

Aufgabe 7: Serielle Schnittstelle (RS232)

G. Kemnitz, C. Giesemann, TU Clausthal, Institut für Informatik

21. Mai 2015

Zusammenfassung

Für den Empfänger einer seriellen Schnittstelle ist eine funktionierende Schaltung vorgegeben. Diese ist zu untersuchen und zu testen. Anschließend ist ein Sender zu entwerfen und der Empfänger um Zusatzfunktionen zu erweitern.

1 Funktion

Die PCs im Übungsraum und unsere Versuchsbaugruppe sind über USB verbunden. Auf der Versuchsbaugruppe hinter dem USB-UART-Stecker befindet sich auf der Unterseite ein FTDI-Chip für die Umsetzung von USB auf das UART-Protokoll. Der FTDI-Chip ist im Wesentlichen über zwei Signalleitungen mit dem FPGA verbunden (Abbildung 1).

- Empfangsleitung RXD (TXD-FTDI nach RXD-FPGA) und
- Sendeleitung TXD (TXD-FPGA nach RXD-FTDI).

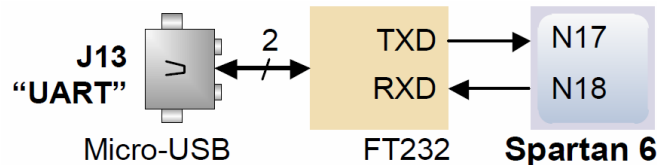


Abbildung 1: UART-Anschluss am FPGA

Die serielle Übertragung in Hin- und Rückrichtung erfolgt byteweise, zeitgleich in beide Richtungen mit dem Protokoll in Abbildung 2. Während einer Übertragungspause ist das jeweilige Sendesignal TXD='1'. Jedes Datenpaket beginnt mit einem Startbit='0' gefolgt von den 8 Datenbits, optional einem Paritätsbit und einem Stoppbit='1' (vgl. Abbildung 2). Danach kann sich eine Pause (Übertragungsleitung gleich Eins) oder das Startbit der nächsten Übertragung anschließen. Das Paritätsbit berechnet sich aus der Modulo-2- oder EXOR-Summe der Datenbits und dient zur Fehlererkennung.

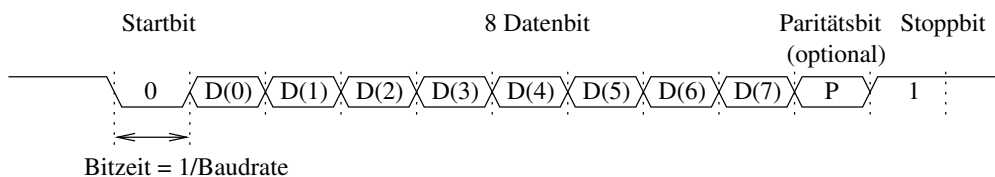


Abbildung 2: Rahmenformat RS232

Der Sender für die serielle Schnittstelle besteht aus

- einer Schaltung zur Bereitstellung des Taktes
- einem Automaten zur Umwandlung eines Bytes in einen Datenstrom entsprechend Protokoll.

Der Basistakt GCLK unserer Übungsbaugruppe hat eine Frequenz von 100 MHz. Dieser Takt wird üblicherweise heruntergeteilt auf einen Takt mit der 16-fachen Baudrate und einem Tastverhältnis 1:1:

```
process(GCLK)
  variable counter : integer range 0 to cTeiler - 1 := cTeiler - 1;
begin
  if rising_edge(GCLK) then
    if counter = 0 then
      clk_16x_baud <= not clk_16x_baud;
      counter := cTeiler - 1;
    else
      counter := counter - 1;
    end if;
  end if;
end process;
```

Gebräuchliche Baudraten und Teilerkonstanten zeigt nachfolgende Tabelle:

Baudrate	cTeiler (Basistakt 100 MHz)
2,4 kBaud	1296
4,8 kBaud	648
9,6 kBaud	324
19,2 kBaud	162

Der Takt `clk_16x_baud` wird in einem weiteren Teiler (4-Bit-Zähler) durch 16 geteilt und als Takt für den Sendeautomaten genutzt.

