COE 405, Term 181

Design & Modeling of Digital Systems

Assignment# 4

Due date: Monday, Nov. 12, 2018

Car Anti-Theft System

You have bought a new car with a built in anti-theft system, but you have concern since this is a standard factory unit and many people know how to disable it. You need to design and build a system with some hidden security features only you will know about!
In this assignment, you will implement an anti-theft system to process sensor inputs and generate the appropriate actuator control signals.

Description of Anti-Theft System

It is required that the anti-theft system be highly automated. The system is armed automatically after you turn off the ignition and exit the car (i.e., the driver's door has opened and closed). The system arms itself T_ARM_DELAY after all the doors have been closed; that delay is restarted if a door is opened and reclosed before the system has been armed.

Once the system has been armed, opening the driver's door the system begins a countdown. If the ignition is not turned on within the countdown interval (T_DRIVER_DELAY), the siren sounds. The siren remains on as long as the door is open and for some additional interval (T_ALARM_ON) after the door closes, at which time the system resets to the armed but silent state. If the ignition is turned on within the countdown interval, the system is disarmed.

As a paragon of politeness, you open the passenger door first if you are transporting a guest. When the passenger door is opened first, a separate, presumably longer, delay (T_PASSENGER_DELAY) is used for the countdown interval, giving you extra time to walk around to the driver's door and insert the key in the ignition to disarm the system.

There is a status indicator LED on the dash. It blinks with a two-second period when the system is armed. It is constantly illuminated either when the system is in the countdown waiting for the ignition to turn on or if the siren is on. The LED is off when the system is disarmed.

So far all this is ordinary alarm functionality. But you're worried that a knowledgeable thief might disable the siren and then just drive off with the car. So you've added an additional *secret* deterrent -- control of power to the fuel pump. When the ignition is off power to fuel pump is cut off. Power is only restored when first the ignition is turned on and then the driver presses both a hidden switch and the brake pedal simultaneously. Power is then latched on until the ignition is again turned off.

The diagram below lists all the sensors (inputs) and actuators (outputs) connected to the system:

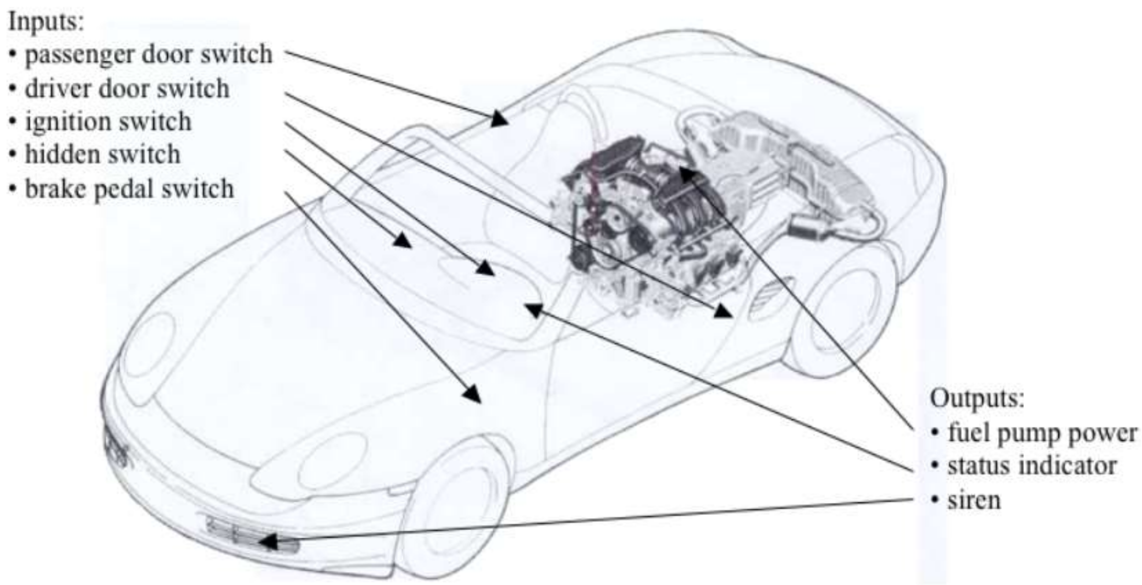

Figure 1: System diagram showing sensors (inputs) and actuators (outputs)

