

Logika cyfrowa

Praktyczna lista zadań nr 11

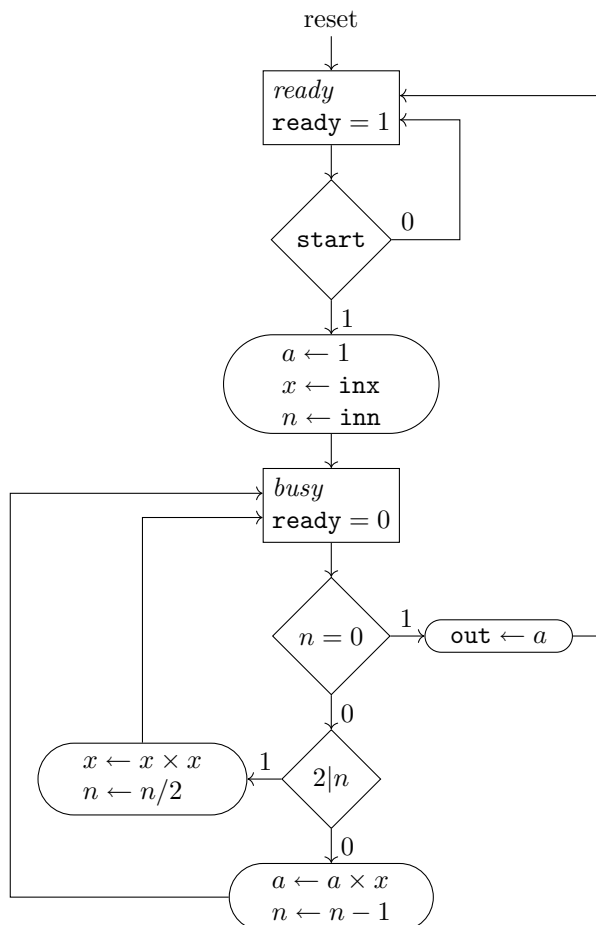
Termin: 22 maja 2024 godzina 30:00

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 skutkowało przyznaniem 0 punktów.**

1. Zaimplementuj obwód potęgujący (modulo rozmiar wyjścia) liczbę naturalną metodą potęgowania szybkiego. Obwód powinien mieć następujące wejścia i wyjścia:

- `clk` – wejście zegara,
- `nrst` – zanegowane wejście resetu asynchronicznego,
- `start` – 1-bitowe wejście uruchamiające obliczenia,
- `inx` – 16-bitowe wejście liczby x ,
- `inn` – 8-bitowe wejście liczby n
- `ready` – 1-bitowe wyjście sygnalizujące gotowość układu do rozpoczęcia pracy,
- `out` – 16-bitowe wyjście x^n , ważne tylko wtedy, gdy `ready = 1`.

Wykorzystaj poniższy diagram algorytmiczny. Zadbaj, aby w układzie występował tylko **jeden** multiplikator – multiplikatory są przecież dużymi, złożonymi układami i byłoby marnotrawstwem budować ich więcej niż potrzeba. Układ nie powinien zawierać (jeszcze bardziej złożonych przecież) układów dzielących i obliczających modulo.



2. Zaimplementuj obwód znajdujący największy wspólny dzielnik dwóch liczb binarnych 8-bitowych bez znaku. Rozdziel ścieżkę sterowania od ścieżki danych – powinny one być zaimplementowane w osobnych modułach. Obwód powinien mieć następujące wejścia i wyjścia:

- **clk** – wejście zegara,
- **nrst** – zanegowane wejście resetu asynchronicznego,
- **start** – 1-bitowe wejście uruchamiające obliczenia,
- **ina** – 8-bitowe wejście liczby a ,
- **inb** – 8-bitowe wejście liczby b
- **ready** – 1-bitowe wyjście sygnalizujące gotowość układu do rozpoczęcia pracy,
- **out** – 8-bitowe wyjście $\text{nwd}(a, b)$, ważne tylko wtedy, gdy **ready** = 1.

Wykorzystaj poniższy diagram algorytmiczny.

Podpowiedź: w ścieżce danych pojawiają się multiplexery.