Der Sendeautomat soll als Graph spezifiziert werden. Solange keine Sendung gestartet ist, bleibt der Automat im Pausezustand und hält die Sendeleitung TXD auf Eins. Wenn eine Übertragung gestartet wird, durchläuft er die Zustandsfolge:

- Sende Startbit: TXD<='0'
- Sende Bit 0: TXD<=D(0)
- ...
- Sende Bit 7: TXD<=D(7)
- Sende Paritätsbit: TXD<=D(0) xor D(1) xor ... xor D(7)
- Sende Stoppbit (TXD<='1')

Die zu versendenden Daten werden in der Regel von einer Schaltung, die nicht mit dem Baudrarentakt arbeitet, bereitgestellt. Das erfordert ein Handshake-Protokoll:

- Datenquelle setzt das Anforderungssignal »req« auf Eins und stellt die Daten auf dem Bus »Din« bereit.
- Der serielle Sender übernimmt, sobald er zur Übernahme bereit ist, die zu versendenden Daten und bestätigt mit `grant<='1'`.
- Versenden der Daten.

- Warten auf Rücknahme der Anforderung und Rücknahme der Bestätigung.
- Warten auf die nächste Anforderung.

Sende- und Handshake-Ablauf sind zu einem Ablaufgraphen zu vereinigen, wobei das Anforderungssignal nur in abgetasteter Form verarbeitet werden darf (Abbildung 4).

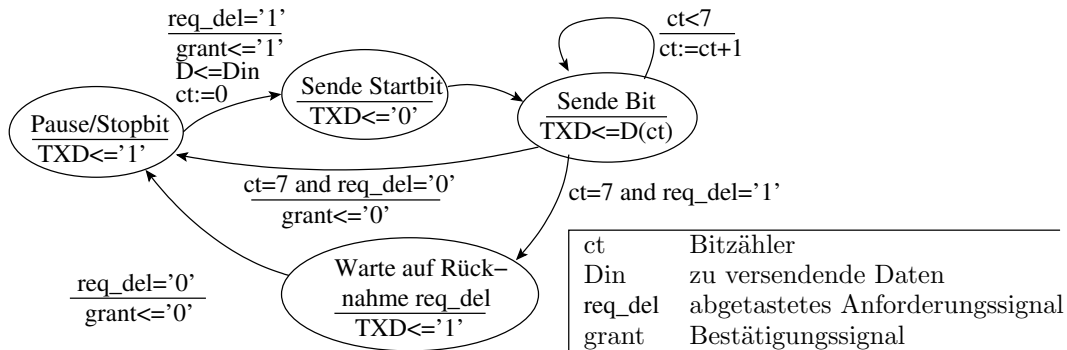


Abbildung 3: Ablaufgraph des Senders mit Handshake

Anmerkungen:

- Der dargestellte Automat versendet die Daten ohne Paritätsbit.
- Der Sender ist durch einen Operationsablaufgraphen beschrieben, in dessen Knoten ein Zähler gesteuert und an dessen Kanten der Zählstand mit ausgewertet wird. Die Automatenbeschreibung könnte auch so abgewandelt werden, dass die versendeten Bits nicht durch einen Zähler, sondern den Automatenzustand unterschieden werden.
- Der Zähler muss vom Typ Integer range 0 to 7 sein, damit er eines von 8 Bits aus D auswählen kann. D muss entsprechend den Typ std_logic_vector(7 downto 0) haben.

Der Empfänger arbeitet mit derselben Baudrate wie der Sender. Dadurch kennt er die Bitzeit. Den Startzeitpunkt einer Übertragung bekommt er vom Sender nur indirekt in Form der fallenden Flanke zu Beginn der Übertragung mitgeteilt. Aus der Kenntnis der Baudrate und der Startflanke muss er die Übernahmezeitpunkte der einzelnen Bits bestimmen. Die Lösung beinhaltet, dass der 1:16-Vorteiler einbezogen wird. Der Empfänger startet wie der Sender im Pausezustand. Statt auf eine Sendeanforderung wartet er auf die fallende Flanke des abgetasteten Empfangssignals »RXD_del«. Das Signal »RXD« wird dabei mit der mehrfachen, im Beispiel mit der 16-fachen Baudrate abgetastet. Sobald die Datenleitung auf Null wechselt, schaltet der Automat einen Zustand weiter und setzt den Taktvorteiler auf Null. Alle weiteren Zustandsübergänge erfolgen, wenn der Vorteiler die Hälfte seines Endwertes erreicht, d.h. etwa in der Mitte zwischen den Signalwechseln des Senders, wenn die gesendeten Daten stabil sind. Beim ersten Zustandswechsel wird eine Null auf der Empfangsleitung erwartet (Startbit). Beim 2. bis 9. Zustandswechsel werden die Datenbits übernommen. Abschließend wird kontrolliert, dass das Stoppbit Eins ist. Falls zur Fehlererkennung zusätzlich ein Paritätsbit mit übertragen wird, ist auch das auszuwerten.

Auch die Übergabe empfangender Daten an die weiterverarbeitende Schaltung erfolgt üblicherweise über Handshake (Abbildung 4):

- Nach fehlerfreiem Empfang eines Bytes eine Schreibanforderung stellen.
- Auf Bestätigung warten und die Anforderung zurücknehmen.

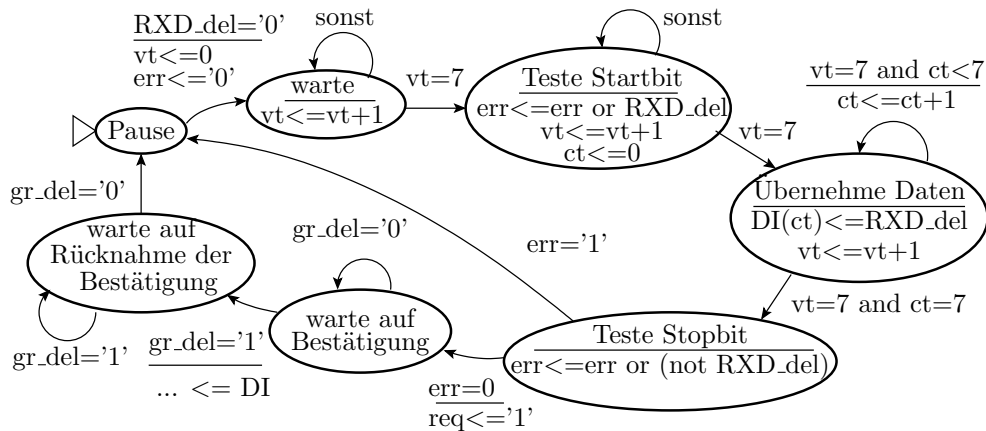


Abbildung 4: Empfänger mit Handshake als Ablaufgraph

- Auf Rücknahme der Bestätigung warten.

Anmerkungen:

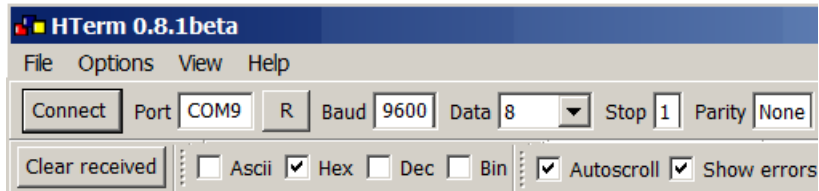
- Kein Paritätsbit
- vt ist ein Zähler modulo 16
- ct muss wie beim Sender vom Typ Integer range 0 to 7 sein, damit es ein Bit aus dem 8-Bit-Vektor DI auswählen kann
- das Bestätigungssignal der nachfolgenden Einheit ist, um laufzeitbedingte Fehlfunktionen auszuschließen, mit dem Takt des Empfängers abzutasten.

