

Logika cyfrowa

Praktyczna lista zadań nr 2

Termin: 13 marca 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 moduł o wejściu czterobitowym i oraz wyjściu jednobitowym `o`. Moduł powinien produkować na wyjściu 1 wtedy i tylko wtedy, gdy co najmniej dwa i nie więcej niż trzy bity wejścia mają wartość 1, w przeciwnym wypadku powinien produkować 0.

Dodatkowo implementacja powinna nie zawierać glitchy. W szczególności: przy zmianie jednego bitu wejścia, jeśli nie powoduje to zmiany liczby zapalonych bitów na wejściu z dwóch na jeden lub odwrotnie, albo z czterech na trzy lub odwrotnie, stan wyjścia powinien pozostać stabilny.

Uwaga: Należy pamiętać, że DigitalJS nie symuluje wszystkich glitchy. W szczególności, jeśli układ będzie tak zbudowany, że wszystkie ścieżki od wejść do wyjść będą miały taką samą liczbę bramek, glitch nie ujawni się w symulacji – jednak wciąż może wystąpić w rzeczywistym układzie.

Aby przetestować rozwiązanie, można zastosować następujący skrypt Lua. Sprawdza on wyłącznie poprawność, nie wykryje glitchy występujących w układzie. Ze względu na uwagę z akapitu powyżej, nie istnieje skrypt testowy, który wykryje wszystkie glitche.

```
for x=0,15 do
  local v = vec(x, 4)
  sim.setinput("i", v)
  sim.sleep(10)
  local n = 0
  for p=0,3 do
    n = n + v(p):tointeger()
  end
  assert(sim.getoutput("o"):ishigh() == (n == 2 or n == 3), "Error: v=" .. v:tobin())
end
print("OK!")
```