

DIJAGNOSTIKA I TESTIRANJE DIGITALNIH SISTEMA SA PROGRAMABILNIM LOGIČKIM KOLIMA

Milan Nikolić, Veljko Malbaša

Fakultet tehničkih nauka, Institut za energetiku, elektroniku i telekomunikacije, Novi Sad

Sadržaj: Rad sa savremenim programabilnim VLSI logičkim kolima zahteva i odgovarajuće metode za dijagnostiku i testiranje, kako u fazi razvoja, tako i u toku eksploatacije, prvenstveno zbog njihove zatvorenosti i nepristupačnosti unutrašnjim signalima i stanjima kola, kao i velike interne složenosti. U toku razvoja, podela kola na manje blokove, sa ili bez određenih modifikacija, a zatim i njihova kvalitetna simulacija ima veliki značaj, jer omogućava sagledavanje i onih detalja koji mogu da naruše rad kola, a u realnom radu ih je nemoguće otkriti. Dodatno ugrađeni blokovi namenjeni za testiranje u realnom radu, iako povećavaju složenost kola, mogu značajno da povećaju dijagnostičke mogućnosti, a time i ubrzaju razvoj kola i testiranje u realnim uslovima.

Ključne reči: Dijagnostika, testiranje, programabilna logička kola

1. UVOD

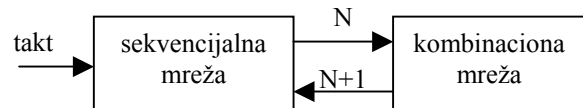
Kao i kod klasičnih elektronskih sistema, dijagnostika i testiranje su nezaobilazni elementi razvoja i eksploatacije digitalnih sistema na bazi složenih programabilnih VLSI logičkih kola. Svakako, veliku razliku u načinu primene ovih elementa kod PLD-a (*Programmable Logic Device*) čini njihova zatvorenost i nepristupačnost unutrašnjim signalima i stanjima kola, kao i velika interna složenost. Zbog toga se, pri razvoju ovakvih sistema, moraju primeniti i odgovarajuće, modifikovane i savremenije metode dijagnostike i testiranja, koje su poznate pod opštim nazivom *Design-For-Test* (DFT) [1]. Neke od ovih metoda su: automatsko generisanje test vektora (ATPG ili *Automatic Test Pattern Generation*), ugrađeni sklopovi za testiranje (BIST ili *Built-In-Self-Test*) i testiranje metodom skeniranja (*Scan-Based Testing*) [1].

U ovom radu su prikazani neki načini implementacije blokova koji su primenljivi kako u fazi razvoja, tako i tokom eksploatacije. Sem toga, date su i određene preporuke korisne u fazi razvoja.

2. DIJAGNOSTIKA I TESTIRANJE U FAZI RAZVOJA

U fazi razvoja digitalnog sistema sa PLD-om, slično kao i pri razvoju softvera, osnovni princip je

projektovanje jednostavnijih pod-blokova i njihovo spajanje u konačnu celinu nakon funkcionalne provere rada. Ovo je posebno važno kod primene simulatora, jer povećanje složenosti simuliranog bloka povećava trajanje simulacije i korišćene resurse računara. Sem toga, vremenski interval u simulaciji je daleko duži nego isti interval u realnom vremenu, što praktično onemogućava simulaciju dužih vremenskih sekvenci. Tako na primer, pri simulaciji brojača od 25 flip-flova sa ulaznim taktom od 20 Mhz i sa intervalom simulacije od 1 ms, u Altera MaxPlus+2 programu na PC računaru P4/1.6 Mhz, simulacija traje oko 16 sekundi, uz izlazni fajl od 640 KB. Za interval simulacije od jedne sekunde, samo postavljanje vrednosti ulaznog takta traje oko tri minuta, a simulacija preko četiri sata, uz veličinu izlaznog fajla od 640 MB. Jedan od razloga se krije u konstrukciji brojača (slika 1), koji je najčešće sastavljen

