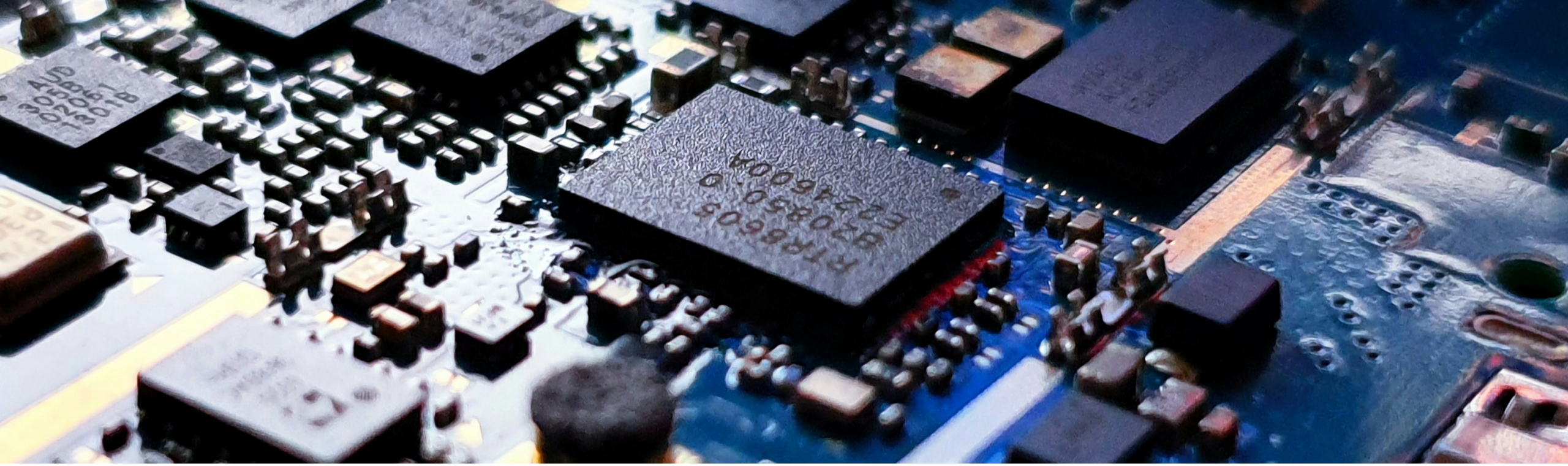


Laboratory 7

UART Design (Part 1)

Computer Architecture

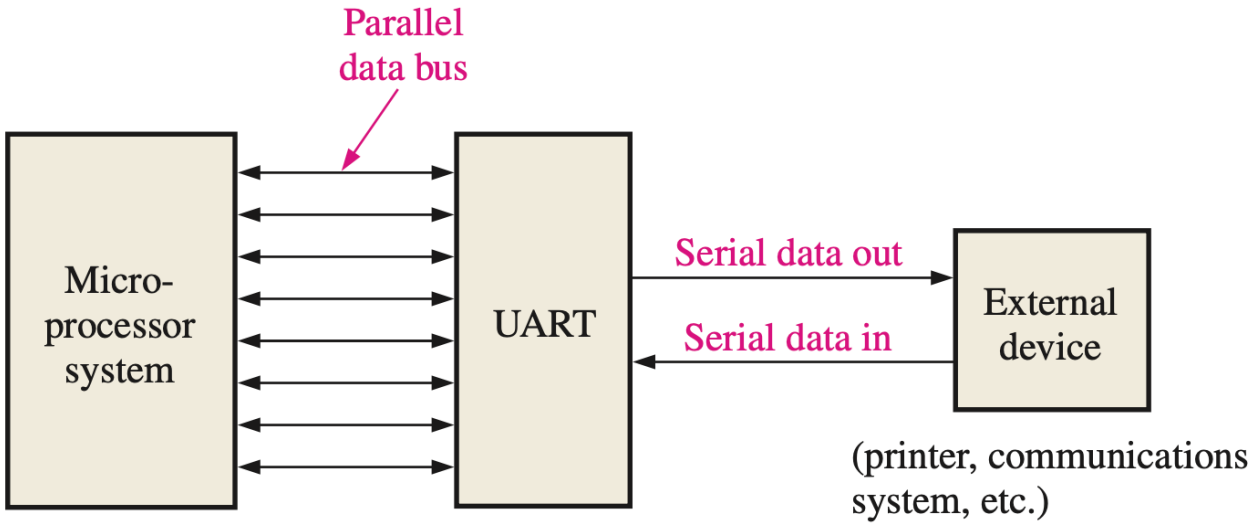
A.Y. 2023/24

Laboratory goals

- Understand the structure and functions of a UART device
- Design and implement said functions in VHDL (over multiple labs)
- In this lab: implement the transmission functionality.

