

NAPREDNO MODELOVANJE KAŠNJENJA SEKVENCIJALNIH KOLA ZA MONTE-CARLO ANALIZU

Miljana Milić, *Elektronski fakultet, Niš, miljana@venus.elfak.ni.ac.rs*

Sadržaj – Ponašanje digitalnih kola sa stanovišta kašnjenja signala slučajne je prirode. Za predviđanje ponašanja kola neophodno je korišćenje statističkih analiza i simulacija, kao što je to Monte-Carlo analiza. Sa druge strane, to zahteva i veliki broj simulacija kola sa različitim parametrima elemenata. Posebnim modelovanjem kašnjenja logičkih kola moguće je ubrzati Monte-Carlo simulacije pri čemu se umesto električnog, za analizu vremenskog ponašanja koristi logički simulator. U ovom radu predložene su napredne tehnike modelovanja kašnjenja sekvencijalnih logičkih elemenata koje omogućavaju ovo ubrzanje. Posebnim modelima opisuju se različiti parametri kašnjenja sekvencijalnih elemenata, dok je u cilju njihove primene u statističkoj analizi vremenskog ponašanja kola, omogućeno slučajno generisanje tih parametara kašnjenja.

1. UVOD

Da bi se svaka faza u projektovanju integrisanih kola uspešno okončala, potrebno je imati detaljan uvid u vremenske karakteristike svakog dela kola, nakon svake projektantske faze. To podrazumeva primenu valjanih tehnika modelovanja i izračunavanja kašnjenja. Da bi se ove tehnike usavršile, a samim tim i celokupan postupak projektovanja digitalnih integrisanih kola, neophodna je i detaljna analiza pojedinih postupaka radi njihove eventualne primene. Različite tehnike razmatraju različite efekte i uticaje tokom različitih faza projektovanja (tehnologija, korisnički zahtevi, tehnika projektovanja, srazmerno smanjivanje dimenzija itd.) i ostvaruju različite tačnosti.

Kod savremenih tehnika projektovanja integrisanih kola precizno određivanje kašnjenja moguće je tek u završnim fazama projektovanja. Kašnjenja se ekstrahuju posebnim programima iz gotovog leajuta kola. To dovodi do zaključka da je radna frekvencija projektovanog kola nepoznata u toku projektovanja. Ako dobijena brzina rada kola nije zadovoljavajuća, potrebno je ponovo projektovati kolo, što može znatno da poveća cenu projektovanja. Zbog toga je pogodno kašnjenje proceniti u nekoj od ranijih faza projektovanja. Pravovremena predikcija kašnjenja omogućuje projektantu da prepozna pogrešna rešenja i preprojektuje kolo, čime se postiže ušteda u vremenu i ceni.

Umesto zahtevnih i dugotrajnih simulacija na tranzistorskom nivou, npr. SPice simulatorom, korišćenje logičkiog simulatora za određivanje kašnjenja puteva signala u kolu, kada su u pitanju digitalna kola, znatno ubrzava postupak. Međutim, iako prelazak na viši nivo apstrakcije opisa smanjuje vreme potrebno da računar odredi kašnjenja, problem nije rešen. On se sada ogleda u tome da je za procenu kašnjenja svih puteva signala u kolu, potrebno kolo simulirati za sve moguće kombinacije ulaznih vektora, kojih ima previše. Logično je da je i takvo rešenje neprihvatljivo.

Dakle, potreban je metod koji će uz pomoć logičkog simulatora biti u mogućnosti da proceni kašnjenja kola opisanih na nivou gejtova. Pri tome, metod treba da bude dovoljno brz da se pri proračunu kašnjenja puteva signala, ne obazire na logičke nivoe, kao ni na moguće kombinacije pobudnih signala, ali i da istovremeno bude dovoljno precizan, odnosno da koristi sofisticirane modele kašnjenja

gejtova. Takav metod razvijen je u Laboratoriji za projektovanje elektronskih kola na Elektronskom fakultetu u Nišu, pod nazivom SSTA For Slog i opisan je u [1], [2], [3]. Proširenje ovog metoda odnosi se na razmatranje uticaja varijacija procesa proizvodnje, i ugrađivanje generatora slučajnih vrednosti u modele kašnjenja gejtova. Time se obezbeđuje primena istih modela u Monte-Carlo petlji.

