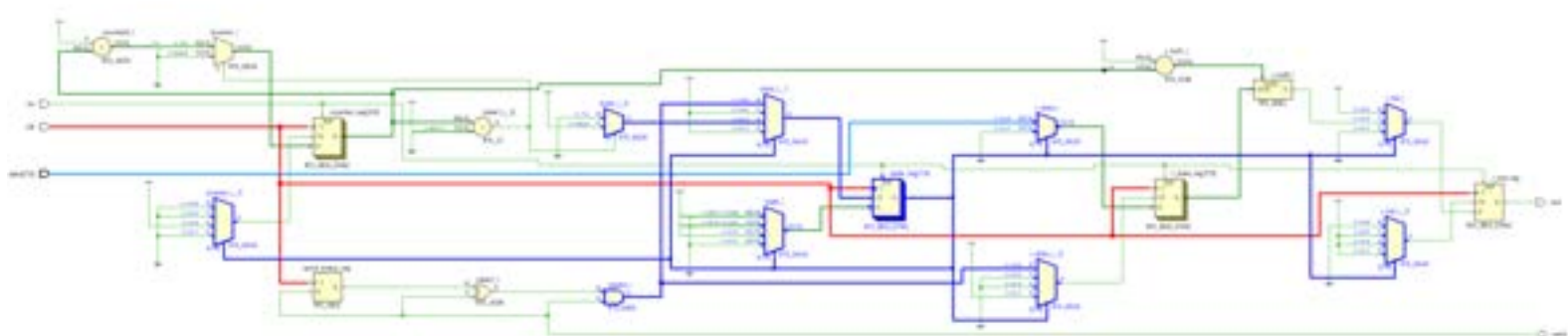
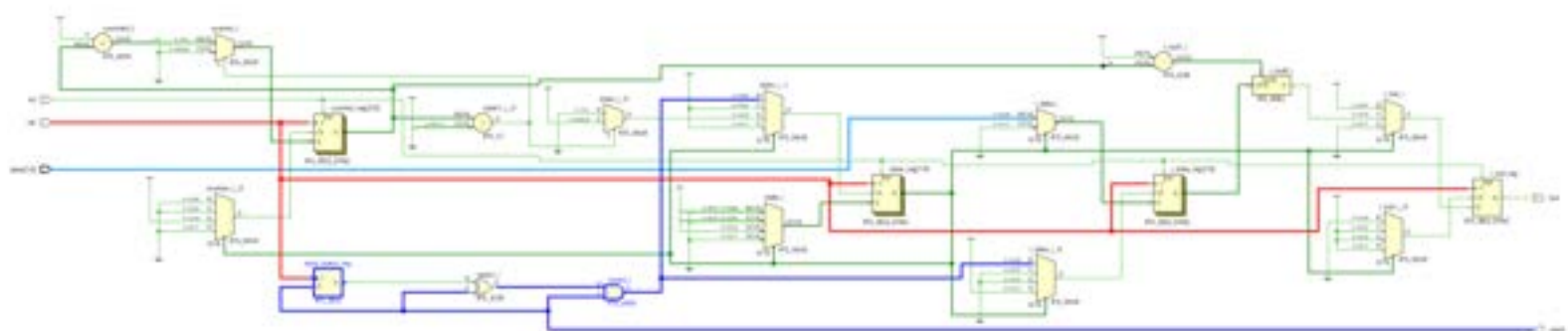


Na przedstawionym schemacie widzimy rejestr (D-flipflop), który jest „wyzwalany” jedynie w jednym stanie, mianowicie stanie 3 (spójrzmy na MUX *counter\_i\_0* – tylko jedno z jego wyjść odpowiadające stanowi 2'b10 jest podpięte pod „1” logiczną, a wyjście jest podawane na CE naszego counter-a).

Wyjście z rejestru counter jest podawane na wejście sumatora, który dodaje do wyniku 1, a następnie poprzez MUX (resetujący) wraca na wejście D rejestru. Dodatkowo wyjście z przerzutnika przechodzi przez blok porównujący typu RTL LT (Less Than), który porównuje wartość z 7 (0111) i jeśli wynik porównania jest „0”, wówczas licznik jest resetowany.



