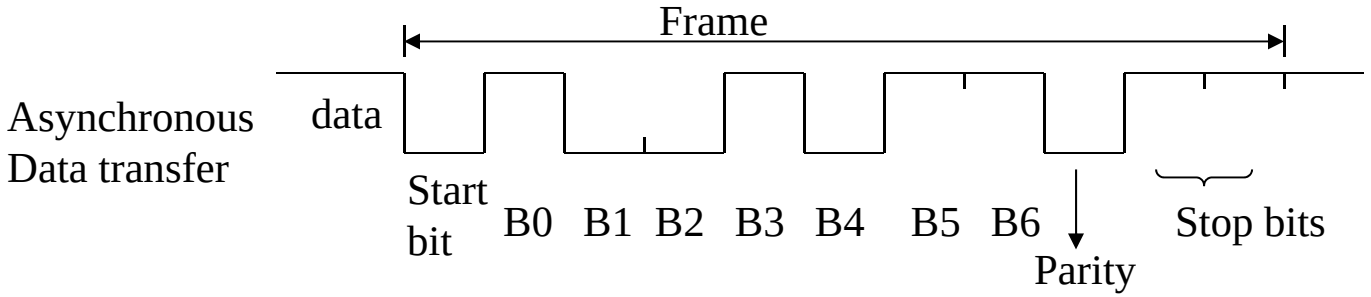
ECE 428 Programmable ASIC Design

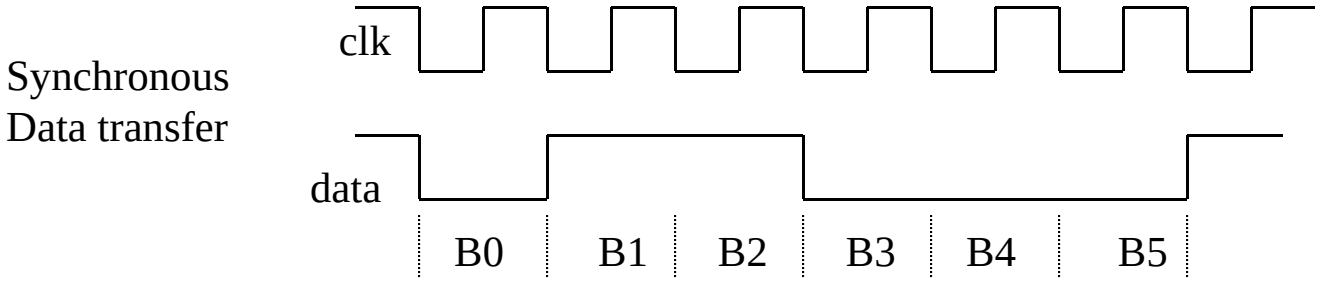
FPGA Implementation of Universal Asynchronous Receiver and Transmitter (UART)

