



LABORATORIUM 1: PRZYPISANIA CIĄGŁE W JĘZYKU VERILOG

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z podstawowymi możliwościami modelowania układów cyfrowych w języku Verilog na przykładzie bramek logicznych.

Zadanie 1. Przygotować opis bramki *AND* w języku Verilog, a następnie przeprowadzić jej symulację.

- ♦ Do realizacji zadania wykorzystać symulator *Active-HDL*.
Model bramki powinien mieć postać następującą:

```
module Bramka_And(Y, A, B);  
    input A;  
    input B;  
    output Y;  
  
    assign Y=A&B;  
endmodule
```

- ♦ Skompilować model a następnie ustawić moduł *Bramka_And* jako *Top-Level*.
- ♦ Utworzyć nowe okno przebiegów czasowych symulacji (*Waveform*), ustawić wymuszenia dla wejść *A* oraz *B*, przeprowadzić symulację, a następnie zapisać wyniki symulacji do pliku..
- ♦ Wygenerować moduł testujący (*Test-Bench*) wykorzystując wbudowane narzędzie programu *Active-HDL*, bazując na pliku z wcześniej zapisanymi wynikami symulacji.
- ♦ Uruchomić symulację bramki *AND* wykorzystując wygenerowany moduł testujący.
Jako *Top-Level* należy ustawić nowo wygenerowany moduł.

Zadanie 2. Przygotować opis bramki *NOT* w języku Verilog, napisać (nie generować!) moduł *Test_NOT*, testujący (weryfikujący poprawność napisanej bramki) i przy jego pomocy przeprowadzić symulację bramki.

- ♦ Aby napisać moduł testujący należy dokładnie przeanalizować moduł z zadania pierwszego.

Zadanie 3. Przygotować opis prostego multipleksera 1 bitowego *Mux_2* (jedno wejście adresowe, dwa wejścia informacyjne, jedno wyjście). Do realizacji zadania wykorzystać operator *?:*

- ♦ Przykład wykorzystania operatora „?:”

```
//gdy Sel=0 wybierz In1, w przeciwnym razie wybierz In2  
assign Out = (Sel==0)?In1:In2;
```

Zadanie 4. Przygotować opis multipleksera 2 bitowego *Mux_4* (dwa wejścia adresowe, cztery wejścia informacyjne, jedno wyjście).