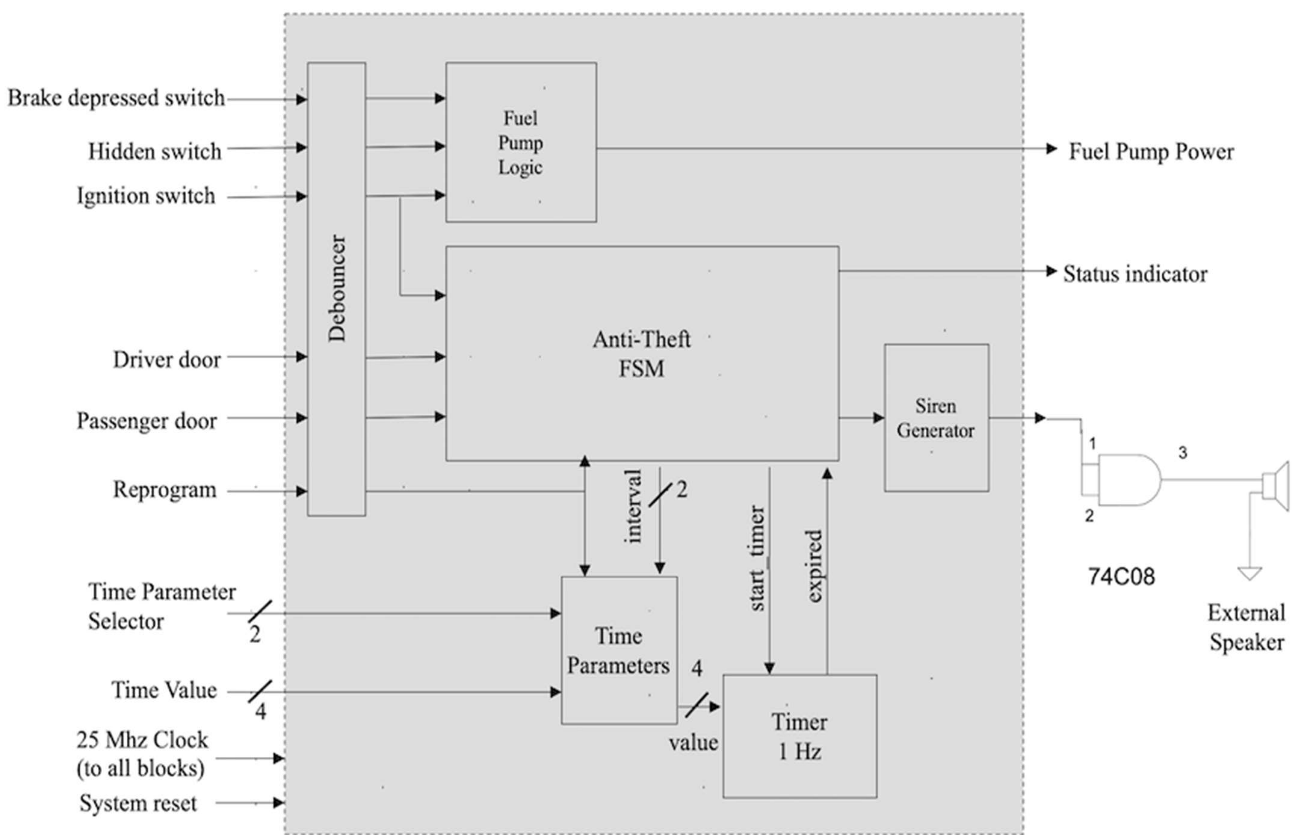
Car Alarm System

Introduction

In today's world, people rely so much on their vehicles so it's important to take steps to improve vehicle's security. A car alarm system tends to work well for this purpose.
The goal of this project is to design a digital Car Alarm System with sequential circuit based on a set of specifications. We have implemented an anti-theft system that uses several interacting FSMs to process sensor inputs and generate the appropriate actuator control signals.

Block Descriptions / Implementations



We have divided the system into five modules. Here is the detailed description of each module.