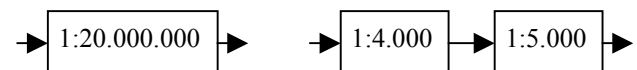


Sl. 1. Sastavni elementi brojača

od niza flip-flova (sekvencijalni deo) i sabirača (kombinacioni deo), pri čemu kombinaciona mreža može biti vrlo složena. Simulator mora računati stanja svih čvorova, nakon svake promene taktnog signala, imajući u vidu sve vrste kašnjenja signala, tolerancije u propagaciji osnovnih elemenata, hazardna stanja i slično.

Iz tih razloga, uz podelu na manje blokove, može se koristiti metoda ubrzanog vremena namernim skraćivanjem intervala internih vremenskih baza, odnosno skraćivanjem modula deljenja ulazne frekvencije do nivoa na kome je realno moguće izvršiti simulaciju.

Drugi način je podela frekventnog delitelja na više manjih delitelja koji se pojedinačno lako simuliraju (slika 2). U datom primeru, podela na dva delitelja, jedan sa

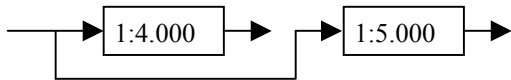


Sl. 2. Podela delitelja na manje blokove

modulom 4000 i drugi sa modulom 5000 je sasvim

prihvatljivo rešenje, jer se trajanje simulacije skraćuje 5000 odnosno 4000 puta u odnosu na kompletan delitelj. Ovaj način podele može se zadržati i u finalnoj verziji, jer ukupna funkcija ostaje ne promenjena.

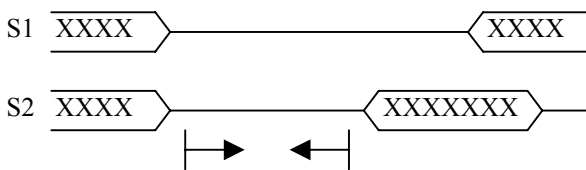
Simulacija ovako podeljenog frekventnog delitelja može se izvršiti i istovremeno, korišćenjem zajedničkog ulaznog takta za oba bloka delitelja (slika 3), ali je ovo



Sl. 3. Paralelna simulacija blokova brojača

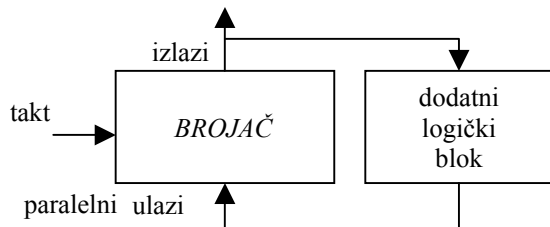
samo privremeno rešenje koje se koristi u fazi simulacije, odnosno razvoja.

Sa druge strane, pri testiranju sekvenci koje sadrže duže intervale nepromenljivih stanja, simulacija se može izvršiti skraćivanjem ovih pauza, slika 4. U ovom



Sl. 4. Skraćivanje trajanja simulacije

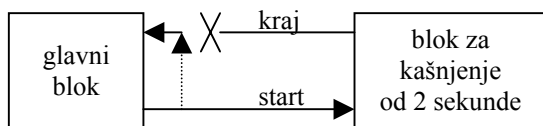
primeru signali S1 i S2 su promenljivi u intervalu označenom sa X, dok su nepromenljiva stanja označena linijom. Strelice označavaju način i mesto skraćivanja trajanja pauze, tj. ubrzanja simulacije. Ovo skraćivanje najčešće zahteva dodatnu kombinacionu mrežu koja je suvišna u finalnoj verziji. Zbog toga je implementaciju ove mreže potrebno izvršiti sa što manje izmena bloka koji se testira, na primer preskakanjem određenih stanja u generatoru vremenske baze. Kao što se na slici 5 vidi,



Sl. 5. Korekcija generatora vremenske baze

dodatni logički blok za korekciju rada generatora vremenske baze projektovan je tako da je njegovo uklanjanje vrlo jednostavno, bez opasnosti od narušavanja funkcionalnosti nakon toga. Sa druge strane, simulacijom su obuhvaćeni samo intervali u kojima postoji promena stanja, što znatno skraćuje trajanje simulacije.

