

ZAVOD ZA ELEKTRONIKU, MIKROELEKTRONIKU, RAČUNALNE I INTELIGENTE SUSTAVE
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
SVEUČILIŠTE U ZAGREBU

DIGITALNA LOGIKA
UPUTA ZA PETU LABORATORIJSKU VJEŽBU
dr.sc. Marko Čupić

Zagreb, 2013.

5. laboratorijska vježba – Sekvencijski sklopovi I

1. Pregled vježbe

Bistabili predstavljaju osnovne memorijske elemente u digitalnim sklopovima te je njihovo modeliranje i ispitivanje tema ove laboratorijske vježbe.

Prije dolaska na laboratorijsku vježbu preporuča se proučiti gradivo vezano uz bistabile te njihovo projektiranje [1,2].

U okviru laboratorijske vježbe potrebno je modelirati dva bistabila te jedan složeniji sekvencijski sklop. Konkretno, potrebno je modelirati i provjeriti sljedeće modele:

- model sinkronog T bistabila sa sinkronim ulazima za postavljanje i brisanje koji reagira na padajući brid signala takta (*sintff*),
- model sinkronog T bistabila s asinkronim ulazima za postavljanje i brisanje koji reagira na padajući brid signala takta (*asintff*),
- model složenijeg sinkronog sklopa (*sekvskl*) definiranog kasnije u sklopu poglavlja *Zadatak*

Za svaki od napisanih modela podrazumijeva se da će biti testirani u odgovarajućem ispitnom modelu.

2. Uvod

Digitalni sklopovi mogu se podijeliti na **kombinacijske** i **sekvencijske** sklopove. **Sekvencijski** sklopovi za razliku od kombinacijskih **sadrže i memorijske elemente** koji služe za pohranu stanja sklopa. **Sekvencijski** sklopovi se nadalje mogu podijeliti na **asinkrone** i **sinkrone**. U asinkronim sklopovima nema globalne koordinacije. Problem asinkronih sklopova je u njihovom dosta teškom projektiranju budući da je tijekom rada sklopa normalna pojava oscilacija signala, kratkotrajne promjene vrijednosti signala, kombinacijske petlje i sl. što lako može dovesti do pogrešnog rada. Alternativu predstavlja uporaba sinkronih sklopova. U **sinkronim** sklopovima sve se promjene odvijaju u točno određenim trenucima koje diktira globalni signal **takta** (signal vremenskog vođenja) – pravokutni signal odgovarajuće periode. Na rastući ili na padajući brid signala vremenskog vođenja odvijaju se sve promjene u sklopu. Maksimalna brzina rada sinkronih sklopova ograničena je brzinom najsporijeg elementa. U sinkronim sklopovima sve se promjene odvijaju sinkrono s rastućim ili padajućim bridom signala takta (ili rjeđe na visoku ili nisku razinu signala takta). Brid na koji se dešavaju promjene naziva se aktivnim bridom. Kako bi sinkroni sklop ispravno radio moraju biti zadovoljeni određeni uvjeti koje postavljaju sekvencijski elementi u sklopu. Ti uvjeti su:

- minimalno vrijeme prije aktivnog brida unutar kojega ulaz sekvencijskog elementa mora biti stabilan – to vrijeme naziva se **vrijeme postavljanja** (engl. setup time),
- vrijeme nakon aktivnog brida tijekom kojega ulazni signal mora ostati stabilan na ulazu sinkronog elementa – to se vrijeme naziva **vrijeme zadržavanja** (engl. hold time).

