

POROČILO DIVS

Generator tonov, intervalov in akordov

Matej Langus

Uvod

Ko sem še hodil v glasbeno šolo sem imel največ težav pri slušnem razpoznavanju akordov in intervalov. Zato sem se za projekt odločil izdelati generator tonov, intervalov in akordov. Za ta namen sem nadgradil projekt oscilatorja, ki smo ga izdelali pri laboratorijskih vajah. Generator se lahko uporablja, kot generator vnesenih tonov, intervalov in akordov ali pa kot naključni generator akordov in intervalov za preverjanje slušnega razpoznavanja.

Opis komponente

Za generator sem predelal komponento PID regulatorja, ki je bila že del Red Pitaye. Moji komponenti lahko prek vodila nastavimo 6 parametrov:

- številko višine tona (od C0 do E8)
- velikost intervala (od prime naprej)
- tip akorda (dur, mol, zvečan, zmanjšan)
- način delovanja (akord, interval, ton s prekinitvami, ton brez prekinitve)
- tempo predvajanja (0.1s do 10s na ton)
- ojačanje izhoda (1-5)

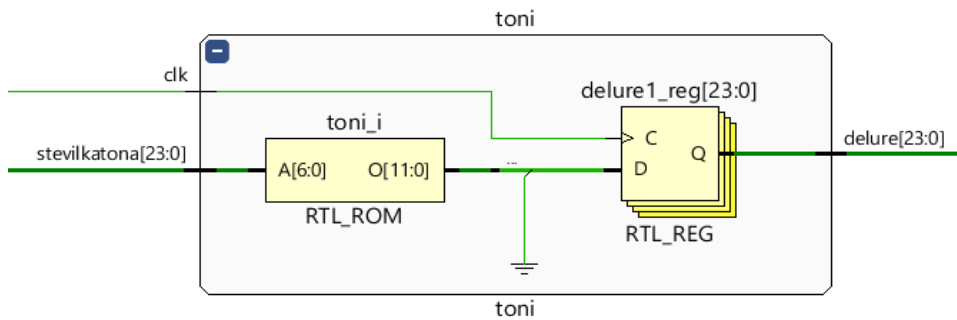
Načrtovanje po komponentah

Ker je bil moj namen, da so frekvence tonov čim bolj pravilne, sem izdelal komponento **toni** v system verilogu, ki podpira 2D array, v katerega sem v (**Zaporedne številke tonov**) shranil preračunane vrednosti (**Zaokroženih faktorjev delitve ure**) s katerimi je potrebno deliti sistemsko uro na Redpitayi (125MHz), da na izhodu sinusnega generatorja dobimo, čim bolj točno frekvenco. Za izračun faktorjev sem uporabil formulo $\frac{125000000}{\text{Frekvenca} * 1024 * 2} - 1$, kjer je 125000000 frekvenca na RedPitayi, Frekvenca je frekvenca željenega tona, 1024 faktor, da iz sinusnega generatorja dobimo celotno periodo, 2 pa je faktor zaradi proženja le na prednjo fronto ure.

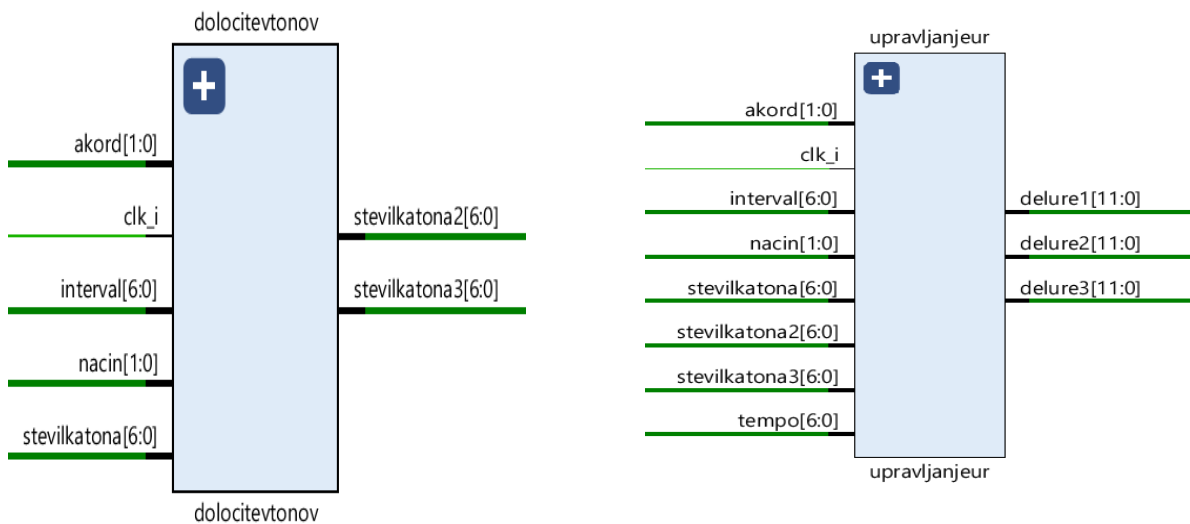
Nota	Frekvenc a (Hz)	Faktor delitve ure $\frac{125000000}{\text{Frekvenca} * 1024 * 2} - 1$	Zaokrožen faktor delitve ure	Zaporedna številka tona	Frekvenca iz zaokroženega faktorja delitve ure	Odstopanje (Hz)	Relativno odstopanje
C ₀	16.35	3732.03708	3732	0	16.3501624	0.000162403	0%
C [#] ₀ /D ^b ₀	17.32	3522.96976	3523	1	17.31985138	-0.000148624	0%
D ₀	18.35	3325.166553	3325	2	18.3509189	0.000918897	0%
D [#] ₀ /E ^b ₀	19.45	3137.054306	3137	3	19.4503366	0.0003366	0%
E ₀	20.6	2961.871663	2962	4	20.59910775	-0.000892254	0%
F ₀	21.83	2794.930199	2795	5	21.82945503	-0.000544975	0%
F [#] ₀ /G ^b ₀	23.12	2638.928904	2639	6	23.11937737	-0.000622633	0%
G ₀	24.5	2490.230867	2490	7	24.50227067	0.002270674	0%
G [#] ₀ /A ^b ₀	25.96	2350.123122	2350	8	25.96135953	0.001359528	0%
A ₀	27.5	2218.460227	2218	9	27.50570358	0.005703583	0%
A [#] ₀ /B ^b ₀	29.14	2093.548945	2094	10	29.13372613	-0.006273866	0%
B ₀	30.87	1976.167355	1976	11	30.87261318	0.002613177	0%
C ₁	32.7	1865.51854	1866	12	32.69156735	-0.008432646	0%