Ova metoda je iskorišćena pri simulaciji uređaja za serijsku komunikaciju [3]. Pri startovanju, uređaj pokreće komunikaciju, pri čemu se prve 2 sekunde šalju nulte vrednosti, a tek nakon toga počinje slanje korisnih



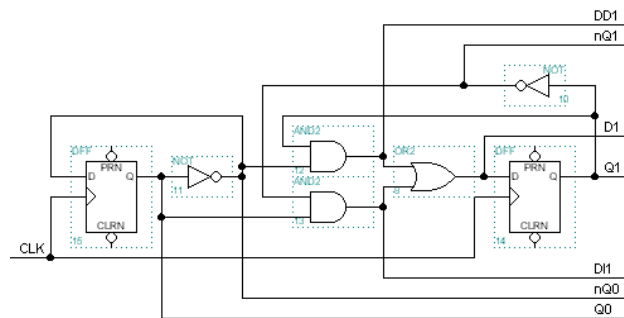
Sl. 6. Isključenje bloka za kašnjenje radi simulacije

podataka. Za svrhe simulacije, blok koji generiše ovo

kašnjenje je privremeno isključen, što je omogućilo simulaciju kratkog perioda prenosa korisnih podataka (slika 6).

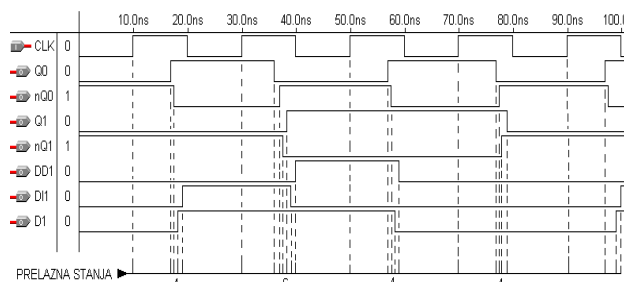
Iako ukljanjanje dodatnog logičkog bloka sa slike 5, koji je dodat radi testiranja u fazi projektovanja, funkcionalno ne narušava rad kola, postoji izvesna mogućnost promene funkcionalnosti u slučajevima kritičnih vremenskih intervala. Razvojni program koji se koristi za projektovanje digitalnog sistema za primenjeni PLD može, nakon uklanjanja dodatnog logičkog bloka, na drugi način da izvrši realizaciju putanja unutar kola, pa time i da promeni pojedina kašnjenja, što kao rezultat može imati poremećenu funkcionalnost. Zbog toga je potrebno izvršiti proveru kritičnih putanja i njihovo poređenje sa i bez dodatne logike.

Na slici 7 prikazan je dvo-bitni sinhroni brojač na



Sl. 7. Primer dvo-bitnog brojača

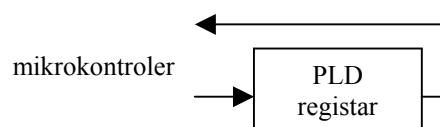
više, dok je odgovarajući prikaz talasnih oblika dat na slici 8, iz koje se jasno vide promene svih internih



Sl. 8. Talasni oblici dvo-bitnog brojača

čvorova, kao i da broj promena nakon prednje ivice takta nije uvek isti. Svaka ova promena se mora računati u simulatoru, što i jeste razlog velikog trajanja simulacije. Primer na slici 8 predstavlja talasne oblike za najgori slučaj, koji uključuje sve tolerancije kašnjenja.