Osim ta dva vremena postoji i vrijeme **kašnjenja** bistabila, tj. vrijeme potrebno da se promijeni izlaz od trenutka kada se je promijenio ulaz bistabila (i kod sinkronih bistabila kada to signal takta dozvoli). Obično su kao ulazna ograničenja prilikom projektiranja nekog digitalnog sklopa zadana i vremena unutar kojih taj sklop mora obavljati svoju funkciju. Primjerice, zadano je da sklop mora raditi ispravno čak i na frekvenciji od 50 MHz. Odgovor da li projektirani sklop zadovoljava ograničenje brzine može se dobiti tek nakon što se implementira u ciljnoj tehnologiji te se iz te implementacije dobiju podaci o kašnjenjima. U tom trenutku izvodi se dinamička i statička analiza,

te ako se ispostavi da signali ne dolaze u zadano vrijeme do sinkronih elemenata, ili da se ne zadržavaju potrebno vrijeme, sa sigurnošću se može tvrditi da sklop neće raditi te ga je potrebno redizajnirati ili implementirati u nekoj bržoj tehnologiji. U predmetu Digitalna logika ne ulazi se predetaljno u probleme vremenske analize sklopova.

Prilikom modeliranja sinkronog sklopa na jedan od ulaza dovodi se signal vremenskog vođenja, ili signal takta (eng. *clock*). Konvencija je da se taj ulaz označava sa `clk`, `cp`, `clock` ili slično. Potom se taj signal razvodi do svih sekvencijskih elemenata koji čine sklop i na taj način se upravlja svim promjenama unutar sklopa. Osnovno pravilo koje je potrebno poštivati tijekom projektiranja jest da se sa signalom takta ne izvode logičke operacije budući da to može dovesti do neispravnog rada sklopa. Primjerice, nije preporučljivo dovoditi signal takta na I sklop, te potom izlaz I sklopa na sinkronizirajući ulaz bistabila, jer se na taj način uvodi raskorak između vremena okidanja tog i ostalih bistabila u sklopu što može dovesti do neispravnog rada sklopa.

Tijekom generiranja ispitnih valnih oblika za T bistabile, odmah nakon kreiranja ispitnog modula, javlja se dijalog u kojemu je potrebno definirati tip dizajna i neke njegove osnovne vremenske parametre. Potrebno je odabrati sinkroni dizajn te podesiti na koji brid signala takta opisani sklop reagira kako bi ispitna okolina ponudila mogućnost dovodenja pobude u pravim trenutcima.

Nakon što se podese svi potrebni parametri dizajna otvara se prozor za generiranje ispitnih uzoraka unutar kojega je u slučaju sinkronih sklopova već predefinirana vrijednost signala vremenskog vođenja. Ako se pogleda odnos tog signala u odnosu na ulazne signale dizajna moguće je primijetiti da se ulazni signali postavljaju na vrijednosti određeno vrijeme prije aktivnog brida signala takta.

3. Uputa

U okviru prve cjeline potrebno je modelirati dva T bistabila, te jedan složeniji sekvencijski sklop. Prvi T bistabil nazovite `shintff` a drugi `asintff`.

Prije dolaska na vježbu popunite tablicu promjene stanja T bistabila. Prikazane su dvije inačice: sažeta tablica te klasična tablica.

T	Q^{n+1}

Q^n	T	Q^{n+1}

`shintff` predstavlja sinkroni T bistabil sa sinkronim ulazima za postavljanje i brisanje koji reagira na padajući brid signala takta. Sučelje sklopa opisuje tablica u nastavku.

<i>Naziv</i>	<i>Tip</i>	<i>Ulaz/Izlaz</i>	<i>Opis</i>
clk	std_logic	ulaz	signal takta
t	std_logic	ulaz	ulaz T
clr	std_logic	ulaz	ulaz za brisanje (djeluje kada je 1)
st	std_logic	ulaz	ulaz za postavljanje (djeluje kada je 1)
q	std_logic	izlaz	izlaz Q
qn	std_logic	izlaz	izlaz Q-komplement

`asintff` predstavlja sinkroni T bistabil s asinkronim ulazima za postavljanje i brisanje koji reagira na padajući brid signala takta. Sučelje sklopa opisuje tablica u nastavku.