...
C8	4186.01	13.58074784	14	96	4069.010417	-116.9995833	-3%
C#8/Db8	4434.92	12.76240299	13	97	4359.654018	-75.26598214	-2%
D8	4698.63	11.9899899	12	98	4695.012019	-3.617980769	0%
D#8/Eb8	4978.03	11.26090567	11	99	5086.263021	108.2330208	2%
E8	5274.04	10.57275187	11	100	5086.263021	-187.7769792	-4%

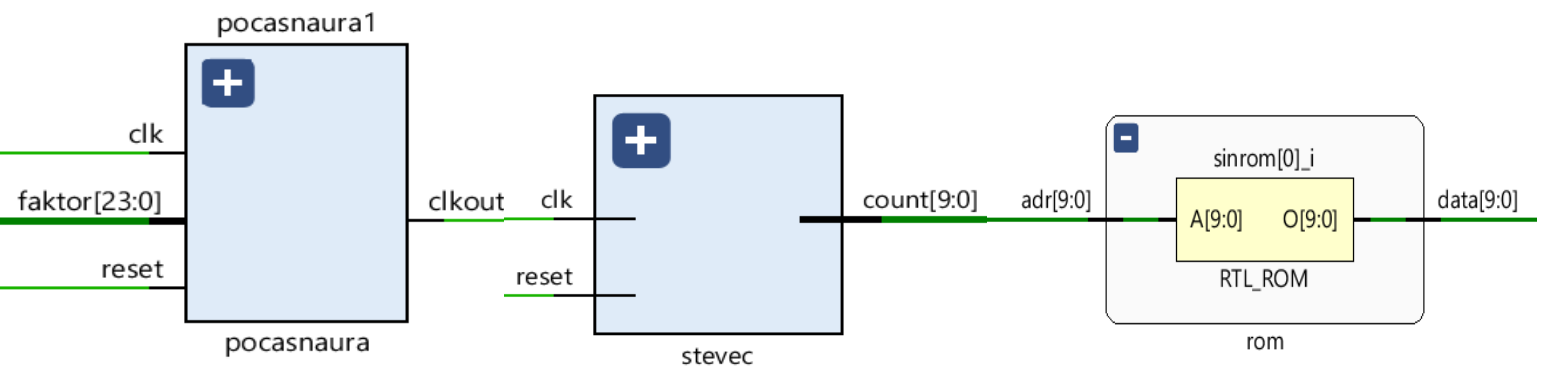
Pri nižjih tonih se frekvenca dobro ujema z željeno, pri višjih pa zaradi zaokrožitve prihaja do vedno večjih napak. Višek napake je pri 99-tem in 100-tem tonu, ki sta zaradi napake kar enaka.



