CS150 Homework 6 Solutions

1) Design a Finite State Machine to receive bytes over an RS232 line running at 9600 baud with one start bit, 8 data bits, one stop bit, and no parity bit. Assume that you have an accurate 1 MHz clock. You may use shift registers (e.g. fig 6.56) and 163 counters (e.g. fig 7.17), as well as simple logic gates and any of the PLAs or PALs shown in the text. The input to your FSM is the rs232 signal converted to logic voltages (from +/- 3 to 24V). You may assume that it has been passed through a synchronizer (e.g. fig 6.37). The output of your FSM should be a "byte_ready" signal, along with the 8 bits of the message available in parallel.

(Answers may vary)
A) Draw the state diagram, state transition table, and the schematic diagram of your FSM. You don't need to show the details of K-maps and implementation.

RS232 Interface

<table>
<thead>
<tr>
<th>Start bit</th>
<th>Bit 0</th>
<th>Bit 1</th>
<th>Bit 2</th>
<th>Bit 3</th>
<th>Bit 4</th>
<th>Bit 5</th>
<th>Bit 6</th>
<th>Bit 7</th>
<th>Stop Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>[S1'BR']</td>
<td>[S1 BR']</td>
<td>[S1 BR']</td>
<td>[S1 BR']</td>
<td>[S1 BR']</td>
<td>[S1 BR']</td>
<td>[S1 BR']</td>
<td>[S1 BR']</td>
<td>STOP</td>
</tr>
<tr>
<td>[S1' BR']</td>
<td>(EN)</td>
<td>(EN)</td>
<td>(EN)</td>
<td>(EN)</td>
<td>(EN)</td>
<td>(EN)</td>
<td>(EN)</td>
<td>(EN)</td>
<td>[S1' BR']</td>
</tr>
</tbody>
</table>

Controller FSM

Design:
- A 1 MHz clock is fed into the counter, which outputs an enable and resets when counter = 104 (1E6 / 9600 = 104). This produces 9600 signal.
- The controller allows bits to be shifted into the shift register upon receiving an enable command and handles start and stop bits accordingly.
B) including the flipflops inside of any registers and counters you may have used, how many flip flops did your design require? (Answers may vary according to implementation)
- Controller uses 4 flip flops (10 states, so \( \log_2(10) \) bits are required to store the states)
- Counter uses 8 flip flops (8 bit counter, although it can be implemented in 7 bits)
- Shift Register uses 8 flip flops (8 bit shift register, 1 flip flop per bit)
- Total 20 flip flops.

C) how would you change your design to allow it to run at two different data rates? 10 different data rates?
- Currently, the counter resets when COUNT = 104 because \( 1000000 / 104 = 9600 \) bps. The divide amount can be changed according to the targeted baud rate.

D) how would you change your design to allow it to work with or without parity, and with different numbers of stop bits?
A third input to the controller can indicate the number of parity bits, and the number of start/stop bits, and extra states in the state machine can keep track of the number of parity and stop bits.

CLD 8.2) Given the state diagram in Figure Ex. 8.2, obtain an equivalent reduced-state diagram containing a minimum number of states. You may use row-matching or implication charts. Put your final answer in the form of a state diagram rather than a state table. Make it clear which states have been combined.

The following are equivalent:

\[
\begin{align*}
A &= B \\
E &= F = G
\end{align*}
\]
Given the state diagram in Figure 8.14, partition the state machine into two communicating finite state machines, one containing the states: \( S_0, S_1, S_2, \) and \( S_3 \), and the other containing: \( S_4, S_5, \) and \( S_6 \).