Jesu li brži sinkroni ili asinkroni sklopovi?
U ovom labosu, brži je sinkroni način rada jer se, u određenom trenutku padajućeg brida clocka, sve promjene dogode paralelno i ispišu na izlazu, dok bi, u asinkronom slučaju, trebalo čekati 5 vremena kašnjenja bistabila B0, 1, 2, 3 i 4 (budući da je izlaz jednog bistabila spojen na clock drugog). Ali, da su bistabili spojeni serijski, u sinkronom bi slučaju trebalo čekat 5 kašnjenja bistabila + još neka određena kašnjenja, a kod asinkronog slučaja bi trebali čekati samo tih 5 bistabila, pa su, u općenitom slučaju, asinkroni sklopovi brži (iako su, ponekad, manje pouzdani).

Koja je razlika između sekvencijskog i kombinacijskog sklopa?
Kombinacijski imaju mogućnost samo generiranja funkcije, dok sekvencijski, uz to, imaju i sposobnost pamćenja stanja, što kombinacijski sklopovi nemaju.

Jesu li korišteni ulazi (clock, set, reset, T), u ovom slučaju, sinkroni ili asinkroni?
Sinkroni su jer su svi T bistabili sinkroni i svi ulazi ovise o clock (svi su izlazi (set, clear, T) unutar naredbe if rising/falling edge (clk), tj. ovise o clk, sinkroni su s njim).

Kako bi se postiglo da, umjesto brojanja unaprijed (0,1, 2, 3…,30, 31, 0, 1, 2…, 30, 31, 0…) postignemo brojanje unatrag (31, 30, 29, …2, 1, 0, 31, 30, …1, 0…)?
Vizualizacije radi, ispiši prva tri broja pri brojanju unaprijed:

00000 (0)
00001 (1)
00010 (2)

A sada unatrag:
11111 (31)
11110 (30)
11101 (29)

Primjeti analogiju: gdje su nule u prvom slučaju, tu su jedinice u drugom slučaju – dakle, potrebno je samo invertirati kraj svakog izlaza Q0, 1, 2, 3, 4 ili, umjesto tog invertitranja, samo spojiti sve izlaze na Q komplement, iz svakog bistabila, umjesto na Q.

Kod 1. točke 4. zadatka, odvija se brojanje koje kreće od 0 → 0,1,2,…30, 31, 0, 1, 2…, a, kod 2. točke 4. zadatka, brojanje kreće od 1 → 1, 2, 3,…30, 31, 0, 1, 2…, 30, 31… Zašto se, u drugom slučaju, ne kreće od 0?
U drugoj točki, zadano je da je p=1 i reset = 1. Pogledaj sliku sklopa, točnije, donji 'I' sklop u kojeg ulaze invertirani p i reset. Invertirani p= 0 i reset = 1 što daje rezultat 0 pa je i clear 0, tj. neće se izvršiti 'čišćenje', stanja bistabila. Pogledaj gornji 'I' sklop u kojeg ulaze p = 1 i reset = 1. Rezultat je 1 pa je i set 1, tj. obavit će se postavljanje stanja bistabila u 1. Dakle, B0 će biti 1, a B1, 2, 3 i 4 će biti 0 jer je početak. Budući da je stanje B4 najveće težine, čita se s desna na lijevo i dobija se 00001 i to je

1. U 1. točki 4. zadatka, set je 0, a clear 1, pa će se izvršiti 'čišćenje' stanja bistabila u 0 pa će biti 00000, što je 0.

Što je vrijeme postavljanja (t setup), što vrijeme kašnjenja (t delay), a što vrijeme zadržavanja (t hold)?

T setup – vrijeme prije brida clocka tijekom kojeg signal treba biti stabilan da bi se upisao u bistabil

T hold – vrijeme kroz kojeg se signal, koji je došao u trenutku brida clocka, mora zadržati da bi sigurno bio upisan u bistabil

T delay – vrijeme koje je potrebno da se izlaz bistabila promijeni od trenutka kad se promijenio ulaz bistabila

Što znače trokutić i kružić ispred clk?