W centrum powyższego schematu (podświetlanego na niebiesko) znajduje się 7-bitowy rejestr stanu, do którego wejść podpięte są wyprowadzenia z dwóch MUX. Pierwszy *state\_i\_1* odpowiada za wejście CE (odpowiada on jednym słowem za ewentualne instrukcje warunkowe, które warunkują przejście do kolejnego stanu). Drugi *state\_i* odpowiada za zdefiniowanie kolejnego stanu, do którego ma nastąpić przejście.



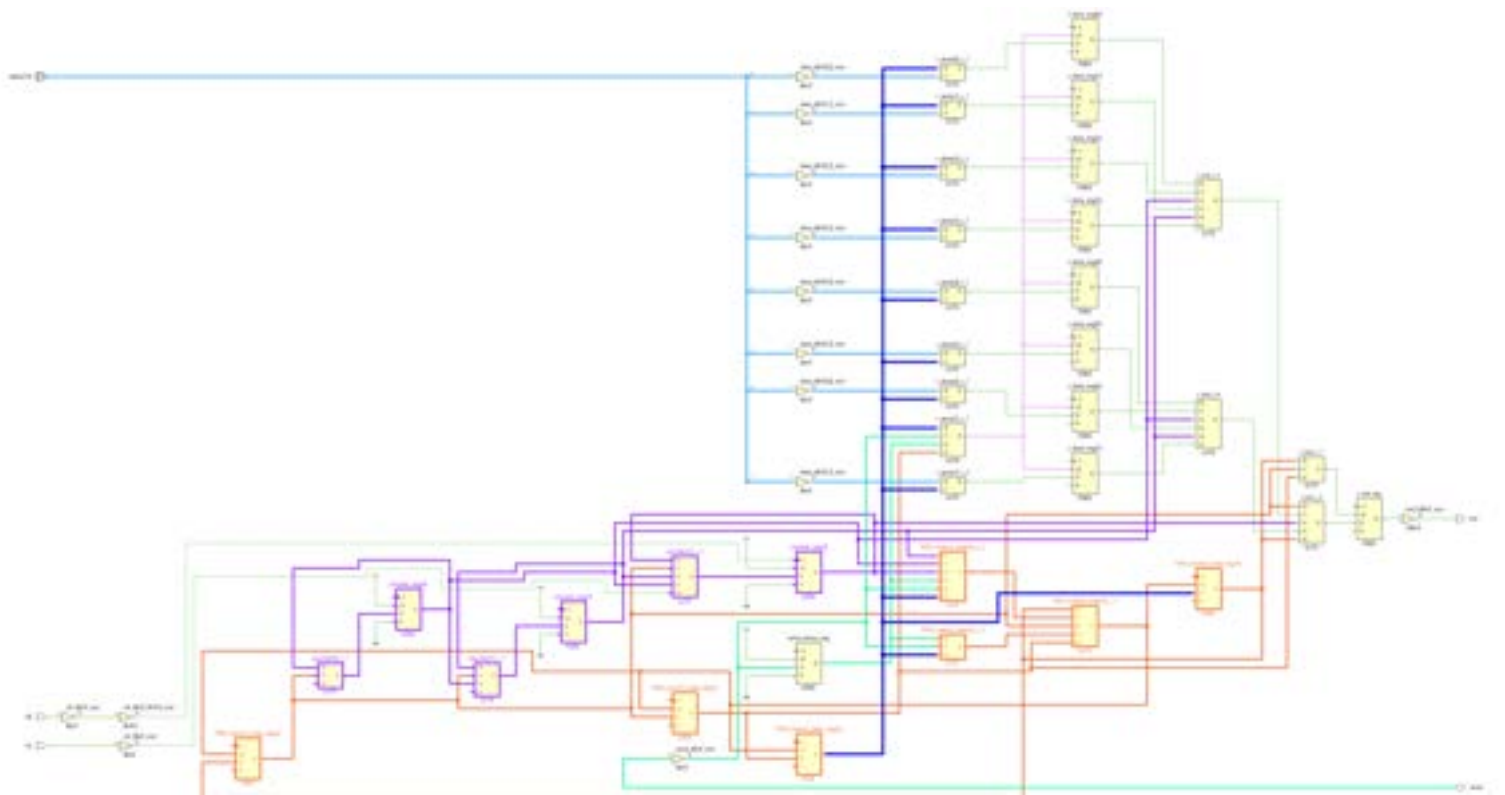
Powyższa część schematu odpowiada za realizację wyzwolenia przejścia do stanu drugiego i przepisania danych:

```

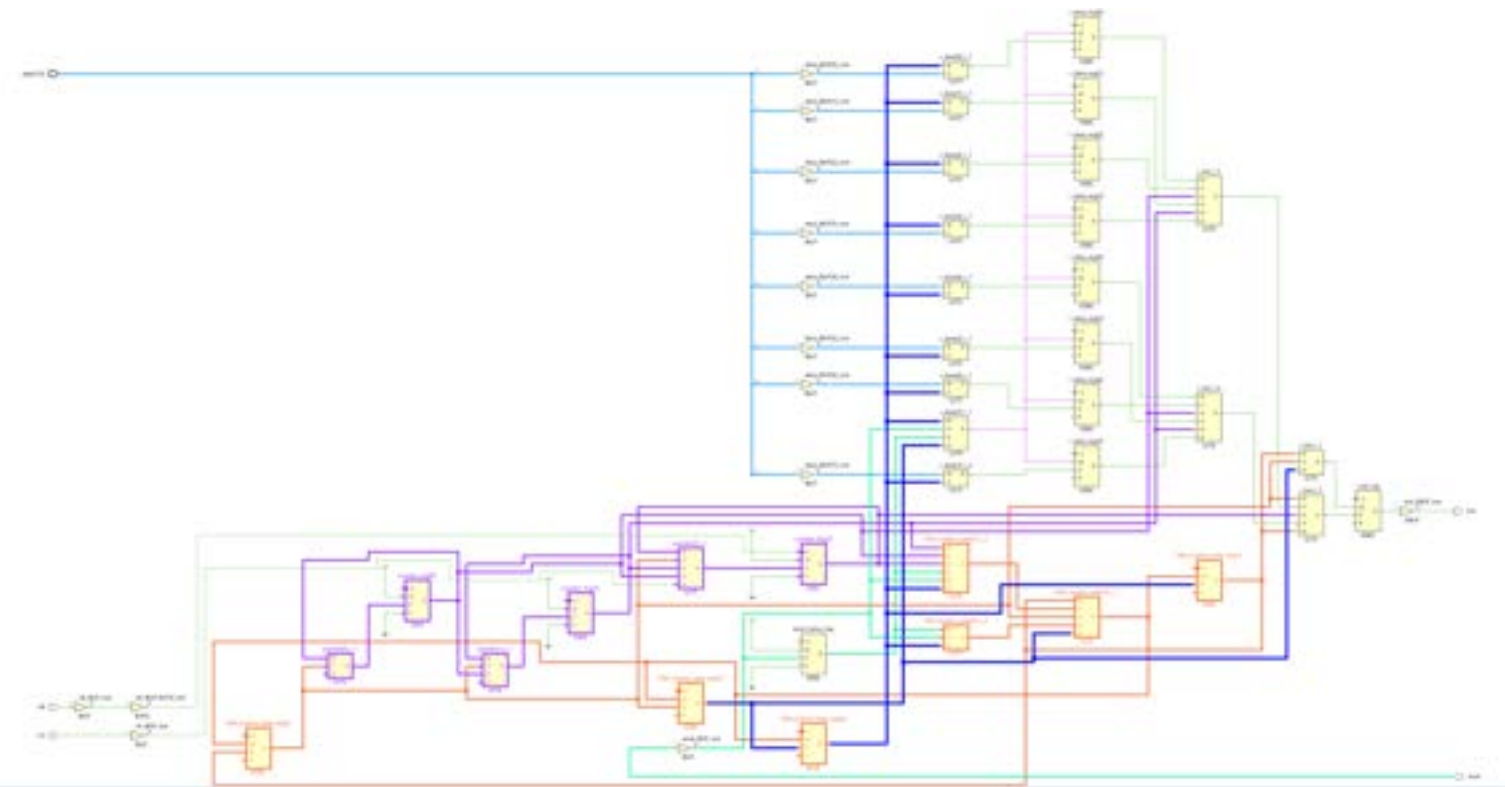
always@(*)
state_i_1
begin
  if (count_state == 3) then
    state_i_1 <= state_i_0
    state_i_1 <= state_i_0
  end
end

```





Wspomniane działanie związane z stanem 4 zwi zana jest tabela LUT  $r\_data[7]_i\_1$  – na wej scia ch przyjmuje ona wyj scia z rejestr w  $reg[0]$  i  $reg[3]$  (stan1 i stan4), a wej scie jest przekierowywane na CE rejestr w danych:



Przej scia pomi dzy stanami s  sterowane za pomoc  tabelle LUT:

