

Systemy wbudowane'18

Lista na laboratorium nr 4 (na zaliczenie)

tydzień od 12 marca 2018

Na tych laboratoriach pojawi się nowy element języka VHDL: *funkcja*. Będzie też pamięć (ROM) oraz generator sumy kontrolnej (CRC-8) a także problem synchronizacji dwóch komponentów.

Funkcja w języku VHDL jest podprogramem posiadającym jako wejście sygnały oraz zwracającym wyłącznie jedną wartość określonego typu. Jest traktowana jako *wyrażenie* i jako takie może być stosowana wyłącznie w ramach procesu. W VHDL istnieje też pojęcie **procedury** – może ona mieć wiele wejść i wyjść i jest *instrukcją* – tak jak np. przypisanie (\leq) i jest wykonywana równoległe bądź sekwencyjnie, w zależności od miejsca jej wywołania.

Podprogram (a więc funkcja) może korzystać wyłącznie z instrukcji sekwencyjnych. Nie może definiować sygnałów wewnętrznych - może używać jedynie zmiennych. I w odróżnieniu od zmiennych w procesie - te w funkcji nie zachowują wartości pomiędzy kolejnymi wywołaniami! Czyli taka normalna funkcja.

Przykład funkcji, która oblicza XOR ze wszystkich bitów w danym wektorze:

```
function mxor(signal A : std_logic_vector) return std_logic is
    variable tmp : std_logic;
begin
    p := '0';
    for i in A'range loop
        p := p xor A(i);
    end loop;
    p := p xor '0';
    return p;
end mxor;
```

Instrukcja **wait** w VHDL służy do kontrolowania działania kodu w czasie i umożliwia określenie operacji “czekaj [dopóki|na zdarzenie|przez]”. Wielokrotnie użyliśmy już **wait for {czas}**. Warto zaznajomić się z pozostałymi możliwościami: **wait on {sygnał}** oraz **wait until {warunek}**. Instrukcji **wait** nie można używać w procesach posiadających listę czułości oraz w procedurach wywoływanych w takich procesach oraz w funkcjach. Wykonywanie **wait** wewnątrz procedury, która jest czuła na jakiś sygnał mogłoby prowadzić do tego, że procedura czeka, a jednocześnie jest uruchamiana od nowa. A ponieważ VHDL opisuje *hardware* – to układ jednocześnie czekający i wykonujący obliczenia - nie jest możliwy. Dodatkowo kod posiadający polecenia **wait** nie jest *syntetyzowalny*, tj. nie można z niego wygenerować “wsadu” do prawdziwego procesora FPGA.

Zadanie 1 Przeanalizuj kod dla `rom.vhd`, `pack.vhd`, `crc8.vhd`. Odpowiedz na pytania:

- Jak wygląda schemat blokowy generatora CRC-8 (narysuj go)?
- Jak jest realizowana pamięć ROM?
- Co jest przechowywane w tej pamięci?

Zadanie 2 Przeanalizuj `crc8_tb.vhd`.

- Co realizują linie 68-72?
- Uruchom ten testbench. Zwróć uwagę na generowane ostrzeżenia (*assertion warning*). Zidentyfikuj i usuń ich przyczynę.

Zadanie 3 Zbuduj test poprawności działania modułu CRC. Pomocne może być narzędzie online: <http://smbus.org/faq/crc8Applet.htm>¹, a do testów modułu CRC-8 z pliku `crc8.vhd` napisz kompletny, automatyczny test korzystający z danych zapisanych w pamięci ROM.

¹Tak, wiem, Java jest wyłączona w przeglądarce...