

## Aplikacje: Zamek cyfrowy

### 1. Cel ćwiczenia

Projektowanie prostych systemów cyfrowych. Praktyczne wykorzystanie wiedzy z zakresu układów kombinacyjnych i sekwencyjnych.

### 2. Zadanie

(jedna dokumentacja na grupę – projekt zespołowy)

A) Wykonaj projekt dekodera znaków ASCII.

(stopień trudności: 1)

Wymagania projektowe:

- sterowanie z programu terminala (np. Hyper Terminal, PuTTY),
- prezentacja wyników na wyświetlaczu 7-segmentowym 4-pozycyjnym lub LCD 2x16,
- minimalny zakres dekodowania:
  - + symbole heksadecymalne (0-9, A-F (a-f)),
  - + symbole sterujące (BS, LF, CR, ESC),(podany minimalny zakres dekodowania można dowolnie rozszerzyć),
- w realizacji należy wykorzystać moduł **uart\_rx\***,
- realizacja przy pomocy dowolnych układów z biblioteki Spartan3E.

B) Wykonaj projekt zamka cyfrowego.

(stopień trudności: 2/3 – w zależności od zrealizowanych funkcji urządzenia)

Wymagania projektowe:

- sterowanie z programu terminala (np. Hyper Terminal, PuTTY),
- prezentacja wyników/komunikatów na wyświetlaczu 7-segmentowym 4-pozycyjnym lub LCD 2x16,
- otwarcie zamka kodem 4-znakowym (4 młodsze cyfry numeru indeksu osoby wykonującej projekt),
- sygnalizacja otwarcia zamka,
- sygnalizacja wprowadzenia błędnego kodu,
- zamykanie zamka wymuszone komendą z terminala sterującego lub automatyczne po stałym czasie opóźnienia  $dT \approx 10s$ ,
- w realizacji należy wykorzystać moduł **uart\_rx\***,
- realizacja przy pomocy dowolnych układów z biblioteki Spartan3E.

Dokumentacja projektowa musi zawierać wyniki symulacji z wyświetleniem przebiegów czasowych w kluczowych punktach pracy urządzenia.

\* moduł **uart\_rx** jest blokiem odbiorczym transmisji szeregowej – źródła i opis na serwerze