The system timings are based on four parameters (in seconds): the delay between exiting the car and the arming of the alarm (T_ARM_DELAY), the length of the countdown before the alarm sounds after opening the driver's door (T_DRIVER_DELAY) or passenger door (T_PASSENGER_DELAY), and the length of time the siren sounds (T_ALARM_ON). The default value for each parameter is listed in the table below, but each may be set to other values using the Time_Parameter_Selector, Time_Value, and Reprogram signals. Time_Parameter_Selector switches specify the number of the parameter to be changed. Time_Value switches are a 4-bit value representing the value to be programmed -- a value in seconds between 0 and 15. Pushing the Reprogram button tells the system to set the currently selected parameter to Time_Value. Note that your system should behave correctly even if one or more of the parameters is set to 0.

Interval Name	Symbol	Parameter Number	Default Time (sec)	Time Value
Arming delay	T_ARM_DELAY	00	6	0110
Countdown, driver's door	T_DRIVER_DELAY	01	8	1000
Countdown, passenger door	T_PASSENGER_DELAY	10	15	1111
Siren ON time	T_ALARM_ON	11	10	1010

Block Descriptions/Implementation

The following diagram illustrates a possible organization of your design into modules:

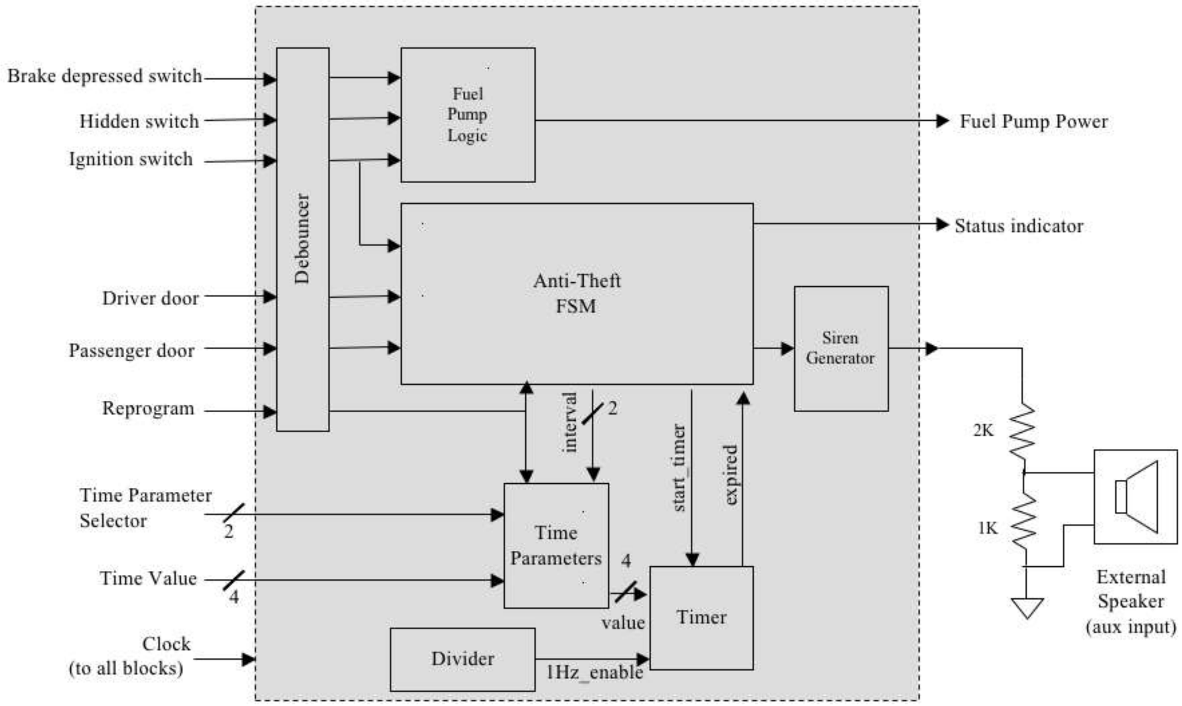

Figure 2: Block Diagram of Anti-Theft System.

