

Programowanie układów FPGA

Marek Materzok

lista zadań nr 3
na zajęcia 29 października 2019

1. Zaimplementuj stoper liczący sekundy i minuty zapisane w postaci 4-bitowych cyfr BCD. Wykorzystaj synteżowalne konstrukcje wysokopoziomowe – bloki **always**, przypisania nieblokujące, arytmetykę.
Napisz testbench prezentujący jego działanie.
2. Zaimplementuj układ synchroniczny z wejściami jednobitowymi **clk**, **rst**, **in** oraz wyjściem jednobitowym **out**. Wyjście powinno mieć wartość 1 wtedy i tylko wtedy, gdy na wejściu, od momentu zresetowania układu, odczytano (na rosnących zboczach zegara) parzystą liczbę jedynek i parzystą liczbę zer. Przykładowo, jedynka powinna się pojawić dla ciągów 1100 i 011000, zaś dla ciągów 101 i 0010 powinno pojawić się zero.
3. Zaimplementuj licznik dwukierunkowy 16-bitowy – układ sekwencyjny o czterech wejściach jednobitowych **clk**, **rst**, **dir** i **zero**, oraz wyjściu 16-bitowym pokazującym stan licznika. Wejścia **dir** i **zero** mają być synchroniczne, **zero** ma wyzerować licznik, natomiast **dir** ma określać kierunek, w którym ma odliczać licznik po wystąpieniu zbocza zegarowego (0 oznacza w górę). Licznik może się zapętlać.
Napisz testbench prezentujący jego działanie.
4. Używając licznika dwukierunkowego zaimplementuj generator PWM. Niech moduł ma pięć wejść: dwa 16-bitowe **top** i **comp**, oraz jednobitowe **clk**, **rst**, **pol**, **mode**. Wejście **mode** określa tryb pracy: dla 0 licznik ma odliczać do wartości **top**, po czym ma być zerowany; dla 1 po osiągnięciu wartości **top** licznik ma zmienić kierunek odliczania na przeciwny, aż do osiągnięcia wartości 0, kiedy ma przywrócić normalny kierunek. Pojedyncze wyjście **pwm** działa w zależności od wejścia **pol**. Jeśli **pol** jest równe 0, **pwm** ma być równe 0 dla wartości licznika mniejszych od **comp**, a równe 1 dla wartości większych lub równych. Wartość **pol** równa 1 oznacza zanegowanie wyjścia **pwm**.
Zaprezentuj działanie generatora za pomocą testbenchu.