

# Systemy wbudowane'18

## Lista na laboratorium nr 5 (na zaliczenie<sup>1</sup>)

### tydzień od 19 marca 2018

Wykonywanie operacji często składa się z kilku etapów. Dane do wysłania muszą być zebrane, załadowane do rejestru przesuwnego, wysłane bit po bicie. Mnożenie (w podstawowej formie, takiej jak mnożenie pisemne) to najpierw wytworzenie poszczególnych iloczynów częściowych, a potem ich zsumowanie. Cykl prania w pralce to nabranie wody, grzanie, obroty bębna, spuszczenie wody, i.t.d.

Naturalnym sposobem reprezentowania takiego mechanizmu w informatyce jest *automat stanów*. Zapewne kojarzysz pojęcie automatu Mealy'ego lub Moore'a. Jeśli nie, to przypomnij je sobie.

Jak taki automat ma się do mechanizmu działania układu? Jeśli w automacie wyróżnimy wszystkie możliwe stany danego systemu, a każdemu z nich przypiszemy pewne wyjściowe sygnały sterujące konkretnymi komponentami, które w danym stanie mają działać – uzyskamy sterowanie *logiką* systemu. Czym jest sygnał sterujący? Przypomnij sobie moduł `simple` licznika z 3. laboratorium. Ten moduł miał sygnał `rst`, który uruchamiał ten licznik – gdy `rst='0'` licznik zliczał. To jest właśnie sygnał sterujący. Innymi sygnałami mogą być linie adresowe multiplekserów, sygnały zapisu lub przesunięcia rejestrów itp.

Niedługo będziemy analizować większy układ, gdzie sygnałów sterujących jest sporo. A tymczasem zajmiemy się podstawową konstrukcją – reprezentacją maszyny stanów w VHDL. Dodatkowo zapoznasz się z podręczną biblioteką ułatwiającą konwersję różnych typów do obiektów typu `string`. Dodatkowo zajmiemy się pojęciem synchronicznego i asynchronicznego działania układu.

W skrócie (i w przypomnieniu): synchroniczne sygnały to te, których działanie warunkowane jest zaistnieniem dodatkowego sygnału (np. zegarowego). Dotychczas często pisaliśmy:

```
process (clk)
begin
    if clk = '1' and rising_edge(clk)
        -- operacje synchroniczne
    end if;
end process;
```

W tym wypadku działania wewnątrz `if` są synchronizowane zdarzeniem “rosnące zbocze `clk`”. Sygnał asynchroniczny nie będzie “czekał” na inne wydarzenie, żeby wywołać jakieś działanie.

**Zadanie 1** Przeanalizuj kod w `statemachine.vhd`.

- Narysuj model tej maszyny w postaci automatu Moore'a oraz Mealy.

**Zadanie 2** Przeanalizuj `statemachine_tb.vhd`.

- Uruchom symulację, zwróć uwagę na wyniki polecenia `assert`. Czy asercje w liniach 46. i 53. ? Jeśli nie, to popraw je.
- Zapoznaj się z biblioteką `txt_util` z pliku `vhdl_txt.vhd`. Przerób `statemachine_tb` tak, żeby kolejne sygnały `driver` były wypisywane na konsoli w postaci liczby ósemkowej.

**Zadanie 3** Dopisz do układu obsługę *asynchronicznego* sygnału `reset` – ustawiającego automat w stan `S0`. Wykaż poprawność działania poprzez symulację.

**Zadanie 4** Zmodyfikuj maszynę stanów zgodnie z poleceniem prowadzącego. Wykaż poprawność działania przez symulację.

---

<sup>1</sup>UWAGA. W przypadku nieoddania listy na bieżących zajęciach zostanie ona rozszerzona o dodatkowe zadanie, podane przez prowadzącego.