

Uniwersalny asynchroniczny nadajnik-odbiornik

UART

Universal Asynchronous Receiver- Transmitter

Cel projektu:

Zbudowanie układu transmisji znaków z komputera na wyświetlacz zamontowany na płycie Spartan-3AN, poprzez łącze RS i program TeraTerm.

Laboratorium Projektowania Systemów Cyfrowych AGH WFiS



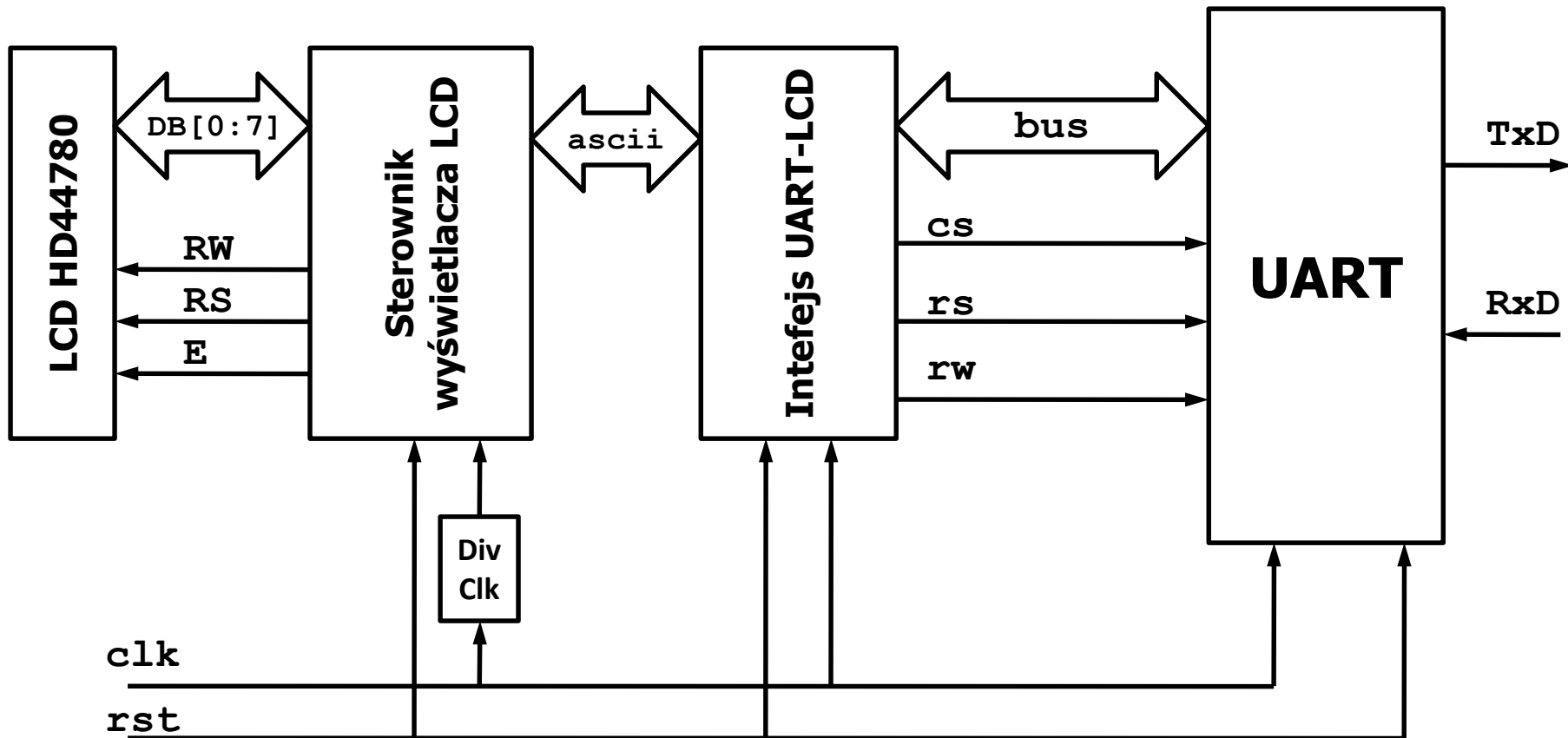
W czasie nadawania UART pobiera 8 bitów danych z magistrali równoległej i przetwarza je na szeregowy strumień danych.

W czasie odbierania UART wykrywa bit startu odbiera 8 bitów danych i przetwarza na formę równoległą kiedy wykryje bit stopu.

Aby nie przesyłać zegara UART musi przychodzący strumień danych zsynchronizować z lokalnym zegarem.

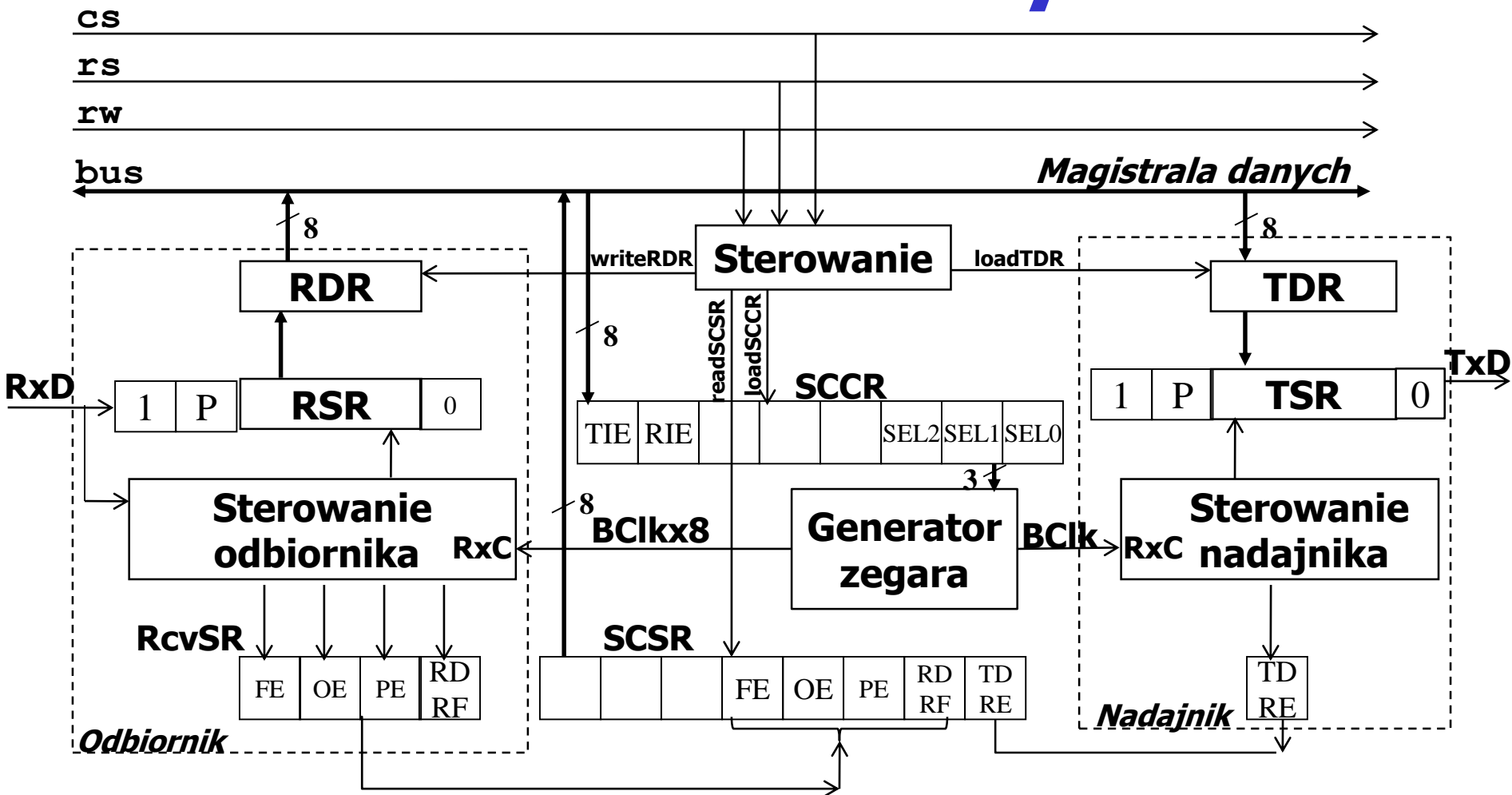
Przykład pokazuje transmisję litery U.

