

Obsługa portu RS-232

Zadanie: Należy zaprojektować moduły nadajnika i odbiornika danych w standardzie RS-232 dla następujących parametrów transmisji: 8 bitów danych, bez bitu parzystości, jeden bit stopu, szybkość transmisji 9600bps bez sprzętowej kontroli przepływu.

Moduły należy połączyć tak, aby odbiornik RS232 po otrzymaniu danych podawał je do sumatora i po dodaniu do nich wartości 20h, moduł nadajnika wysyłał je.

Opis sygnałów:

clk_i - zegar 100 MHz,

rst_i - reset asynchroniczny (przycisk **BTNR**) (w trybie zdalnym - wirtualny przycisk **BTN3**),

RXD_i - wejście danych RS232,

TXD_o - wyjście danych RS232.

Aby wysyłać dane RS232 na płytkę FPGA należy uruchomić program Putty, wybrać połączenie „Serial”, wpisać prędkość (9600) oraz port: COMx (x to numer portu, należy go odczytać w programie „device manager” (program ten można łatwo znaleźć na komputerze za pomocą klawisza szukania „lupa”) – pod pozycją „Ports (COM&LPT)” należy znaleźć urządzenie o nazwie „USB serial Port” – numer portu będzie podany obok). Należy jeszcze włączyć echo lokalne, aby znaki wysyłane do płytki były widoczne na ekranie – trzeba kliknąć w kategorię „Terminal” i zaznaczyć opcję „Force on” dla „Local echo”. Potem kliknąć w klawisz „Open” w celu otwarcia połączenia. Dzięki lokalnemu „echu” będzie widać znaki wysyłane do płytki FPGA, a zaraz za każdym znakiem wysłanym powinien się pojawić znak odebrany z płytki – dzięki temu można łatwo sprawdzić czy projekt prawidłowo odbiera, przetwarza i wysyła znaki.

Uwaga! W projekcie musi być zawarty układ synchronizujący sygnał **RXD_i** z zegarem (szczegóły w screenaście wykładowym „Stany zabronione”)!

Należy wykonać symulację funkcjonalną oraz zweryfikować układ praktycznie poprzez zaprogramowanie płytki testowej.

Minimalne wymagania dotyczące symulacji i weryfikacji na płytce: wykonać reset na początku symulacji, następnie wysłać daną z przykładu (Rys. 1) trzy razy, za pierwszym razem z prędkością nominalną (9600 bps), za drugim z prędkością mniejszą o 4% od nominalnej, za trzecim z prędkością większą o 4% od nominalnej. We wszystkich przypadkach odpowiedź układu powinna być taka sama (53h + 20h = 73h).

W celu weryfikacji na płytce FPGA należy wpisać w terminalu wielkimi literami imię, następnie wprowadzić spację, potem nazwisko, a następnie numer indeksu zaliczającego studenta (lub jednego ze studentów zaliczających ćwiczenie).

Prawidłowa zmiana kodów znaków (+20h) powinna być widoczna w oknie terminala. Częstotliwość zegara clk_i w symulacji – 100 MHz.

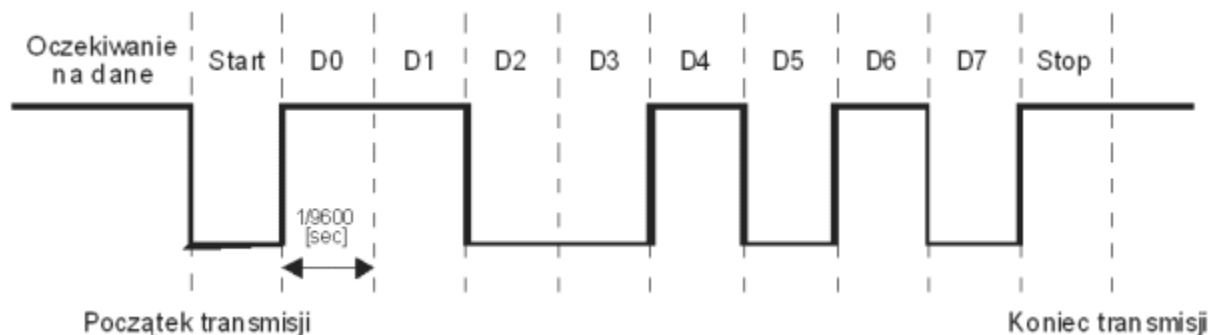
Fragment głównego pliku projektowego VHDL z deklaracją sygnałów:

```
entity top is
    Port ( clk_i : in STD_LOGIC;
          rst_i : in STD_LOGIC;
          RXD_i : in STD_LOGIC;
          TXD_o : out STD_LOGIC);
end top;
```

Plik z ograniczeniami projektowymi dla płytki Nexys-A7 (układ FPGA xc7a100tcs9324-1): [iup9s.xdc](#)

Wersja pliku z ograniczeniami projektowymi przeznaczona dla trybu zdalnego (obsługa przycisków wirtualnych): [iup9z.xdc](#)

Nadawanie i odbiór sygnałów w standardzie RS232 Nadawanie i odbiór sygnałów w standardzie RS232 odbywa się w sposób szeregowy, oddzielnie na dwóch liniach w kierunkach do i od urządzenia. W czasie braku transmisji sygnał na danej linii jest w stanie wysokim. Rozpoczęcie transmisji inicjowane jest przez tzw. bit startu będący "0" logicznym, który powinien trwać przez okres równy odwrotności prędkości transmisji, w naszym przypadku wynoszący 1/9600 [sekund]. Wszystkie następne informacje są przesyłane z identycznym okresem trwania. Następnie nadawane są szeregowo dane począwszy od najmniej znaczącego bitu aż do bitu najbardziej znaczącego (**D0-D7**). Później występuje bit parzystości, będący operacją logiczną **XOR** na danych **D0-D7**. Bit parzystości jest opcjonalny i w przypadku niniejszego zadania nie występuje. Zakończenie transmisji sygnalizowane jest bitami stopu, w ilości zazwyczaj od 1 do 2. Przykład sygnału, zgodnego z wymaganiami zadania, przenoszącego kod **01010011** przedstawiony jest na poniższym rysunku.



Rys.1 Przykładowa transmisja kodu 01010011 przez RS-232

Informacje dodatkowe o standardzie RS232:

<http://www.fizyka.umk.pl/~ptarg/labview/folie/RS232.pdf>

<http://pl.wikipedia.org/wiki/RS-232>