Haibo Wang
ECE Department
Southern Illinois University
Carbondale, IL 62901

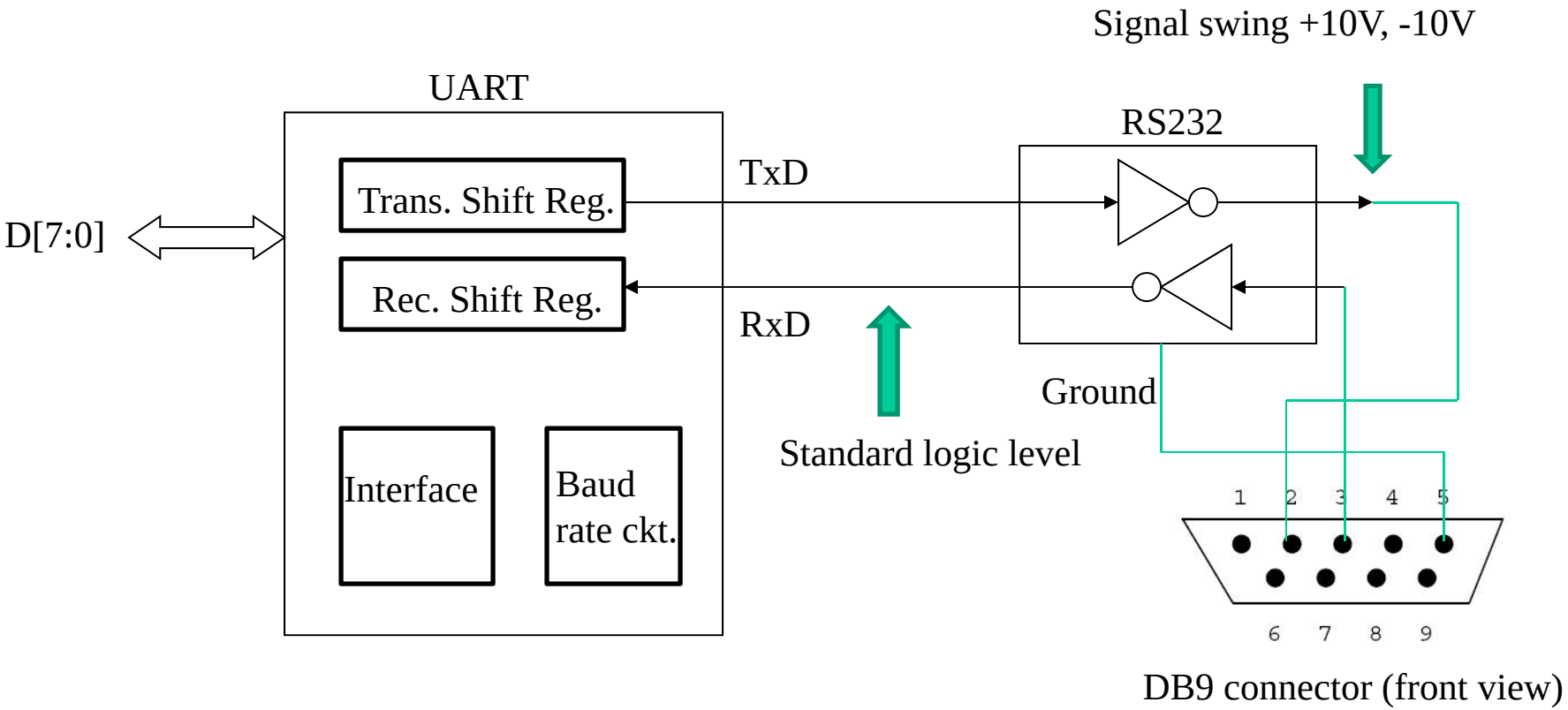
Serial Data Transfer

- Asynchronous ***v.s.*** Synchronous
  - — Asynchronous transfer does not require clock signal. However, it transfers extra bits (start bits and stop bits) during data communication
  - — Synchronous transfer does not transfer extra bits. However, it requires clock signal



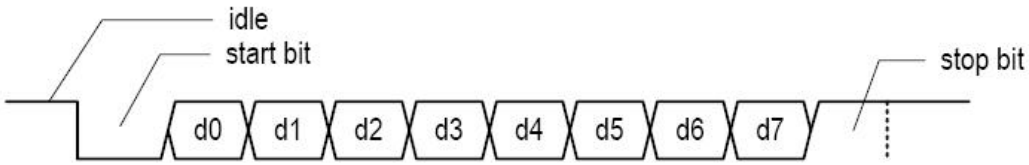


Overview of UART and RS-232

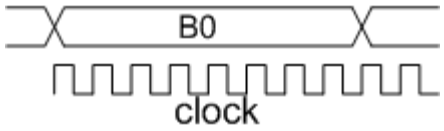


- Most materials presented here are from “FPGA Prototyping by Verilog Examples” by Pong Chu.
- You can download the UART chapter of the book at: http://academic.csuohio.edu/chu_p/rtl/fpga_vlog.html