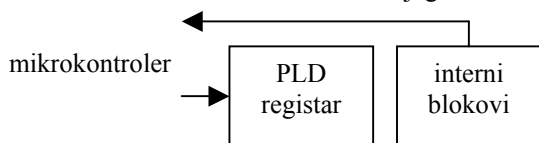
Sem u retkim slučajevima, PLD radi u kooperaciji sa mikrokontrolerom ili mikroprocesorom. U takvom sistemu mikrokontroler se koristi kao 'host' sklop koji, sem ostalih funkcija, vrši i nadzor rada PLD-a. Ova osobina se može iskoristiti i u fazi razvoja. Naime, prvi korak rada na realnom hardveru je provera samih veza između mikrokontrolera i PLD-a. Kako u takvoj konfiguraciji PLD mora imati neku vrstu interfejsa prema



Sl. 9. Testiranje komunikacije µC-PLD

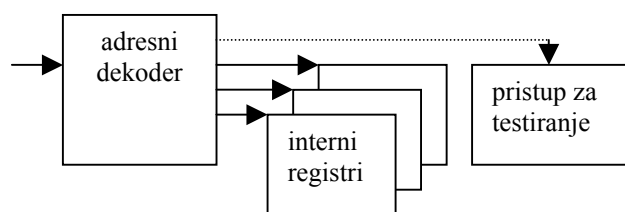
mikrokontroleru, osnovna ideja je iskorišćavanje dvosmerne komunikacije između ovih kola za svrhe testiranja. U tom smislu, prvi test u fazi testiranja je baš provera ove komunikacije, slika 9. Svaki registar upisnog tipa unutar PLD-a treba da poseduje mogućnost čitanja upisanog sadržaja. Dodatni logički sklop koja omogućava ovo čitanje je najčešće vrlo jednostavan i predstavlja malu potrošnju internih resursa PLD-a, a ima znatan doprinos u testiranju i dijagnostici rada. Ovakav dodatni logički sklop pripada klasi BIST slopova i primenljiv je i u fazi eksploatacije. Za svako testiranje je neophodno raspolagati bar jednim ispravnim pod-blokom koji se može koristiti u svrhe testiranja. Zbog toga mogućnost testiranja veze između mikrokontrolera (ili nekog drugog 'host' sklopa) i PLD-a ima poseban značaj.

Za svrhe razvoja, primer sa slike 9 se može modifikovati i koristiti kao dodatni dijagnostički alat,



Sl. 10. Dodatne mogućnosti komunikacije μ C-PLD

slika 10. U ovom slučaju, povratna veza od PLD-a ka mikrokontroleru se koristi kao 'prozor' kroz koji se mogu dobiti informacije u unutrašnjim stanjima kola. Ovakva veza se može projektovati i zasebno, ne narušavajući postojeće veze, ali korišćenjem što većeg broja već postojećih linija međusobnih veza μ C-PLD, kao i kontrolnih blokova za pristup PLD-u (slika 11). U



Sl. 11. Korišćenje postojećih resursa za dodavanje dijagnostičkih kola

ovom primeru, postojeći adresni dekodera unutar PLD-a se proširuje dodatnom izlaznom linijom, preko koje se pristupa internim stanjima kola, tj. odgovarajućim dijagnostičkim blokovima. Ovo proširenje adresnog dekodera zauzima male dodatne resurse, jer najveći deo dekoderskog kola već postoji.

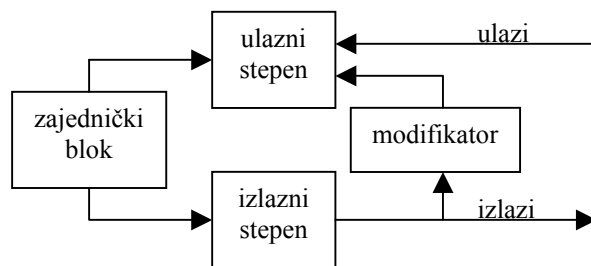
U fazi razvoja može se koristiti i metoda skeniranja, kod PLD-a koji ima fabrički ugrađene odgovarajuće sklopove (na primer JTAG). Ova metoda se koristi da se napravi snimak svih stanja od interesa u određenom trenutku. Kako se informacije o stanjima prenose serijski do 'host' sklopa, što zahteva izvesno vreme, ova metoda se može efikasno koristiti za analizu statičkih stanja. U dinamičkom radu treba imati u vidu da je presek dobijenih stanja vremenski nepouzdan ukoliko se pre snimanja stanja ne izvrši zaustavljanje svih dinamičkih generatora u kolu, kao što su takt generatori. Zbog toga se ova metoda ređe koristi u eksploataciji PLD-a. Sem toga, u slučajevima gde to sam hardver PLD-a dozvoljava, moguće je ne samo čitanje nego i postavljanje stanja internih sekvencijalnih elemenata, pa time i generisanje test vektora i analiza pojedinih kritičnih sekvenci.

