

## Projekat na predmetu PNRS2 -UVM

Priložena arhiva *Projekat\_UVM\_2022.zip* sadrži projekat za ispitivanje APB GPIO Master bloka.

1. Kompletan projekat je preuzet kao otvoreni kod sa GITHUB portala. Cilj projekta je upoznavanje sa Arm-ovom APB perifernom magistralom, analizom rešenja koje liči na savremena komercijalna rešenja kakva se mogu očekivati u inženjerskoj praksi.
2. Simulacija je pisana u SV jeziku i pokreće se putem run.do skripte iz Questa okruženja sa lokacije:

*Projekat\apb4\_gpio-master\sim\questa\run.do*

3. Potrebno je proučiti sledeće:

Functionisanje osnovnog RTL bloka koji se ispituje:

*Projekat\apb4\_gpio-master\rtl\verilog\apb\_gpio.sv*

zatim magistrale koja je formirana kao BFM, testova i konačno testbenča:

*Projekat\apb4\_gpio-master\bench\verilog\*

*apb\_bfm.sv*

*test.sv*

*testbench\_top.sv*

Dokument koji opisuje funkcionalnost bloka koji se ispituje:

*Projekat \apb4\_gpio-master\docs\apb4\_gpio\_datasheet.pdf*

Za detaljnije upoznavanje sa APB4 magistralom proučiti izvorni ARM-ov document postavljen na:

*Projekat \apb4\_gpio-master\docs\IHI0024C\_amba\_apb\_protocol\_v2\_0\_spec.pdf*

4. Pokrenuti datu simulaciju iz tačke 2 i proveriti da li se u konzoli Questa simulatora dobija sledeći ispis:

```
# APB GPIO Testbench Initialized
# -----
# Checking reset values ...
# Basic IO test ...
# Random IO test ...
# Clear Status Register test ...
# Trigger Level-Low test ...
# Trigger Level-High test ...
# Trigger Level-Random test ...
# Trigger Falling-Edge test ...
# Trigger Rising-Edge test ...
# Trigger Random-Edge test ...
# IRQ test ...
# -----
# APB GPIO Testbench finished successfully @19383990
# -----
```

5. Projekat koji je na u ovoj arhivi urađen u SV treba prebaciti u UVM. To znači:
  - a) Formirati model APB magistrale kao interface u UVM-u ukoliko je moguće.
  - b) Sve postojeće SV testove formirati kao testove naslednike uvm\_test klase, odnosno kao UVM testove.

- Odlučiti kako formirati transakcije, zapravo sekvence (sequence\_item-e) iz perspektive APB magistrale i slično tome kako formirati transakcije, zapravo sekvence (sequence\_item-e) iz perspektive GPIO portova.
  - Kad je prva stavka opredeljena, treba formirati potrebne sequence\_item, sequence, sequencer, driver, monitor, scoreboard.
  - Predložiti i realizovati Coverage koji će pokriti sistemski bitne stavke u okviru ovog projekta.
  - Randomizovati sve što smatramo za bitno da prođe kroz randomizaciju u okviru ovih testova.
  - Obezbediti u okviru \*.do skripte da se testovi pokreću jedan za drugim i da se objedinjeno prati pokrivenost.
- c) Naravno i sam tesbenč prilagoditi UVM metodologiji, odnosno uskladiti ga sa zahtevanim izmenama definisanim pod a) i pod b) u okviru ove stavke. Isto važi i za samu \*.do skriptu kao što je ranije navedeno.

Literatura se može koristiti, kao i prethodno rađene vežbe, udžbenik i internet. Dozvoljena je komunikacija između projektnih timova uz veoma važnu napomenu **da će nakon predaje projekta kodovi biti međusobno poređeni i ukoliko se uoči prepisivanje kodova između dva tima, oba tima završavaju projekat sa nula bodova! Nije dozvoljeno deljenje kodova sa rešenjem zadatka između timova, sankcije slede i izvoru i primaocu bez razlike!**

**Nije dozvoljeno preuzimanje tuđeg rešenja sa interneta, čak u slučaju da uspete da nađete dostupan projekat koji pokriva sličnu tematiku u UVM-u!**