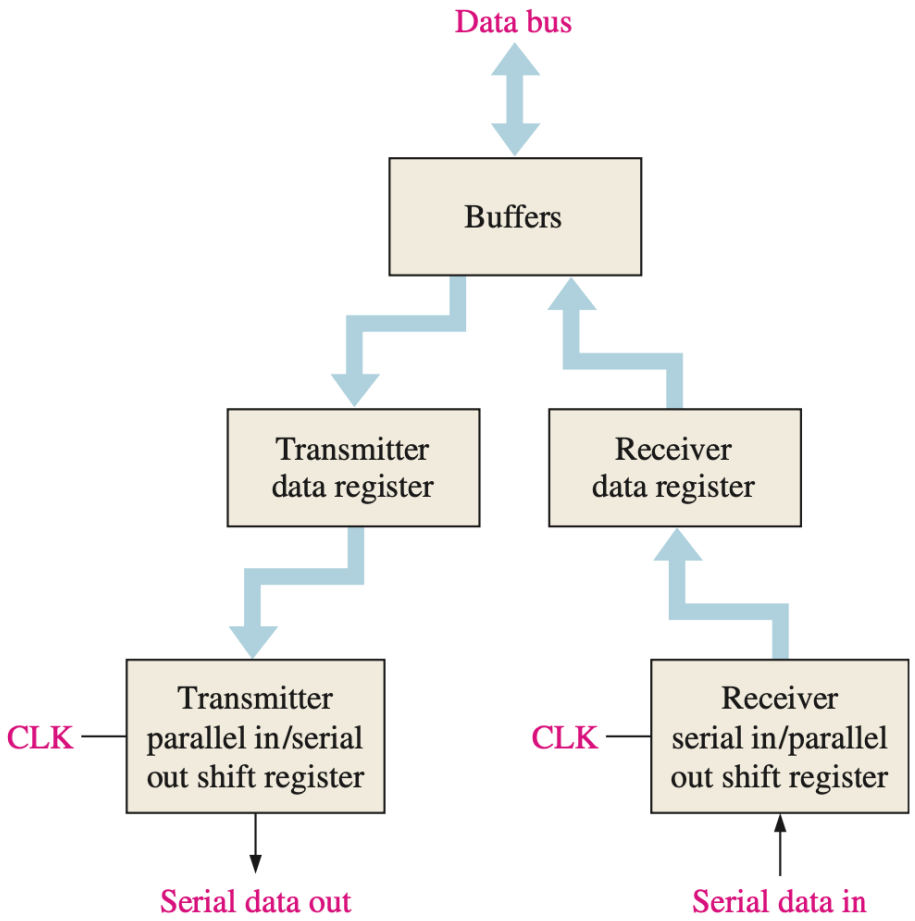
UART – Definition

A Universal Asynchronous Receiver Transmitter (UART) device is used to communicate between a microprocessor, or device that uses parallel data buses, and devices that communicate in serial data.



UART Block Diagram

As such, a UART must include parallel-to-serial as well as serial-to-parallel conversion blocks, to both send and receive data in serial format. The parallel data bus is interfaced by buffers.



UART Data Transfer Protocol

To enable data transfer without synchronization, both the UART and the device at the other end must agree on:

- Transmission speed (i.e. bit rate or baud)
- Data format: send 8-bit characters, framed by a start bit and a stop bit.

Receiver behavior

All operations of the UART hardware are controlled by an internal clock signal, which typically runs 8 or 16 times the bit rate.

- The receiver tests the state of the incoming signal on each clock pulse, looking for the beginning of the start bit. If the apparent start bit lasts at least one-half of the bit time, it is valid and signals the start of a new character. If not, it is considered a spurious pulse and is ignored.
- After waiting one more bit time, the state of the line is sampled again, and the resulting level is clocked into a shift register. After the required number of bit periods for the character length have elapsed (usually 8), the contents of the shift register are made available to the receiving system. The UART will set a flag indicating new data is available.

Transmitter behavior

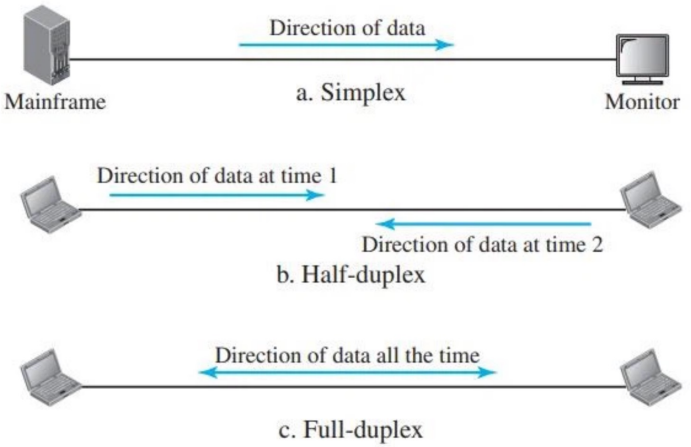
Transmission is more straightforward, as the timing does not have to be determined from the line state nor is it bound to any fixed time intervals.

- As soon as the sending module puts a character in the shift register (after completion of the previous transmission), the UART generates a start bit, shifts the required number of data bits out to the line, generates and sends the parity bit (if used), and sends the stop bits.
- Since transmission of characters may take a long time relative to the CPU speed, the UART maintains a busy status flag, so the host system knows transmission is in progress.

Design of a UART – Requirements

Design a digital circuit implementing a UART in half-duplex mode. Consider the following parameters for the serial communication:

- 1 Start Bit (always LOW)
- 8 Data Bits (LSB sent first)
- 1 Stop Bit (always HIGH)
- 9600 baud/s or bps (tx/rx speed)



Design of a UART – Requirements (2)

The circuit must have the following ports:

- **t_data_in**: an 8-bit input signal containing the data to be transmitted (parallel input)
- **t_start**: an input signal that makes the transmission start when HIGH. (This takes one clock cycle, and if the device is in a busy state, it is ignored).
- **t_status**: an output signal indicating if the transmitter is busy (LOW) or idle (HIGH).
- **r_data_out**: an 8-bit output signal containing the received data (parallel output)
- **r_new_available**: an output signal indicating a newly received data is available. (is LOW by default, goes to HIGH in one clock cycle)
- **tx**: the output signal of the UART device employed for transmission
- **rx**: the input signal of the UART device employed for the reception
- **clk**: the input signal for the clock (Assume an input clock of 50MHz)
- **rst**: Asynchronous reset signal. It restarts the device, clears all data to transmit or receive, and suspends the transmission/reception.

