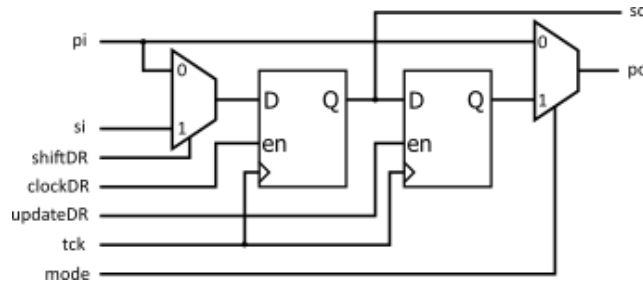


7. vaja: Generična testna linija



Univerzalna osnovna testna celica

7.1 Izdelava modela vhodne in izhodne testne celice

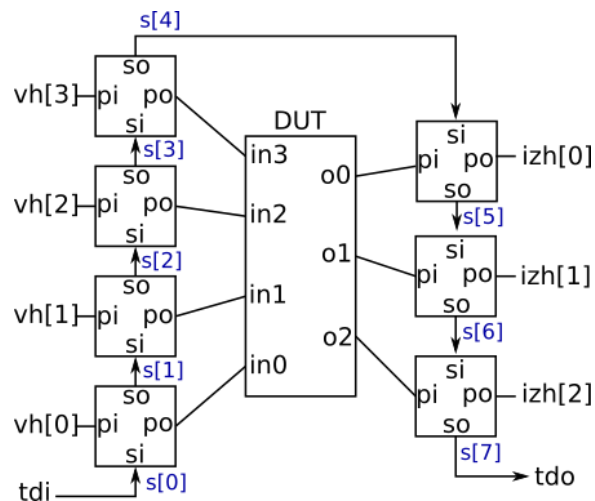
Optimiziraj univerzalno osnovno testno celico (5. vaja) tako, da bo delovala le kot vhodna ali pa le kot izhodna celica. Testna celica ima v vsakem primeru izbiralnik z vhodom **mode**, ki omogoča delovanje logike, ko nismo v načinu testiranja. Nekaj izmed ostalih gradnikov pa lahko izhpustimo.

- Vhodna celica med testiranjem nastavlja vrednost, ki pride na vhod testiranega vezja (po). Razmisli, kaj lahko v univerzalni celici izpustimo in naredi model vhodne celice.
- Izhodna celica je namenjena serijskemu branju izhodov testiranega vezja (izhodi so vezani na pi). Naredi tudi model izhodne celice.

7.2 Izdelava generične testne linije

Naredi generično testno linijo, ki je narejena iz N vhodnih celic, ki so povezane v serijo z M izhodnimi celicami. Blokovna shema prikazuje primer testne linije s štirimi vhodnimi in tremi izhodnimi celicami, ki je vezana na vezje DUT.

Na shemi so prikazani le paralelni in serijski podatkovni signali. Krmilni signali testnih celic na poenostavljeni blokovni shemi niso prikazani (shiftDR, clockDR...). Kako jih povežemo, smo videli v zadnjih dveh vajah.



Za opis generične (splošne) linije vhodnih in izhodnih celic uporabi stavek **generate**. Ker imamo dve vrsti celic, je najlažje uporabiti 2x **generate**. Deklarirajte notranji vektor (npr. s[]), s katerim povežite serijske vhode in izhode celic. Primer opisa povezav vhodnih celic s stavkom **generate**:

```

genvar i;
generate
  for(i=0; i<4; i=i+1)
    begin : vhodne_celice
      vh_celica u(vh[i], s[i], shiftDR, clockDR, updateDR, tck, mode, s[i+1], in[i]);
    end
endgenerate
    
```