Te oznake ukazuju da je bistabil bridom okidan, tj. da se, isključivo na padajući ili rastući brida clocka (signala takta), zateknuta stanja na ostalim ulazima upisuju na bistabil. Trokut označava upis na rastući brid, a trokut i kružić upis na padajući brid signala takta. Ako nema nikakvih oznaka, bistabil nije bridom okidan i signali, dovedeni na ulaz, upisuju se odmah u bistabil, bez čekanja nekog brida clocka.

Koja je razlika sinkronog i asinkronog sklopa?

Razlika je u povezivanju clk. Kod sinkronog sklopa, clk je, od jedne točke, razveden na svaki pojedini bistabil sklopa (tj. na svim bistabilima je 'isti clk') te se promjene, na svakom pojedinom bistabilu, događaju istovremeno u točno određenim trenucima (na padajućem ili rastućem bridu, ovisno o vrsti clk, o kružiću i trokutu). Kod asinkronog sklopa, izlaz Q jednog bistabila spojen je na clk drugog bistabila, a izlaz Q drugog bistabila spojen je na clk trećeg bistabila itd. Kod ovakvog sklopa, potrebno je čekati zboj svih vremena kašnjenja pojedinih bistabila da bi se promijenilo i dobilo željeno stanje krajnjeg bistabila

Koja je razlika između varijable i signala?

Signal je interni (unutarnja žica), tj. nalazi se unutar sklopa i nije vidljiv okolini sklopa, dok je, kod varijable, obrnuti slučaj. Na sklopu (čipu) mogu postojati nožice upravo za tu varijablu i njenom deklaracijom i promjenom, tijekom rada s bistabilom, opisuje se (pamti) stanje bistabila, umjesto da se, za pamćenje i provođenje stanja, koristi signal. Na neki je način funkcionalno istovjetna internom signalu, ali je shvatljivija kao pojam.

Uvodi se 'noviji' simbol ':=' koji se koristi prilikom inicijalizacije varijable, tj. odmah joj se daje određena početna vrijednost. ':=' se razlikuje od '<=' jer se, kod '<=', izračunata vrijednost s desne strane '<=' (koja ne mora odmah imati inicijaliziranu, poznatu vrijednost, već se računa u ovisnosti o radu sklopa) upisuje u izraz s lijeve strane '<=', dok je, kod ':=', već deklarirana poznata vrijednost.

Izraz 'open' označuje izlaz koji je prisutan, ali se ne koristi, tj. nema svrhu u sklopu pa je to poznato naglasiti i ostaviti ga 'otvorenim' u stanju beskonačne impendancije (ovdje je to slučaj s Q(n)', znak ' označuje komplement)

Kod sintff i asintff, oznaka 'after 10 ns' predstavlja kašnjenje bistabila, tj. vrijeme koje prođe dok varijabla, koja je zapamtila stanje bistabila, preda to stanje izlazu Q bistabila.

Naredba process ukazuje da se, izrazi unutar zagrade, prilikom simulacije nužno odvijaju po redoslijedu.

Kod sekvencijskog sklopa, izrazi s 'port map', 'signal' i deklaracije varijabli predstavljaju strukturne modele. Svi ostali izrazi (if, then, elsif…) koji pridružuju i računaju vrijednosti, jesu ponašajni modeli. Entity je opis sučelja.

Kod postavljanja Testbencha, treba postaviti ove parametre:

Označi kružić uz 'single clock' (clk)
Clock time high (50 ns)
Clock time low (50 ns)
Input setup time (10 ns)
Initial lenght of testbench (pošto mora biti bar 40 padajućih bridova u zadatku 4., 4100 ili više ns, tj. 40 x 100 ns, 100 = vrijeme periode clk, 50 ns je u 0, a 50 ns u 1)

Kod postavljanja reset u 0 ili 1 (1. Ili 2. točka), skripta kaže da se postavi u 0 ili 1 u 100. ns što je nemoguće zbog onog 'after 10 ns' u sintff. Dakle, može se promijeniti u 90. ns ili 190. ns. Promijeni reset u 190. ns i pripazi da sivi naopaki trokutić (smješten na brojevnom pravcu iznad) bude negdje iza 4100. ns kako bi simulacija dovoljno trajala.