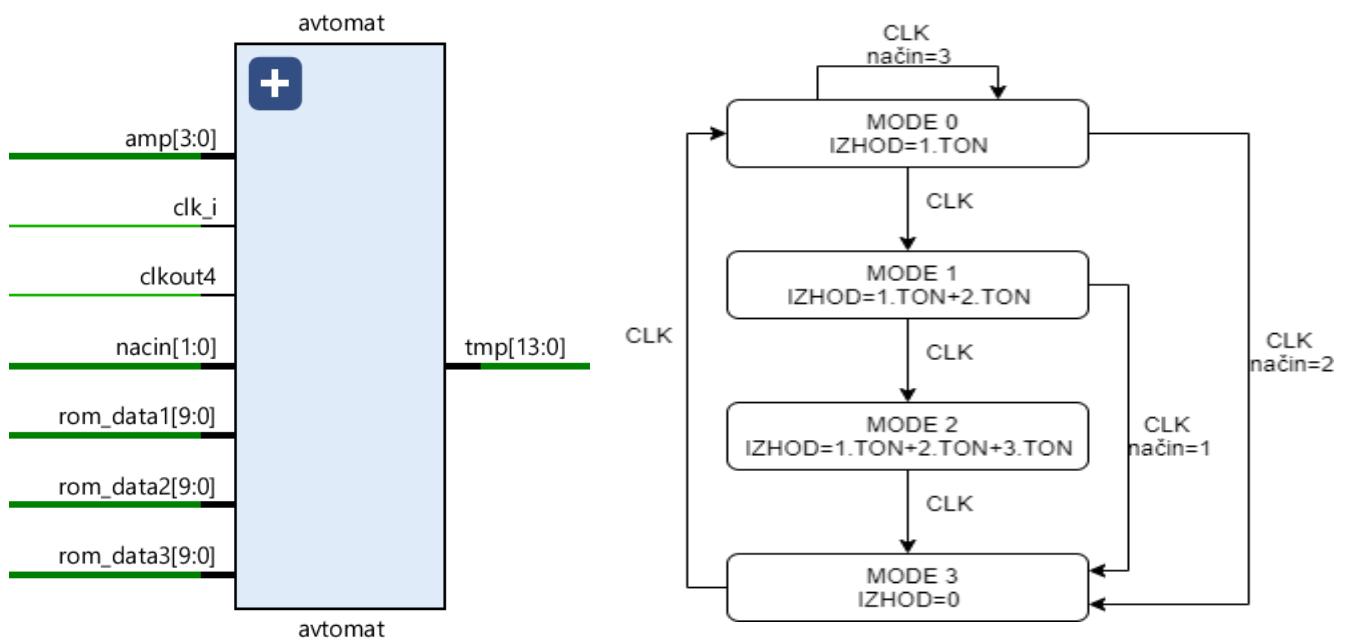
Komponenta **toni**, ki sinhrono za vhodno *stevilkatona* določi faktor deljenja sistemske ure *delure*, je del komponente **upravljanjeur**. Komponenta **upravljanjeur** ob vsaki spremembi akorda, intervala, načina, izhodiščnega tona ali tempa ponovno izračuna faktorje za vse tri tone (poleg izhodiščnega še enega za interval in še dva za akord, ki ju določi komponenta **dolocitevtonov**).



Naslednja komponenta je bila **pocasnaura**, ki iz sistemske ure ustvari za *faktor*, ki ga kot *delure* dobi iz komponente **upravljanjeur**, znižano frekvenco ure. To je realizirano s števcem, ki prešteje do faktorja in invertira izhod. Ta znižana frekvenca krmili 10-bitni števec v komponenti **stevec**, katerega izhod *count* je vezan na *adr* vhod komponente **rom**. Komponenta **rom** generira sinus.



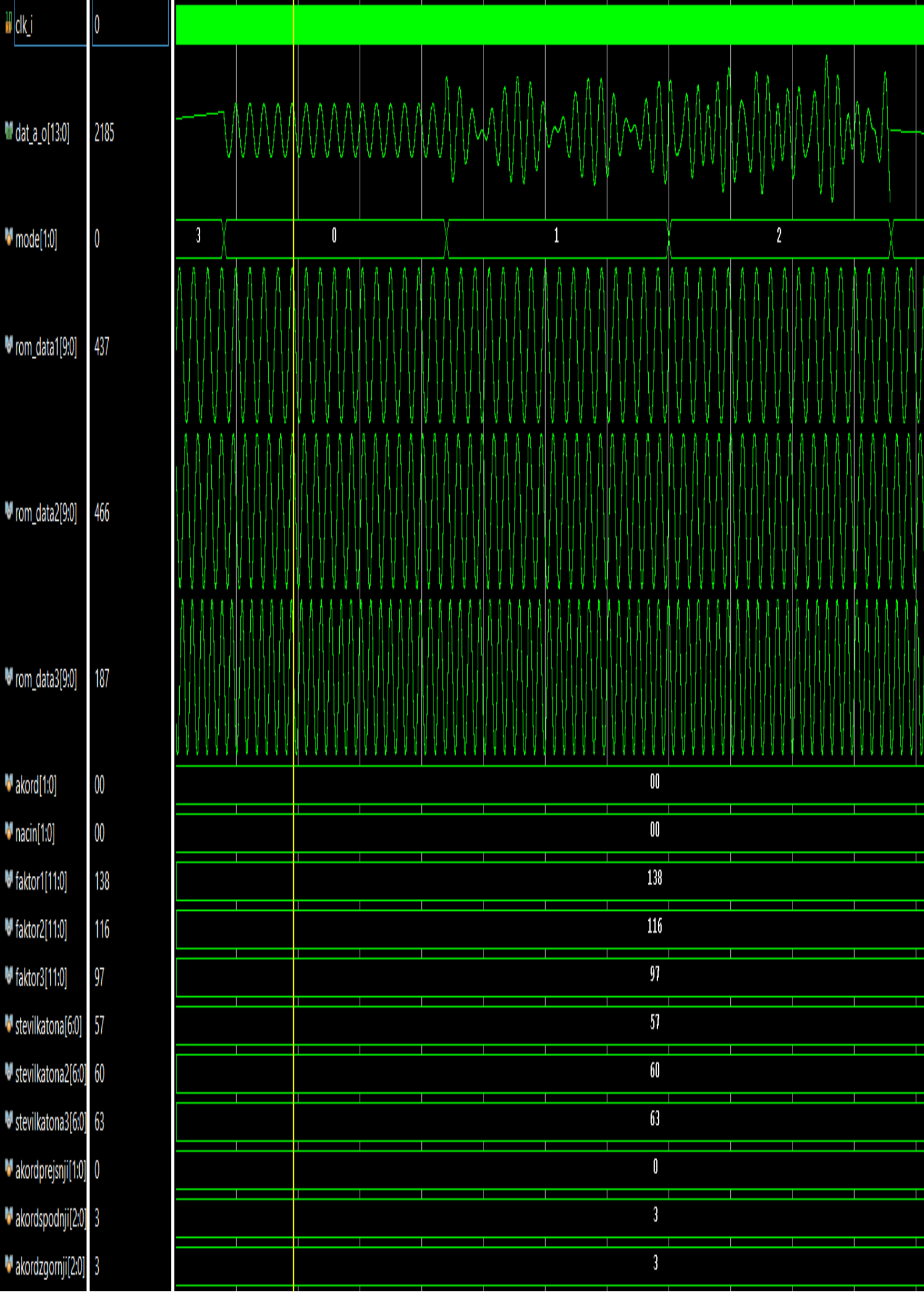
Avtomat je komponenta, ki skrbi za izhod. V odvisnosti od izbranega *načina* (akord, interval, ton s prekinitvami in ton brez prekinitvev) prehaja med štirimi stanji, ki so prikazani na diagramu. Hitrost prehajanja določa četrta komponenta **pocasnaura**, katere frekvenco lahko spreminjamo z vhodnim parametrom *tempo*. Poleg tega avtomat pomnoži izhod še z ojačanjem *amp*, kar določa glasnost.

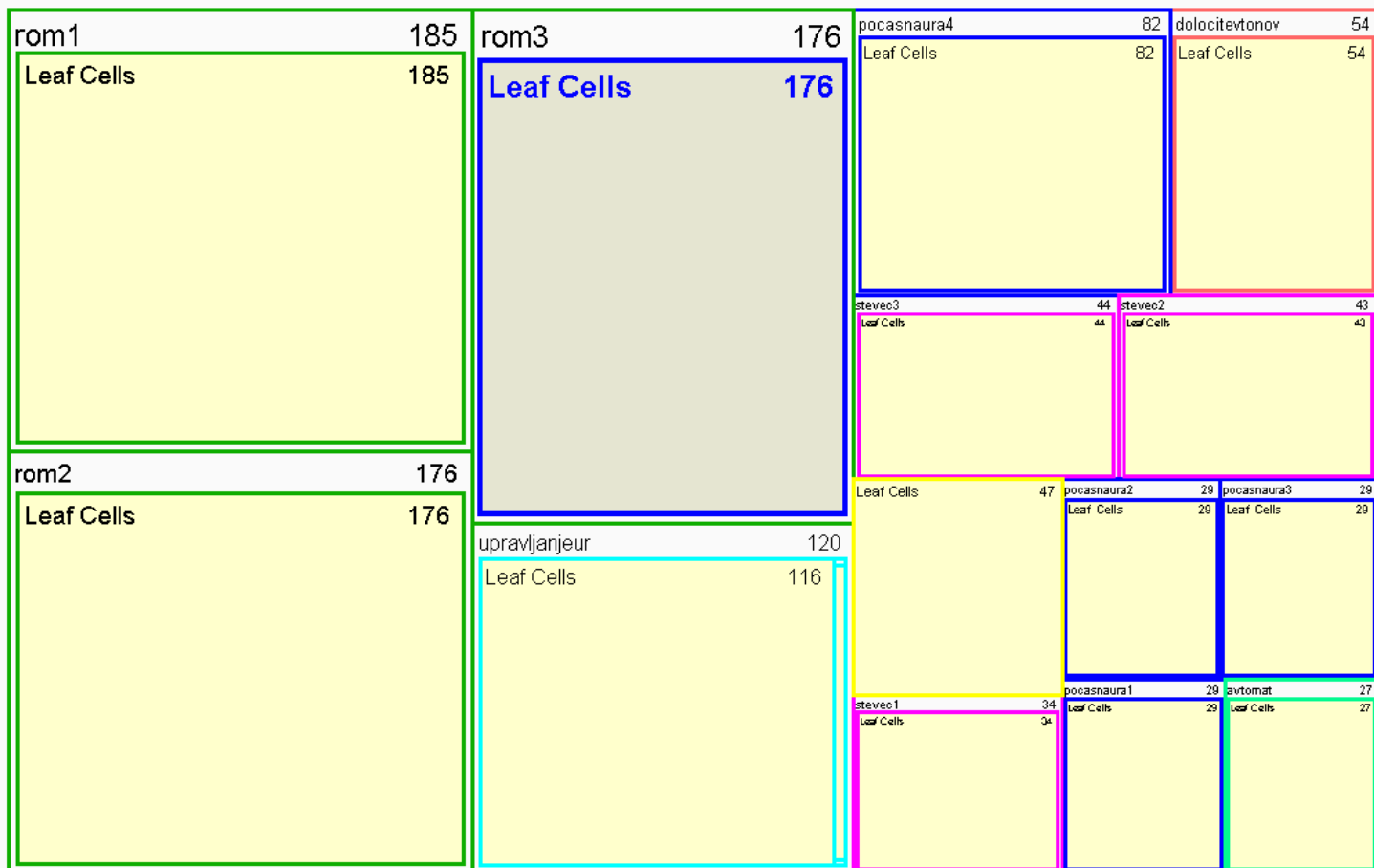


Avtomat prehaja med štirimi možnimi stanji, ko je na izhodu en, dva, trije sinusi ali pa nič. Ob vsakem urnem ciklu se stanje zamenja. V primeru akorda kroži čez vsa štiri stanja, v primeru intervala stanje s tremi sinusi izpusti, v primeru tona s prekinitvama menja med stanji brez signala in stanjem z enim sinusom, v primeru tona brez prekinitvev pa ves čas ostaja v stanju z enim sinusom.

