Finite state machine design

You’ve been tasked by an airline to build a device that smashes every 3rd suitcase that is checked.

To implement this circuit, design a finite-state machine with a 1-bit input, sensor, and a 1-bit output, div₃, which is 1 when the number of cycles that sensor has been 1 is a number divisible by 3.
Finite state machine (FSM) design

A universal serial bus (USB) transmits data across a cable one-bit at a time. Sometimes we can process the data as it is received from the cable. For example, rather than wait for all 32 bits to arrive and then process all of the data through a single large 32-bit ALU, we will process the data using a bit-serial design, processing each bit as it arrives. The design lets us use a 1-bit ALU to process 32-bits of data and in turn lets us run the circuit that processes incoming data at an extremely high clock speed.

Design a bit-serial adder finite state machine (FSM) that will control a 1-bit full adder. Your finite state machine will be a 1-bit data input $c_{out}$ from the Full Adder. The $Go$ signal will become 1 for less than one clock cycle and will asynchronously reset your FSM and begin performing addition on a new set of data. Your FSM will produce a 1-bit output ($c_{in}$) that will be sent to the full adder. The full adder will receive 2 data-inputs from two serial buses $a$ and $b$ and should send out a string of sum bits $s$ serially. The series of $a$ and $b$ inputs should be interpreted as coming in order from least-significant bit to most-significant bit. The following is an example input and output stream ($5 + 12 = 17$ followed by $3 + 23 = 26$).

```
time = 0 1 2 3 4 5 6 7 8 9 10 11
Go    1 0 0 0 0 1 0 0 0 0 0 1
a     1 0 1 0 0 1 1 0 0 0 0 0
b     0 0 1 1 0 1 1 1 0 1 0 0
s     1 0 0 0 1 0 1 0 1 1 0 0
```

Design the sequential logic circuit that will control the Full Adder.
Scaling Decoders

Decoders are an important circuit for implementing memories (like register files). Below is the truth table for a 1-to-2 decoder.

<table>
<thead>
<tr>
<th>enable</th>
<th>in</th>
<th>out[1:0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2'b00</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2'b00</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2'b01</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2'b10</td>
</tr>
</tbody>
</table>

Design a 2-to-4 decoder using only 1-to-2 decoders.

Continued on the following page.
Design a 3-to-8 decoder using only 1-to-2 and 2-to-4 decoders.
Finite state machine (FSM) design

In this problem, you will design the finite state machine for the alarm system for a car. The alarm system is controlled by three wireless buttons (U = Unlock, L = Lock, and P = Panic). Each button is asserted to be 1 when pressed, 0 when not pressed. The alarm system uses a three-bit code to encode the output behaviors of the FSM: Do nothing (ABC = 000), sound the alarm and lock the car doors (ABC = 110), lock all car doors (ABC = 010), unlock the driver’s door (ABC = 001), unlock all passengers’ doors (ABC = 111). Any button may be pressed at any time and multiple buttons may be pressed at the same time. For maximum security, the panic button overrides the lock and unlock buttons (i.e., the system ignores lock and unlock when panic is pressed) and the lock button overrides the unlock button. The motors that control the locks on the door take one clock cycle to switch from locked to unlocked or from unlocked to locked. The motors are controlled by the FSM outputs.

The alarm must meet the following specifications.

1. Continually running the door motors can damage them, so the FSM must tell the motors to unlock or lock the doors for only one clock cycle per button press.
2. The FSM starts in a state in which all passengers’ doors are already unlocked.
3. If the panic button is pressed, all car doors should be locked and the alarm should sound.
4. If the alarm is sounding, pressing any button should turn off the alarm.
5. Pressing the lock button should lock all of the doors.
6. Pressing the unlock button once should unlock only the driver’s door.
7. Pressing the unlock button twice in a row should unlock all passengers’ doors.