Do 10. ns, u simulaciji traje 'UUUUU' što je nedefinirani signal zbog 'after 10 ns' (izlazi bistabila još nisu poprimili određenu vrijednost do 10. ns).

5. str. pripreme:

Sažeta tablica

T	Q(n+1)
0	Q(n)
1	Q(n)'

Q(n)	Q(n+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

Izlazni test.

1.Vrijeme potrebno da se promijeni izlaz sinkronog sklopa od trenutka nailaska aktivnog brida signala takta naziva se:

- vrijeme otpuštanja (engl. release time)
- vrijeme postavljanja (engl. setup time)
- vrijeme kašnjenja (engl. delay time)
- vrijeme zadržavanja (engl. hold time)

(Gore pogledaj definicije)

---

2. Označimo trenutno stanje JK bistabila sa Qn, a sljedeće sa Qn+1. Za kombinaciju J=0 i K=1, sljedeće stanje JK bistabila bit će jednako:

- NOT Qn
- 1
- Qn
- 0

Ponoviti tablicu stanja JK bistabila

Q(n)	Q(n+1)	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

3. Zadan je sljedeći VHDL kod:

```
entity Element is port(
   cp, P, Q: in std_logic;
   Qout: out std_logic);
end Element;

architecture beh of Element is
   signal Qint: std_logic;
begin

   process(cp, P)
   begin
      if falling_edge(cp) then
         if Q= '0' then
            Qint<= not Qint;
         end if;
      end if;
      if P='0' then
         Qint<= '0';
      end if;
   end process;

   Qout<= Qint;

end beh;
```

Označite sve ulaze bistabila koji djeluju asinkrono.

☐ Q

☑ P

Asinkroni su oni ulazi koji nisu obuhvaćeni uvjetom if rising/falling edge, u ovom slučaju, tom naredbom je obuhvaćen Q (ako je brid padajući/rastući onda je Q ___ pa Q djeluje skupa s njim, tj. sinkrono), a p je 'neovisan' o toj naredbi pa je asinkron

4. Označimo s fs maksimalnu frekvenciju rada sinkronog binarnog brojila unaprijed sa serijskim prijenosom, a s fp maksimalnu frekvenciju rada sinkronog binarnog brojila unaprijed s paralelnim prijenosom. Ako promatramo fs i fp za 10-bitno brojilo, tada vrijedi:

- fs može biti jednak fp, a ako nije jednak, onda je veći od fp
- fs je strogo manji od fp
- fs može biti jednak fp, a ako nije jednak, onda je manji od fp
- fs je strogo veći od fp

Ako je prijenos serijski, potrebno je čekati n kašnjenja bistabila (n = broj bistabila u cijelom sklopu), a, ako je prijenos paralelan, potrebno je čekati vrijeme kašnjenja samo 1 bistabila jer se sve promjene, na svakom pojedinom bistabilu, događaju istovremeno, dakle, kod serijskog prijenosa (gdje je frekvencija fs) potrebno je duže vrijeme T, a, prema formuli za frekvenciju (f=1/T), što je vrijeme duže, frekvencija je manja, pa, ako za slučaj fs treba više vremena, u tom će slučaju frekvencija fs biti strogo manja od druge frekvencije fp

5.Zadan je sljedeći VHDL kod:

```
entity Element is port(
   clk, Q, R, S, T: in std_logic;
   Qout: out std_logic);
end Element;

architecture beh of Element is
   signal Qint: std_logic;
begin

   process(Qint, clk, Q, R)
      variable sel: std_logic_vector(1 downto 0);
   begin
      if Q='0' then
         Qint<= '0';
      elsif R='1' then
         Qint<= '0';
      elsif rising_edge(clk) then
         sel:=S&T;
         case sel is
           when "00"=> Qint<= '0';
           when "01"=> Qint<= '0';
           when "10"=> Qint<= not Qint;
           when "11"=> Qint<= not Qint;
           when others=> null;
         end case;
      end if;
      Qout<= Qint;
   end process;

end beh;
```

Označite asinkroni ulaz najvišeg prioriteta.

