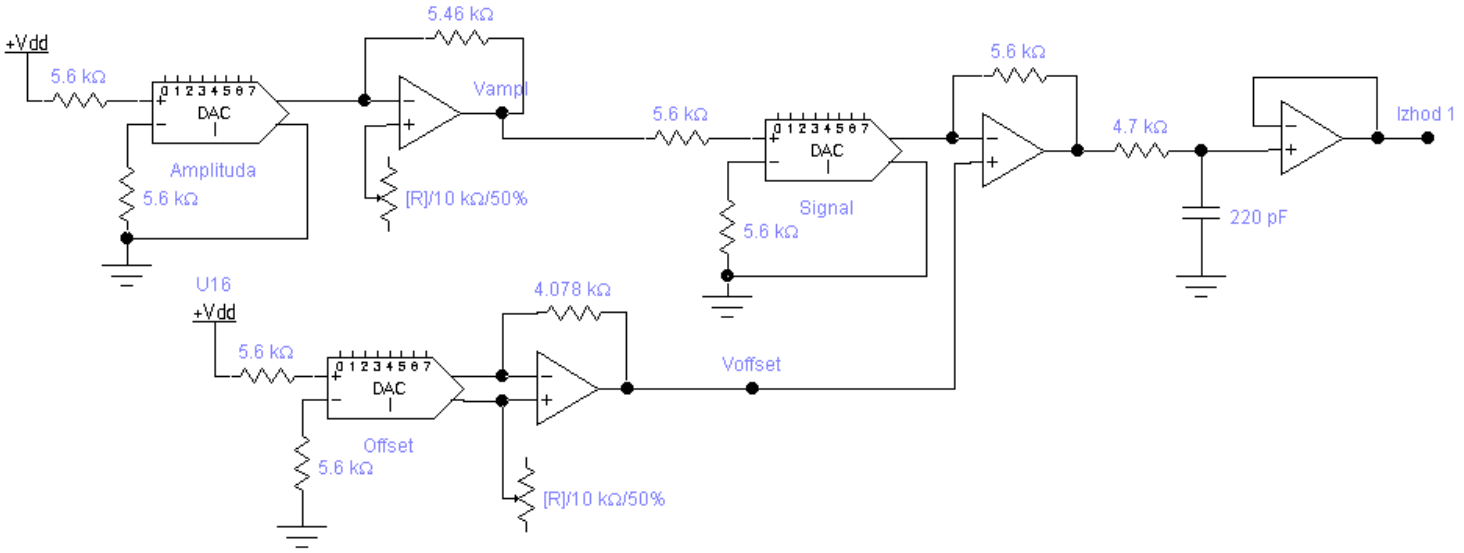
Pri čemer je clk=40MHz, beseda je 16-bitna (fstart, če je sweep onemogočen). Če pa signal obračamo je frek=frek/4. Korak spreminjanja fekvence je Δf=2.384 Hz.
Enačba za hitrost spreminjanja frekvence (sweep):

$$\frac{\Delta f}{\Delta t} = clk \times \frac{speed}{65536} \times 2.384 Hz \left[\frac{Hz}{s}\right]$$

TABLEA je dual port ram, kateri omogoča hkratno vpisovanje in izpisovanje signala iz različnih naslovov. Obe tabeli sta enaki. Imata 256 8-bitnih lokacij katere so direktno povezane na naslov iz generatorja naslova. Posebnost predstavlja dodatek, ki skrbi za obračanje signala kadar je to omogočeno. Ta mora signal prestaviti za neko enosmerno vrednost in odvisno katero ¼ periode opisujemo signal prišteti/odšteti tej enosmerni vrednosti. Princip prikazuje spodnja simbolična slika:



D/A pretvorniki predstavlajo edini analogni del vezja. Skrbijo za pretvorbo digitalnih veličin v analogne. Sestavljeni so iz dve identičnih delov, vsak za svoj kanal. Vezava je prikazana na spodnji sibolični sliki:



Podkrepimo z enačbami:

Vdd = 17.5V
5.46K = 5.6K || 220K
4.078K = 5.6K || 15K
DAC0800LCN

$$V_{ampl} = \frac{V_{dd}}{5.6K} \times 5.46K \times \frac{beseda}{256}\Big|_{beseda=256} \cong 12.75V$$

$$V_{offset}* = \frac{V_{dd}}{5.6K} \times 4.078K \times (-\frac{255}{256} + \frac{2 \times beseda}{256})\Big|_{beseda=256} \cong 12.7V$$

$$V_{izhoda} = \frac{V_{ampl}}{5.6K} \times 5.6K \times \frac{beseda}{256}$$

Potencialometri služijo za nastavitev ničelne amplitude in offseta (in izničitev neničelnih veličin operacijskih ojačevalnikov). Izhodnji NF filter skrbi za dušitev visoko frekvenčnih konic, ki se pojavijo ob hitrih prehodih, frekvence 1.61MHz. Največja izhodna frekvenca pa je 156.250KHz, kar pomeni da imamo na voljo samo eno dekado, da konice potrebno zmanjšamo.Potrebovali bi filter višjega reda, da bi uspešno zadušili konice. A uporabil sem filtr prvega reda s mejno frekvenco približno 156KHz.

2.1 Analiza delovanja

Nad generatorjem frekvence bedi kontrolni register, katerega vpisujemo preko ukaza za 8-bitno vpisovanje podatkov. Spodnja tabela prikazuje pomen bitov v kontrolnem registu:

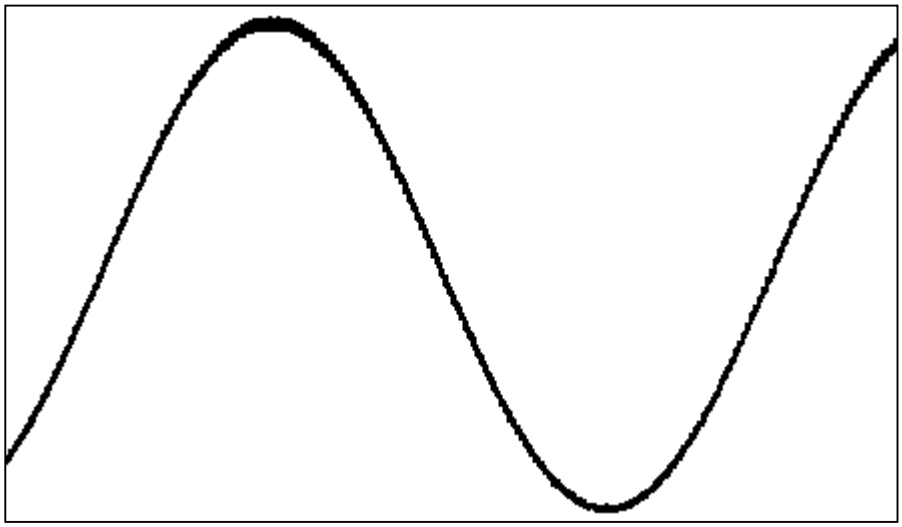
BIT	pomen bita v kontrolnem registru
0	1: obračanje ON (za prvi kanal)
1	not used
2	0: sweep OFF (za prvi kanal)
3	0:faza × 1, 1: faza × 4
4	1:obračanje ON (za drugi kanal)
5	not used
6	0: sweep OFF (za drugi kanal)
7	1: faza ON

Obračanje pomeni, da iz ¼ periode sestavimo celotno periodo, kar lahno nastavimo za vsak kanal posebej. Problem je, ker če je samo en kanal nastavljen na obračanje, frekvenca signala ni več enaka in to je potrebno popraviti z novim vpisom v fstart.
Bit 3 je potreben, ker ko določujemo fazno zakasnitev med dvema signaloma, bi radi dosegli zakasnitev do 360°. Vendar če signal zakasnimo s samo 8-bitnim podatkom pri obračanju, nikakor ne moremo doseči do 360° zakasnitve. V tem primeru je potrebno fazno zakasnitev pomnožiti s 4.
Faza ON pa preprosto omogoči fazno sklenitev obeh signalov z določeno fazno zakasnitvijo.