<i>Naziv</i>	<i>Tip</i>	<i>Ulaz/Izlaz</i>	<i>Opis</i>
clk	std_logic	ulaz	signal takta
t	std_logic	ulaz	ulaz T
clr	std_logic	ulaz	ulaz za brisanje (djeluje kada je 1)
st	std_logic	ulaz	ulaz za postavljanje (djeluje kada je 1)
q	std_logic	izlaz	izlaz Q
qn	std_logic	izlaz	izlaz Q-komplement

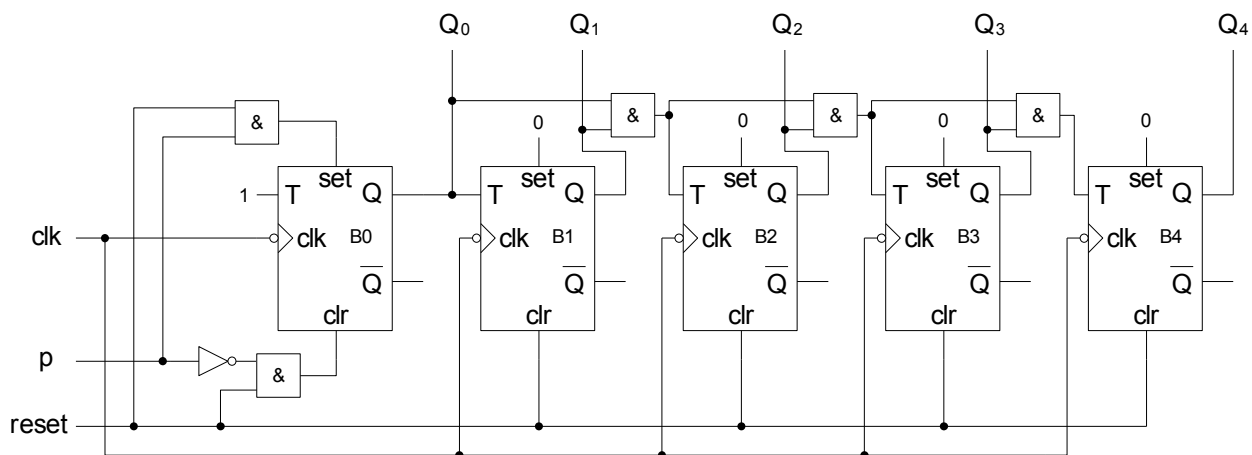
Uočite da je sučelje oba sklopa jednako – ono što se razlikuje jest ponašanje (rad) sklopa. U oba slučaja sklop treba modelirati tako da u slučaju istovremenog djelovanja ulaza za postavljanje i ulaza za brisanje bistabil obavi brisanje.

Jezik VHDL omogućava čitav niz različitih načina modeliranja bistabila. U okviru ove vježbe potrebno je napraviti model koji za čuvanje stanja koristi *varijablu*. Bistabil treba modelirati tako da mu se definira vrijeme kašnjenja od 10 ns.

Sklop **sekvsklop** prikazan je na sljedećoj slici. Temeljem te slike potrebno je napisati odgovarajući strukturni model. Sučelje sklopa definirano je sljedećom tablicom.

<i>Naziv</i>	<i>Tip</i>	<i>Ulaz/Izlaz</i>	<i>Opis</i>
clk	std_logic	ulaz	signal takta
reset	std_logic	ulaz	ulaz za brisanje
p	std_logic	ulaz	???
q	std_logic_vector(4 downto 0)	izlaz	izlaz

Prilikom modeliranja ovog sklopa kao T bistabile koristiti `sintff`. Logičke sklopove ne treba modelirati zasebnim komponentama – koristite ugrađene logičke operatore.



Prije dolaska na laboratorijsku vježbu odgovorite na sljedeće pitanje. Odgovor upišite u polje za upis odgovora.

Pogledajte shemu sklopa `sekvsklop`. Je li taj sklop sinkron ili asinkron? Objasnite!

4. Zadatak

Napišite VHDL modele sklopova `sintff` i `asintff`. Za svaki sklop napravite ispitni sklop i provjerite ispravnost rada.