U narednom poglavlju, najpre će biti ukratko opisani različiti modeli kašnjenja koji se upotrebljavaju u različitim fazama projektovanja elektronskih kola. VHDL simulator i jezik za opis hardvera mogu se koristiti za Monte-Carlo analizu. Time se ostvaruje ubrzanje Monte-Carlo postupka u odnosu na korišćenje standardnih električnih simulatora na tranzistorskom nivou. O metodu SSTA For Slog biće više reči u trećem poglavlju. U [1] je detaljno opisana primena metode na kombinacionim i asinhronim kolima. U četvrtom poglavlju, biće razvijeni precizni modeli kašnjenja sekvencijalnih digitalnih kola za SSTA For Slog statističku analizu kašnjenja. Na kraju sledi VHDL implementacija razvijanih modela i zaključak.

2. MODELI KAŠNJENJA DIGITALNIH KOLA

Kašnjenje se definiše kao vremenski interval između pojavljivanja događaja na ulazu nekog kola i pojavljivanja događaja na izlazu tog kola kao posledica događaja na ulazu [4], [5]. Ukupno kašnjenje kola zavisi od dve komponente: kašnjenja gejtova i kašnjenja veza.

Postoji mnogo modela kašnjenja u digitalnim kolima. Oni se razlikuju po tačnosti i upotrebi; različita izračunavanja zahtevaju različite mehanizme uključene u opis kašnjenja gejta. Ako razmatramo logičku simulaciju nekog digitalnog kola, najjednostavniji model kašnjenja koji možemo da upotrebimo je model nultog kašnjenja. Ovako opisani gejtovi koriste se samo za verifikaciju logičke funkcije simulacijom. Sledeći, takođe veoma jednostavan model jeste model jediničnog kašnjenja. Kod njega se pretpostavlja da rastuća i opadajuća ivica izlaznog signala kasne tačno jednu jedinicu vremena u odnosu na trenutak promene ulaznog signala. Model kašnjenja kojim se definišu različita kašnjenja prednje i zadnje ivice izlaznog signala nešto je komplikovaniji, ali je mnogo bliži realnosti. Kao posledica tolerancija u tehnološkom postupku izrade kola, kao i promena u okruženju, ovaj model treba proširiti tako da se umesto fiksnih vrednosti, kašnjenja rastuće i opadajuće ivice izlaznog signala gejta definišu kao opsezi. To znači da se vrednost kašnjenja rastuće ili opadajuće ivice izlaznog signala može naći između određene minimalne i maksimalne vrednosti.

Dalji razvoj modela kašnjenja uzima u obzir i opterećenje posmatranog gejta. Ukoliko je izlaz jednog gejta vezan na ulaz jednog gejta, njegovo kašnjenje je manje nego kada bi pobuđivao ulaze dva ili više gejtova istovremeno. Ovo nas dovodi do modela kašnjenja gejta koji zavisi od konkretne vrednosti njegovog fanout-a. Primenom ovakvih modela, polako se u proračun kašnjenja uvodi i konkretna realizacija kola. Da bi se ovakav model mogao koristiti, mora se znati kompletna netlista sistema u kome je instanciran posmatrani gejt. Kašnjenje neke ivice signala na izlazu gejta zavisi

takođe od toga šta je prouzrokovalo tu tranziciju. Nekoliko različitih događaja na ulazu gejtta može da prouzrokuje isti događaj na izlazu tog gejtta. To drugim rečima znači da pojava određene tranzicije na izlazu gejtta može da ima različite uzroke. Tako npr. jedno trouglasto NI kolo, ima tri različita kašnjenja rastuće ivice na izlazu u zavisnosti od toga šta je uzrokovalo tu rastuću ivicu.

