Obraz zawierający diagram, linia, Plan, Równolegle

Opis wygenerowany automatycznie

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.

Obraz zawierający diagram, linia, Plan, Równolegle

Opis wygenerowany automatycznie

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.

Obraz zawierający diagram, linia, Plan, Równolegle

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, zrzut ekranu, Czcionka, algebra

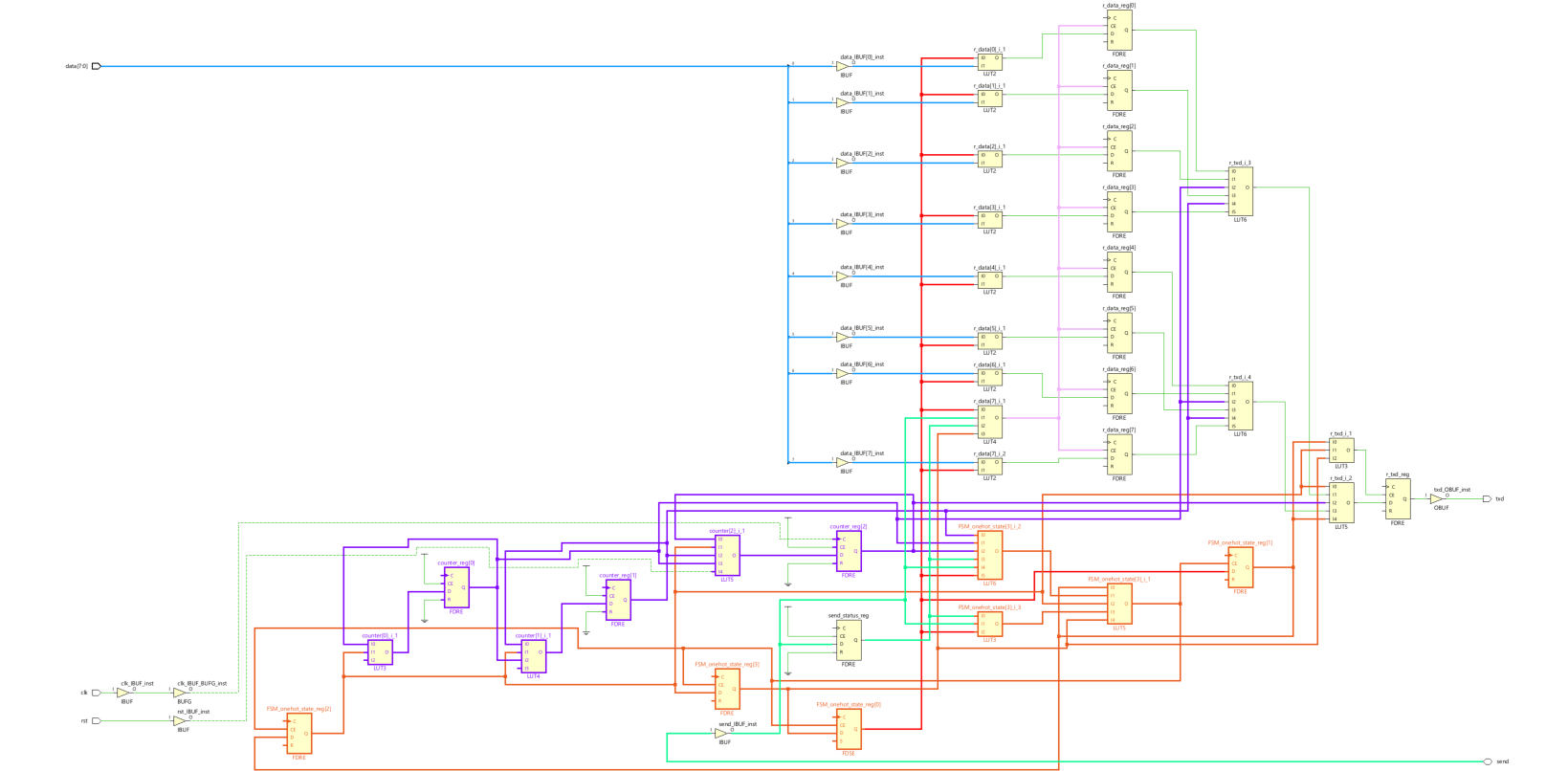
Opis wygenerowany automatycznie

Obraz zawierający diagram, linia, Plan, Rysunek techniczny