Finalna provera dizajna zahteva testiranje na realnom sistemu, što podrazumeva priključivanje svih spoljašnjih veza i proveru svih ulazno-izlaznih signala. Kako u nekim slučajevima spoljašnji uređaji koji se priključuju na PLD nisu uvek dostupni, moguće je primeniti i hardverski simulator spoljašnjeg uređaja [2], koji takođe može biti realizovan pomoću PLD-a. Naravno, kako i ovo kolo treba projektovati i testirati, i ovde je potrebno predvideti adekvatne mehanizme za dijagnostiku i testiranje.

3. DIJAGNOSTIKA I TESTIRANJE U FAZI EKSPLOATACIJE

U ovoj fazi dodatni logički blokovi koji se koriste za dijagnostiku i testiranje unutar PLD-a delimično zavise od primene samog kola. Princip povratne informacije, kao što je opisano u prethodnom poglavlju, je primenljiv uvek kada je PLD povezan sa nekim 'host' sklopom (najčešće mikrokontrolerom ili mikroprocesorom), nezavisno od primene samog PLD-a.

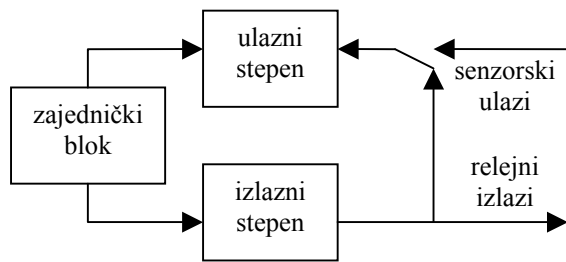
Kako realizovani PLD najčešće ima određeni broj ulaza i izlaza, može se primeniti interna povratna sprega, najčešće poznata pod nazivom *Loop-Back*, slika 12. U



Sl. 12. Realizacija 'Loop-Back' metode

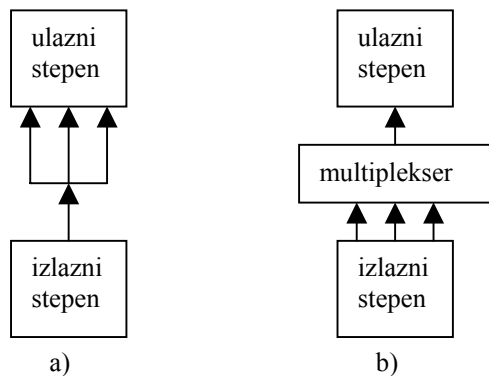
ovom režimu rada ulazni stepen se prespaja na izlazni, direktno ili preko kola za modifikaciju signala. Direktna veza je moguća samo ako su u normalnom radu izlazni i očekivani ulazni signali identični, što je vrlo redak slučaj, ili ako između ulaznih i izlaznih signala ne postoji nikakva zavisnost i pri tome ulazni signali mogu biti proizvoljni. U ostalim slučajevima se mora primeniti odgovarajuće kolo koje će modifikovati izlazne signale tako da rezultat odgovara obliku ulaznih signala. Ovakav modifikator može biti jednostavan, ali i vrlo komplikovan, zavisno od funkcije transformacije izlaznih u ulazne signale. Svrha modifikatora je što je moguće jednostavnija simulacija spoljašnjeg kola na koje su ulazi i izlazi PLD-a povezani, odnosno perifernih sklopova priključenih na ulaze i izlaze, što omogućava punu proveru ulazno-izlazne jedinice bez potrebe za potpuno ispravnim spoljašnjim signalima. Za razliku od eksternog simulatora ili realnog uređaja, testiranje ovom metodom može se vršiti tokom eksploatacije, u pauzama ili pre rada sa realnim sistemom.

Primer direktne interne veze između izlaza i ulaza dat je na slici 13. U ovom slučaju je dovoljan ulazni multiplekser koji obezbeđuje prespajanje ulaza na izlaze. U opštoj varijanti sa slike 12 ovakav multiplekser je sastavni deo ulaznog bloka, pa u ovom slučaju skoro nikakvi dodatni resursi nisu potrebni. Kako broj ulaza i izlaza ne mora biti isti, potpuno testiranje se može izvršiti primenom nekoliko multipleksera kojima se



Slika 13. Realizacija 'Loop-Back' metode direktnom vezom

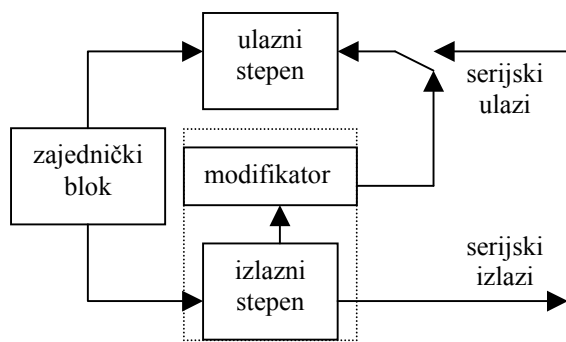
moгу obuhvatiti svi ulazi i svi izlazi (slika 14). U slučaju kada je broj ulaza veći od broja izlaza, više ulaza se spaja



Slika 14. Direktna veza između izlaza i ulaza
a) više ulaza nego izlaza
b) više izlaza nego ulaza

sa jednim izlazom. Ako je slučaj obrnut, koristi se multiplekser ukoliko je potrebno testirati sve izlaze.

Drugi slučaj, koji odgovara međusobno zavisnim ulazima i izlazima prikazan je na slici 15, na primeru serijske komunikacije [3]. Ovde je serijska komunikacija takvog tipa da se nakon svakog poslatog paketa tačno



Slika 15. Realizacija 'Loop-Back' metode indirektnom vezom

određene veličine i proizvoljnog sadržaja očekuje prijem jednog manjeg paketa, takođe određene dužine i proizvoljnog sadržaja. Modifikator koji je ovde primenjen ima za svrhu selektovanje određenog dela predajnog paketa i njegovo prosleđivanje prijemu zakašnjeno, tek nakon završenog slanja predajnog paketa. U toku rada, pored se poslati i primljeni paketi, a na osnovu rezultata moguće je odrediti ispravnost rada i eventualne greške. Realizacija modifikatora kao samostalnog bloka u ovom slučaju može da bude veoma složena. Značajna ušteda resursa PLD-a može se ostvariti ako se ovakav modifikator najvećim delom implementira

u okviru samog izlaznog stepena, tj. kao dodatna varijanta načina rada predajnika. Uređaj serijske komunikacije [3] je realizovan sa tri PLD integrisana kola. Podela je izvršena na predajni, prijemni i kontrolni blok. Logički sklop modifikatora je izveden većim delom kao deo predajnika i zauzima 20, dok preostali deo predajnika zauzima 313 logičkih ćelija od ukupno 336 postojećih, odnosno oko 6% resursa PLD-a. U prijemnom PLD-u je smešten ostatak modifikatora koji zauzima 17, dok preostali deo prijemnika zauzima 267 logičkih ćelija od ukupno 336 postojećih, odnosno 5% resursa PLD-a. PLD u kome je realizovan kontrolni (zajednički) blok nije obuhvaćen elementima modifikatora, pa kod njega nema dodatne potrošnje resursa. Iako je funkcija modifikatora u ovom primeru složena, korišćenjem već postojećih rešenja u predajniku i prijemu, uz minimalno proširenje, postignuta je efikasna implementacija modifikatora uz minimalnu dodatnu potrošnju resursa oba PLD-a. Sa druge strane, mogućnosti koje ovakav modifikator pruža u dijagnostici i testiranju uveliko opravdavaju njegovu ugradnju.