Simulacija

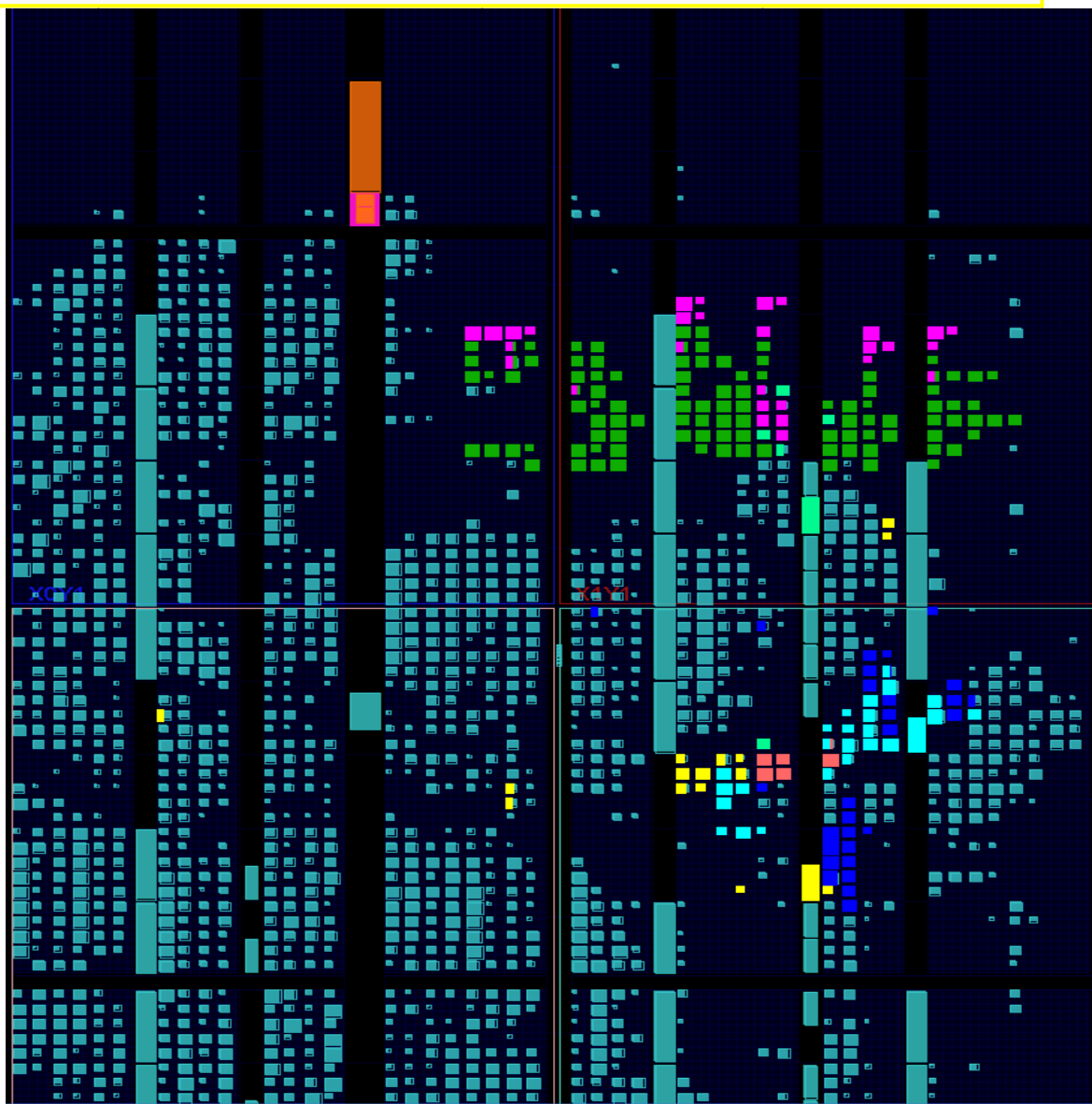
Na simulaciji na naslednji strani vidimo sistemsko uro *clk_i* 125 MHz. Na *dat_a_o* je izhod, ki se spreminja v odvisnosti od signala *mode* (pri 0 en sinus, pri 1 dva sinusa, pri 2 trije sinusi in pri 3 je izhod nič). Sinusi iz katerih je sestavljen izhod so v signalih *rom_data1*, *rom_data2* in *rom_data3*. Signal *akord* je 0, kar pomeni zmanjšan akord, signal *nacin* je 0, kar pomeni generiranje akordov. Signali *faktor1*, *faktor2* in *faktor3* so faktorji za deljenje sistemske ura za generiranje sinusov pravih frekvenc. Signal *stevilkatona* je 57, kar pomeni A4 (440Hz). Signala *stevilkatona2* in *stevilkatona3* sta izračunana iz signalov *akordspodnji* in *akordzgornji*.