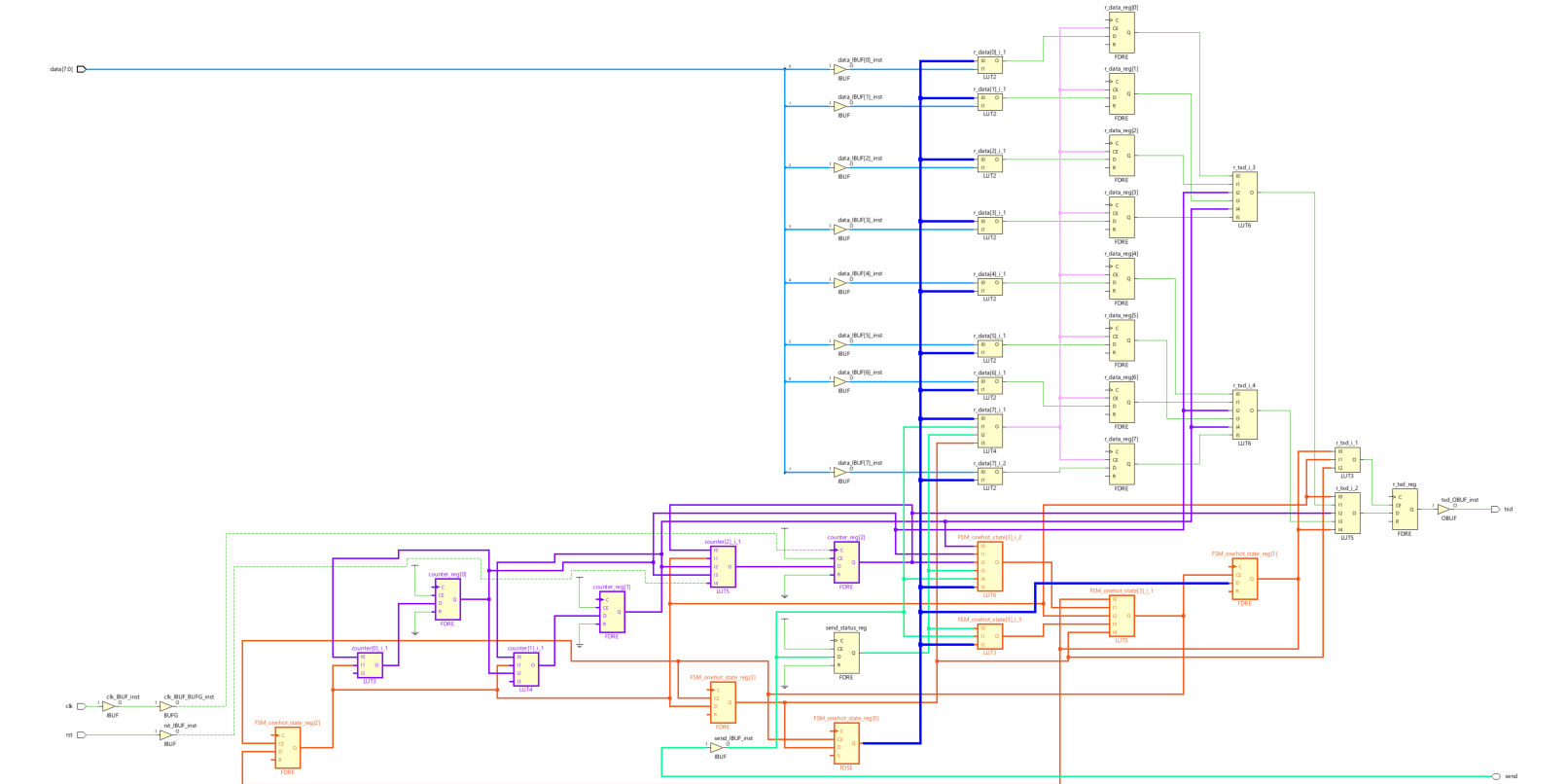
Opis wygenerowany automatycznie

Powyżej zaznaczona część schematu odpowiada za przepisywanie danych z wejścia. MUX *r\_data\_i\_0* odpowiada za obsługę CE rejestru *r\_data\_reg* w poszczególnych stanach. A selektor RTL\_BSEL podpięty do wyprowadzenia rejestru odpowiada za wybór odpowiedniej komórki na podstawie countera (aby odczyt następował w odwrotnej kolejności wprowadzono przesunięcie związane z blokiem RTL\_SUB – nie jestem pewien, ale to chyba suboptymalne, niemniej widzę to dopiero teraz).

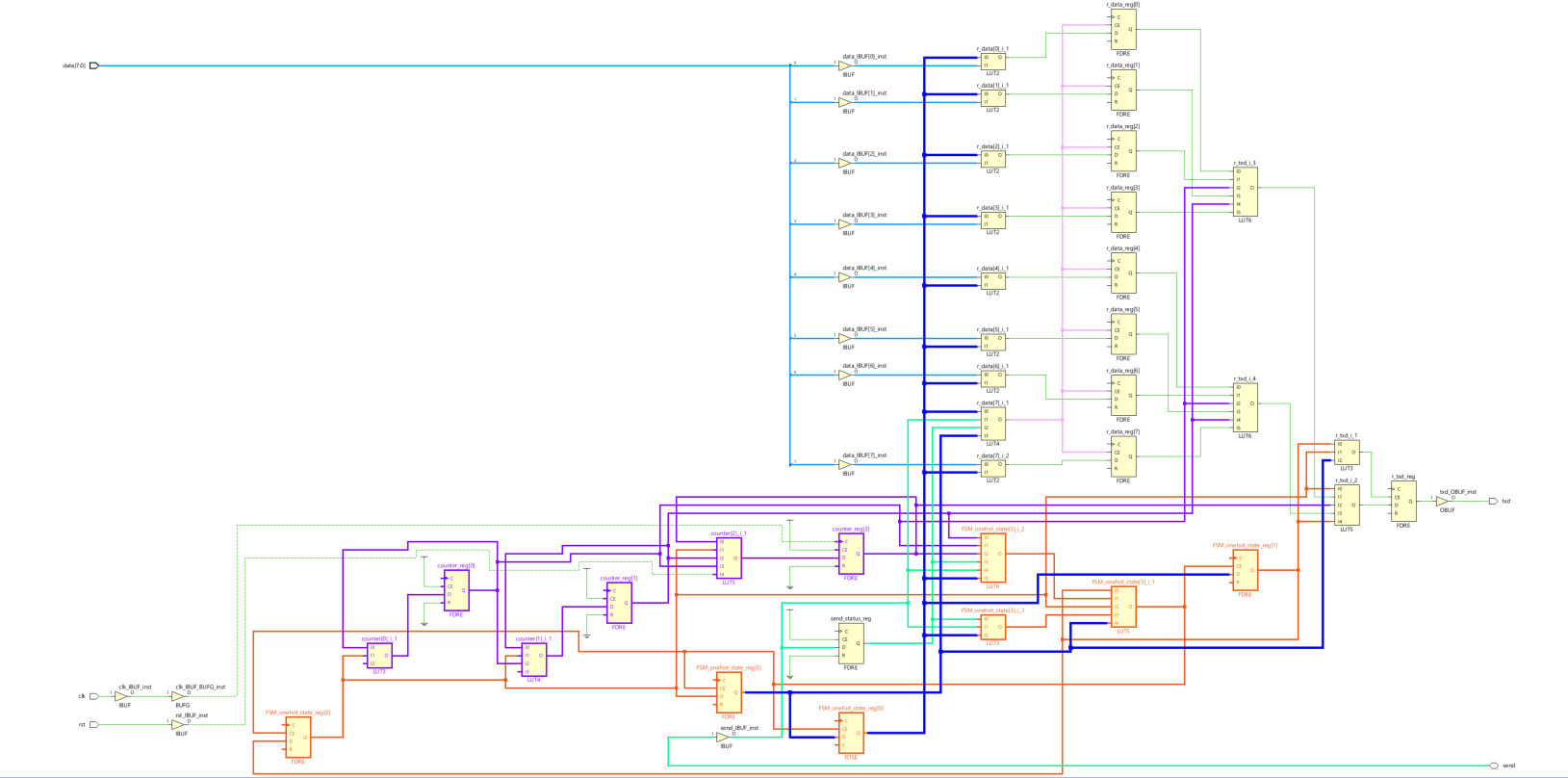
Analogicznie spróbujmy przeanalizować *Synthesis Design*:



Na niebiesko mamy zaznaczoną linię danych, na pomarańczowo linie związane ze stanem maszyny, a na fioletowo z licznikiem. Linie danych są podpięte poprzez bufory do tabeli LUT. Zakładam, że ich celem jest przepuszczanie sygnałów jedynie w stanie 1, a w stanie 4 (kiedy cały rejestr danych jest resetowany) sygnał z linii danych jest ignorowany, a na wejścia D przerzutników podawane są zera:



Wspomniane działanie związane z stanem 4 związana jest tabela LUT *r\_data[7]\_i\_1* – na wejściach przyjmuje ona wyjścia z rejestrów reg[0] i reg[3] (stan1 i stan4), a wejście jest przekierowywane na CE rejestrów danych:



Przejścia pomiędzy stanami są sterowane za pomocą tabeli LUT:

