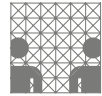


# 64-211 Übung Eingebettete Systeme



## Aufgabenblatt 4 Termine: 15.05. / 16.05. + 29.05. / 30.05.

Gruppe	
Name(n)	Matrikelnummer(n)

### Aufgabe 4.1 (Punkte 20)

Machen Sie sich mit der VHDL-Simulationsumgebung vertraut. Simulieren Sie beispielsweise die Fußgängerampel aus dem tar-Archiv: [tams.informatik.uni-hamburg.de/lectures/2013ss/vorlesung/es/doc/vhdlIntro.tgz](http://tams.informatik.uni-hamburg.de/lectures/2013ss/vorlesung/es/doc/vhdlIntro.tgz).

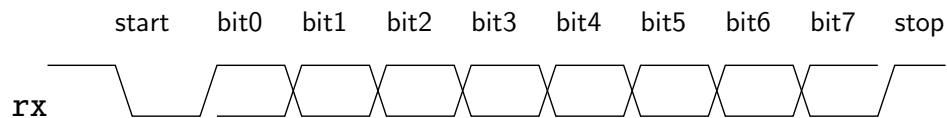
### Aufgabe 4.2 (Punkte 10+30+10)

Informieren Sie sich im Netz über die serielle Schnittstelle (RS-232). Mögliche Quellen zum RS-232 Protokoll und der Umwandlung der seriell empfangenen Bit in ein Byte (ASCII-Zeichen) sind:

- [en.wikipedia.org/wiki/RS-232](http://en.wikipedia.org/wiki/RS-232)
- [en.wikipedia.org/wiki/Universal\\_asynchronous\\_receiver/transmitter](http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter)
- [www.rn-wissen.de/index.php/RS232](http://www.rn-wissen.de/index.php/RS232)
- ...

1. Machen Sie sich mit der seriellen Schnittstelle vertraut.

2. Ein endlicher Automat soll die *logischen Pegel* einer RS-232 Schnittstelle (RxD) abtasten und ein Byte Nutzdaten „auf sammeln“. Dazu werden Startbit, 8-Datenbit und ein Stoppbit übertragen, wie in der Grafik skizziert:



byteEn

Abtastzeitpunkt

FSM-Zustand

Zeichnen Sie dazu die Abtastzeitpunkte und Automatenzustände ein. Außerdem soll für nachfolgende Verarbeitungseinheiten ein Signal `byteEn` aktiviert werden, das dem Datenempfang anzeigt. Skizzieren Sie den zugehörigen Zustandsübergangsgraphen. Analog zu dem Beispiel der Ampelschaltung kann ein Zähler benutzt werden, um Wartezeiten in den Zuständen zu realisieren.

3. Wenn die Datenübertragung bei 9 600 Baud erfolgen soll und die Schaltung mit 8 MHz getaktet ist, welche Werte müssten Sie für die Zählerstände annehmen?

**Aufgabe 4.3** (Punkte 30)

Codieren Sie den Automaten in VHDL und simulieren Sie die Schaltung mit einer Testumgebung.<sup>1</sup> Vervollständigen Sie folgende VHDL-Entity:

```

-- serRx.vhd
--
-- entity      serRx          -serial receiver
-- architecture behaviour    -RS232 receiver, 1-start bit, half-stop
--                                     -new byte signal: byteEn [H]
--
--          rx          \___+___+___+___+___+___+___+___/
--                      start 0  1  2  3  4  5  6  7  stop
--
--          byte        ..... ? (depends on implementation) ....._data_
--
--          byteEn     _____ ? (depends on implementation) .....
--
-----
library ieee;
use ieee.std_logic_1164.all;

-- entity      -----
-----

entity serRx is
port ( clk          : in  std_logic;          -- clock          [R]
      rstN         : in  std_logic;          -- reset           [L]
      rx           : in  std_logic;          -- rx line
      byte         : out std_logic_vector(7 downto 0); -- data
      byteEn       : out std_logic);        -- data enable [H]
end entity serRx;

-- architecture -----
-----

architecture behaviour of serRx is
...

end architecture behaviour;
-----
-- serRx.vhd      - end

```

[tams.informatik.uni-hamburg.de/lectures/2013ss/vorlesung/es/doc/serRx.vhd](http://tams.informatik.uni-hamburg.de/lectures/2013ss/vorlesung/es/doc/serRx.vhd)

<sup>1</sup>Die Testumgebung wird später bereitgestellt.