i_pid (red_pitaya_pid)

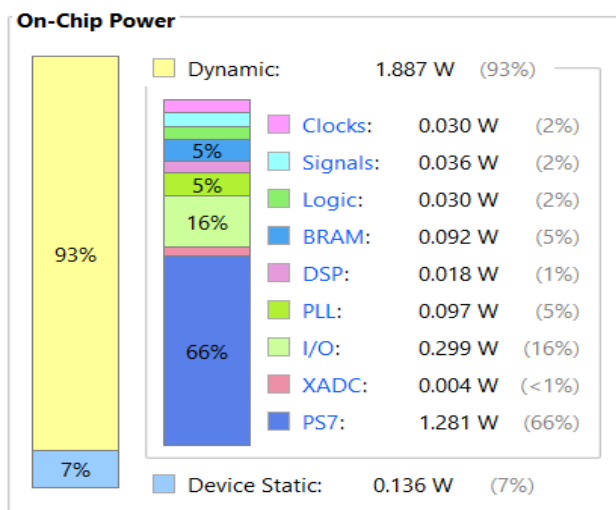
- > Nets (359)
- > Leaf Cells (47)
- > avtomat (avtomat)
- > dolocitevtonov (dolocitevtonov)
- > pocasnaura1 (pocasnaura)
- > pocasnaura2 (pocasnaura_5)
- > pocasnaura3 (pocasnaura_6)
- > pocasnaura4 (pocasnauratempo)
- > rom1 (rom)
- > rom2 (rom_7)
- > rom3 (rom_8)
- > stevec1 (stevec)
- > stevec2 (stevec_9)
- > stevec3 (stevec_10)
- > upravljanjeur (upravljanjeur)



Implementacija

Na prejšnji strani je prikaz uporabe celic na FPGA-ju. Komponente so označene z različnimi barvami. Največ zasedejo komponente **rom**, ki iz naraščajočega signala na izhodu generirajo sinusni signal. Naslednja je komponenta **upravljanjeur**, ki vsebuje tudi komponento **toni** z arreyem stotih vrednosti za posamezne tone. Vse ostale skpaj pa porabijo manj kot polovico.

Poraba



Poraba je po izračunih programa Vivado 2.023 W, od tega se večino porabi v prehodih (več kot 90 %). Od tega komponenta PID, kjer je implementiran generator tonov akordov in intervalov porabi 0.021 W, kar je približno 1%.

Na tabeli so predstavljeni uporabljeni elementi na FPGA na Red Pitayi.

Name	Slice LUTs (17600)	Slice Registers (35200)	F7 Muxes (8800)	F8 Muxes (4400)	Slice (4400)	LUT as Logic (17600)	LUT as Memory (6000)	Block RAM Tile (60)	DSPs (80)
i_pid (red_pitaya_pid)	512	270	126	38	205	512	0	0.5	2
avtomat (avtomat)	13	2	0	0	11	13	0	0	1
dolocitevtonov (dolocitevtonov)	23	25	0	0	9	23	0	0	0
pocasnaura1 (pocasnaura)	9	13	0	0	7	9	0	0	0
pocasnaura2 (pocasnaura_5)	9	13	0	0	7	9	0	0	0
pocasnaura3 (pocasnaura_6)	9	13	0	0	7	9	0	0	0
pocasnaura4 (pocasnaurate)	19	33	0	0	14	19	0	0	0
rom1 (rom)	122	0	40	18	36	122	0	0	0
rom2 (rom_7)	117	0	43	10	32	117	0	0	0
rom3 (rom_8)	118	0	43	10	36	118	0	0	0
stevec1 (stevec)	9	20	0	0	7	9	0	0	0
stevec2 (stevec_9)	18	20	0	0	11	18	0	0	0
stevec3 (stevec_10)	19	20	0	0	11	19	0	0	0
upravljanjeur (upravljanjeur)	40	74	0	0	24	40	0	0.5	0
toni (toni)	1	0	0	0	1	1	0	0.5	0

Za primerjavo podani še uporabljeni elementi originalnege komponente PID na Red Pitayi.

Name	Slice LUTs (17600)	Slice Registers (35200)	F7 Muxes (8800)	Block RAM Tile (60)	DSPs (80)
i_pid (red_pitaya_pid)	502	539	0	0	12

Nastavljanje preko vodila

Vseh šest parametrov lahko nastavljamo in beremo z ukazom »monitor« in sicer:

- 0x40000000 akord
- 0x40000004 nacin
- 0x40000008 interval
- 0x4000000c amp
- 0x40000010 stevilkatona
- 0x40000014 tempo

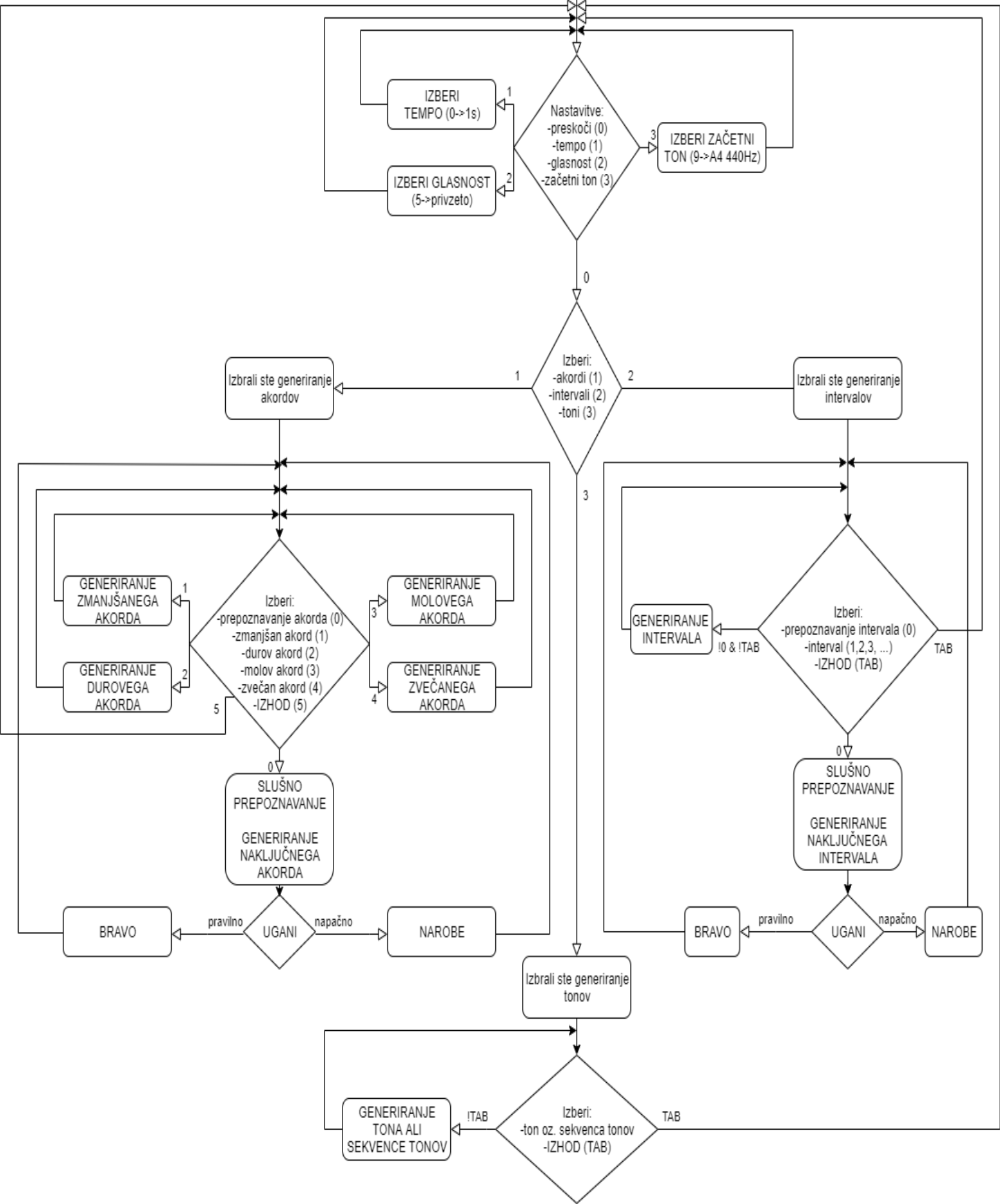
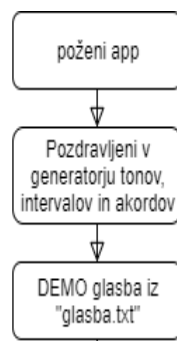
Program v C

Za lažjo uporabo sem izdelal tudi program v jeziku C, ki se ga naloži na Red Pitayo in avtomatizira nastavljanje registrov. Za komunikacijo se uporablja serijska povezava prek terminal z Red Pitayo. Program v terminal izpisuje navodila za posamezne korake, upravljamo pa ga z vnosom ASCII znaka in pritiskom na ENTER, saj je za sprejem uporabljena funkcija *getchar()*. Samemu vnosu nisem posvečal velike pozornosti, zato je za nastavljanje (npr. izhodiščnega tona) potrebno poznati desetiške vrednosti ASCII znakov ki so podani tudi v tabeli (Tabela ASCII znakov in njihovih desetiških vrednosti).

Tabela ASCII znakov in njihovih desetiških vrednosti

Dec	Char	Dec	Char	Dec	Char	Dec	Char
0	NUL (null)	32	SPACE	64	@	96	`
1	SOH (start of heading)	33	!	65	A	97	a
2	STX (start of text)	34	"	66	B	98	b
3	ETX (end of text)	35	#	67	C	99	c
4	EOT (end of transmission)	36	\$	68	D	100	d
5	ENQ (enquiry)	37	%	69	E	101	e
6	ACK (acknowledge)	38	&	70	F	102	f
7	BEL (bell)	39	'	71	G	103	g
8	BS (backspace)	40	(72	H	104	h
9	TAB (horizontal tab)	41)	73	I	105	i
10	LF (NL line feed, new line)	42	*	74	J	106	j
11	VT (vertical tab)	43	+	75	K	107	k
12	FF (NP form feed, new page)	44	,	76	L	108	l
13	CR (carriage return)	45	-	77	M	109	m
14	SO (shift out)	46	.	78	N	110	n
15	SI (shift in)	47	/	79	O	111	o
16	DLE (data link escape)	48	0	80	P	112	p
17	DC1 (device control 1)	49	1	81	Q	113	q
18	DC2 (device control 2)	50	2	82	R	114	r
19	DC3 (device control 3)	51	3	83	S	115	s
20	DC4 (device control 4)	52	4	84	T	116	t
21	NAK (negative acknowledge)	53	5	85	U	117	u
22	SYN (synchronous idle)	54	6	86	V	118	v
23	ETB (end of trans. block)	55	7	87	W	119	w
24	CAN (cancel)	56	8	88	X	120	x
25	EM (end of medium)	57	9	89	Y	121	y
26	SUB (substitute)	58	:	90	Z	122	z
27	ESC (escape)	59	;	91	[123	{
28	FS (file separator)	60	<	92	\	124	
29	GS (group separator)	61	=	93]	125	}
30	RS (record separator)	62	>	94	^	126	~
31	US (unit separator)	63	?	95	_	127	DEL

Sam potek programa opisuje diagram:



Zagon programa

Na Red Pitayo najprej naložimo seveda bitstream projekta. C program pa poženemo tako, da s pomočjo programa WinSCP na Red Pitayo naložimo kodo app.c in jo kar s kompilerjem na Red Pitayi prevedemo z ukazom v ter »gcc -o app app.c« v terminalu Program pa potem zaženemo z ukazom »./app«.

Poleg tega v isto mapo kot app.c shranimo tudi datoteko glasba.txt. V to datoteko je mogoče v naprej zapisati sekvenco tonov, ki se zgenerirajo pred glavnim delom programa nekakšen glasbeni uvod v program.

```
Pozdravljeni v generatorju tonov, intervalov in akordov
IZBERI: NAPREJ 0, tempo 1, amplitudo 2, izhodiščni ton 3
0
Za generiranje akordov pritisnite      1
Za generiranje intervalov pritisnite   2
Za generiranje tonov pritisnite       3
```

Najprej je možno nastaviti tempo (1) amplitudo (glasnost) (2) in izhodiščni ton (3). Če pa izberemo (0) se pojavi izbira med generiranjem akordov (1), intervalov (2) in tonov (3).

```
1
Izbrali ste generiranje akordov
Za slusno prepoznavanje akorda        0
Za generiranje ZM akorda pritisnite   1
Za generiranje DUR akorda pritisnite  2
Za generiranje MOL akorda pritisnite  3
Za generiranje ZV akorda pritisnite   4
Za IZHOD pritisnite                   5
0
UGANI! (ZMANJSAN-1, DUR-2, MOL-3, ZVECAN-4)
1
NAROBE (pravilno-4)
```

V primeru, da smo izbrali (1) se pojavi izbira za tip akorda. Izbiramo lahko med zmanjšanim (1), durovim (2), molovim (3) in zvečanim (4). Če ne želimo generirati akorda pritisnemo (5) za izhod. V primeru, da pritisnemo (0), se bo generiral naključni akord izmed štirih in lahko bomo ugibali kateri je in s tem vadili slušno prepoznavanje akordov. Če je odgovor napačen dobimo tudi pravilno rešitev.

```
Za slusno prepoznavanje akorda      0
Za generiranje ZM akorda pritisnite  1
Za generiranje DUR akorda pritisnite  2
Za generiranje MOL akorda pritisnite  3
Za generiranje ZV akorda pritisnite   4
Za IZHOD pritisnite                   5
```

```
5
IZBERI: NAPREJ 0, tempo 1, amplitudo 2, izhodišni ton 3
```

```
0
Za generiranje akordov pritisnite    1
Za generiranje intervalov pritisnite  2
Za generiranje tonov pritisnite      3
```

```
2
Izbrali ste generiranje intervalov
```

```
Za prepoznavanje intervala pritisni 0, za primo pritisni 1 in po pol tona naprej
```

```
0
UGANI! (prima-1, mala sekunda-2, velika sekunda-3,...)
```

```
1
NAROBE (pravilno-10)
```

Če zapustimo generiranje akordov (5) pridemo zopet na začetni meni, kjer je možno nastaviti tempo (1) amplitudo (glasnost) (2) in izhodiščni ton (3). Če nastavitve zopet preskočimo (0) in v naslednjem meniju izberemo generiranje intervalov (2) lahko izbiramo med različnimi intervali (1-prima, 2-mala sekunda, 3- velika sekunda, ...). Če pa izberemo (0) se nam generira naključni interval med primo (1) in oktavo (12), ki ga lahko poskusimo uganiti.

```
Izbrali ste generiranje tonov
```

```
Za A4 (440 Hz) pritisni 9 po pol tona gor in dol glej ASCII tabelo
Za IZHOD pritisni TAB
```

Zadnja možnost pa je generiranje tonov. Za generiranje tona je potrebno uporabljati ascii znake, kjer je številka devet ton A4 s 440 Hz.

```
5
Za A4 (440 Hz) pritisni 9 po pol tona gor in dol glej ASCII tabelo
Za IZHOD pritisni TAB
```

Lahko generiramo le en ton.

```
987654321123456789
Za A4 (440 Hz) pritisni 9 po pol tona gor in dol glej ASCII tabelo
Za IZHOD pritisni TAB
```

Lahko pa vpišemo več znakov in ko pritisnemo enter se v prej nastavljenem tempo zaporedoma generirajo vpisani toni.