Potom napišite strukturni VHDL model sklopa `sekvsklop`. Izlaz `q` ovog sklopa tumačite kao binarno zapisani cijeli broj, pri čemu je `q(4)` bit najveće težine. Mapravite ispitni sklop i ispitajte kako se mijenjaju izlazi u sljedeća dva slučaja:

1. na početku postavite `p=0`, `reset=1`; u 100. ns postavite `reset=0` i više ga nemojte mijenjati. Signal takta treba biti simetričan, 50 ns u 1 pa 50 ns u 0. Obavite simulaciju koja sadrži barem 40 padajućih bridova signala `clk`. 20 ns nakon svakog padajućeg brida signala takta očitajte izlaz `q` odnosno broj koji je na njemu. Zapišite taj niz brojeva. Što se događa nakon 32. padajućeg brida signala takta?
2. na početku postavite `p=1`, `reset=1`; u 100. ns postavite `reset=0` i više ga nemojte mijenjati. Signal takta treba biti simetričan, 50 ns u 1 pa 50 ns u 0. Obavite simulaciju koja sadrži barem 40 padajućih bridova signala `clk`. 20 ns nakon svakog padajućeg brida signala takta očitajte izlaz `q` odnosno broj koji je na njemu. Zapišite taj niz brojeva. Što se događa nakon 32. padajućeg brida signala takta?

Možete li zaključiti koju funkciju obavlja modelirani sklop? Kako ulaz `p` modificira njegovo ponašanje?

Sljed brojeva koji se pojavljuju na izlazu sklopa `sekvsklop`, za točku 1 (popuniti na vježbi)

Sljed brojeva koji se pojavljuju na izlazu sklopa `sekvsklop`, za točku 2 (popuniti na vježbi)

Kako ulaz `p` utječe na rad sklopa? (odgovorite na vježbi)

Važno: sljedeću vježbu (6. laboratorijsku vježbu) morat ćete nastaviti raditi u ovom istom projektu jer ćete trebati VHDL modele sklopova koje ste upravo napisali.

5. VHDL

U laboratorijskoj vježbi, prilikom modeliranja nekih bistabila, neophodno je inicijalizirati interne signale na neku poznatu vrijednost kako bi simulacija uopće uspjela. Inicijalizacija u VHDL-u se obavlja prilikom deklariranja signala i to tako da se doda inicijalizacijski izraz. Primjer deklaracije i inicijalizacije jednobitnog signala je:

```
SIGNAL sigA: std_logic := '0';
```

Vrijednost na koju se signal inicijalizira koristi se isključivo u simulaciji, dok se prilikom sinteze ta vrijednost ignorira.

Tijekom sinteze model sklopa opisan u VHDL-u implementira se upotrebom kombinacijskih i sekvencijskih elemenata. Svaki signal (a također i varijabla) može se preslikati u registar, kombinacijsku mrežu ili se može jednostavno ukloniti iz dizajna. Signal za koji se analizom modela utvrdi da mu se dodjeljuje vrijednost u svakom mogućem slučaju implementira se upotrebom kombinacijske mreže, dok se signali za koje se utvrdi da im se ne dodjeljuje vrijednost u svakom slučaju – a istovremeno se čita vrijednost tog signala – implementiraju upotrebom memorijskih elemenata. To drugim riječima znači da je analizom modela utvrđeno kako dotični signal mora čuvati svoju vrijednost, a sposobnost čuvanja, tj. memoriranja, vrijednosti imaju samo memorijski elementi – bistabili.

Primjerice, sintezom sljedećeg VHDL modela generirao bi se bistabil koji pohranjuje vrijednost signala *b*. To je nužno budući da se vrijednost signala *b* koristi kako bi se dodijelila vrijednost signalu *c*, a istovremeno se može dogoditi da signal *a* postane 0, te se deaktivira kod unutar izraza IF.

```
IF (a = '1') THEN
    b <= '0';
END IF;
c <= b;
```

Za razliku od tog ispisa, u sljedećem ispisu ne koristi se memorijski element budući da se signalu *b* uvijek dodjeljuje neka vrijednost i nema potrebe za pamćenjem.

```
IF (a = '1') THEN
    b <= '0';
ELSE
    b <= '1';
END IF;
c <= b;
```

Prethodna razmatranja ponešto se kompliciraju činjenicom da se tijekom sinteze obavljaju različite minimizacije i optimizacije dizajna.