UART Parameters

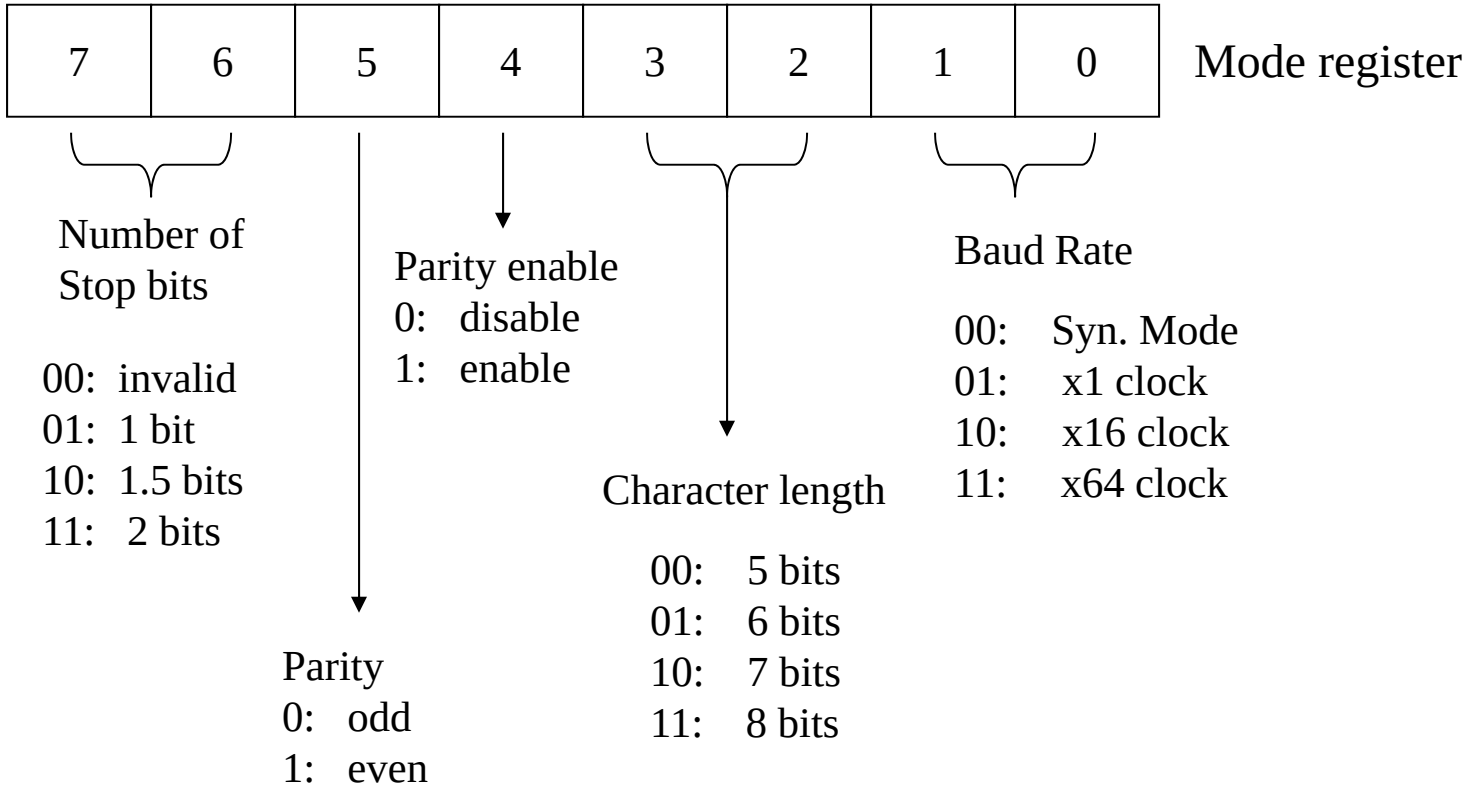


- Signal is 1 (high voltage) when the system is idle
- Start bit is placed before the data and, optionally, stop bits are placed at the end of data
  Start bit is 0 and stop bits are 1
- LSB is first transmitted or received
- **Baud rate**: number of bits per second; frequently used baud rate: 9600, 19,200
- **Number of Data bits**
- **Stop bits**
- Whether parity check is enabled?
- Multiplication factor for clock, e.g. x8 clk (means baud rate x 8 = system clock freq.)

