ECE/CS 5710/6710

Verilog

Name:

Instructions:

- This homework is designed to review your Verilog coding skills. There are two simple problems. Please submit the Verilog behavioral code along with appropriate testbench. You don't need to provide simulation result, but it is probably a good idea to do sanity-check using a simulator. Modelsim is a simulator that you can use.

  This quiz is open book, and you may use and review previous course materials or other resources when taking this quiz. However, *this work must be done independently* without collaboration from other past or present students.

- **Cheating policy will be strictly enforced. Fill in the following form.**

DISCLAIMER: I,……(insert name) hereby confirm that the answers provided in this exam are solely my own work and were not generated with the help from others, nor were any non-permitted aids used. I am aware that the consequences of me cheating in this exam or in any part of my classwork as part of this course could lead to failing the class and potentially being excluded from the program as per university policy. Conversations during the exam are not permitted and are considered academic misconduct, unless explicitly allowed on a case by case basis by the instructor.

Date:…….Signature:……

Grading Rubric:

Question	Score
Problem 1 (30 pts)	
Problem 2 (70 pts)	
Total (100 pts)	

1. ***Binary to BCD Converter***

In digital systems, it is possible to represent a decimal number simply by encoding each bit in its binary form. This is called ***Binary-Coded-Decimal*** (BCD). For example, $(43)_{10}$->0100 0011. In this exercise, we will convert a binary number to its equivalent BCD representation. You will probably be using an algorithm called **'Double dabble'**. This is a link to the wiki page, where you can read about the algorithm. You will also find the example in C and VHDL in the same wiki page.

Draw a block diagram that describes the steps to convert a binary number to its BCD representation. Write a behavioral Verilog that uses the same steps. Also include a Verilog testbench for simulating the behavioral Verilog (you don't need to do any simulation) in your answer, where you convert all the integer number between 100 and 255.

2. ***Transmit block for UART***

Universal Asynchronous Receiver and Transmitter (UART) is very popular for serial data transmission where a host processor sends ASCII coded serial data to another processor. At the same time, it can receive the data. The transmitter encodes an alphabetical character in ASCII representation. The 8-bit word is added with a pair of start-stop bits, where start bit is added as ***LSB*** (least significant bit) and stop bit as ***MSB*** (most significant bit). Here is a link to the wiki description.

An UART transmitter's operation can be divided into several parts: (a) a data register where the original data is stored, (b) a shift register that will serially shift the output data (c) a status register that counts the number of transmitted bits, and (d) a state machine which will control the overall operation such as loading the shift register, clearing the status register when 10 bits are successfully transmitted after a 'transmit' operation is done. ***The output of the transmitter will be the LSB of the shift register.*** The very first bit that will go to the channel is start bit (as it is the LSB of 10-bit codeword), and the last bit to go is the stop bit (as it is the MSB of 10-bit codeword)

You are required to write a Verilog code for the overall operation. Also, provide a testbench for simulating the Verilog. ***Hint*****: State controller is the part which controls the overall operation. So, pay attention to the ASM chart description for state controller attached with the homework.** Remember that it is often a good practice to implement a system involving a state machine using multiple ***always*** blocks.

Here is a list of inputs and output that you might want to use:

Data_Bus	Input ASCII character
Byte_ready	Input from the host processor to signal the byte is ready
Load_XMT_datareg	Used by the host processor to load the data register
T_byte	Used by the host processor to signal the start of the transmission
Clk	Clock speed of the transmission
reset	resetting all the register when high
Serial_out	1-bit output to the channel

You can declare all other variables as '***reg***'. ***Hint: fig. 1 and 2 contains information about other possible variables.***