Sinkroni elementi mogu reagirati na vrijednost signala takta ili na njegov rastući ili padajući brid. Sinkroni elementi koji obavljaju svoje operacije na razinu signala (primjerice, pohranjuje ulaznu vrijednost dok je sinkronizirajući signal u logičkoj jedinici) rjeđe se koriste budući da se donekle ponašaju kao asinkroni elementi i mogu se pojaviti zatvorene kombinacijske petlje, što pak može uzrokovati pojavu oscilacija (prisjetite se primjera razinom upravljanog JK bistabila; što se događa u slučaju kada je $J=K=1$ i kada se CP konstantno drži u 1?). Daleko češća je sinkronizacija na rastući ili padajući brid signala takta. U VHDL-u se to modelira vrlo jednostavno. Provjera da li je na nekom signalu rastući brid obavlja se upotrebom funkcije `rising_edge()`, dok se padajući brid detektira funkcijom `falling_edge()`. Te funkcije vraćaju logičku vrijednost *istinito* u slučaju da je detektiran rastući brid na signalu koji se ispituje (`rising_edge`), odnosno padajući brid (`falling_edge`) te se mogu direktno upotrebljavati u izrazu IF. Kako bi se dobio bistabil bitno je da u tom izrazu nisu definirani svi mogući slučajevi. Primjerice, model T bistabila sastoji se od sljedećih linija:

```

IF (rising_edge(clk)) THEN
  IF (t = '1') THEN
    tq := not tq;
  END IF;
END IF;
q <= tq;
qn <= not tq;

```

U navedenom modelu pretpostavlja se da je tq interna varijabla koja pohranjuje stanje bistabila, t je informacijski ulaz, a q i qn su izlaz i njegov komplement. Ovom je prilikom potrebno napomenuti da se korištenje varijabli preferira u odnosu na signale, jer je semantika varijabli daleko shvatljivija te je zbog toga manja vjerojatnost pogreške u modelu. Vrijednost varijable pohranjena je između dva izvršavanja procesa, tj. deklariranje i inicijalizacija varijable obavlja se samo jednom i to prilikom prvog izvršavanja procesa. Također je potrebno pripaziti na način modeliranja jer se može desiti da se ne dobije željeni element. To proizlazi iz nesavršenosti današnjih alata koji se koriste za sintezu. Kao primjer možemo uzeti sljedeći model T bistabila s asinkronim ulazom za brisanje stanja koji će sintezom vrlo vjerojatno biti preveden u dva bistabila od kojih će jedan davati izlaz q , a drugi qn . Razlog takvoj sintezi leži u zasebnom modeliranju izlaza q i qn , gdje nigdje u opisu nije dana veza između njih. Stoga je teško očekivati da će sintetizator primijetiti da bi ta dva izlaza međusobno trebala biti komplementarna, i da je dovoljno pamtit stanje samo jednog od njih.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity t_bistabil is
  Port (
    t, clk, reset: in std_logic;
    q, qn: inout std_logic
  );
end t_bistabil;

architecture Behaviorall of t_bistabil is
begin
  process (t, clk, reset)
  begin
    if (reset = '1') then
      q <= '0';
      qn <= '1';
    elsif (rising_edge(clk)) then
      q <= q xor t;
      qn <= qn xor t;
    end if;
  end process;
end Behaviorall;

```

Međutim, ako se taj isti bistabil opiše korištenjem jedne pomoćne varijable kao što je to prikazano u sljedećem ispisu (ovaj puta samo arhitekture), tada se taj bistabil implementira u samo jednom memorijskom elementu, tj. onako kako je i zamišljeno.