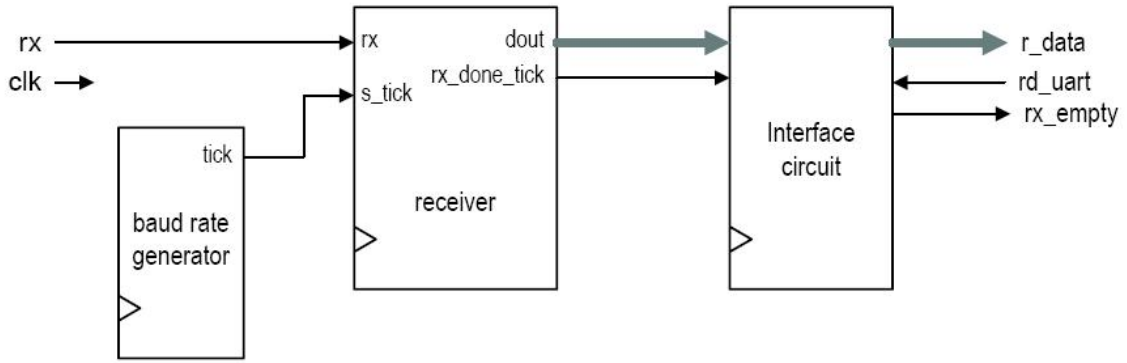


How to Assign UART Parameters

- Fix the parameters at the design stage (this example)
- Make it programmable with using configuration (mode) register
  Example: 8251 chip



