

Systemy wbudowane'17

Lista na laboratorium nr 3 (zad. 1, 2 – na zaliczenie, zad. 3 – 10 pkt.) 15, 16 marca 2017

Ostatnio okazało się, że VHDL pozwala na tworzenie układów komponentów przez łączenie ich sygnałami za pomocą instrukcji `port map` oraz `generic map`. Były to na razie proste układy logiczne. W odróżnieniu od nich układy *sekwencyjne* są to układy pamiętające – ich wyjście zależy od aktualnego stanu. Najprostszym przykładem są liczniki – następne wyjście zależy od poprzednich wejść (a więc stanu licznika).

Pojęcie *procesu* już się pojawiło przelotnie, teraz pora na bliższą znajomość. Proces jest zestawem instrukcji wykonywanych sekwencyjnie (tj. w kolejności ich zapisania w procesie). Jednakże sam proces jest wykonywany współbieżnie z innymi instrukcjami współbieżnymi (np. przypisanie podstawowe (`a <= "010"`), warunkowe:

```
y <= a when s = '0' else  
    b when s = '1' else 'X';
```

(Nb. – jakie działanie opisuje ten kod?). Inną instrukcją współbieżną jest przypisanie selektywne:

```
with s select  
y <=    a when "00",  
        b when "01",  
        c when "10",  
        "XXX" when others;
```

Ponawiam pytanie: co opisuje ten kod?

Tak czy inaczej, są to instrukcje, które występują bezpośrednio w ciele opisu architektury i są wykonywane równolegle. Wszystkie. Razem z procesami.

A wracając do procesów, te są wykonywane linia po linii. To trochę zagmatwane, ale proces jest po prostu bardziej złożoną pojedynczą instrukcją – umożliwia podejmowanie decyzji, wykonywanie operacji itd.

Z procesem powiązana jest jego *lista czułości* – zestaw sygnałów podanych w nawiasie zaraz za jego nazwą. Jeśli którykolwiek z tych sygnałów zmieni swój stan - powoduje to uruchomienie procesu. To nie jest lista argumentów funkcji, chociaż na to wygląda. Jeśli proces nie ma listy czułości, to by znaczyło, że będzie działał zawsze i w kółko. Tak być nie może, więc VHDL nakazuje, by wewnątrz takiego procesu została użyta instrukcja `wait [for|until]`.

Zadanie 1 Przeanalizuj kod dla `simple.vhd`, odpowiedz na zadane w komentarzu pytanie. Uruchom `simple_tb.vhd` modyfikując czasy zadania sygnału `rst`, aby licznik doliczył do 195 a potem do 182.

Zadanie 2 Zapoznaj się z realizacją układu zawartego w `twoway.vhd`. Postaw tezę na temat działania tego licznika, napisz test (możesz wykorzystać ten z zad. 1.). Czy wszystko działa jak powinno? Jeśli nie, to popraw.

Zadanie 3 W pliku `lfsr.vhd` znajdziesz prostą implementację rejestru LFSR (*Linear Feedback Shift Register*).

- Napisz program testowy, pozwalający na wydobycie z LFSR sekwencji bajtów pseudo-losowych.
- Użyj programu z <http://lfsr-generator.sourceforge.net/>, aby wygenerować rejestr LFSR i porównać go do swojego modelu w VHDL (możesz zmodyfikować kod w `lfsr.vhd`

Porównaj wyniki z obu programów. Narysuj wykres generowanych bajtów w czasie dla różnych wartości początkowych.