```

architecture Behavioral2 of t_bistabil is
begin
  process (t, clk, reset)
    variable reg: std_logic;
  begin
    if (reset = '1') then
      reg := '0';
    elsif (rising_edge(clk)) then
      reg := reg xor t;
    end if;
    q <= reg;
    qn <= not reg;
  end process;
end Behavioral2;

```


Napomena: u prethodnom primjeru, zbog jednostavnosti, pristupi q i qn sklopa `t_bistabil` modelirani su kao ulazno/izlazni signali. U laboratoriju i u pripremi taj se način modeliranja **ne smije koristiti**.

Prilikom korištenja bistabila za izgradnju složenijih sekvencijskih sklopova može se desiti da neki od **izlaza** bistabila **nisu potrebni**. U tom slučaju moguće ih je ostaviti nepovezanim i to tako da se prilikom definiranja primjerka (instance) komponente na mjestu gdje se nalazi izlazni signal navede rezervirana riječ **OPEN**. Primjerice, neka imamo komponentu `T` bistabila definiranu na sljedeći način:

```
component t_bistabil
  Port (
    clk, t, sn, rn: in std_logic;
    q, qn: out std_logic
  );
end component;
```

Kada tu komponentu koristimo u nekom sklopu, ali pri tome nam nije potreban izlazni signal qn , tada možemo to napisati na sljedeći način:

```
b0: t_bistabil port map (clk, '1', iset, reset_n, bint(0), open);
```

U ovom primjeru qn nije povezan niti na jedan interni ili izlazni signal već “visi” u zraku, što je određeno rezerviranom riječju `open`.

U ovoj vježbi zahtijeva se da bistabili i logički sklopovi imaju određena kašnjenja. Ta kašnjenja, kao i općenito bilo kakvo kašnjenje definirano u nekom modelu, imaju smisao samo prilikom ponašajne simulacije, dok se prilikom sinteze ona zanemaruju te se uzimaju nova kašnjenja koja se dobivaju iz tehnologije u kojoj je implementiran sklop. Modeliranje kašnjenja logičkih sklopova je jednostavno, i već je bilo opisivano u prethodnim laboratorijskim vježbama. Primjerice, troulazni `I` sklop čije kašnjenje treba biti 5 ns bi napisali na sljedeći način:

```
res <= A and B and C after 5 ns;
```

Slična situacija je i kod bistabila. U tom slučaju preporučljivo je definirati unutar procesa internu varijablu koja sadrži vrijednost bistabila. Vrijednost te varijable ispituje se i mijenja unutar `IF` ili `CASE` izraza. Pred kraj procesa vrijednost te varijable samo se dodijeli izlaznim signalima q i qn s odgovarajućim kašnjenjem. Na taj način kašnjenje će biti isto za bilo kakve promjene u bistabilu (ponišćavanje stanja, postavljanje stanja, “regularna” promjena vrijednosti na odgovarajući brid signala takta) što je potrebno dobiti u ovoj laboratorijskoj vježbi.

6. Priprema

Na samu vježbu potrebno je donijeti ovu uputu u kojoj sva polja za unos rješenja trebaju biti popunjena. Sva rješenja trebaju biti napisana rukom, običnom ili kemijskom olovkom; VHDL modeli te ispitni sklopovi trebaju biti uneseni u VHDLLab2.

7. Pitanja za vježbu

1. Što znači da je neki digitalni sustav sinkron?
2. Što znači da je neki digitalni sustav asinkron?
3. Što znači da je neki digitalni sklop sinkron?
4. Što znači da je neki digitalni sklop asinkron?
5. Što znači da neki ulaz X digitalnog sklopa djeluje sinkrono?
6. Što znači da neki ulaz X digitalnog sklopa djeluje asinkrono?
7. Koja je razlika između sinkronih i asinkronih ulaza digitalnog sklopa?
8. U koju vrstu ulaza spada ulaz signala takta, u sinkroni ili asinkroni ulaz? Argumentirajte odgovor!
9. Kako u VHDL opisu sklopa možemo prepoznati djeluje li neki ulaz sinkrono ili asinkrono?
10. Kako izgledaju tablice promjene stanja za druge bistabile?
11. Kako izgledaju dijagrami promjene stanja za druge bistabile?
12. Kako izgledaju jednadžbe promjene stanja za druge bistabile?

Literatura:

- [1] Peruško, Glavinić: *Digitalni sustavi*. Školska knjiga, 2005.
- [2] Čupić, *Digitalna elektronika i Digitalna logika. Zbirka riješenih zadataka*. Kigen, 2006.