

Logika cyfrowa

Programistyczna lista zadań nr 8

Termin: 26 kwietnia 2020

Uwaga! Poniższe zadania należy rozwiązać przy użyciu języka SystemVerilog, sprawdzić w DigitalJS oraz wysłać w systemie Web-CAT na SKOS. Należy pamiętać, aby nazwy portów nadesłanego modułu zgadzały się z podanymi w treści zadania. Wysłany plik powinien mieć nazwę `toplevel.sv`. **Nie przestrzeganie tych zasad będzie skutkować przyznaniem 0 punktów.**

1. Zaimplementuj generator sygnału PWM (Pulse Width Modulation – modulacja szerokości impulsu). Generator taki składa się z licznika, rejestru porównania, rejestru wartości szczytowej oraz układów porównujących. Rejestry oraz licznik są podłączone do wspólnego sygnału zegara. Działanie jest następujące:

- licznik rozpoczyna odliczanie od wartości 0;
- gdy wartość licznika jest większa lub równa wartości szczytowej, licznik jest resetowany do 0 (synchronicznie);
- sygnał wynikowy jest w stanie niskim, gdy wartość licznika jest większa lub równa wartości rejestru porównania, w przeciwnym wypadku jest w stanie wysokim;
- do każdego z elementów składowych (włącznie z licznikiem) można załadować nową wartość, nie zatrzymując przy tym działania licznika (chyba, że właśnie licznik jest ładowany).

Układ powinien mieć następujące wejścia i wyjścia:

- `clk` – wejście sygnału zegara,
- `d` – szesnastobitowe wejście ładowania,
- `sel` – dwubitowe wejście wybierające ładowany rejestr, znaczenie:
 - 0 – nic nie jest ładowane,
 - 1 – ładowany jest rejestr porównania,
 - 2 – ładowany jest rejestr wartości szczytowej,
 - 3 – ładowany jest licznik,
- `cnt` – szesnastobitowe wyjście bieżącej wartości licznika,
- `cmp` – szesnastobitowe wyjście bieżącej wartości rejestru porównania,
- `top` – szesnastobitowe wyjście bieżącej wartości rejestru wartości szczytowej,
- `out` – jednobitowy sygnał wyjściowy generatora.

W tym zadaniu nie używaj modeli bramkowych przerzutników ani liczników. Do modelowania elementów synchronicznych wykorzystaj bloki `always_ff`.