Elektrotehnički fakultet
Univerzitet u Banjoj Luci

UPOZNAVANJE SA MOGUĆNOSTIMAQUARTUS SOFTVERA U POGLEDU RADA SA AVALON-MM INTERFEJSOM

1. Uvod

Nakon što smo dizajnirali module u VHDL jeziku u skladu sa tim da komuniciraju koristeći Avalon-MM interfejs, postavlja se pitanje kako verifikovati dizajn. Za rad se koristi Quartus Prime softver. Unutar Quartus softvera postoji nešto što se zove Avalon Verification IP Suite, koji daje modele pod nazivom BFM (Bus Functional Model) koji su u mogućnosti simulirati ponašanje željenog interfejsa i olakšati verifikaciju. Između ostalog daje podršku za Avalon-MM master i slave interfejse. Također postoji i Avalon-MM Monitor koji u suštini verifikuje željeni protokol.

2. BFM

Svaki BFM sastoji se iz nekoliko dijelova, pri čemu je nama ključni dio koji se odnosi na VHDL paket. Ovaj paket sadrži VHDL API koji se koristi za kontrolu BFM i interfejsa sa našim testnim programom. Ovaj paket sadrži VHDL procedure. Dakle ideja je da imamo neki testbenč programa zatim da imamo sa jedne strane BFM (bio to master ili slave) i sa druge strane tzv. DUT odnosno komponentu koja je dizajnirana da koristiti Avalon-MM interfejs i koju treba verifikovati.

BFM uključuje proceduralni interfejs za implementiranje sljedećih funkcionalnosti

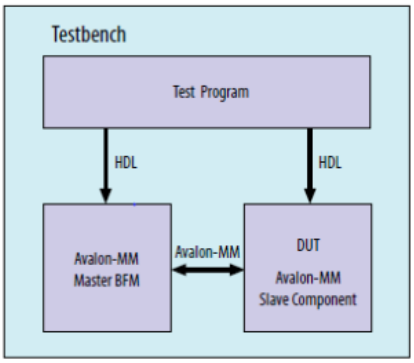
- Praćenje komandi
- Prosljeđivanje pristiglih komandi testnom programu (koji se piše u VHDL)
- Prihvatanje odgovora transfera pristiglih od testnog programa

2.1 Avalon-MM Master BFM

Avalon-MM Master BFM implementira Avalon-MM interfejs protocol, uključujuči čitanje, pisanje, kao i burst mod. Na ovaj način mi kreiramo Avalon-MM BFM koji predstavlja master komponentu, a komponenta koju verifikujemo je slave komponenta koja je dizajnirana, često se na engleskom naziva DUT (Device Under Test). Kreira se testbenč koji sadrži:

- Avalon-MM Master BFM
- Testbenč program
- DUT koji uključuje Avalon-MM slave interfejs.

Na slici je prikazana ta konfiguracija:



BFM uključuje i ilegalne transfere tako da se može testirati i funkcionalnost dizajna DUT-a u slučaju dešavanja grešaka prilikom komunikacije.

BFM podržava čitav spektar signala definisanih za Avalon-MM interfejs. Avalon-MM interfejs se može podesiti po želji koristeći niz parametara kao što su:

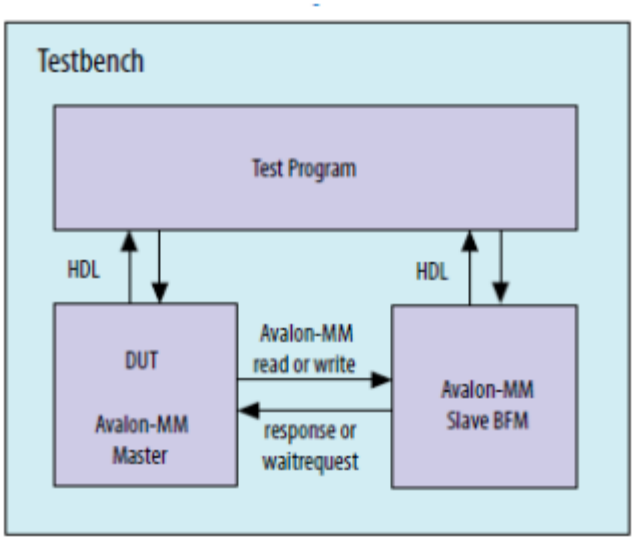
- Address width
- Symbol width
- Use the read signal
- Use the write signal
- Use the addres signal
- Use the byteenableblock signal
- i drugi …

API koji podržava Avalon-MM Master BFM se sastoji iz niza funkcija kao što su:

- signal_write_response_complete() – signalizira da je primljen write response signal
- set_command_address – postavlja adresu transkacije
- init() – inicijalizuje Avalon-MM interfejs
- i mnoge druge ..

2.2 Avalon-MM Slave BFM

Avalon-MM Slave BFM implementira slave strane interfejs protokola. Sada imamo obrnutu situaciju u odnosu na prethodno izloženu. Sada je sa slave strane BFM, a sa master strane je DUT.



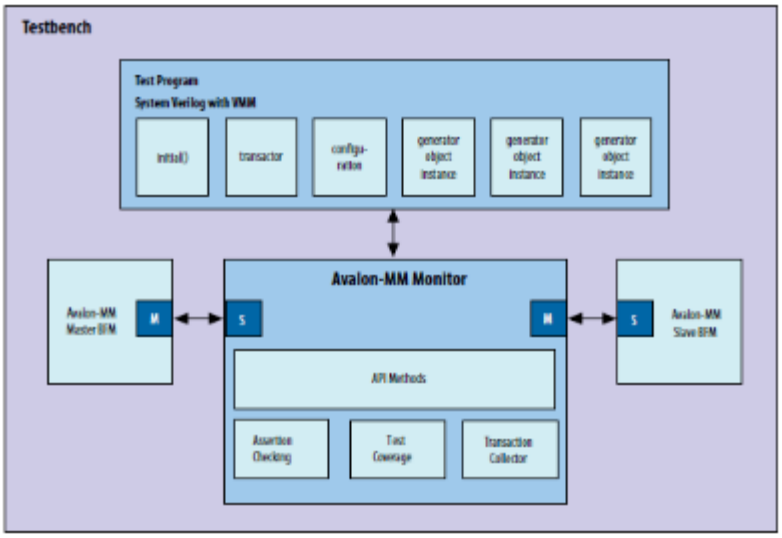
Potrebno je napisati testni program u VHDL koji programira Avalon-MM master da šalje transkacije, programira Avalon-MM Slave BFM da odgovara na njih te analizirati rezultate.

Ostatak priče za parameter i API je slična kao za Master BFM.

2.3 Avalon-MM Monitor

Također postoji nešto što se zove Avalon-MM Monitor komponenta. Služi za verifikaciju Avalon-MM interfejsa. Kreira se testbenč koji koristi Avalon-MM Monitor tako da je monitorov slave interfejs povezan na Avalon-MM master interfejs komponente, a njegov master interfejs povezan na slave interfejs komponente. Testni program zatim komunicira sa monitorom i može koristiti monitorove provjere (assertions) da bi se pokrile sve funkcionalnosti DUT-a. (vidi sliku).

Zatim imamo podesive sve moguće parametre Avalon-MM interfejsa da se prilagode specifičnoj implementaciji. A zatim API sa nizom funkcija koje se mogu pozivati i na taj način provjeravati šta se dešava u komunikaciji.



Više informacija može se naći na link

Također postoji i primjer kako može izgledati jedan testebenč ukoliko se koristi Avalon Verification IP Suite za verifikaciju dizajna. Može se naći linku