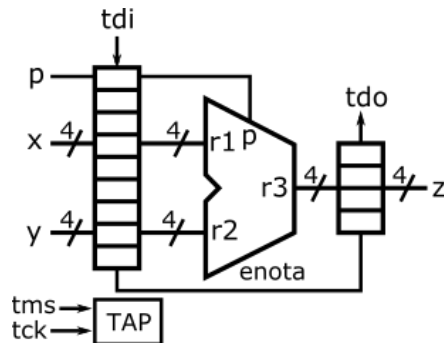


8. vaja: Preizkušanje vezja z robno testno linijo

Robna testna linija omogoča preizkušanje digitalnih integriranih vezij med proizvodnjo in tudi ko so vgrajena na tiskano vezje. Naredili bomo model preprostega kombinacijskega vezja, mu dodali testno linijo s krmilnikom in simulirali postopek preizkušanja s testno linijo.

8.1 Model integriranega vezja z robno linijo



Model vezja naj vsebuje 4-bitno kombinacijsko računsko enoto s celicami linije za robno testiranje na vseh priključkih enote in testni krmilnik. Računska enota ima dva 4-bitna vhoda (r1 in r2) in krmilni vhod p, s katerim izbiramo operacijo. Pri $p=0$ izvede $r3 = r1 \mid r2$, sicer pa $r3 = r1 \& r2$.

Devet vhodnih signalov računske enote je povezanih na vhodne celice, štiri izhodni signali pa na izhodne celice testne linije. V modelu integriranega vezja je še testni krmilnik z zunanjimi signali za upravljanje linije: tdi, tck, tms in tdo.

8.2 Simulacija vezja z robno linijo

Naredimo testno strukturo za simulacijo preizkušanja integriranega vezja z robno testno linijo. V prvi izvedbi testne strukture določi zaporedje signalov testnega krmilnika, tako da bodo v stanju shift nastavljene vse vhodne testne celice na željeno vrednost.

Primer: nastaviti želimo: $p=0$, $r1=0001$, $r2=0010$. Glede na zaporedje signalov v testni liniji, sestavimo 9-bitni testni vektor¹: $di = \{4'b0010, 4'b0001, 1'b0\}$;

V testni strukturi opiši generator ure (tck) in v bloku **initial** prehode krmilnega signala tms, tako bo testni krmilnik prišel v stanje shift. V tem stanju naj bo 9 ciklov ure in v vsakem ciklu naj se postavi tdi na vrednost ustreznega bita testnega vektorja:

```
for (i=8; i>=0; i=i-1) begin
  tdi = di[i];
  ...
```

¹ V splošnem bi potrebovali 13-bitni testni vektor za pomik čez vse celice. Ker pa nastavljamo le vhodne celice, ki so na začetku testne linije, je dovolj 9-bitni vektor s katerim nastavimo izhode vhodnih celic.

Ob prenosu zadnjega bita preklopi vrednost tms, tako da gre krmilnik še v stanja exit1 in update. Preveri na simulaciji ali se po stanju update pojavijo ustrezne vrednosti na vhodu in izhodu testirane računske enote.

8.3 Simulacija preizkušanja s testnimi vektorji

V testni strukturi deklariraj zbirko 9-bitnih testnih vektorjev in nastavi vrednosti testnih vektorjev (zaporednje signalov je z, r2, r1, p), npr. zbirka dveh testnih vektorjev:

```
reg [8:0] testv[1:0];  
testv[0] = {4'b0010, 4'b0011, 1'b0};  
testv[1] = {4'b0010, 4'b0011, 1'b1};
```

Dopolni testno strukturo tako, da bo naredila več prehodov stanj avtomata čez stanja capture, shift in update. Pri prvem prehodu naj se pomika vrednost prvega testnega vektorja čez vezje, pri drugem prehodu drugega itn. Izhodni podatek iz robne testne linije naj se prenese iz tdo na izhodni vektor (glej vajo 5). Po končanem pomikanju izpiši tiste bite² izhodnega vektorja, ki vsebujejo vrednost signala r3.

Naredi simulacijo in s pomočjo izpisov ugotovi, ali kombinacijsko vezje pravilno opravlja svojo funkcijo³.

Dopolni testno strukturo tako, da bo omogočala avtomatsko preizkušanje kombinacijske enote s primerjavo izhodov enote z izhodi referenčnega vezja, ki ga opiši v testni strukturi.

² Izhodni vektor (do) mora v splošnem imeti toliko bitov, kolikor je ciklov pomikanja – v našem primeru 9.

³ Testna linija v enem prehodu čez stanja avtomata najprej zajame izhod in nato nastavi vhode. Izhod vezja, ki je posledica prvega testnega vektorja, bomo zato lahko opazovali šele po tem, ko bomo pomikali drugi vektor...