Dodatna proširenja modela kašnjenja gejtova odnose se na statističku vremensku analizu (SSTA –Statistical Static Timing Analysis). Modeli kašnjenja koji se koriste za statističke analize ponašanja nekog kola u vremenskom domenu treba da sadrže generatore slučajnih vrednosti sa zadatim funkcijama raspodele verovatnoće. Generisanje slučajnih vrednosti treba obaviti nekoliko puta da bi se na pravi način simulirale realne varijacije parametara gejtova iz masovne proizvodnje. Drugim rečima, ovakvi modeli omogućavaju pokretanje Monte-Carlo analiza.

3. STATISTIČKA PROCENA KAŠNJENJA

Vrednosti parametara elektronskih kola i komponenata predmet su varijacija iz različitih razloga. Od tehnološkog procesa proizvodnje, preko uslova ambijenta tokom njihove eksploatacije, pa do varijacija koje su posledica starenja. Zbog promena vrednosti parametara, odziv projektovanog kola u jednom trenutku može da postane neprihvatljiv. Takođe se može desiti da u uslovima masovne proizvodnje usled neznatnih varijacija parametara, prinos bude manji od 100% i pored toga što u kolima ne postoje defekti. U tom smislu, treba napomenuti da je priroda odstupanja parametara statistička, pa se dobijene vrednosti parametara nalaze u okviru intervala koji se naziva tolerancija parametara. Međutim, i vrednosti odziva komponente čiji su parametri predmet varijacija, sada su slučajne veličine koje pripadaju intervalu koji se naziva tolerancija odziva.

Preslikavanje prostora tolerancija parametara u prostor tolerancija odziva naziva se analiza tolerancija [6]. Ove analize rade se iz dva razloga. Prilikom projektovanja složenog elektronskog sistema, ukoliko dobijeno kolo ne zadovoljava stroge zahteve, nema mogućnosti da se proizvedeni sistem koriguje jeftinim postupkom. Umesto ispravljanja grešaka nakon proizvodnje, bolje je pre proizvodnje simulirati kolo sa parametrima koji su predmet varijacija i utvrditi da li dobijeni odziv zadovoljava postavljene zahteve. Druga situacija odnosi se na masovnu proizvodnju gde je jedan od najvažnijih zahteva postizanje niske cene. Korišćenje komponenata i tehnologija sa širokom tolerancijom parametara, jeftinije je.

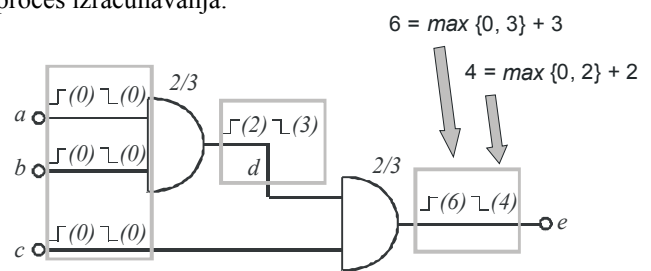
Za analizu tolerancija koriste se metode uzorkovanja i direktne metode. Kod direktnih postoje formule za preslikavanje tolerancija parametara u tolerancije odziva. Oni se upotrebljavaju u slučajevima kada su priraštaji parametara mali. Među metodama uzorkovanja prepoznaje se najmoćniji metod statističkog uzorkovanja pod imenom Monte-Carlo. On teži da simulira prirodni fenomen uzorkovanja koji se dešava tokom montaže elektronskih uređaja.

SSTA For Slog metod kao parametre razmatra isključivo kašnjenja logičkih gejtova i njihov uticaj na kašnjenja puteva kompleksnih digitalnih kola. On predstavlja algoritam statističke procene najnepovoljnijih slučajeva kašnjenja svih puteva u jednom kolu. U suštini to je Monte-Carlo analiza koja kao parametre tretira samo različite tipove i aspekte kašnjenja, pri čemu se kao medijum za simulaciju koristi logički simulator. I ovaj metod zasnovan je na primeni nekoliko stotina simulacija, istovremeno. Tačan broj simulacija određen je kao i kod Monte-Carlo analize, tj.

zahtevanom preciznošću i tačnošću rezultata. Podrazumeva se da je kolo opisano na strukturnom nivou i da su u modelima gejtova dostupni opsezi kašnjenja svih gradivnih blokova za rastuće i opadajuće ivice signala. Sve vrednosti najnepovoljnijih kašnjenja jednog gejtta, slučajno se generišu, saglasno sa datom standardnom devijacijom, koju je moguće menjati. Vrednost standardne devijacije zadaje se na osnovu tolerancija parametara za zadata tehnologiju izrade integrisanih kola.

