

Programowanie układów FPGA

Marek Materzok

lista zadań nr 2
na zajęcia 22 października 2019

1. Używając jednego bloku `assign`, zamodeluj układ obliczający bit parzystości dla słowa wejściowego o zadanej długości (użyj parametru o domyślnej wartości 8). Podpowiedź: wykorzystaj operator redukcji.
Napisz (również sparametryzowany) testbench sprawdzający, czy dla każdego możliwego wejścia bit parzystości jest obliczany poprawnie; należy wejścia testowe konstruować tak, aby wiadomo było z góry, jaka jest jego parzystość. Podpowiedź: kody Graya.
2. Zamodeluj enkoder priorytetowy o 4 wejściach na dwa sposoby: używając bloków `assign` lub bloków `always`. Napisz testbench wyczerpująco porównujący obie implementacje.
3. Zamodeluj (używając bloków `assign` lub `always`) enkoder priorytetowy o 2^n wejściach. Podpowiedź: użyj generowania.
Napisz sparametryzowany testbench testujący go wyczerpująco (sprawdzając wszystkie 2^n wejść). Podpowiedź: jakiej postaci są liczby, które mają pierwszą jedynkę na pozycji k ? Dobierz kolejność bitów, aby zadanie było łatwe.
4. Zamodeluj proste ALU dla słów 8-bitowych. Niech ma dwa wejścia 8-bitowe, jedno wyjście 8-bitowe oraz wejście sterujące wybierające, którą operację chcemy wykonać: dodawanie, odejmowanie, bitowe AND, bitowe OR, lub negację jednej z poprzednich czterech (wybór negacji powinien być osobnym bitem wejścia sterującego).
Napisz testbench, który wczyta za pomocą `$readmemh` plik, w którym każdy przypadek testowy ma cztery bajty: wejście 1, wejście 2, wejście sterujące, wyjście; a następnie przetestuje model, wykorzystując wczytane dane. Przyjmijmy, że koniec pliku oznaczają cztery bajty `8'hff`.