- T
- R
- Q
- S

Kao prvo, asinkroni su ulazi oni koji nisu obuhvaćeni naredbom if falling/rising edge (nisu ovisni o sinkronizaciji clocka), a to su, u ovom slučaju Q i R, Q je većeg prioriteta (tj. prije se izvršava) jer je, u bloku process (koji sve izvršava redom), naveden prije R, sinkroni ulazi su obavezno obuhvaćeni naredbom falling/rising edge (naredba bi trebala biti i uvučena u kodu)

6.Zadan je sljedeći VHDL kod:

```
entity Element is port(
   cp, L, M: in std_logic;
   Qout: out std_logic);
end Element;

architecture beh of Element is
   signal Qint: std_logic;
begin

   process(Qint, cp, L)
   begin
      if falling_edge(cp) then
         if M= '0' then
            Qint<= not Qint;
         end if;
      end if;
      if L='1' then
         Qint<= '1';
      end if;
      Qout<= Qint;
   end process;

end beh;
```

Na što djeluje signal takta cp?

- (•) na padajući brid
- ( ) na logičku razinu 0
- ( ) na rastući brid
- ( ) na logičku razinu 1

Pogleda se ključna riječ uz cp (falling edge – padajući brid, rising – rastući brid)

7.Kod kojeg/kojih se automata ulazi pišu na lukovima?

- niti kod jednog od ova dva
- samo kod Mooreovog
- samo kod Mealyjevog
- i kod Mooreovog, i kod Mealyjevog

---

8. U ovoj laboratorijskoj vježbi, sklop Dekoder stanja je:

- kombinacijski sklop
- sinkroni sekvencijski sklop
- asinkroni sekvencijski sklop
- ništa od navedenoga

9. Zadan je sljedeći VHDL kod:

```
entity Element is port(
   clk, P, Q, R: in std_logic;
   Qout: out std_logic);
end Element;

architecture beh of Element is
   signal Qint: std_logic;
begin

   process(...)
   begin
      if rising_edge(clk) then
         if R= '1' then
            Qint<= not Qint;
         end if;
      end if;
      if P='0' then
         Qint<= '1';
      end if;
      if Q='0' then
         Qint<= '1';
      end if;
   end process;

   Qout<= Qint;

end beh;
```

Označite sve signale koji čine minimalnu listu osjetljivosti.

- [x] P
- [ ] R
- [x] Q
- [ ] Qint
- [x] clk

Izrazi, koji čine minimalnu listu osjetljivosti, jesu oni izrazi o kojima ovisi stanje nekog drugog izraza, a da, pri tome, i oni sami ne ovise o nekom drugom izrazu, pogledaj izraz 'if clk rising edge then R=1', to znači da R ovisi o clk, tj. clk nešto uvjetuje pa je clk u listi, dalje – R isto uvjetuje nešto, tj. utječe na Q int, pa bi trebao biti u listi, ali nije, zbog toga što i sam ovisi o clk, dakle, R nije 'neovisan' i nije u listi, P i Q su izvan naredbe rising edge pa su neovisni i uvjetuju na Qint pa su u listi

10. Kod kojeg/kojih je automata izlaz u potpunosti definiran stanjem (tj. ako znate samo stanje, onda odmah znate i izlaz)?

- niti kod jednog od ova dva
- i kod Mooreovog, i kod Mealyjevog
- samo kod Mooreovog
- samo kod Mealyjevog

Teoretski, kod Mooreovog modela izlazi ovise o ulazima (ili su jednaki, uglavnom, postoji neka pravilnost), dok, kod Mealyjevog automata, izlazi ne ovise o ulazima ni u kakvom smislu

11. Vrijeme potrebno da se promijeni izlaz sinkronog sklopa od trenutka nailaska aktivnog brida signala takta naziva se:

- vrijeme otpuštanja (engl. release time)
- vrijeme postavljanja (engl. setup time)
- vrijeme kašnjenja (engl. delay time)
- vrijeme zadržavanja (engl. hold time)

(Pogledaj tekst prije pitanja)