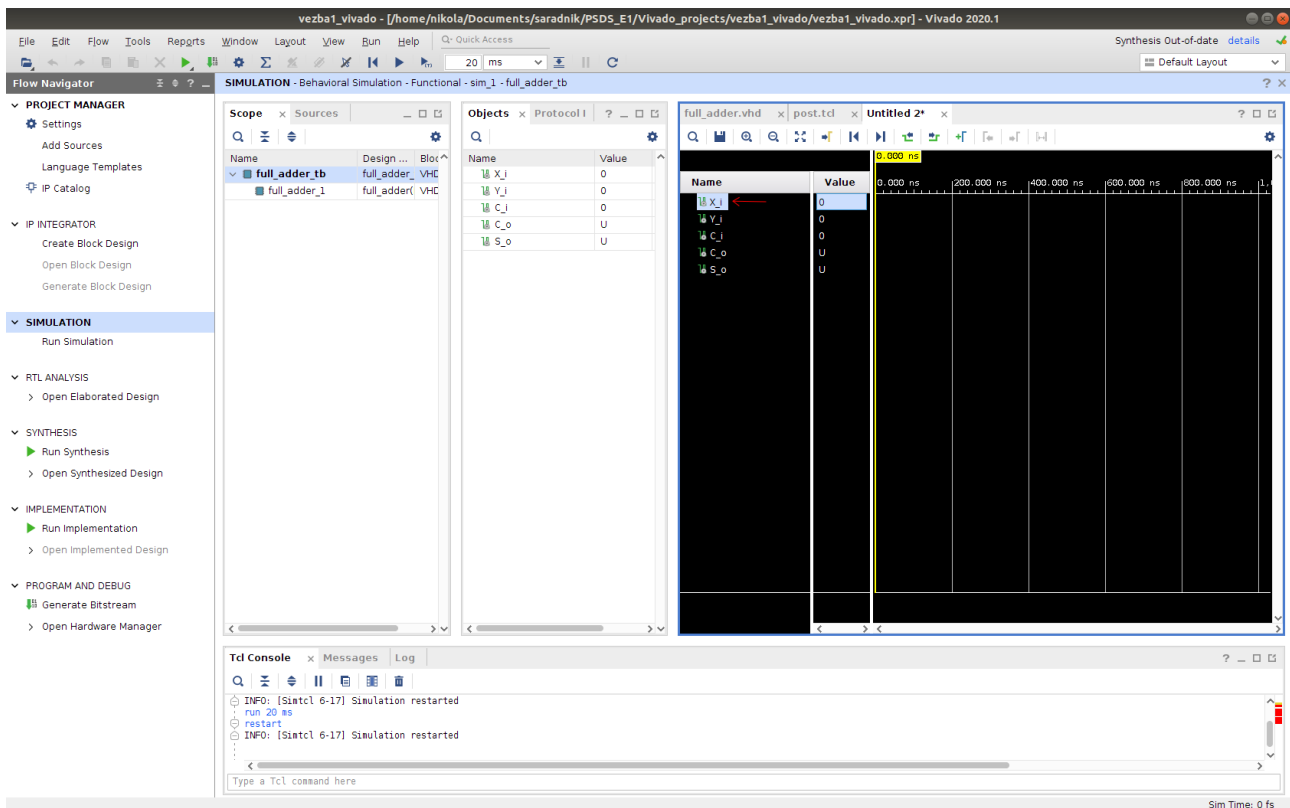


Uputstvo za forsiranje vrednosti na primarnim ulazima, unutrašnjim linijama i primarnim izlazima u Vivado alatu

1 Forsiranje vrednosti kroz Vivado Gui

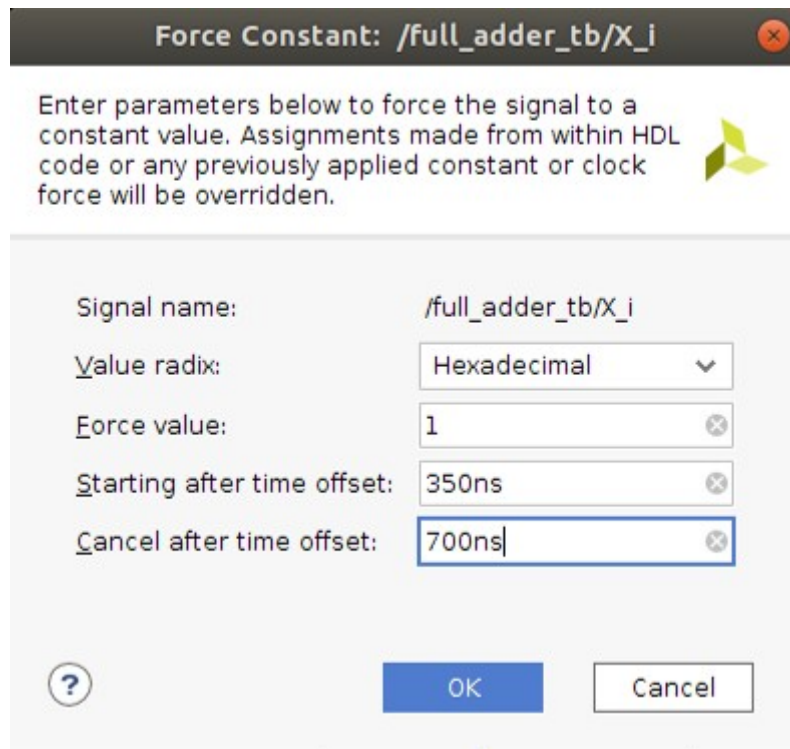
Vivado alat omogućava forsiranje vrednosti na primarnim ulazima, primarnim izlazima i unutrašnjim signalima projektovanog modula. Na taj način moguće je utvrditi kako se modul ponaša kada se pojavi greška kao što je SA1 (*eng. stuck at 1*) ili SA0 (*eng. stuck at 0*).

Najjednostavniji način da se to uradi jeste kroz GUI (*eng. Graphical user interface*) Vivado alata i u nastavku je ilustrovano forsiranje vrednosti na jednobitnom primarnom ulazu „X_i“ „full adder“ modula:



Slika 1. Forsiranje vrednosti na signalu „X_i“.

Prvo što je neophodno uraditi jeste pokrenuti simulaciju. Nakon toga otvoriti prozor u kojem je moguće posmatrati promene na signalima modula (*eng. wave window*) i desnim klikom odabrati signal na kojem želite da forsirate vrednost (X_i u prethodnom primeru). Odabrati opciju *force constant* i otvoriće se sledeći prozor:

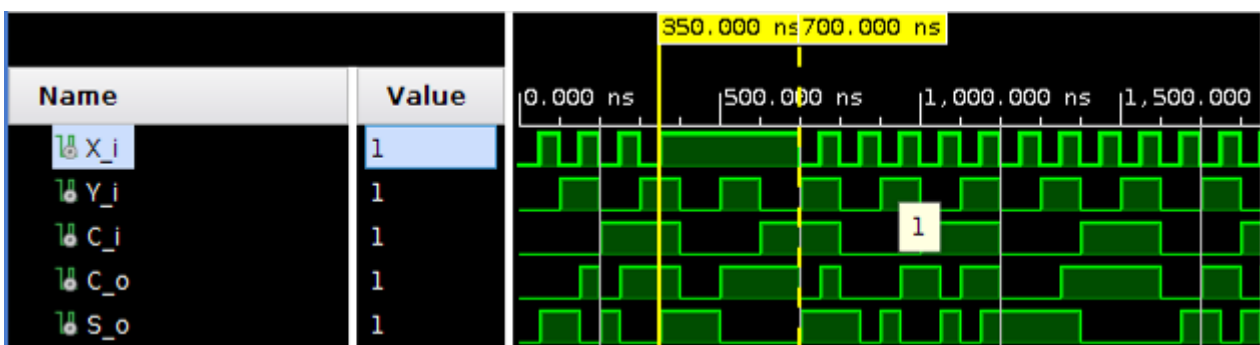


Slika 2. Force constant prozor

Polja prozora sa prethodne slike imaju sledeće značenje:

- „Signal name“ predstavlja hijerarhijsku putanju do signala X_i (sa prethodne slike X_i se nalazi unutar modula „full_adder_tb“).
- „Value radix“ predstavlja brojni format u kom će biti zapisan „Force value“ parametar
- „Force value“ je vrednost na koju će signal biti forsiran.
- „Starting after time offset“ jeste simulaciono vreme od kog forsiranje vrednosti treba da počne da važi.
- „Cancel after time offset“ jeste simulaciono vreme od kog forsiranje treba da prestane da važi.

U prethodnom primeru na signalu „X_i“ forsirana je logička jedinica, zapisana u heksadecimalnom formatu, i to treba da važi od 350ns do 700ns simulacionog vremena. Sledeća slika to ilustruje:



Slika 3. Talasni dijagram forsiranja vrednosti na signalu „X_i“

2 Forsiranje vrednosti pomoću .tcl komandi

Kako je Vivado alat zasnovan na izvršavanju .tcl komandi, tako se i forsiranje signala može izvršiti pomoću njih. Detaljan opis Vivado .tcl komandi može se pronaći u:

https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug835-vivado-tcl-commands.pdf

Komanda koja će ovde biti opisana jeste „add_force“, čija je sintaksa sledeća:

```
add_force [-radix <arg>] [-repeat_every <arg>] [-cancel_after <arg>] [-quiet] [-verbose]
<hdl_object> <values>
```

Argumenti prethodne .tcl komande:

- *radix* – Specifira brojni format u kom će biti zapisana vrednost koju treba forsirati na signalu (opcionni argument).
- *repeat* – Specifira nakon kog vremenskog intervala forsirana vrednost treba da se ponovi (opcionni argument).
- *cancel_after* – Specifira nakon kog vremenskog intervala forsirana vrednost treba da prestane da važi (opcionni argument).
- *quiet* – Ukoliko je ovaj argument prosleđen „add_force“ komandi neće biti prikazana poruka o njenom izvršenju u .tcl prozoru (opcionni argument).
- *verbose* - Ukoliko postoje neka ograničenja ispisa poruka, ova komanda ih nadjačava i omogućava ispis poruka „add_force“ komande (opcionni argument).
- *hdl_object* – Specifira hdl objekat nad kojim treba izvršiti forsiranje vrednosti (obavezan argument).
- *value* – Vrednost koja treba da se forsira (obavezan argument).

U nastavku su prikazani načini korišćenja „add_force“ .tcl komande. Prvi primer ilustruje forsiranje logičke jedinice na reset signalu u trenutku 300ns. Takođe, nakon izvršene komande se u promenljivoj for10 sačuvava hdl objekat nad kojim je izvršeno forsiranje (reset).

```
set for10 [ add_force reset 1 300 ]
```

Naredni primer pokazuje kako periodično forsirati vrednosti na određenom signalu i kako završiti to periodično forsiranje:

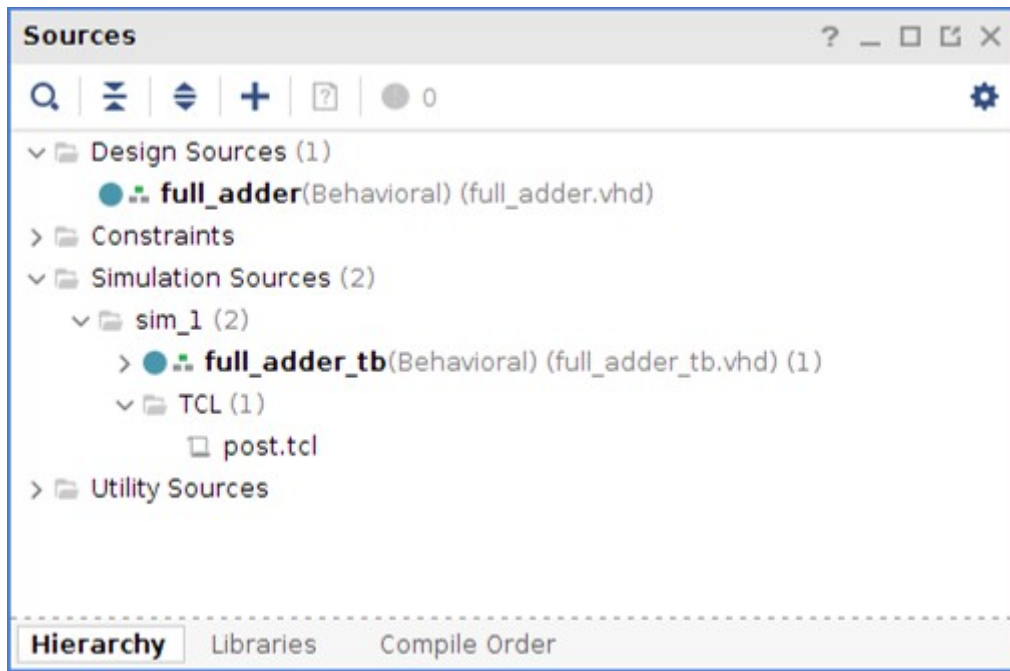
```
add_force mySig {0} {1 50} {0 100} {1 150} -repeat_every 200 -cancel_after
10000
```

Obratiti pažnju da je u vitičastim zagradama navedena vrednost koja se forsira i vremenski trenutak u kom se ona forsira. Takođe, u prvoj vitičastoj zagradi navedena je samo vrednost i na taj način će se ona forsirati u trenutnom simulacionom vremenu.