Block Diagram of UART Receiving Subsystem

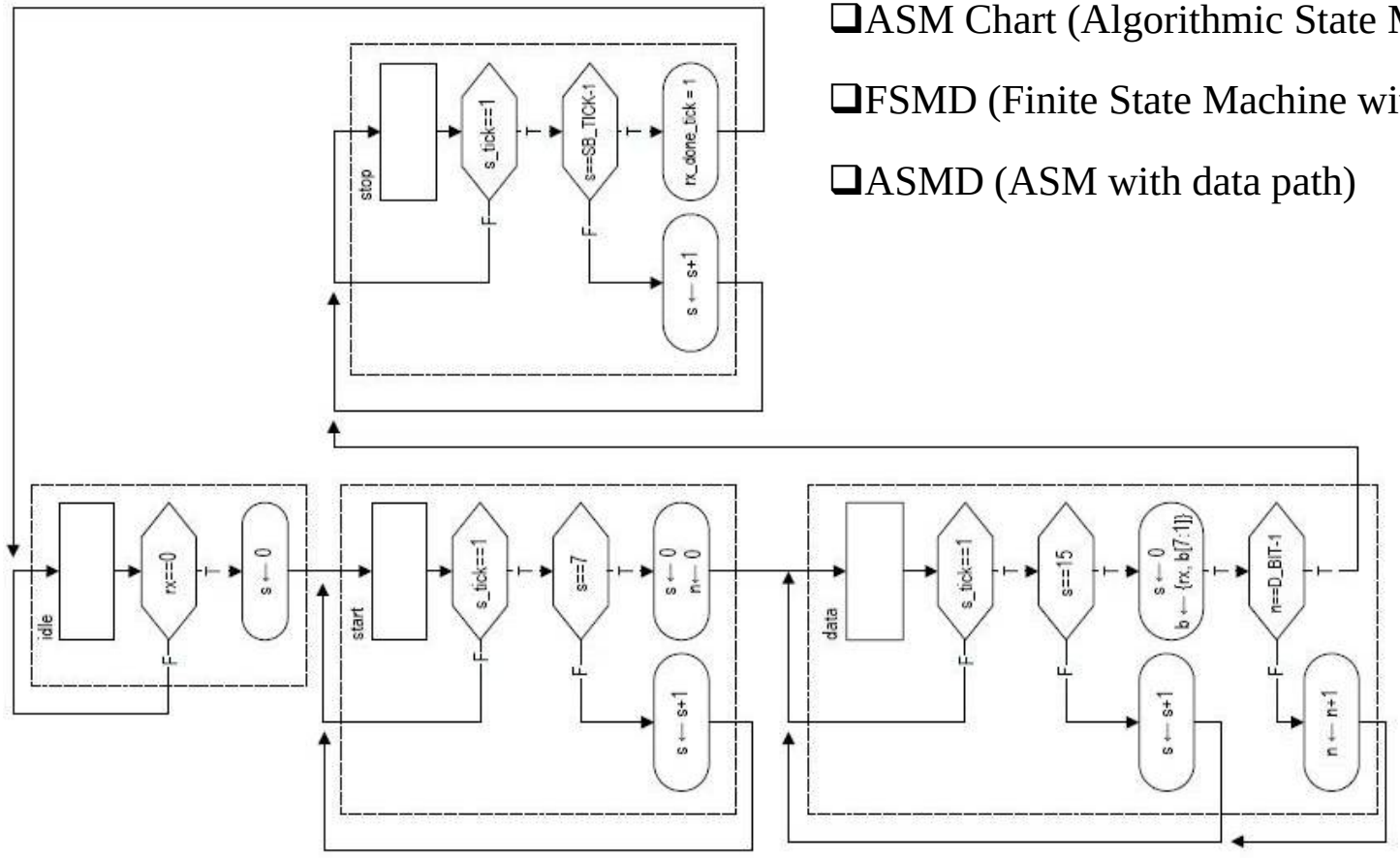


- Assume N data bits, M stop bits and f_{CLK}=16*baud rate

1. Wait until the incoming signal becomes 0, the beginning of the start bit, and then start the sampling tick counter.
2. When the counter reaches 7, the incoming signal reaches the middle point of the start bit. Clear the counter to 0 and restart.
3. When the counter reaches 15, the incoming signal progresses for one bit and reaches the middle of the first data bit. Retrieve its value, shift it into a register, and restart the counter.
4. Repeat step 3 N–1 more times to retrieve the remaining data bits.
5. If the optional parity bit is used, repeat step 3 one time to obtain the parity bit.
6. Repeat step 3 M more times to obtain the stop bits.

ASMD of UART Receiving Subsystem

- FSM (Finite State Machine)
- ASM Chart (Algorithmic State Machine)
- FSMD (Finite State Machine with Data Path)
- ASMD (ASM with data path)



HDL Coding for UART Receiving Subsystem

- The ASMD chart contains four states: ***idle, start, data, stop***

```
// symbolic state declaration
localparam [1:0]
   idle  = 2'b00,
   start = 2'b01,
   data  = 2'b10,
   stop  = 2'b11;
```

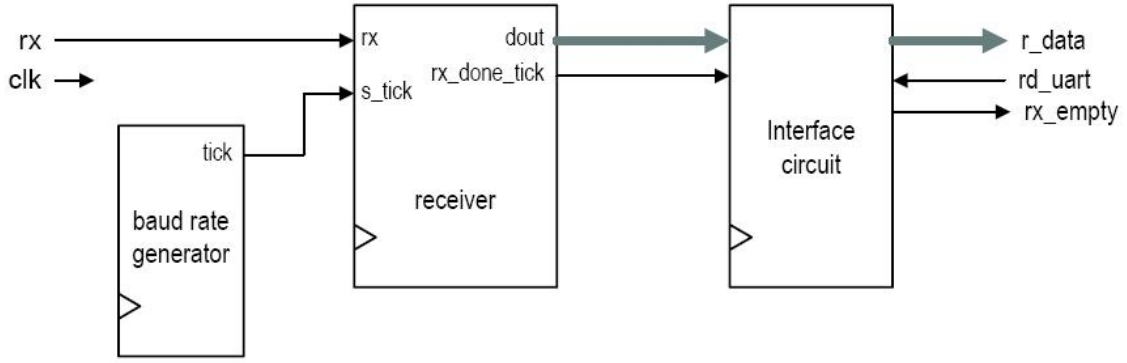
```
// FSMD state & data registers
always @(posedge clk, posedge reset)
   if (reset)
      begin
         state_reg <= idle;
         s_reg <= 0;
         n_reg <= 0;
         b_reg <= 0;
      end
   else
      begin
         state_reg <= state_next;
         s_reg <= s_next;
         n_reg <= n_next;
         b_reg <= b_next;
      end
```

HDL Coding for UART Receiving Subsystem

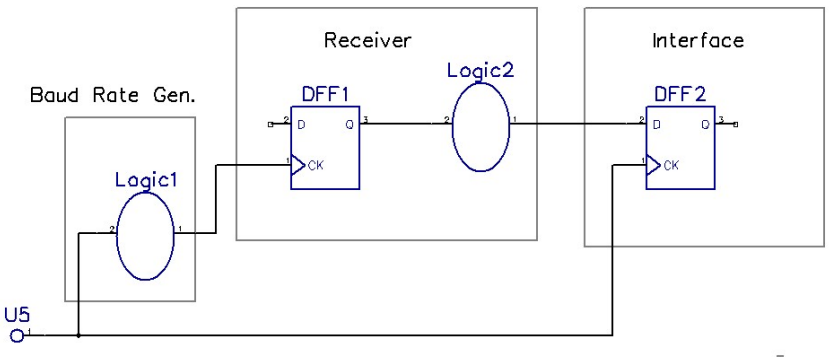
- Coding for operations within one state

```
case (state_reg)
   idle:
      if (~rx)
         begin
            state_next = start;
            s_next = 0;
         end
   start:
      if (s_tick)
         if (s_reg==7)
            begin
               state_next = data;
               s_next = 0;
               n_next = 0;
            end
         else
            s_next = s_reg + 1;
   data:
      if (s_tick)
         if (s_reg==15)
            begin
```