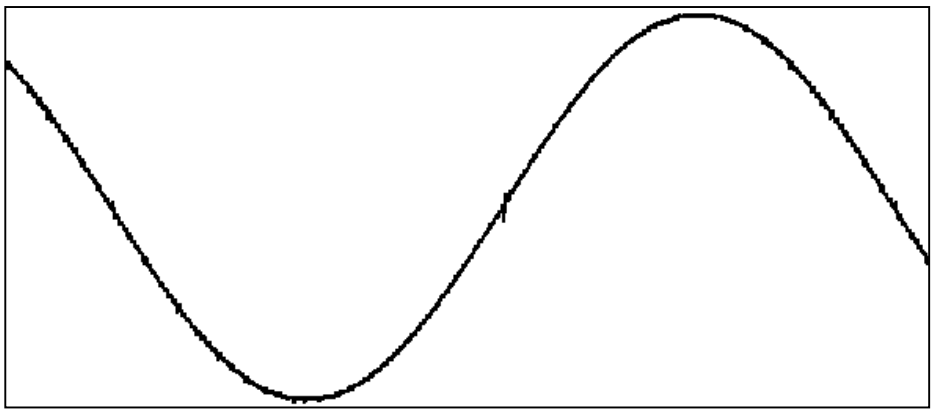
Problem se pojavi tudi pri obračanju signala. Ko je signal (sinus) na prehodu iz ene v drugo ¼ periode se na prehodu pojavita dva enaka podatka za dva × 256/freq. Ampak je amplituda signala spremenjena zaradi preoblikovalnika signala. Torej dobimo namesto npr. dveh 0 (pri prehodu iz druge v tretjo ¼ periode) enkrat 128 in drugič 127. Srednja vrednost signala je tako med 128 in 127. Pri maksimalni amplitudi je negotovost 25mV.

2.2 Izmerjene karakteristike

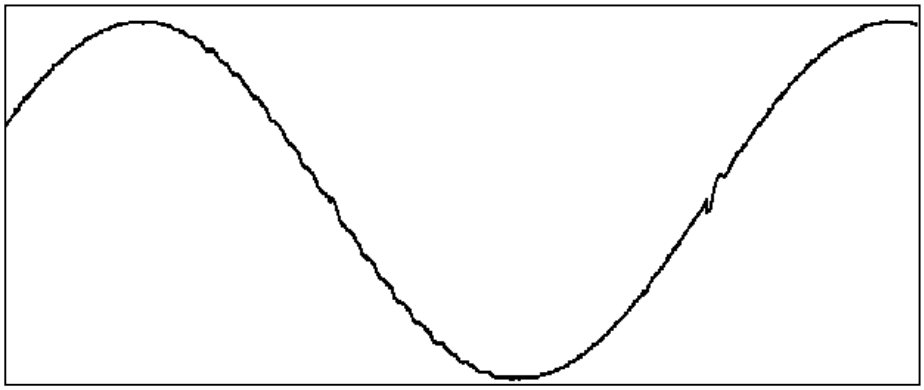
Izmeril sem potek sinusnega signala pri treh frekvencah:
- pri 0.6Hz



- pri 10KHz



- pri 39KHz



Vsi signali so enake amplitude. A pri veliki frekvenci, se pojavijo rahle oscilacije.

2.3 Navodila za uporabo

Zaporedja ukazov in delovanje je bilo razloženo v opisu vezja.
Pred uporabo je potrebno napravo umeriti. Amplitudo signala postavimo na 0 in nastavimo potencialometer (bolje je če uporabimo multiturn nastavljiv upor) pri amplitudnem O.O. tako, da je na izhodu O.O. 0V.
Ravno tako naredimo za offsetni del, s to razliko, da je 0V offseta pri 128 oziroma 127 besedi, pravzaprav nekje umes. Katero si izberemo za ničlo, je stvar okusa, v vsakem primeru bo napaka enaka pri obračanju signala.

