

Logika cyfrowa

Praktyczna lista zadań nr 3

Termin: 20 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 skutkować przyznaniem 0 punktów.**

1. Zaimplementuj układ dodająco-odejmujący dla liczb dwucyfrowych w formacie BCD. Operacje powinny być wykonywane modulo 100; przykładowo, wynikiem dodawania liczb 60 i 50 powinno być 10. Nie używaj operatorów arytmetycznych SystemVeriloga.

Główny moduł powinien posiadać wejścia ośmiobitowe `a` oraz `b`, jedno wejście jednobitowe `sub` oraz jedno wyjście ośmiobitowe `o`. Wejścia oraz wyjścia oznaczają liczby BCD, gdzie każdy półbajt oznacza jedną cyfrę dziesiętną. Gdy wejście `sub` ma stan wysoki, układ powinien wykonać operację odejmowania; gdy ma stan niski, dodawania.

Rozwiązanie można przetestować za pomocą poniższego skryptu Lua.

```
for x=0,99 do
    local a = math.random(0, 99)
    local b = math.random(0, 99)
    sim.setinput("a", '16h' .. tostring(a))
    sim.setinput("b", '16h' .. tostring(b))
    sim.setinput("sub", false)
    sim.sleep(100)
    assert(tonumber(sim.getoutput("o"):tohex()) == (a + b) % 100,
        "Error: a=" .. a .. " b=" .. b)
    sim.setinput("sub", true)
    sim.sleep(100)
    assert(tonumber(sim.getoutput("o"):tohex()) == (a - b) % 100,
        "Error: a=" .. a .. " b=" .. b)
end
print("OK!")
```

2. Zaimplementuj 16-bitowy sumator hierarchiczny z przewidywaniem przeniesienia wykorzystujący grupy czterobitowe. Ścieżka krytyczna w zaimplementowanym układzie nie powinna być dłuższa niż 8 bramek. Nie używaj operatorów arytmetycznych SystemVeriloga.

Główny moduł powinien posiadać wejścia szesnastobitowe `a` oraz `b` oraz jedno wyjście szesnastobitowe `o`.