2 Test des Beispielentwurfs

Kopieren Sie die Dateien aus dem Archiv `PrakVHDL1_7.zip` in ein neue anzulegendes Verzeichnis für das Projekt. Öffnen Sie dann das Projekt in ISE.

1. Rekonstruieren Sie den Graphen des Empfangsautomaten aus der Datei `Aufg7.vhdl` und zeichnen Sie ihn auf das Abgabebblatt. Vergleichen Sie den gefundenen Graphen mit dem oben beschriebenen Graphen.
 - (a) Empfang mit oder ohne Auswertung des Paritätsbits?
 - (b) Welche Baudrate ist eingestellt?
 - (c) Enthält der Automat eine Handshake-Funktion?
 - (d) Wie funktioniert die eingebaute Echofunktion?
2. Übersetzen Sie den Entwurf und programmieren Sie den Schaltkreis.
3. Auf dem PC HTerm starten die richtige »COM« auswählen¹. 9600 Baud, 8 Daten, 1 Stopp- und kein Paritätsbit einstellen. Connect:

¹Die »COM« die im Auswahlmenü von HTerm hinzukommt, wenn das Kabel zur USB-UART-Buchse eingesteckt wird.



4. Verbindung testen: Nacheinander die Ziffern 0 bis 9 senden und kontrollieren, dass auf LD(7 downto 0) die zugehörigen ASCII-Codes 30h bis 39h angezeigt werden.

3 Entwurfsaufgaben

1. Stecken Sie an Stecker JA des NEXYS-Boards ein PmodTPH2 (Test Point Header), an P1 des Headers (RXD) den Logikanalysatoreingang 0 und an P2 (TXD) den Logikanalysatoreingang 1. Laden Sie die Datei Aufg7.xml aus dem Netz herunter. Tragen Sie in die Datei eine sinnvolle Abtastrate und Triggerbedingung ein, um die Übertragung vom PC zur Baugruppe aufzeichnen zu können. Begründen Sie Ihre Wahl der Abtastrate.
2. Entwerfen Sie einen Sendeautomaten, der bei Betätigung der Taste BTND das Byte SW(7 downto 0) an den PC sendet. Achtung: Entprellung nicht vergessen! Schreiben Sie den Sendeautomaten in eine neue Datei und verwenden Sie folgende Entity-Beschreibung:

```
entity a3_rs232_sender is
  port (
    clk_16x_baud : in  std_logic;
    TXD          : out std_logic;
    data         : in  std_logic_vector(7 downto 0);
    req          : in  std_logic;
    grant        : out std_logic
  );
end entity;
```

Simulieren Sie ihren Automaten mit Hilfe des Testrahmens Aufg7_tb.vhdl aus dem Netz und testen Sie die Schaltung auf der Baugruppe. Der Logikanalysator kann zur Problemfindung verwendet werden.

3. Modifizieren Sie die Empfängerschaltung so, dass auf den beiden linken 7-Segmentziffern das letzte empfangene Byte als 2-stellige Hexadezimalzahl dargestellt wird. Ergänzen Sie die Überprüfung des Paritätsbits im Zustand 9 des Automaten. Bei einem Fehler soll die DUO-Led auf der Ansteckbaugruppe PmodAMPEL an Stecker JC rot und sonst grün leuchten.

4 Abnahmekriterien

- Zeichnungen des Sende- und Empfangsautomaten auf dem Abgabebblatt
- Vorführung der Messergebnisse des Logikanalysators
- Simulation und Vorführung des Sendeautomaten
- Änderungen am Empfangsautomaten.