3 ZAKLJUČEK

3.1 Morebitne težave

Največja težava bi bila, da se pojaviljao prenihaji ob hitrih preskokih, prenihaji so pa tudi do nekaj 2 V, odvisno od amplitude signala. Za to rešitev, bi bila seveda potrebna uporaba filtra višjega reda (2. red bi zadostoval), da zaduši prenihaje,ki so frekvence 16.1MHz.
Težava so tudi upori v D/A enoti, katerim vrednost odstopa in je tako nemogoče določiti točno vrednost izhodne amplitude ali offseta. Problem bi bil seveda rešljiv z uporabo bolj naračnih uporov, kar bi tudi zadostovalo.
Ena od negativnih lastnosti NCO-ja je jitter. Negotovost periode željenje frekvence je enaka eni periodi taktne frekvence NCO-ja. V konkretnem primeru je to ± 12.5ns. Kar sicer ni opazno, vendar je prisotno. Dalo bi se to rešiti, vendar bi v tem primeru dobili konstanten premik od željene frekvence za en takt NCO-ja. Ampak mogoče bi se s kompleksnejšim algoritmom dalo tudi temu izogniti.

3.2 Sklepne ugotovitve

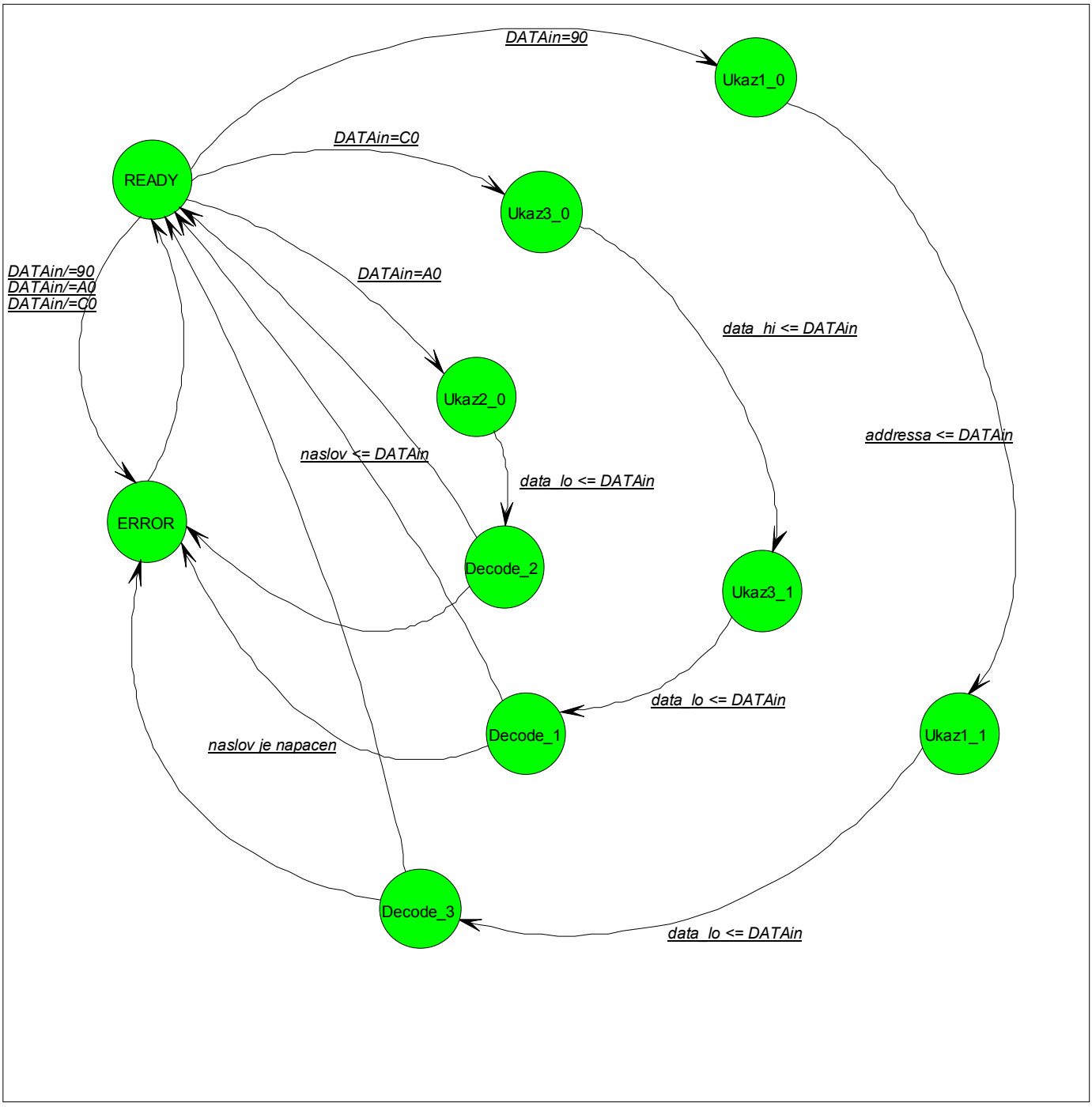
Digitalni funkcijski generator je kar zajeten izziv, če ga želiš narediti do profesionalnega nivoja. Seveda bi potreboval še nekaj ostrega pogleda, vendar v osnovi je kar dobro narejen. Pa tudi z dobrim PC-vmesnikom, bi se začel zgledal že kar solidno.

