Lab 2 – Digital Systems Design

In contrast to a combinational circuit, the output of a sequential circuit can depend on the current as well as previously applied inputs. Consequently, to design sequential circuits, we need memory elements that retain the information about previous inputs. Sequential systems can be implemented using synchronous and asynchronous circuits. However, synchronous circuits are preferred due to the ease of their implementation as meeting timing requirements become easier compared to meeting them in asynchronous circuits. In this section, we will study synchronous sequential circuit design and their verification.

Basic Memory Elements and their SystemVerilog Description

All sequential circuits require some sort of memory element to store information about previous inputs. There are two basic memory elements:

1. Latch
2. Flip-Flop

Each of these elements can store 1-bit of data. We can combine multiple flip-flops to store a word (data of width more than 1) and the resulting storage element is called a register.

In this session, we will learn about the behavior of these basic memory elements and their SV descriptions.

Basic Building Blocks in Synchronous Circuits and their SystemVerilog Description

In this section of the training, we will learn different building blocks for synchronous circuits such as:

1. Counters
2. Shift Registers
3. Register Files

alongwith their SV description.

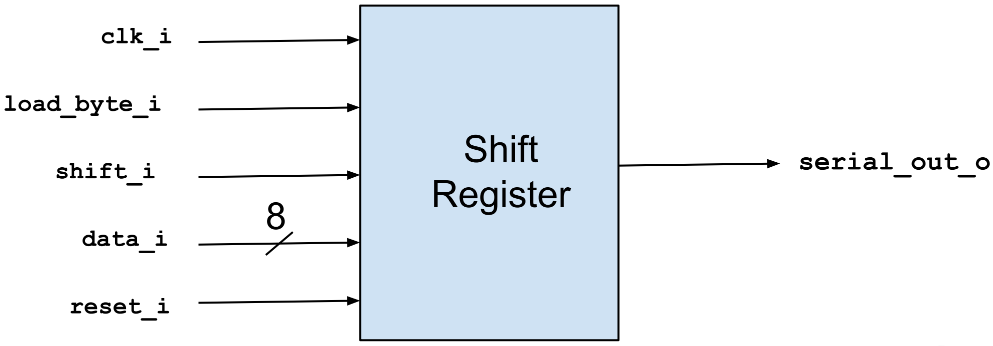
Basic Testbench Design for Synchronous Circuits

The testbench design for a synchronous circuit is different from a combinational circuit as all the outputs are changed wrt the synchronizing clock's edge. As a result, the test signals that we need to apply to synchronous circuits should also be synchronized to the clock. In this section, we will learn how to design stimulus for synchronous circuits.Design and SV Description of Finite-State Machines

We start from the design of synchronous circuits using finite-state machine (FSM) and will discuss its two types: Mealy and Moore. We will further discuss how the two state machines differ in their timing and finally learn their SV description.

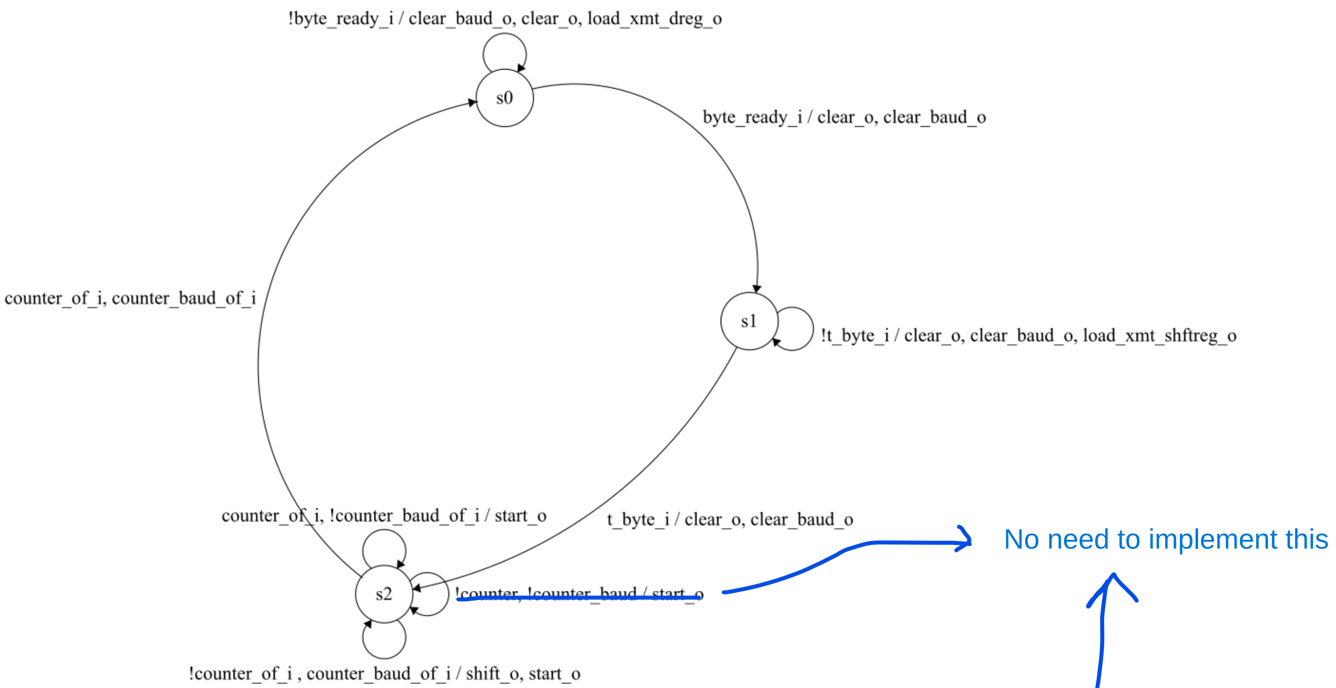
Tasks

1. A 8-bit parallel-input serial-output shift register takes in 8-bit parallel values as input and takes out one bit at a time starting from the LSB. A block diagram with input/output specifications is given below:

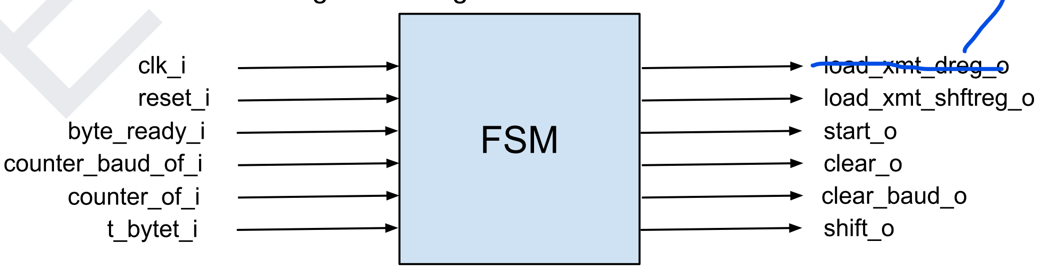


The input **data_i** is loaded into the shift register and when **shift_i** is asserted, one bit comes out of the **serial_out_o** pin with LSB first.

2. Write SystemVerilog description of a 8-bit parallel-input serial-output shift register.
3. Verify the 8-bit shift register by applying random inputs and observing the output.
4. Write a SystemVerilog description of the following state-transition graph (STG).

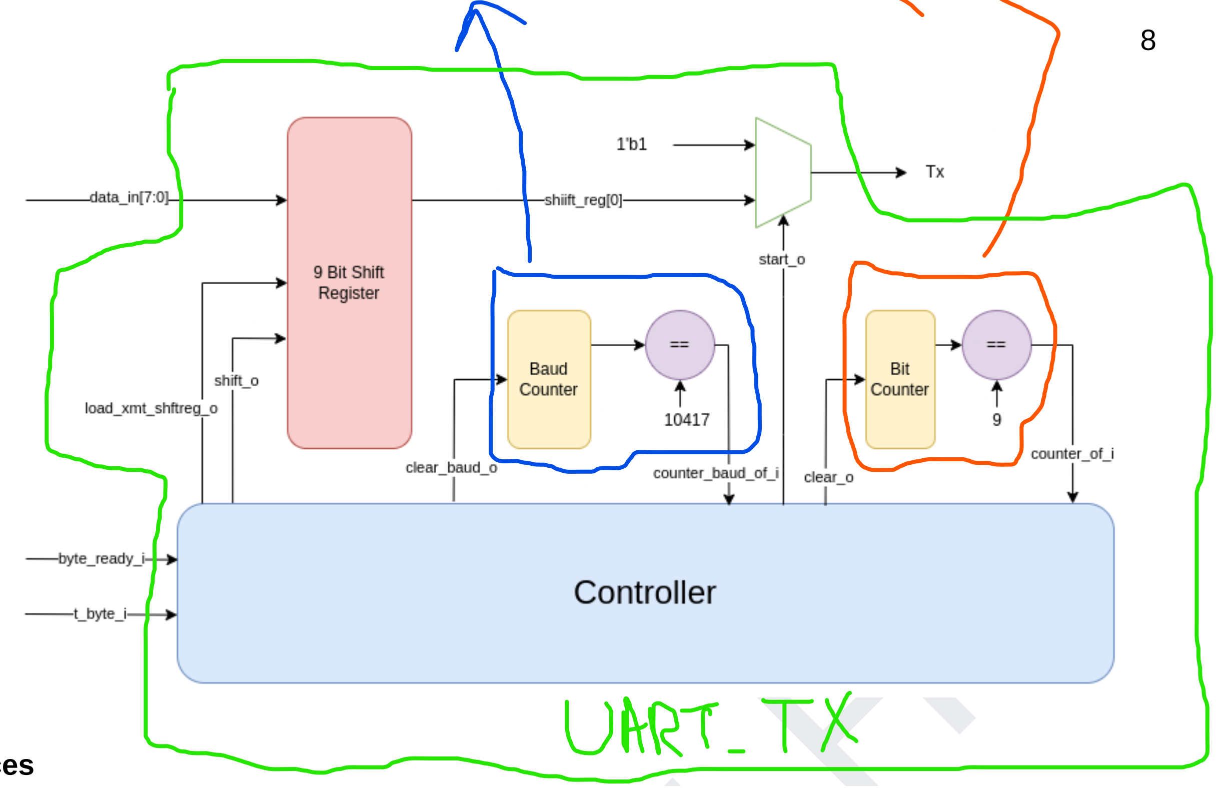


The FSM has the following block diagram.



5. Design the UART Tx module using the FSM created in the above task as its controller and the following datapath.

Merge them in a single module



References

[1] Harris, Sarah L., and David Harris. *Digital Design and Computer Architecture, RISC-V Edition*. Morgan Kaufmann, 2021.