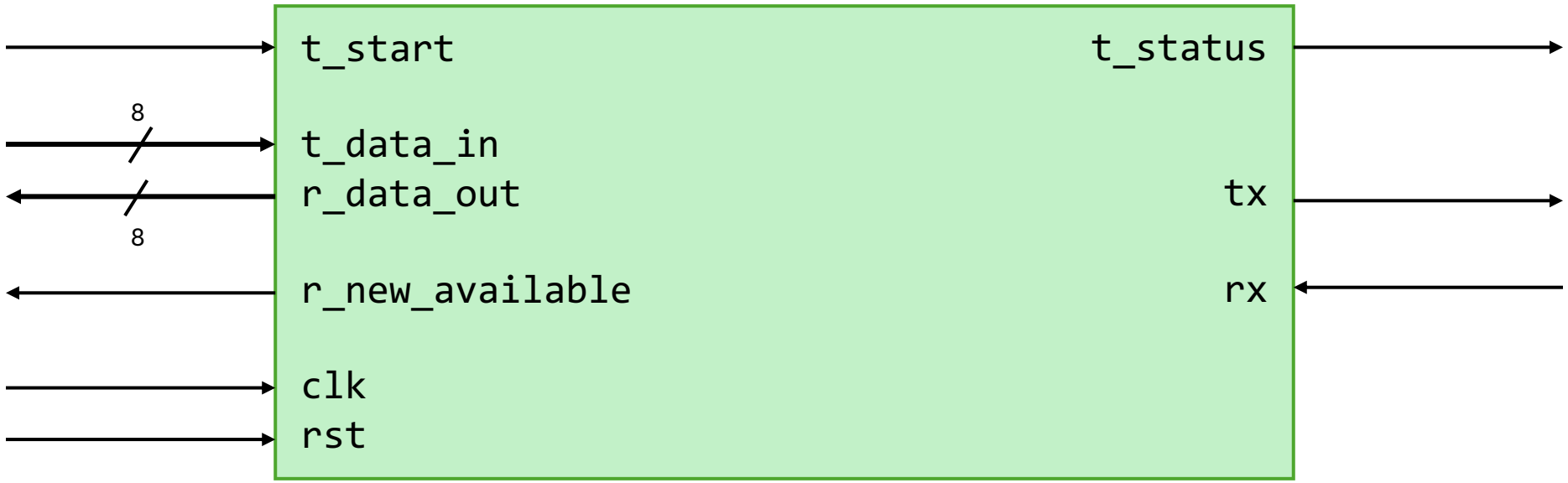
Design of a UART – Tips

Try to identify which essential hardware components (sequential or combinational) you need to use and how to control them (FSM).
Divide the problem into fundamental blocks:

1. Essential block diagram for your design.
2. Finite State Machine diagram.
3. Implement the transmitter in VHDL.
4. Design and run a testbed for your design.

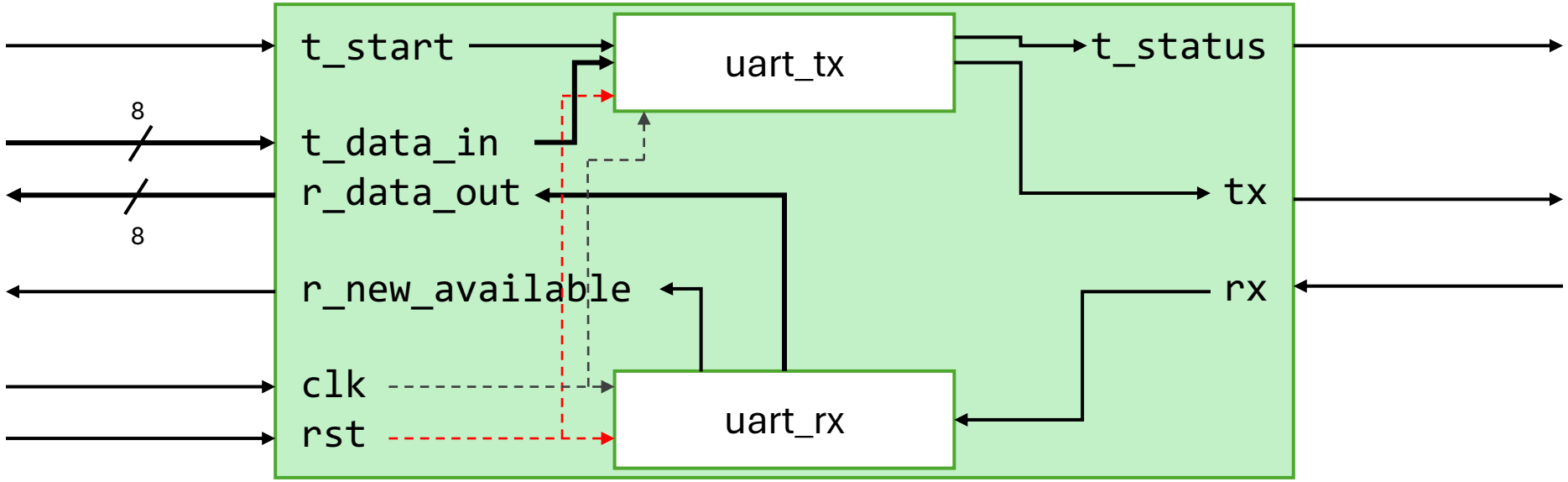
Design of a UART – Example

1. *Essential block diagram for your design.*



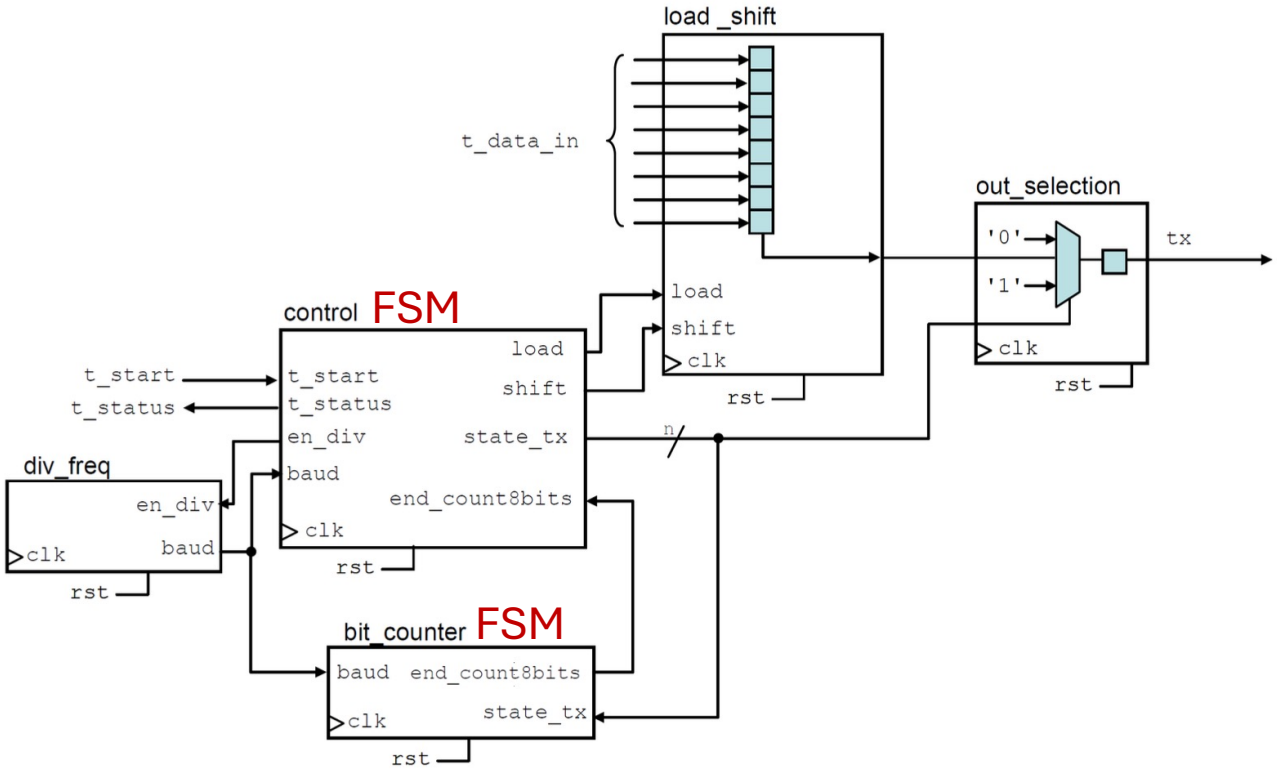
Design of a UART – Example

1. *Essential block diagram for your design.*



Design of a UART – Example

2. Finite State Machine diagram (TX).



Design of a UART – Exercise

This design exercise will span three lab sessions. For now, we will focus on the transmission (TX) part.

Exercise 1: Implement a shift register in VHDL and validate its functionality.

Exercise 2: Implement the frequency divider in VHDL and test its functionality.

Ex. 2b: If your design has other blocks, implement and test them in VHDL.

Exercise 3: Design the transmission control block and the bit counter FSMs, define the number of states and their truth tables.

Exercise 4: Implement the transmission control block in VHDL.

Exercise 5: Join all the blocks and design a testbed to validate the transmission functionality.

References

- Tokheim, R. Digital Electronics: Principles And Applications
- Floyd,Thomas L. Digital Fundamentals, 2016. Pearson
- Mano, Morris. Digital Desing.
- Intel QUARTUS Help Manuals
- Tocci, Ronald. Digital Systems Principles and Applications.