Całość systemu



Waznym elementem jest projekt układu sprzęgającego UART z wyświetlaczem. Ścieżka danych sterownika wyświetlacza LCD wymaga przeprojektowania.

Schemat blokowy



P – bit parzystości analizowany przez odbiornik i generowany przez nadajnik.

Układy te pominięto na rysunku

Rejestry wewnętrzne

RSR	Receive Shift Register
RDR	Receive Data Register
TDR	Transmit Data Register
TSR	Transmit Shift Register
SCCR	Serial Communication Control Register
SCSR	Serial Communication Status Register

Każdy z rejestrów jest adresowany przez zewnętrzny procesor

RDR, SCSR, SCCR – sterują magistralę danych przez bufor trybów trójstanowych

TDR, SCCR – ładowane z magistrali danych

Cztery z rejestrów są dostępne dla zarządzającego procesora:

cs	rw	rs	opis
0	x	x	Układ niekatywny
1	1	0	Odczyt SCSR
1	0	0	Zapis SCCR
1	1	1	Odczyt RDR
1	0	1	Zapis TDR

Rejestry wewnętrzne

Opis bitów rejestru **SCSR**:

Naprawdę są to czterobiowy rejestr w odbiorniku i jeden przerzutnik w nadajniku.

Nazwa	Bit	RcvSR	Opis	Opis
TDRE	0	–	Transmit Data Register Empty	Przerzutnik w nadajniku
RDRF	1	0	Receive Data Register Full	Rejestr statusowy RcvSR w odbiorniku
PE	2	1	Parity Error	
OE	3	2	Overrun Error	
FE	4	3	Frame Error	

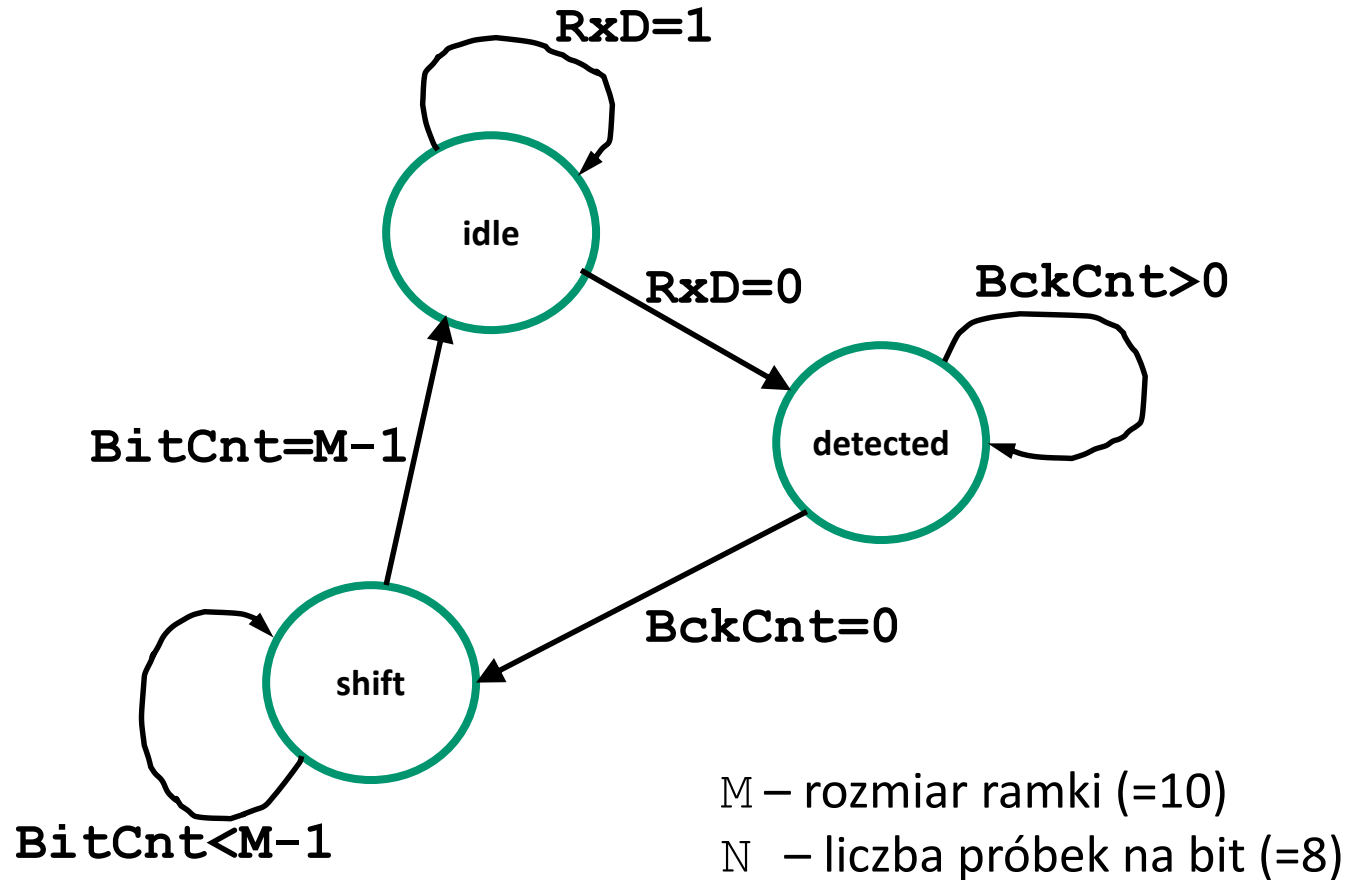
Opis bitów rejestru **SCCR**:

Nazwa	Bit	Opis	Opis
TIE	7	Transmit Interrupt Enable	Odblokuj przerwania
RIE	6	Receive Interrupt Enable	Odblokuj przerwania
sel2	2	Data rate selector	Wybór prędkości transmisji
sel1	1		
sel0	0		

Moduły projektu

Nazwa modułu	input	output	inout	Opis
uart	clk, rst, rw, cs, rs, RxD		[7:0]bus	Moduł szczytowy
transmitter	clk, rst, loadTDR, [7:0]bus, TxC	TE, TxD		Moduł nadajnika
receiver	clk, rst, readRDR, RxC, RxD	[7:0]RDR, [2:0]RcvSR		Moduł odbiornika
brate	clk, rst, [2:0]sel,	Bclk, Bclkx8		Moduł ustalania prędkości przesyłu

Graf automatu odbiornika



Opis pracy odbiornika

Działanie odbiornika:

- ❑ Gdy zostanie wykryty bit startu kolejne bity zostają wczytane szeregowo do RSR.
- ❑ Gdy wszystkie bity i bit stopu zostały wczytane, RSR jest ładowany do RDR i ustawiana jest flaga RDRF w SCSR.
- ❑ Gdy procesor wykryje „1” w RDRF wczytuje zawartość RDR i zeruje RDRF.

Ponieważ wejściowy strumień bitów jest nie zsynchronizowany lokalnym zegarem wystąpią problemy z rozpoznaniem gdzie bity się kończą, a gdzie zaczynają. Aby uniknąć tych trudności linia RxD jest próbkowana 8 razy na bit tzn. na każdym narastającym zboczu $BclkX8 \uparrow (RxC)$. Po wykryciu stanu niskiego na RxD (bit startu) odczekujemy 4 okresy $BclkX8$ i w ten sposób trafiamy na środek bitu. Środek kolejnego bitu jest po kolejnych 8 okresach $BclkX8$. Następnie kontynuujemy odczyt co 8 okresów zegara $BclkX8$, aż napotkamy bit stopu.

Opis pracy odbiornika c.d.

Potrzebne są dwa liczniki:

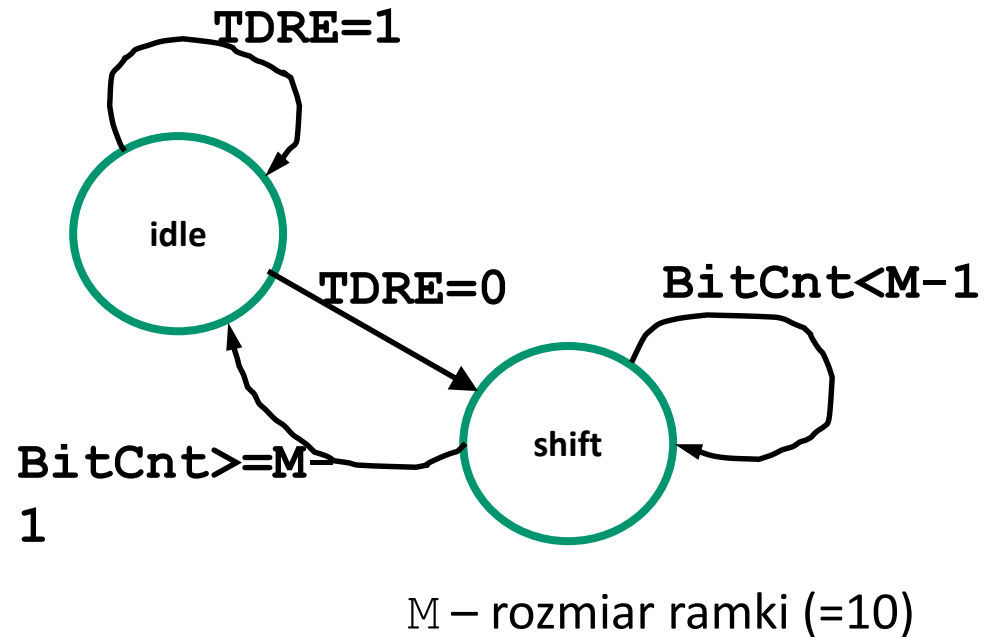
- ❑ `BckCnt` – liczy okresy zegara `BclkX8` (próbki), modulo `N`
- ❑ `BitCnt` – liczy bity przyjęte po bicie startu, modulo `M`.

W stanie **idle** automat oczekuje na bit startu $RxD=0$ i przechodzi do stanu **Detected**. W tym stanie automat próbuje linię RxD na kolejnych narastających zboczach $RxC \uparrow$ i za każdym razem dekrementowany jest licznik `BckCnt`. Licznik ten jest początkowo inicjowany na 4, ale potem pracuje jako modulo 8. Gdy `BckCnt` wyzeruje się układ przechodzi do stanu **shift**. Czas przebywania w tym stanie jest wyznaczony przez inkrementację licznika `BitCnt` modulo 10. Licznik ten jest inkrementowany tylko w chwilach wyzerowania się licznika `BckCnt`.

Na koniec stanu **shift**:

- ❑ zawartość 7 bitów informacji w `RSR` jest przepisywana do `RDR`,
- ❑ znacznik `RF` jest ustawiany by zasygnalizować do interfejsu zewnętrznego, że dana w `RDR` jest gotowa do pobrania,
- ❑ znacznik `FE` jest ustawiany jak zaprzeczenie aktualnego stanu linii `RxD`,
- ❑ znacznik `OE` jest ustawiany ???,
- ❑ znacznik `PE` jest ustawiany na wynik obliczenia parzystości danych.

Graf automatu nadajnika



Opis pracy nadajnika

Bit `TDRE` w rejestrze statusowym odbiornika `SCSR` jest ustawiany na 1 gdy `TDR` jest pusty. Wtedy zewnętrzny interfejs ładuje `TDR` sygnałem `loadTDR` co powoduje wyzerowanie `TDRE`.

Kiedy zewnętrzny interfejs jest gotowy do transmisji:

- ❑ zewnętrzny interfejs czeka na `TDRE=1` i ładuje słowo do `TDR` i zeruje `TDRE`
- ❑ UART przesyła dane z `TDR` do `TSR` i ustawia `TDRE=1`
- ❑ UART wysyła bit startu „0” i następnie przesuwa `TSR` w prawo aby wysłać siedem bitów, bit parzystości i bit stopu „1”.

W stanie **idle** automat oczekuje na wyzerowanie `TDRE` i utrzymywany jest stan wysoki na `TxD`. Gdy to się stanie rejestr `TSR` jest ładowany zawartością `TDR` wraz bitami startu, stopu i parzystości, a licznik bitów `BitCnt` jest inicjowany na zero. Równocześnie automat przechodzi do stanu **shift**. W tym stanie na każdym narastającym zboczach `TxC` \uparrow `TSR` jest przesuwany by wysłać kolejny bit i licznik bitów jest inkrementowany. Gdy `BitCnt=9` układ wraca do stanu **idle**.

Baud rate generator

Trzy niższe bity rejestru SCCR wybierają jedną z ośmiu szybkości transmisji: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400.

Maksymalna potrzebna częstotliwość wynosi: $38400 \times 8 = 307\,200$

Ponieważ zegar jest 50MHz to trzeba go podzielić przez 162,8 czyli albo szybkości będą trochę różne albo zegar trzeba dostroić.