2.1 Primer korišćenja .tcl komandi za forsiranje vrednosti nad signalima „full adder“ modula.

U sekciji 1 je objašnjeno kako koristeći GUI vivado alata može da se forsira logička jedinica nad signalom „X_i“, a u nastavku je to urađeno pomoću .tcl komandi.

Prvo što je neophodno uraditi jeste napraviti .tcl fajl i uključiti ga u Vivado projekat. Odabirom opcije „Add sources -> Add new simulation sources“ dodati <ime>.tcl fajl (**Napomena: U verziji 2020.1 Vivado alata neophodno je navesti celu putanju do .tcl fajla, jer se fajl neće prikazati ukoliko pokušate ga ručno, kroz GUI pronađete**). Nakon što je fajl dodat, u sources prozoru biće prikazano sledeće:



slika 3. Dodavanje .tcl fajla

Kao što se može videti, u direktorijumu Simulation Sources pojavio se dodatni pod-direktorijum unutar koga je smešten .tcl fajl (*post.tcl*). Komande unutar njega će biti pokrenute kada se pokrene simulacija (*Simulation -> Run simulation*).

Nakon što je .tcl fajl kreiran, može se preći na pisanje .tcl komandi i u narednom primeru ilustrirano je forsiranje logičke jedinice na unutrašnjem signalu C4_s „full_adder_1“ instance nakon što protekne 400ns simulacionog vremena.

```
add_force /full_adder_tb/full_adder_1/C4_s -radix hex 400ns 1
```

Kao što se može videti, neophodno je specificirati celu hijerarhijsku putanju do signala kome se forsira vrednost, počevši od modula na vrhu hijerarhije (u ovom primeru *full_adder_tb*). Nakon što je komanda napisana, pokrenuti simulaciju i posmatrati vrednost na signalu C4_s.

Napomena:

*Prilikom pokretanja simulacije obavezno podesiti podrazumevano vreme koje protekne na početku, jer komande napisane u dodatom .tcl fajlu kreću da se primenjuju tek nakon što ono protekne. U prethodnom primeru, ukoliko bi na početku simulacije proteklo 1000ns, add_force tcl komanda ne bi imala smisla, jer bi forsirala vrednost u trenutku 400ns, a on je već protekao. **Podešavanje vremena koje protekne na početku simulacije se vrši promenom vrednosti u settings -> simulation -> simulation -> xsim.simulate.runtime* prozoru.***

Forsiranje vrednosti pomoću .tcl komandi je jako korisno, jer sa par linija koda mogu da se podese vremenski trenuci forsiranja vrednost nad više signala i u sledećem primeru to je ilustrovano:

```
#set x 4
set time_i 400

for {set x 4} {$x < 17} {incr x 1} {

    # postavljanje hijerarhijske putanje do signala nad kojim vrši forsiranje
    set -force_path /full_adder_tb/full_adder_1/C
    append force_path $x _s

    #postavljanje vremenskog trenutka u kome je potrebno izvršiti forsiranje vrednosti
    set time_c $time_i
    set time_string [append time_c ns]

    #postavljanje vremenskog trenutka u kome treba prekinuti forsiranje vrednosti
    set cancel_f_time [expr {$time_i + 400}]
    append cancel_f_time ns

    # forsiranje vrednost
    add_force $force_path -radix hex 1 $time_string -cancel_after $cancel_f_time

    # Uvećavanje vrednosti time_i za 400.
    incr time_i 400
}
```

Kodni listing. Primer korišćenja tcl komandi kako bi se forsirale vrednosti nad više signala

U prethodnom primeru se vrši forsiranje vrednosti na unutrašnjim signalima C4_s do C16_s u vremenskim trenucima pomerenim za 400ns. Ovo je moguće zbog „pametnog“ imenovanja unutrašnjih signala „full_adder“ modula.

Napomena: Detaljnije o .tcl skriptovanju možete pronaći u materijalu za 13 nedelju laboratorijskih vežbi na predmetu Projektovanje složenih digitalnih sistema:

<https://www.elektronika.ftn.uns.ac.rs/projektovanje-slozenih-digitalnih-sistema/wp-content/uploads/sites/120/2018/03/Vezba-13-Design-Constraining-and-Tcl-Scripting.pdf>