3.3 Možnosti nadgradnje

Dalo bi se uporabiti boljše DAC-e, ki bi imeli boljši odzivni čas (kateri so uporabljeni v vezju komaj spremljajo hitre preskoke pri višjih frekvencah).
Komunikacija je samo enosmerna, kar bi se dalo tudi popraviti v dvosmerno. Tako bi pa lahko popravili komunikacijo med notranjimi enotami, uporabili bi 16-bitno vodilo in tako bi tudi odpadel sekvenčni avtomat na kontrolni logiki. Vse bi opravljal protokol. Hitrost prenosa podatkov bi se močno povečala.
Kot sem že omenil, bi se dalo programsko rešiti jitter NCO-ja.

Potem pa se najdejo ideje, kako bi se dalo z razširtvijo narediti mostični RLC meter. Saj na voljo sta dva signala medsebojnega faznega zasuka in poljubne frekvence.

4 SLIKE


DATAin=90
Ukaz1_0
DATAin=C0
READY
Ukaz3_0
DATAin/=90
DATAin/=A0
DATAin/=C0
DATAin=A0
data_hi <= DATAin
Ukaz2_0
addressa <= DATAin
naslov <= DATAin
data_lo <= DATAin
ERROR
Decode_2
Ukaz3_1
data_lo <= DATAin
Decode_1
Ukaz1_1
naslov je napacen
Decode_3
data_lo <= DATAin

Reference:

V prilagodilnem vezju sem uporabil integrirani vezji:

- 74HC245 line driver
  http://www.philipslogic.com/products/fast/pdf/74f245.pdf

- 74HC00 nand logična vrata
  http://www.philipslogic.com/products/hc/pdf/74hc00.pdf

Krmilna logika, generator frekvenc in tabeli so implementirani v FPGA vezje, ki ga predstavlja Xilinx-ov Spartan-II XC2S100
http://direct.xilinx.com/partinfo/ds001_2.pdf

V zadnji enoti D/A pretvornik sem uporabil:

- DAC0800 D/A pretvornik
  http://www.national.com/ds/DA/DAC0800.pdf

- TL084 operacijske ojačevalnike
  http://www-s.ti.com/sc/ds/tl084.pdf