

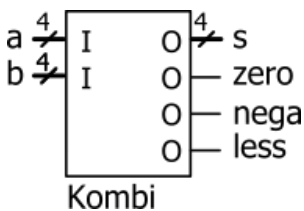


## 1. Vaja: kombinajska vezja

Ponovi: osnove jezika VHDL, opis vezij na nivoju pretoka podatkov v jeziku VHDL: [vhdl pretok](#)

Dvojiški zapis pozitivnih in negativnih števil. Kakšen je obseg 4-bitnih števil: 0111 predstavlja desetiško \_\_\_\_, 1000 predstavlja vrednost \_\_\_\_, 1111 predstavlja \_\_\_\_

## Kombinacijski seštevalnik in primerjalnik



Naredi opis 4-bitnega seštevalnika in primerjalnika celoštevilskih vrednosti. Signali a, b in s naj bodo deklarirani s podatkovnim tipom **signed** (3 **downto** 0). Na izhodu vezja naj bodo poleg vsote (s) tudi enobitni izhodi:

- zero, ki se postavi na '1', kadar je vsota enaka 0
- nega, ki se postavi na '1', kadar je vsota manjša od 0
- less, ki se postavi na '1', kadar je a < b

Vsoto vektorjev tipa **signed** opišemo z operatorjem seštevanja, primerjave pa s pogojnim prireditvenim stavkom **when ... else**. Za začetek opisa vezja v jeziku VHDL si pomagaj s spletnim orodjem: <http://lniv.fe.uni-lj.si/grafTB.htm>

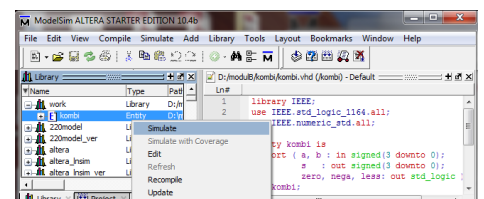
### Grafični Test Bench

Circuit type:  Sequential circuit

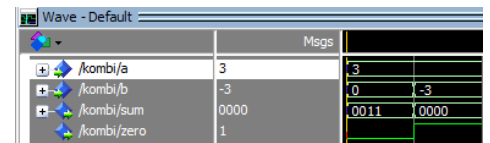
Name	In/Out	Type	MSB	LSB
a	in	signed	3	0
b	in	signed	3	0
s	out	signed	3	0
zero	out	std_logic		0
nega	out	std_logic		0
less	out	std_logic		0

V tabeli določi vse vhode in izhode vezja, odstrani kljukico pri **Sequential circuit** (ker imamo kombinajsko vezje) in določi ime entitete (npr. Kombi). Ob kliku na gumb **Update** se pojavi grafični prikaz signalov, ob kliku na **Generate Entity** pa dobimo prvi opis vezja v jeziku VHDL. Prevajanje in simulacijo bomo izvedli v orodju ModelSim:

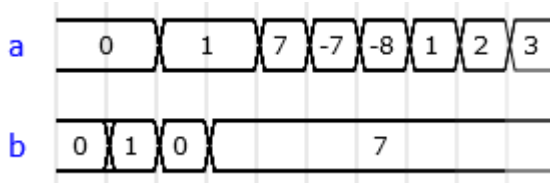
- Pripravi mapo v kateri bodo datoteke projekta
- V programu ModelSim ustvari nov projekt: **Jumpstart**, **Create Project** ali **File > New > Project**
- Naredi novo datoteko **Create New File**, jo odpri **File > Open** in prekopiraj prvi opis vezja s spletne stranio, nato pa dopolni VHDL kodo
- Prevedi datoteko s klikom na ikono **Compile** in popravi morebitne napake



- Startaj simulacijo **Simulate** in izberi opis vezja iz work
- Dodaj signale v okno za opazovanje **Add > to Wave**
- Klikni z desnim gumbom na vhodne signale in nastavi vrednost **Force**
- Zaženi simulacijo **F9** in opazuj rezultate

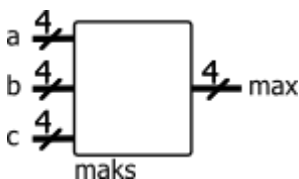


- Naredi opis seštevalnika z uporabo notranjega signala. Ugotovi, zakaj ne moremo vsote prirediti kar izhodnemu signal, ki je deklariran kot **out**.
- Preizkusi delovanje vezja z negativnimi števili v dvojiškem komplementu. Vsem večbitnim signalom nastavi decimalen prikaz (**Radix > Decimal**). Negativno desetiško vrednost določimo v programun ModelSim v obliki **-10#vrednost**. Preizkusi različne kombinacije vhodnih vrednosti, tako se da bodo vsi izhodi spreminjali. V praktičnih vezjih ne moremo preizkusiti vseh možnih vrednosti, za se omejimo na nekaj tipičnih in mejnih primerov.
- Uporabi spletno orodje za izdelavo simulacijske testne strukture. Nastavi vrednosti vhodoma a in b in klikni na gumb **Generate Test Bench**. Vsebinsko kopiraj v novo datoteko v programu ModelSim (npr. TestKomb), ki jo uporabi za izvedbo simulacije.



- Popravi opis seštevalnika, tako da bo izračunal 5-bitno vsoto, pri kateri ne bo prišlo do preliva.
- Razmisli, ali za izhod nega potrebuješ stavek **when...else** s primerjalnim operatorjem ali bi ga lahko opisal enostavneje.  
Razmisli, kako bi z operacijo odštevanja naredil primerjavo  $a < b$  !

## Vaja: Maksimum



Naredi vezje za funkcijo maksimum s tremi 4-bitnimi vhodi. Na 4-bitnem izhodu naj bo vedno maksimalna izmed vseh vhodnih vrednosti, ki so predstavljene kot nepredznačena števila.

- Najmanj koliko primerjav potrebuješ, da najdeš največjo vrednost ?

Število primerjalnih operatorjev v opisu vpliva na velikost vezja, zato je dobro razmisliti kako s čim manj operatorji opisati vezje.