RĪGAS TEHNISKĀ UNIVERSITTĀTE

ELEKTRONIKAS UN TELEKOMUNIKĀCIJU FAKULTĀTE

[](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwj9zKXEs6fgAhWF1iwKHceJAlAQjRx6BAgBEAU&url=https://lv.wikipedia.org/wiki/Att%C4%93ls:RTU_logo_2017.svg&psig=AOvVaw2YftzNrZHl3gHZWIEXy3Ih&ust=1549552627423236)

Funkcionālo un loģisko shēmu modelēšana, RTR532

Laboratorijas darbs Nr.4

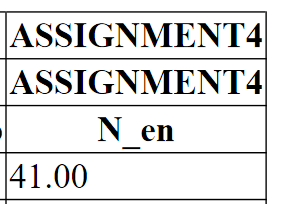
Bogdans Žukovskis

161REC020

Riga, 2021

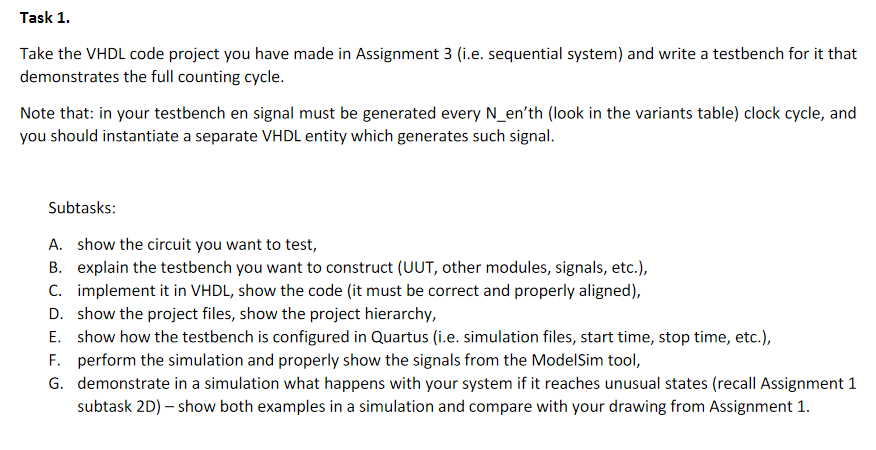
# Ievads

Variants:



4. laboratorijas darbs ir veltīts secīgas shēmas testēšanas realizēšanai ar testēšanas vidi ModelSim.

# Uzdevums



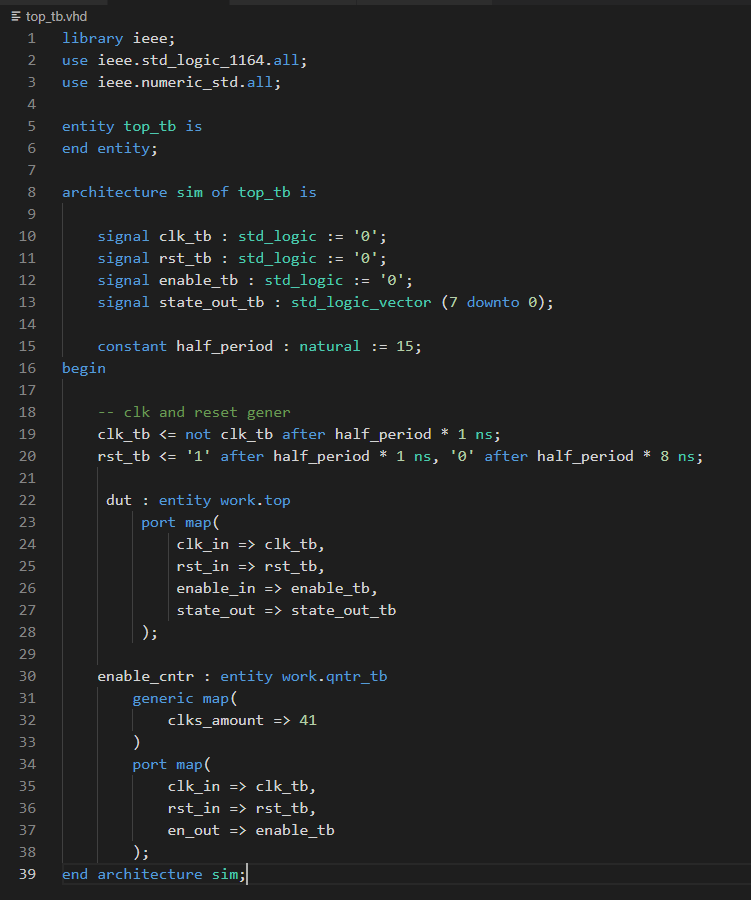
## B.

Klasiska testbench struktūra ietver sevī ieejas ģenerēšanu, moduli kas tiek testēts un izejas rezultātu salīdzināšana ar sagaidāmo. Mūsu gadījumā tiek testēta vienkārša secīga shēma , kura uzvadās kā skaitītais ar dažiem ekstra stāvokļiem , kur stāvokļu parēja nav regulāra (+1). Tāpēc arī testbench būs vienkāršs – funkcionāla manuāla simulācija. Tas nozīme ka izejas tiek pārbaudītas bez automātikas un signālu pārvadīšanā nav iesaistītas aiztures. Tā kā testējamas shēmas stāvokļa reģistram ir arī enable ieeja, būs arī vajadzīgs skaitītāja modulis, kurš ģenerē enable izeju uz uzdoto taktu numuru. Skaitītajā enable izeja testbench iekšā tiek pievienota testējamam top modulim. Virs tam mūsu testbench fails sastāv tikai no taktēšanas signāla un reset signāla ģenerēšanas. Signāli kuri ir definēti testbench iekšā ir ar *\_tb* pierakstu. Ir vērts pamanīt, ka rakstot simulācijas failu var arī parastajiem signāliem uzdot sākuma vērtību. Ko labāk nedarīt sintezējamā koda gadījumā. Dēļ tā ka skaititājam bija definēts generic parametrs ar taktēšanas ciklu skaitu pec kurā ir jāģenerē enable izeju, bija vajadzīgs uzdefinēt funkciju kura definēs cik plātu vajag paņemt signālu lai tajā būtu iespējams ievietot vajadzīgu taktēšanas ciklu skaitu. Tāda funkcija sastāv no logarifma aprēķināšanas un bija aprakstīta *util\_pckg.vhd* package faila iekšā *work* library iekšā.

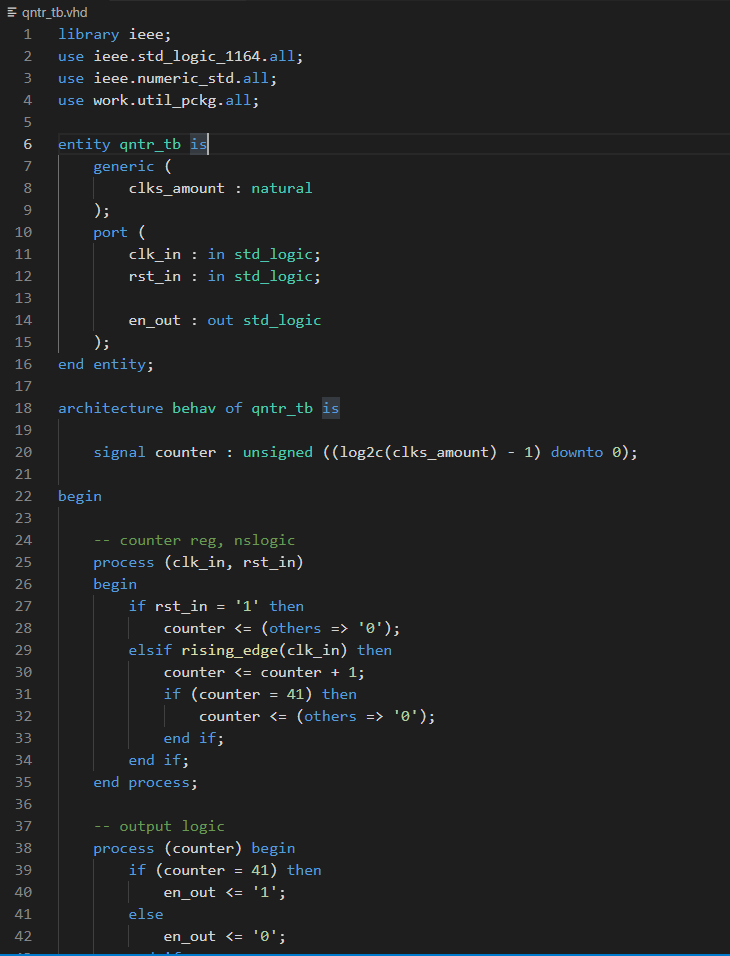
## C.

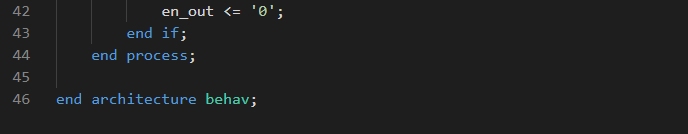
Visiem aprakstītiem failiem kods izskatās sekojoši:

*top\_tb.vhd*

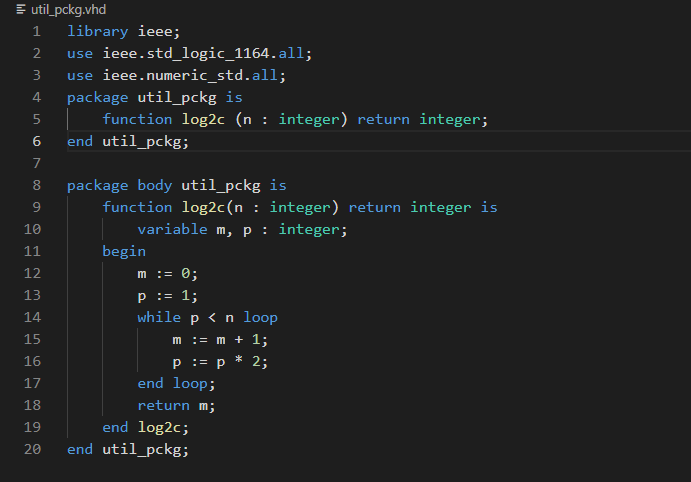


Skaitītajā modulis:





*util\_pckg.vhd* package fails:

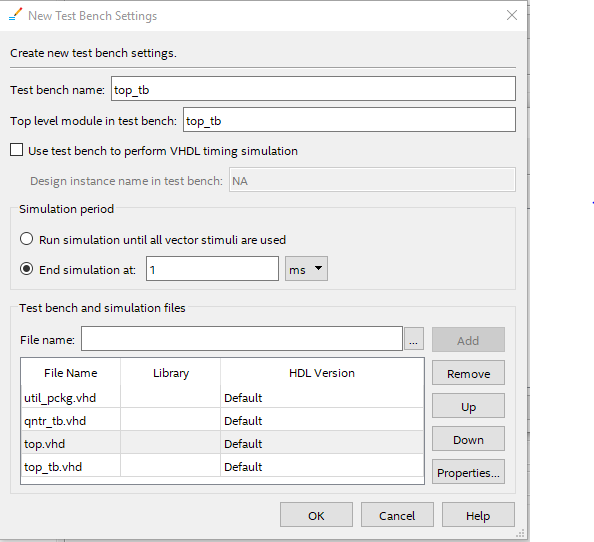


## D.

Projekta hierarhija un failu saraksts atkārtojas no 3. lab. darba.

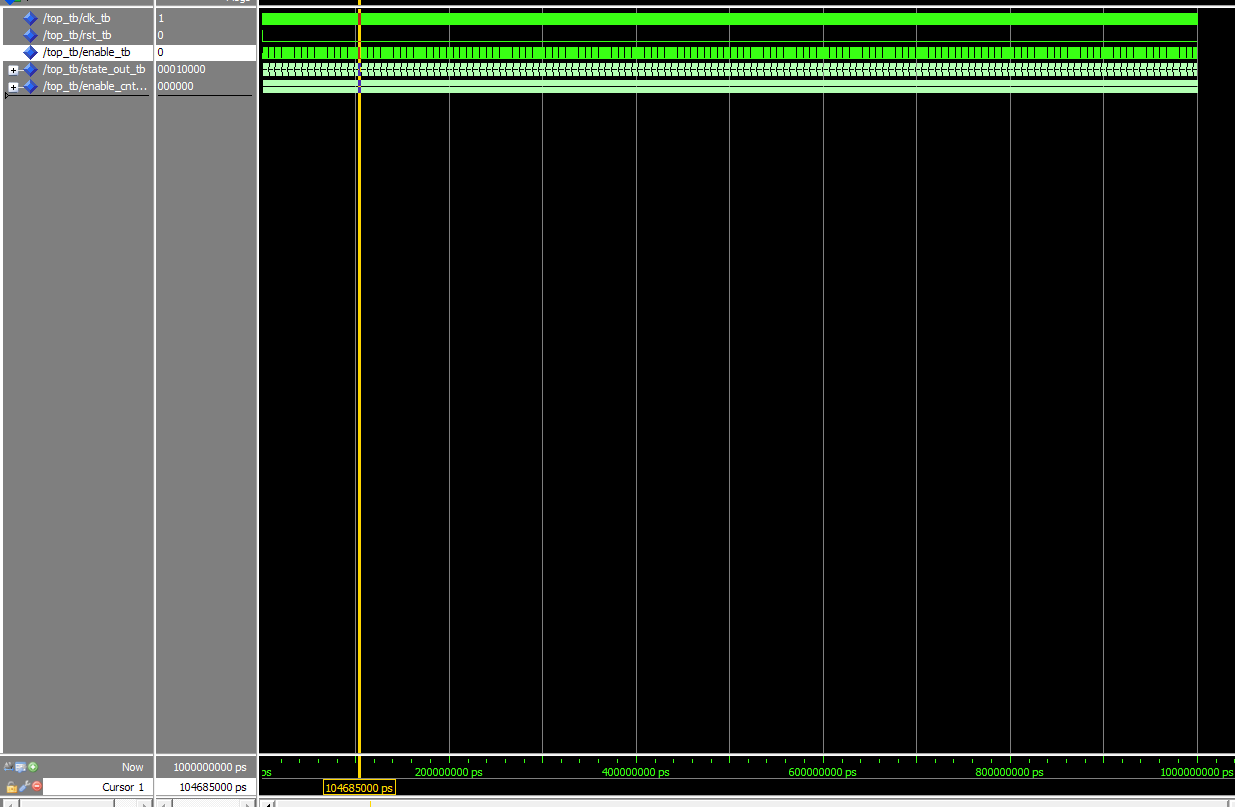
## E.

Quartus vidē testbench ir nokonfigurēts sekojoši:

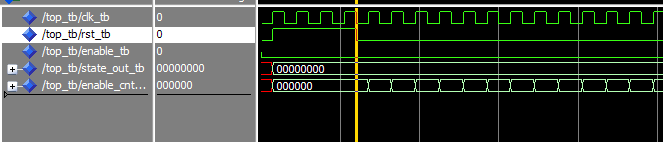


## F.

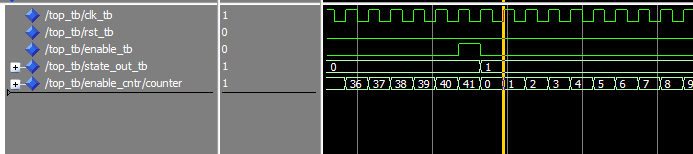
ModelSim vidē grafiskie rezultāti ir sekojoši:



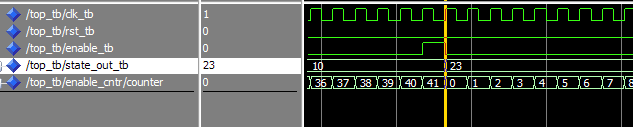
Sākumā ir redzamā *reset* signāla darbība :



*Enable* signāla darbība, kopa ar kuru mainās testējamas secīgas shēmas stāvoklis. *Enable* signāls aktivējas uz 41 skaitītāja taktu.

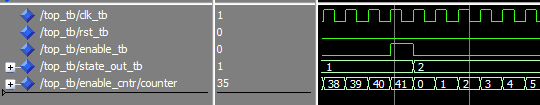


Stāvokļa maiņa nestandarta stāvokļa parejas gadījumam:

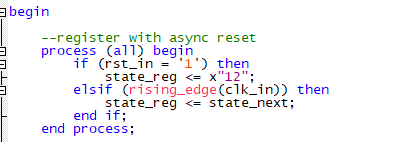


## G.

Gadījumā , ja sistēma būs stāvokļi 1 , nākamais būs 2.



Gadījumam ar stāvokļa reģistra vērtību 19 , bija uzdota stāvokļa reģistra reset vertība(18) , lai būtu iespējams aiziet stāvokļi 19 un izpētīt kas notiks tālāk.



## Rezultātu analīze

Laboratorijas darba ietvaros bija izpētītas sintēžu iespējas Quartus vidē secīgajiem shēmām realizētajiem VHDL apraksta veidā. Secīga shēma bija aprakstīta FSM veida , tomēr , ne klasiskajā case veidā, bet šajā gadījumā dabiskājā if veidā.