Mogućnosti *Loop-Back* metode mogu biti vrlo značajne. Tako se u uređaju [3], koji predstavlja PC karticu za brzu višekanalnu serijsku komunikaciju, ova metoda koristi za proveru internog sistema detekcije prisustva ulaznog signala, provera ispravnosti ugrađenih memorija, određivanje maksimalne brzine prenosa zbog ograničenja PC računara i operativnog sistema, provera internih detektora grešaka serijske komunikacije (greška višestrukog prijema, greška zakasnelog prijema, greška prekomernog prijema), prekomerni upis u prijemnu FIFO memoriju, prekoračenje čitanja iz predajne memorije i slično [4]. Zahvaljujući konstrukciji modifikacionog kola kao na slici 15, resursi PLD-a potrošeni za ove svrhe predstavljaju manji deo kompletne ugrađene logike.

4. ZAKLJUČAK

Prikazane su neke metode testiranja u razvoju složenih programabilnih logičkih kola. Dato je nekoliko primera poboljšanja simulacije u fazi razvoja digitalnog sistema na bazi PLD-a, kao što su podela složenog bloka na jednostavnije blokove koji se lako simuliraju, skraćivanje trajanja simulacije izbacivanjem pasivnih intervala i podelom sekvencijalnih jedinica na manje blokove. Ugrađivanjem povratne informacije prema host sistemu omogućeno je praćenje izabranih unutrašnjih stanja PLD-a, koja na drugi način nije moguće analizirati. Metoda internog spajanja izlaza i ulaza PLD-a, tkz. 'Loop-Back', omogućava veći broj različitih testova u toku razvoja i eksploatacije.

5. LITERATURA

- [1] Alfred L. Crouch, "Design-for-Test for Digital IC's and Embedded Core Systems", Prentice Hall, 1999.
- [2] Ivan Mezei, Milan Nikolić, "Sistemska verifikacija pomoću FPGA emulatora", ETRAN 2002, 3-6 juni 2002, Teslić, Banja Vrućica, Republika Srpska.
- [3] Milan Nikolić, Veljko Malbaša, Mladen Kezunović, Tomo Popović, "Sistem za testiranje uređaja relejne zaštite u elektrodistribuciji", INDEL 2002, 14-16 novembar 2002, Banja Luka, Republika Srpska.

[4] Milan Nikolić, Veljko Malbaša, Mladen Kezunović, "Primer projektovanja složenog digitalnog sistema sa posebnim mogućnostima za testiranje", INFOTEH-JAHORINA 2003, 27-30 juni 2003., Srpsko Sarajevo, Republika Srpska.

DIAGNOSTIC AND TESTING OF PLD BASED DIGITAL SYSTEMS

Abstract: *Complex digital systems based on VLSI programmable logic devices are hard to test due to the fact that their internal signals and states are not directly accessible from the external pins. To perform diagnostic and testing we need innovative test methods that can be*

applied both during the development and use of PLD based digital systems. A possible approach is to partition the digital system into subsystems to provide for the individual subsystem simulation and testing. The approach facilitates capturing the details that may affect correct operation of the whole system and that are hard to detect during the operation in real environment. Another approach is to add on blocks that are specially designed to support the testing during the real mode of operation. Even though these blocks increase the system complexity, they improve the diagnostic and testing capabilities and thus speed up the system design.

Key Words: *Diagnostic, testing, programmable logic circuits*