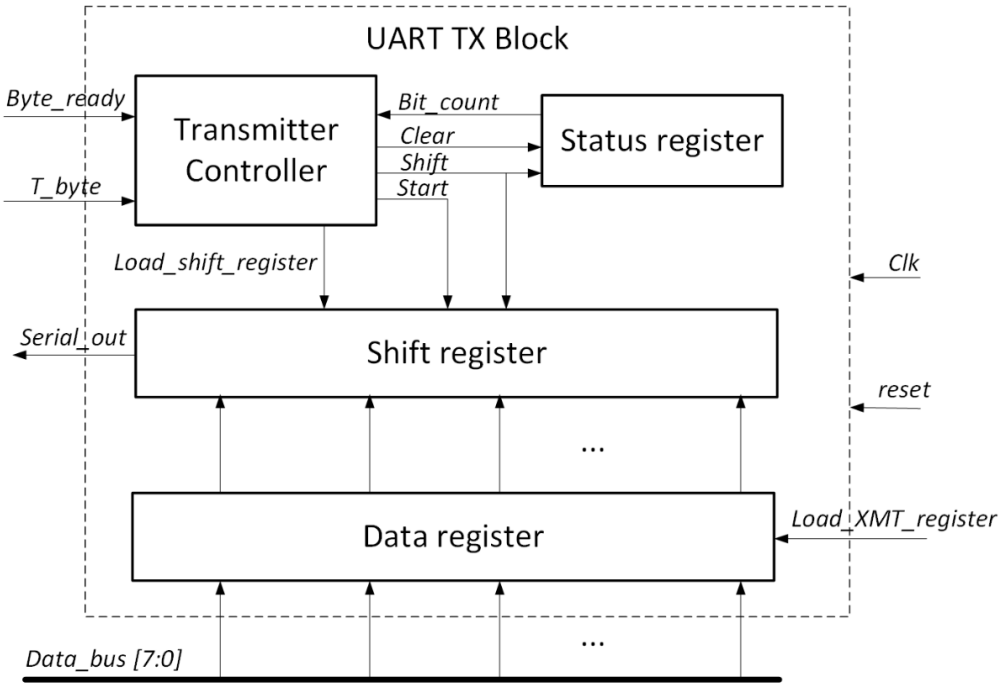

Fig. 1. Block diagram for an UART transmitter.

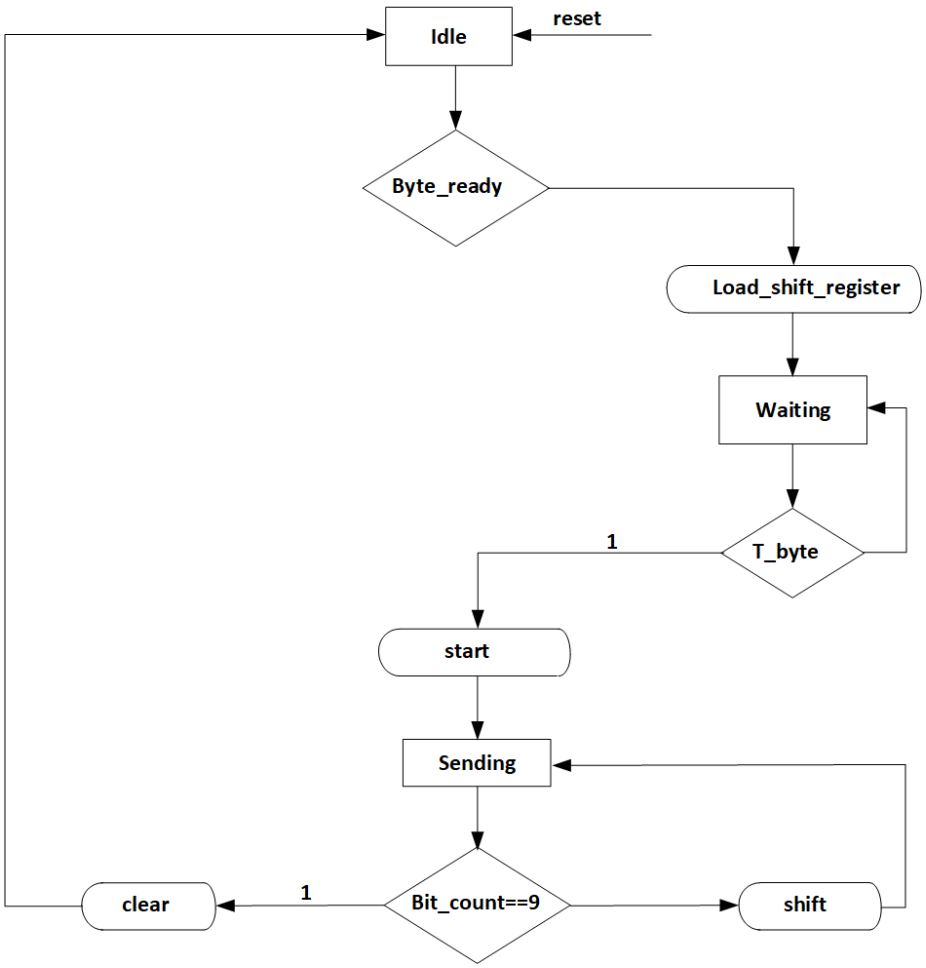

Fig. 2. ASM chart for the state controller.