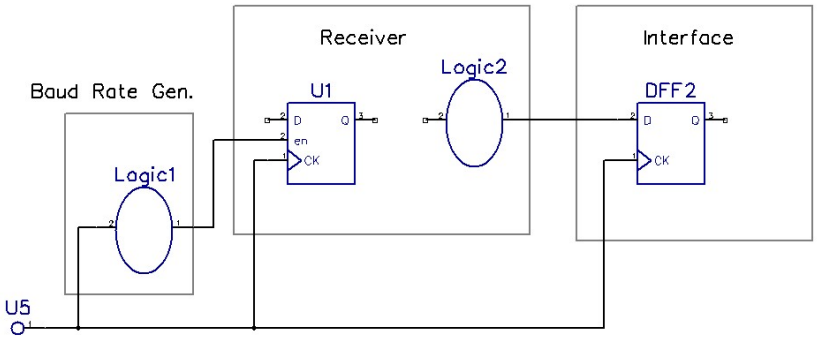
Clocking Issue of Receiver Design



- Should we use s_tick as clock signal or clock enable signal?

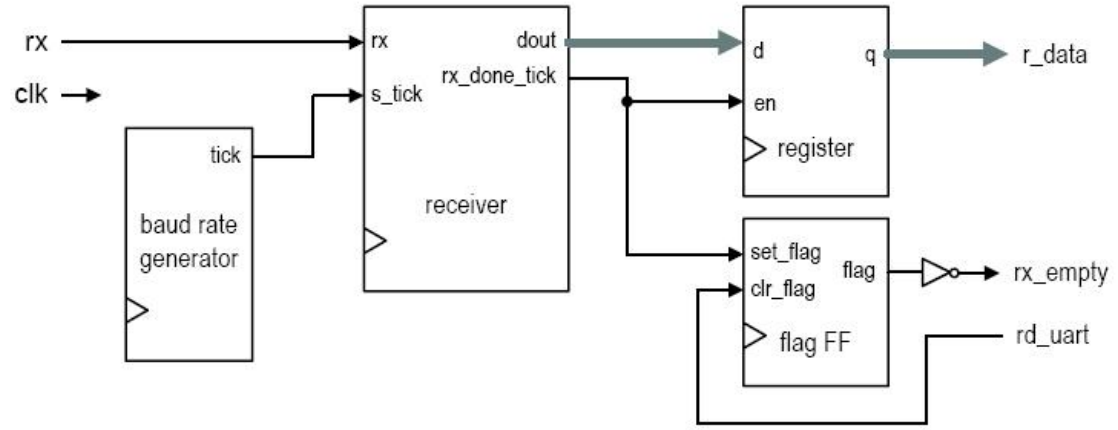


Functionally OK

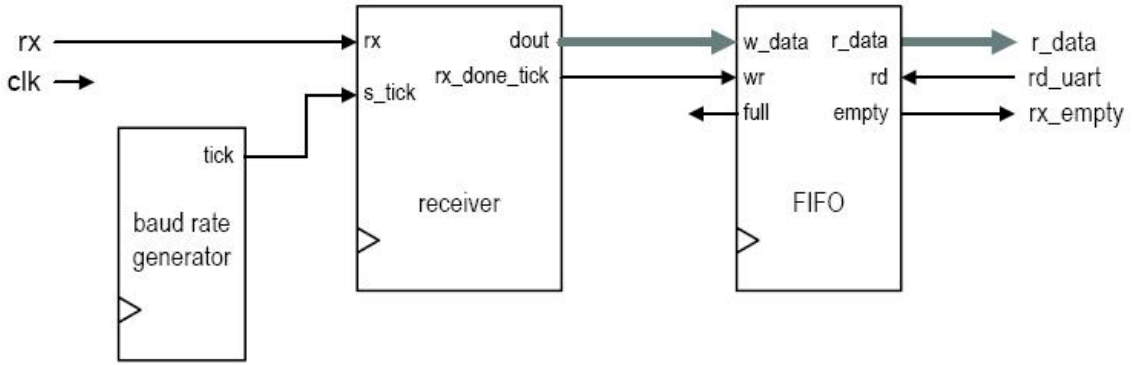


Better

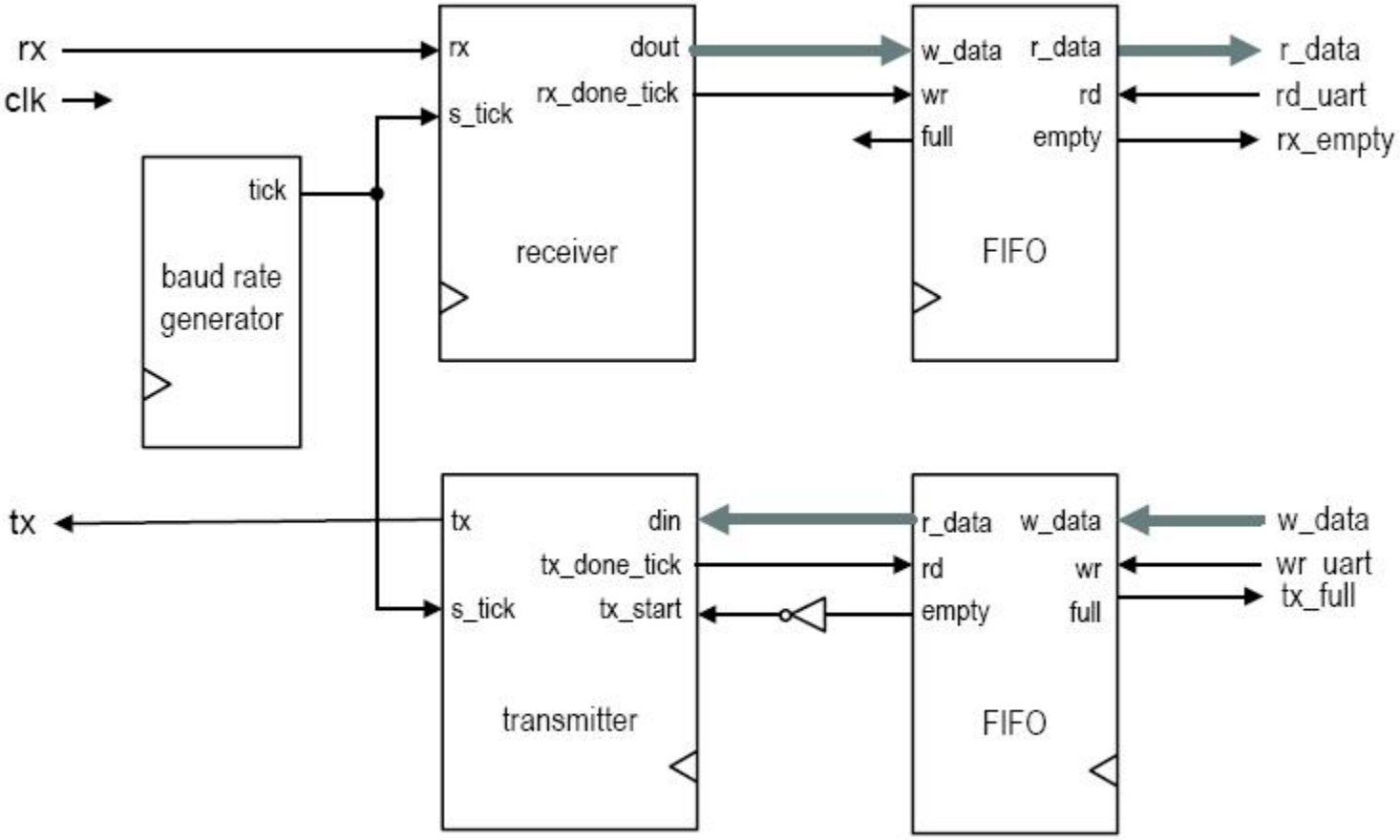
Interface Circuit of UART Receiving Subsystem



(b) Flag FF and one-word buffer



Complete UART Circuit



Configure HyperTerminal to test UART Circuit