Simulacije po SSTA For Slog metodi ne zahtevaju razvoj ni primenu nikakvih pobudnih vektora ili sekvenci, jer se unutar procesa procene kašnjenja u opisima gejtova ne razmatraju neke pojedine, već sve moguće tranzicije signala. U praktičnom smislu to ne znači da se kolo pobuđuje svim mogućim kombinacijama pobudnih vektora, već se u svakom gejtu pojedinačno razmatraju sve moguće kombinacije i biraju one najnepovoljnije. Tokom jednog prolaska simulatora kroz celo kolo, dobijaju se istovremeno minimalna i maksimalna kašnjenja svih rastućih i opadajućih ivica svih puteva signala kroz kolo.

Osnovni princip akumulacije kašnjenja u jednom prolasku simulatora kroz kolo prikazan je na Sl. 1. Tu je ilustrovan proces izračunavanja maksimalnih kašnjenja do svih izlaza gejtova kola. Istovremeno se svi ulazi pobuđuju rastućim i opadajućim ivicama signala. Informacija o kašnjenju prostire se paralelno kroz sve topološke nivoe kola, čineći pri tome front izračunavanja, sve dok se ne dođe do poslednjeg topološkog nivoa, tj. do izlaza kola. Tada prestaje proces izračunavanja.



Slika 1. Ilustracija principa akumulacije u procesu određivanja maksimalnog kašnjenja do svih izlaza kola

Treba napomenuti da kašnjenje svakog tipa gejtta u svakoj iteraciji analize nije konstantno, već se ono slučajno generišu po zadatoj raspodeli (npr. Gauss-ovoj). Time se omogućava statistička analiza rezultata.

4. NAPREDNI MODELI KAŠNJENJA SEKVENCIJALNIH ELEMENATA

Modeli kašnjenja gejtova svoj rad zasnivaju na posebno modelovanim signalima. Signali u gejtovima ne sadrže logičke vrednosti, što bi se koristilo u slučaju funkcionalne simulacije i analize kola. U njima je dostupna samo informacija o kašnjenjima i vremenskim trenucima događaja koji iniciraju izračunavanja kašnjenja u gejtovima iz netliste.

Postupkom procene kašnjenja logičkim simulatorom za svaki signal S u logičkom sistemu, procenjuju se 4 tipa kašnjenja [7], [8], [1]:

- kašnjenje najkraćeg puta za rastuću ivicu signala S
- kašnjenje najkraćeg puta za opadajuću ivicu signala S
- kašnjenje najdužeg puta za rastuću ivicu signala S
- kašnjenje najdužeg puta za opadajuću ivicu signala S.

Sve ove informacije sadržane su u signalu S u vidu atributa. Pored njih, neophodno je obezbediti mehanizam za aktiviranje izračunavanja svih tipova kašnjenja u svakom gejtu, kada se za to steknu pravilni uslovi. To je realizovano sa dodatnih četiri atributa, koji predstavljaju flegove, i to:

- fleg pristizanja informacije o kašnjenju rastuće ivice duž najkraćeg puta
- fleg pristizanja informacije o kašnjenju opadajuće ivice duž najkraćeg puta
- fleg pristizanja informacije o kašnjenju rastuće ivice duž najdužeg puta
- fleg pristizanja informacije o kašnjenju opadajuće ivice duž najdužeg puta.

Dodeljivanje konkretne vrednosti kašnjenja logičkom elementu, kompleksan je zadatak. Postoje dve komponente ovog kašnjenja. Jedna komponenta uzima u obzir činjenicu da se takav model logičkog elementa koristi za statističku procenu najnepovoljnijih slučajeva kašnjenja svih puteva signala u kolu. Zbog toga je neophodno uvesti "slučajnost" određivanja kašnjenja, pa zato parametri kojima se definiše svaki tip kašnjenja svakog od logičkih kola pojedinačno, predstavljaju u stvari srednju vrednost raspodele verovatnoće.

Druga komponenta kašnjenja mora da vodi računa o tome da se u razmatranje uzima i konkretna realizacija kola, odnosno položaj pojedinih gejtova unutar celog kola. Naime, dobro je poznato da kašnjenje izlaznog signala jednog gejta zavisi od broja gejtova koji se pobuđuju posmatranim gejtom. Ovakvim proširenjem, informacija o fanout-u svakog gejta uključuje se u proces izračunavanja kašnjenja, čime se znatno dobija na tačnosti celokupnog metoda.

Kada se radi o modelovanju kašnjenja konkretnih sekvencijalnih elemenata, pristup je takav da se konačno kašnjenje izlaznog signala računa kao zbir kašnjenja pristiglih na ulaze flipflopa, i novogenerisanih slučajnih vrednosti kašnjenja odgovarajućih ivica. Pri tome se u razmatranje uzima i vrednost fanout-a konkretnog sekvencijalnog elementa. U našem slučaju, razvijeni su modeli kašnjenja za 4 sekvencijalna elementa. To su DFF, TFF, RS-latch i JKFF. Svi modeli razvijeni su u VHDL jeziku za opis hardvera. Više detalja o svakom od njih biće dato u narednom poglavlju, gde je detaljno opisana njihova konkretna realizacija.

5. VHDL IMPLEMENTACIJA

VHDL jezik za opis hardvera pruža jednostavna rešenja za implementaciju modela kašnjenja. Signali koji ne nose logičku vrednost, već informacije o kašnjenju mogu se implementirati korišćenjem kompozitnog tipa signala, dok se informacije o kašnjenjima unutar signala opisuju u vidu atributa. Na Sl. 2 prikazana je VHDL implementacija D flip flopa. U opisu entiteta flip flopa, vidi se da ovaj sekvencijalni element karakteriše 6 tipova kašnjenja (6 generika): kašnjenja taktnog signala i kašnjenja uspostavljanja signala sa ulaza za podatke. Treba primetiti da ovdje ne postoje specifični uslovi za aktiviranje procesa izračunavanja, zbog toga što ova kola imaju jedan ulaz za podatke koji je od interesa za prostiranje signala kroz njega.

Na Sl. 3 prikazan je opis T flip flopa. I on ima samo jedan ulaz za podatke, pri čemu se stanje izlaza komplementira dolaskom svake opadajuće ivice ulaznog signala. Ovaj flip flop nema taktni signal, pa zato ima manji broj generika.

Na Sl. 4, prikazan je deo VHDL opisa kašnjenja RS-latch kola. Tu se situacija prilično komplikuje. Najpre treba pretpostaviti da kolo ima dva ulaza, R i S, i dva izlaza, Q i njegov komplement NQ. Na početku opisa, može se primetiti da kolo u svojoj listi ima 18 različitih generika. Njih čini 16 generika za različite tipove kašnjenja i to minimalno i maksimalno kašnjenje između četiri moguće kombinacije ulaz-izlaz (R-Q, R-NQ, S-Q i S-NQ) i za obe moguće

tranzicije signala. Preostala dva generika (prvi i drugi generik u listi) predstavljaju vrednosti fanout-a za oba izlaza RS-latch-a. Kao i kod opisa ostalih gejtova, ove dve vrednosti inicijalno su jedinične. Prilikom instanciranja ovog kola, ove vrednosti dobiće konkretne cele brojeve koji odgovaraju pravoj topologiji kola.

```
entity DFF is
generic (ifo_izl: integer:= 1;
tr_ck_qmn : real := 1.0e-9;
tf_ck_qmn : real := 0.9e-9;
tsu_d_ckmn : real := 0.45e-9;
tr_ck_qmx : real := 1.05e-9;
tf_ck_qmx : real := 0.95e-9;
tsu_d_ckmx : real := 0.55e-9);
.
architecture only of DFF is
begin
p1:process (ck.d0mn,ck.d1mn,ck.arr0mn,ck.arr1mn,
ck.d0mx,ck.d1mx,ck.arr0mx,ck.arr1mx)
.
multipl := real(ifo_izl);
f<=fanout_func(multipl);
i:= (f* tr_ck_qmn + (0.03*(gauss_rng)));
j:= (f* tf_ck_qmn + (0.03*(gauss_rng)));
k:= (f* tsu_d_ckmn + (0.03*(gauss_rng)));
l:= (f* tr_ck_qmx + (0.03*(gauss_rng)));
m:= (f* tf_ck_qmx + (0.03*(gauss_rng)));
n:= (f* tsu_d_ckmx + (0.03*(gauss_rng)));
q.arr1mn <= true;
q.arr0mn <= true;
q.d1mn <= ck.d1mn + i + k;
q.d0mn <= ck.d1mn + j + k;
q.arr1mx <= true;
q.arr0mx <= true;
q.d1mx <= ck.d1mx + l + n;
q.d0mx <= ck.d1mx + m + n;
end process;
end only;
```

Slika 2. VHDL implementacija kašnjenja D flip flopa

Opis arhitekture kola ima dva procesa, jedan za određivanje svih maksimalnih tipova kašnjenja (proces p1), i jedan za određivanje svih minimalnih tipova kašnjenja. Tako će se rastuća tranzicija na izlazu Q dogoditi ako se se dese opadajuće tranzicije na oba ulaza R i S. Maksimalno kašnjenje rastuće ivice na izlazu Q se onda izračunava kao zbir maksimuma između maksimalnih kašnjenja opadajućih tranzicija pristiglih na ulazima R i S, i maksimuma od slučajnih novogenerisanih maksimalnih kašnjenja rastućih tranzicija između portova S i Q, i portova R i Q. Slični uslovi moraju se uspostaviti za postizanje opadajuće ivice signala na izlazu NQ, pri čemu se maksimalna kašnjenja opadajuće ivice na ovom izlazu računaju korišćenjem drugih vrednosti. Opadajuća tranzicija signala na izlazu Q dešava se ukoliko se istovremeno na ulazu R javila rastuća ivica, a na ulazu S opadajuća ivica signala. Maksimalno kašnjenje opadajuće ivice signala na izlazu Q, izračunava se kao zbir maksimuma između maksimalnog kašnjenja rastuće ivice pristigle na ulaz R i maksimuma kašnjenja opadajuće ivice pristigle na ulaz S, i maksimalnog slučajno novogenerisanog kašnjenja opadajuće tranzicije između portova S i Q, i između portova R i Q. I u ovom slučaju, da bi se postigla rastuća ivica na izlazu NQ, moraju biti uspostavljeni slični uslovi. Analogni procesi važe i za izračunavanje minimalnih tipova kašnjenja.

Najkomplikovaniji model kašnjenja ima JK flip flop. Tu ukupno kašnjenje određuju kašnjenja taktnog signala i kašnjenja koje ima sam flip flop. Informacija o kašnjenjima signala koji dolaze na ulaze J i K nemaju uticaj na ukupno kašnjenje flip flopa. U modelu kašnjenja JK flip flopa ima 20 generika: 16 generika za različite tipove kašnjenja i to minimalno i maksimalno kašnjenje između četiri moguće kombinacije ulaz-izlaz (J-Q, J-NQ, K-Q i K-NQ) i za obe moguće tranzicije signala (t_r za rastuću i t_f za opadajuću ivicu); dva generika predstavljaju vrednosti fanout-a za oba izlaza flip flopa; poslednja dva generika se odnose na min/max kašnjenje opadajuće ivice taktnog signala, kojom se

```

entity TFF is
generic (ifo_izl: integer:= 1;
tr_en_qmn : real := 1.0e-9;
tf_en_qmn : real := 0.9e-9;
tr_en_qmx : real := 1.05e-9;
tf_en_qmx : real := 0.95e-9);
.
.
architecture only of TFF is
.
.
multipl := real(ifo_izl);
f<=fanout_func(multipl);
i:= (f* tr_en_qmn + (0.03*(gauss_rng)));
j:= (f* tf_en_qmn + (0.03*(gauss_rng)));
k:= (f* tr_en_qmx + (0.03*(gauss_rng)));
l:= (f* tf_en_qmx 5 + (0.03*(gauss_rng)));
q.arrlmm <= true;
q.arr0mn <= true;
q.dlmm <= t.d0mn + i;
q.d0mn <= t.d0mn + j;
.
.
q.arrlmm <= true;
q.dlmm <= t.d0mn + k;
q.d0mx <= t.d0mx + l;
.
.

```

Slika 3. VHDL implementacija kašnjenja T flip flopa

```

entity RS-latch is
generic (ifo_izl_1: integer:= 1;
ifo_izl_2: integer:= 1;
tr_rq_mn : real := 1.0e-9;
tf_rq_mn : real := 0.9e-9;
tr_rq_mx : real := 1.05e-9;
tf_rq_mx : real := 0.95e-9;
tr_rmq_mn : real := 1.0e-9;
tf_rmq_mn : real := 0.9e-9;
tr_rmq_mx : real := 1.05e-9;
tf_rmq_mx : real := 0.95e-9;
tr_sq_mn : real := 1.0e-9;
tf_sq_mn : real := 0.9e-9;
tr_sq_mx : real := 1.05e-9;
tf_sq_mx : real := 0.95e-9;
tr_snq_mn : real := 1.0e-9;
tf_snq_mn : real := 0.9e-9;
tr_snq_mx : real := 1.05e-9;
tf_snq_mx : real := 0.95e-9);
.
.
architecture only of RS-latch is
.
.
p1:process(r.d0mx,r.dlmm,r.arr0mx,r.arrlmm, s.d0mx,
s.dlmm, s.arr0mx, s.arrlmm)
.
.
multipl1 := real(ifo_izl1);
multipl2 := real(ifo_izl2);
f1<=fanout_func(multipl1);
f2<=fanout_func(multipl2);
i:= (f1* tf_rq_mx + (0.03*(gauss_rng)));
j:= (f1* tr_sq_mx + (0.03*(gauss_rng)));
k:= (f2* tf_rmq_mx + (0.03*(gauss_rng)));
l:= (f2* tr_snq_mx + (0.03*(gauss_rng)));
if (r.arr0mx and s.arrlmm) then
q.dlmm<=max(r.d0mx,s.dlmm)+max(i, j);
q.arrlmm <= true;
nq.d0mx<=max(r.d0mx,s.dlmm)+max(k, l);
nq.arr0mx <= true;
end if;
m:= (f1* tr_rq_mx + (0.03*(gauss_rng)));
n:= (f1* tf_sq_mx + (0.03*(gauss_rng)));
o:= (f2* tr_rmq_mx + (0.03*(gauss_rng)));
p:= (f2* tf_snq_mx + (0.03*(gauss_rng)));
if (r.arrlmm and s.arr0mx) then
q.d0mx<=max(r.dlmm,s.d0mx)+max(m, n);
q.arr0mx <= true;
nq.dlmm<=max(r.dlmm,s.d0mx)+max(o, p);
nq.arrlmm <= true;
end if;
.
.

```

Slika 4. VHDL implementacija kašnjenja RS-latch-a

okida ovaj flip flop. Deo procesa procene maksimalnih kašnjenja izlaznih signala flip flopa dat je jednačinama 1-4.

$$q.dlmm \leq tf_ck_mx + \max[\max(tr_jq_mx, tf_kq_mx), \max(tr_jq_mx, tr_kq_mx)] \quad (1)$$

$$nq.d0mx \leq tf_ck_mx + \max[\max(tr_jnq_mx, tf_knq_mx), \max(tr_jnq_mx, tr_knq_mx)] \quad (2)$$

$$q.d0mx \leq tf_ck_mx + \max[\max(tf_jq_mx, tr_kq_mx), \max(tr_jq_mx, tr_kq_mx)] \quad (3)$$

$$q.ndlmm \leq tf_ck_mx + \max[\max(tf_jnq_mx, tr_knq_mx), \max(tr_jnq_mx, tr_knq_mx)] \quad (4)$$

Rastuća ivica na izlazu JK flip flopa dobija se kada je na J stigla rastuća, a na K opadajuća ivica taktnog signala, ili kada su na oba ulaza stigle rastuće ivice. Da bi se odredilo ukupno maksimalno kašnjenje rastuće ivice signala na izlazu Q (q.dlmm), flip flopa, najpre treba uračunati kašnjenje opadajuće ivice taktnog signala (tf_ck_mx), a zatim izabrati maksimum između dve moguće kombinacije puteva signala od ulaza do izlaza koji daju rastuću ivicu na Q.

6. ZAKLJUČAK

U radu su razvijeni novi VHDL modeli sekvencijalnih logičkih kola za Monte-Carlo, odnosno SSTA For Slog analizu. Njih odlikuje jednostavnost za primenu u logičkom simulatoru, ali i dovoljna složenost kojom bi se obezbedila tačnost procena kašnjenja. Primenom ovakvih modela i metoda, moguće je rano proceniti kašnjenja složenih digitalnih sistema, ali se rezultati isto tako mogu iskoristiti za otkrivanje defekata kašnjenja i loših projektantskih rešenja.

LITERATURA

- [1] M. Milić, „*Određivanje statistički najnepovoljnijeg slučaja kašnjenja u digitalnim kolima upotrebom logičkog simulatora*“, Doktorska disertacija, Elektronski fakultet u Nišu, 2009.
- [2] M. Sokolović, V. Litovski, M. Zwolinski, “Efficient and realistic statistical worst case delay computation using VHDL”, *Electrical Engineering*, Vol. 91, No. 4-5, December 2009., pp. 197-210.
- [3] M. Sokolović, V. Litovski, M. Zwolinski, “New concepts of worst case delay and yield estimation in asynchronous VLSI circuits”, *Microelectronics Reliability*, Vol. 49, Issue 2, Feb. 2009., pp. 186-198.
- [4] M. Sokolović, M. Damnjanović, “Digital circuits delay analysis,” *Zbornik radova 51. konferencije ETRAN-a*, Herceg Novi, 2007, (na CD-u).
- [5] M. Sokolović, V. Litovski, “Using VHDL simulator to estimate logic path delays in combinational and embedded sequential circuit”, *Proc. of the International Conference on Computer as a Tool, EUROCON 2005*, Belgrade, Nov. 2005, pp. 1683-86.
- [6] V. Litovski, „*Projektovanje elektronskih kola*, DGIP “Nova Jugoslavija“ – Vranje, Niš, 2000.
- [7] J. Ousterhout, “Crystal: A Timing Analyzer for NMOS VLSI Circuits”, *Proc. of the 3rd Caltech Conference on VLSI*, Caltech, March 1983. pp. 57-69.
- [8] D. Maksimović, V. Litovski, “Tuning Logic Simulators for Timing Analysis”, *Electronic Letters*, Vol. 35, No. 10, May 1999, pp. 800-802.

Abstract – Digital circuits delay behavior has a random nature. To estimate the circuit’s behavior, statistical analysis and simulations must be used, such as Monte-Carlo analysis. On the other hand, it requires large number of circuit simulations with different element parameters. When we apply a special modeling of logic circuit delay, it’s possible to speed up Monte-Carlo simulations by using a logic simulator for timing behavior analysis instead of an electrical simulator. This paper suggests some advanced techniques of sequential circuits delay modeling that enables this acceleration. Special models describe different sequential circuits delay parameters, while, in order to use them for statistical analysis of the timing behavior, random delay generation is enabled.

ADVANCED MODELING OF SEQUENTIAL CIRCUITS DELAY FOR MONTE-CARLO ANALYSIS

Miljana Milić