

Programowanie układów FPGA

Marek Materzok

lista zadań nr 1
na zajęcia 15 października 2019

W poniższych zadaniach zaprojektowane układy należy zaprogramować w Verilogu (w formie bramkowej, z opóźnieniami równymi 1, chyba że treść zadania mówi inaczej), przetestować i rozumieć ich zachowanie. Należy używać wyłącznie konstrukcji z wykładu nr 1, chyba, że zadanie mówi inaczej.

1. Przerzutnik JK zachowuje się jak przerzutnik RS wyzwalany zboczem, z jednym dodatkowym zachowaniem – gdy oba wejścia są równe 1, wyjście ma zmienić stan na przeciwny. Można go opisać następującą tabelą:

<i>clk</i>	<i>j</i>	<i>k</i>	<i>q</i>
-	?	?	<i>q</i>
↑	0	0	<i>q</i>
↑	1	0	1
↑	0	1	0
↑	1	1	\bar{q}

Zaprojektuj realizację przerzutnika JK używającą przerzutnika D oraz multiplexera o 4 wejściach. Podpowiedź – użyj wejścia JK jako wejście adresowe dla multiplexera.

2. Przerzutnik typu T zmienia stan wyjścia na przeciwny po pojawieniu się zbocza zegarowego, jeśli wejście *enable* jest w stanie wysokim; w przeciwnym wypadku nie zmienia stanu. Można go skonstruować z przerzutnika JK przez podłączenie jednego sygnału do dwóch jego wejść *j* i *k*. Taki przerzutnik powinien posiadać możliwość inicjalizacji, w tym celu wykorzystuje się wejście *reset*, które typowo jest asynchroniczne – ustawia stan przerzutnika na 0 niezależnie od sygnału zegarowego. Przerzutnik typu T można opisać tabelą:

<i>clk</i>	<i>rst</i>	<i>en</i>	<i>q</i>
?	1	?	0
-	0	?	<i>q</i>
↑	0	0	<i>q</i>
↑	0	1	\bar{q}

Zaimplementuj przerzutnik typu T jako element podstawowy (**primitive**).

3. Enkoder priorytetowy to układ kombinacyjny, który jako wynik daje numer pierwszego wejścia z wartością niezerową, oraz dodatkowy bit *valid*, informujący czy którekolwiek z wejść miało wartość niezerową.

Zaprojektuj układ enkodera priorytetowego o dwóch wejściach. Przy jego pomocy zaprojektuj enkoder priorytetowy o czterech wejściach.

4. Zaimplementuj 4-bitowy licznik odliczający wartości od 0 do 9. Możesz wykorzystać przerzutniki typu D lub typu T (lub inny).

Wejście zegarowe w module testującym możesz zaimplementować w następujący sposób:

```
reg clk;  
always begin  
    #1 clk = 0;  
    #1 clk = 1;  
end
```

Okres zegara zwiększ odpowiednio do opóźnień w użytej logice, aby licznik działał prawidłowo.