7. vaja: Generator sinusnega signala

Naredili bomo numerični oscilator za generiranje vzorcev sinusnega signala. Vrednosti signala bomo brali iz vpogledne tabele, ki bo imela na vhodu fazni akumulator – števec z nastavljivim korakom štetja.

Generator z vpogledno tabelo

Vpogledna tabela je pomnilnik v katerem so zapisane izračunane vrednosti signala pri določenem kotu. Če tabelo beremo po vrsti, bomo na izhodu dobili zaporedne vzorce sinusnega signala. Za branje tabele uporabimo števec. V tabeli lahko shranimo le četrtino signala in z upoštevanjem simetrije izračunamo ustrezen izhod.

 Naredimo model vezja s 5-bitnim števcem st, ki spreminja vrednosti med 0 in 31. Deklarirajmo še 3-bitni signal adr za naslov vpogledne tabele. Razmisli, kako bi iz vrednosti števca dobil naslov, ki se spreminja takole: 0,1,2,3,4,5,6,7,7,6,5,4,3,2,1,0, 0,1,2,3,4,5,6,7,7,6,5,4,3,2,1,0... Namig: pomagaj si s primerjavo binarnih vrednosti števca in naslova:

clk																		
st	00000	00001	00010	00011	00100	00101	00110	00111	01000	01001	01010	01011	01100	01101	01110	01111	10000	10001
adr	000	001	010	011	100	101	110	111)		110	101	101 100		010	010 001		00	001
t=0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

2. Vzemimo vzorce ene četrtine sinusnega signala z amplitudo 127. Interval med 0 in 90° razdelimo na osem delov in izračunamo vrednost sinusa, ki ga zaokrožimo na celo število:

fi	127 sin(fi)
5,625	
16,875	
28,125	
39,375	
50,625	
61,875	
73,125	
84,375	

Deklaracija tabele predznačenih vrednosti v SHDL:

tab: 8s8 = 1,2,3,4,5,6,7,8;

.. in v jeziku VHDL:

type tabela is array (0 to 7) of signed(7 downto 0); constant tab : tabela := ("00000001", "00000010", "00000011", "00000100", "00000101", "00000110", "00000111", "00001000");

V modelu vezja deklariraj tabelo osmih predznačenih 8-bitnih vrednosti in jih nastavi na izračunane vrednosti (namesto na 1,2,3...8, kot je v primeru kode).

3. Dodaj v model vezja 8-bitni predznačen izhod **data**. Izhod naj bo enak vrednosti iz tabele tab(adr), kadar je najvišji bit števca enak 0, ali pa negirani vrednosti tabele, da dobimo cel sinusni val:

clk	Л	Ц	Г	Ų	U	Ų	U	Ų	U	U		Ц			П	Ц	Л	Ц	Л	Ц	Л	Ц	П	ų	Γ	Ų	Г	LΠ	Л	Л	Л	Л	П	П	П		Ц	U	ų	T	Ų	Л	Л	Л	Л	U	IL
st	0	1	2	з (4	5	6	7)	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24 2	25 2	26 2	72	8 29	9)30	31	0	1	2	3	4	5	6	7	8	9)	10 1	1)1:	2 13	14	15 1	16 17
adr	0	1	2	з (4 (5	6	7		6	5	4	3	2	1)	1	2	3	4	5	6	7		6	5 (4	4 (3	3 2	1		0	1	2	З	4	5	6	7		6	5 4) з	2)1)	0	1
data	F	_	_											_	_	_	_		_	_	_						_	-	_	-	-	-	_		-	_						_		-	-	-	
t=0	0		2		4		6		8		10		12		14		16		18		20		22		24	2	26	2	8	30)	32		34		36		38		40		42	44	4	46	4	18

Spreminjanje frekvence

Frekvenco izhodnega signala lahko spreminjamo na dva načina: s spreminjanjem vhodne ure (delilnik ure) ali pa z drugačno izvedbo števca za določanje faznega kota.

- 1. Popravi deklaracijo števca tako, da bo sedaj 16-biten. Popravi tudi stavke, ki so odvisni od vrednosti števca: najvišji bit števca je ima sedaj indeks 15, za naslov pomnilnika pa uporabi bite med 11 in 14. Za začetek nastavi korak štetja na 2048, da bo vezje delovalo enako prej in preveri delovanje na simulaciji.
- 2. Dodaj nov 16-biten vhod, ki bo predstavljal korak števca. Nastavi različne korake števca (npr. med 1000 in 5000) in opazuj izhodni signal na simulaciji.



Ugotovi, kakšna je povezava med frekvenco ure, korakom in frekvenco izhodnega signala.

*Simulacija in sinteza generatorja

 Preveri delovanje vezja še v orodju ModelSim. Naredi nov projekt, v katerega dodaj datoteko z VHDL modelom vezja. Za simulacijo tokrat ni potrebna testna struktura, saj je potrebno nastaviti le uro in konstanten vhod. Npr. nastavimo uro frekvence 100 kHz, korak na 1000 in čas simulacije 5 ms:

```
force clk 1 0, 0 {5 us} -r {10 us}
force korak 10#1000
run 5ms
```

🧼 /s	sin2/dk	1										
🖃 🌛 /:	sin2/korak	1000	(1000									
	sin2/data	-99	$\overline{\ }$	$\left \right\rangle$	\bigcirc		\frown	\square	\int		\bigwedge	\bigwedge
😐 🔶 /s	sin2/st	1010	-++++++++++	++++++++++	******	++++++++++	++++++++++	******	********	++++++++++		+++++++++++
. → /s	sin2/adr	100	ເງພາງງາຫນັງພາງ	<u>(nantinantina)</u>	, MAALAAN MAALAAN	נעעענענענענ		נוממת ממש נוומ	ענעענעענעע		(ע) ער איני איני איני איני איני איני איני אינ	າກເງິນການເງິນແມ່ງຫຼ
. 🖅 /:	sin2/t	0110	ເຫັນເປັນຫຼັງຫຼັງກ	<u>anninnn inni</u>	אמנומענומע	(ממשרת המור המור המור המור המור המור המור המור	<u>www.www.www</u>	נוממת ממש נוומ	ັນພົງນັ້ພງງນັ້ນງິງນ	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	עמעעכשענש	າກເງິນການງານຫຼາງກ
4,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	Now	000 ps	i i DS			200000	0000 ps	1 1		400000	0000 ps	
🔒 🎸 😑	Cursor 1	1000 ps										50000(

4. Naredi tudi nov projekt v orodju Quartus, izberi poljubno vezje FPGA iz družine Cyclone IV E in dodaj v projekt VHDL datoteko. Naredi sintezo vezja (Ctrl+K) in poglej koliko logičnih elementov in registrov zasede generator sinusnega signala.

Ime datoteke na disku mora biti enako imenu entitete vezja. Enako naj bo tudi ime projekta ali pa vsaj glavne entitete v Quartusu. Nastavitev je na: Assignments, Settings, General, Top-level entity.