A more detailed description of each module is given below:

Debouncer

Your clocked state machine is controlled by several asynchronous inputs that might be changed by the user at any time, potentially creating a problem with metastability in the state registers if one of the inputs changes too near a rising clock edge. In general, asynchronous inputs need to be synchronized to the internal clock before they can be used by the internal logic. You need to add necessary logic to care of that effect.

A second problem arises from the mechanical "bounce" inherent in switches: as a metal contact opens and closes it may bounce a couple of times, creating a sequence of on/off transitions in rapid succession. So you need to use debouncing circuitry to filter out these unwanted transitions. Make sure to add necessary logic to debounce any switch inputs you use in your design.

Time Parameters

The time parameters module stores the four different time parameter values. The module acts like a 4-location memory that's initialized with default values at power on, but may be reprogrammed by the user at any time. Using the 2-bit Interval signal, the Anti-theft FSM selects one of the four parameters to be used by the Timer module.

On power on, the parameters should be set to the default values specified above. However the user may modify any of the values by manipulating Time_Parameter_Selector (2 bits), Time_Value (4 bits), and Reprogram. Whenever a parameter is reprogrammed, the FSM should be reset.

Divider

The divider converts the master clock (100MHz) into an 1HZ_enable signal that's asserted for just 1 cycle out of every 100,000,000 cycles (i.e., once per second). The 1HZ-enable is used by the Timer module and for making the LED blink with a two-second period. The divider needs to reset when Start_Timer is asserted (see Timer module below) so that the first 1HZ_enable after the timer starts to count comes a full second after the timer has been started.

Timer

The timer counts down the number of seconds specified by the Time Parameter module. It initializes its internal counter to the specified Value when Start_Timer is asserted and decrements the counter when 1Hz_enable is asserted. When the internal counter reaches zero, the Expired signal is asserted and the countdown halts until Start_Timer is once again asserted.

Anti-theft FSM

This finite state machine controls the sequencing for the system. The system has four major modes of operation:

1. **Armed**. The status indicator should be blinking with a two-second period; the siren is off. If the ignition switch is turned on go to Disarmed mode, otherwise when a door opens start the appropriate countdown and go to Triggered mode. This is the state the FSM should have when the system is powered on.
2. **Triggered**. The status indicator light should be constantly on; the siren is off. If the ignition switch is turned on, go to Disarmed mode. If the countdown expires before the ignition is turned on, go to Sound Alarm Mode.
3. **Sound Alarm**. The status indicator light and siren should be constantly on. The alarm should continue to sound until either T_ALARM_ON seconds after all the doors have closed (at which point go to Armed mode) or the ignition switch is turned on (at which point go to Disarmed mode).

4. **Disarmed**. The status indicator light and siren should be off. Wait until the ignition switch is turned off, followed by the driver's door opening and closing, then after T_ARM_DELAY seconds go to Armed mode.

Note that more than one FSM state may be needed to implement the required functionality of each mode, i.e, your state transition diagram will have many more than 4 states.

Fuel Pump Logic

The fuel pump FSM controls the power to the fuel pump. Power is disabled when the ignition switch is turned off and only re-enabled when the appropriate sequence of sensor values is received (see description above).

Siren Generator

Use a LED to indicate that whether Siren is on or off.

(i) Design the Anti-Theft FSM and the Fuel Pump FSM (described using state diagram or ASM chart).

(ii) Write Verilog models for modeling the various components in your system.

(iii) Implement the Car Anti-Theft System on FPGA and demonstrate its correct functionality. Include a link for a video illustrating the correct functionality of your implementation.