1. ***Debouncer***

Since our clocked state machine is controlled by several asynchronous inputs that might be changed by the user at any time. It may create problem with metastability in the state registers if any input changes too near a rising clock edge. Hence asynchronous inputs need to be synchronized to the internal clock with a debouncer before they can be used by the internal logic.

2. ***Time Parameters***

The time parameters module stores the four different time parameter values. It acts like a 4 location memory that's initialized with default values when system resets, but may be reprogrammed by the user at any time. The user can modify any of the values by manipulating Time Parameter Selector (2 bits) , Time Value (4 bits) and Reprogram.

3. ***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 one_hz 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.

4. ***Anti Theft FSM***

This FSM controls the sequencing for the system with four major modes of operation :

- ***Armed*** *: It is the state the FSM should have when the system is reset. In this state , the Status Indicator LED blinks with a two second period and the Siren is off. If the ignition switch is turned on , the system goes to Disarmed Mode otherwise when a door opens it starts the appropriate countdown and goes to Triggered Mode.*

- ***Triggered*** *: In this state, the Status Indicator LED remains constantly on and Siren off. If the ignition switch is turned on system goes to Disarmed Mode and if the countdown expires before the ignition is turned on it goes to Sound Alarm Mode.*

- ***Sound Alarm*** *: Here , the Status Indicator LED and Siren remain constantly on and alarm continues to sound until either T_ALARM_ON seconds after all doors have closed or the ignition switch is turned on.*

- ***Disarmed*** *: The status Indicator LED and Siren remain off and system waits here until the ignition switch is turned off, followed by the driver's door opening and closing, then after T_ARM_DELAY seconds system goes to Armed Mode.*

5. ***Fuel Pump Logic***

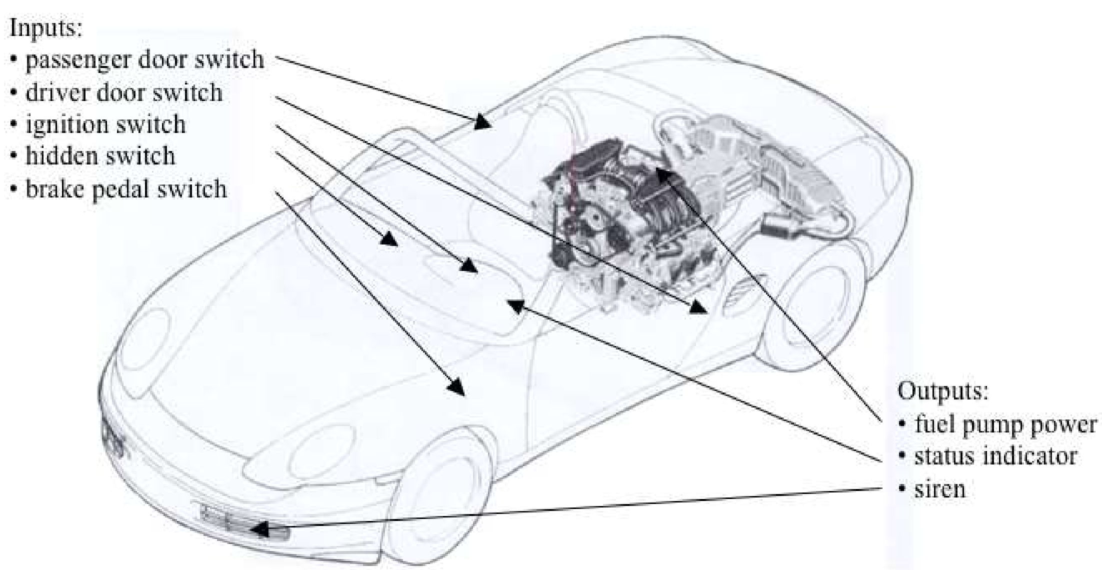
This simple FSM controls the power to the fuel pump. Power is disabled when the ignition switch is turned off and only reenabled when the appropriate sequence of sensor values is received.

Description of Anti-Theft System

The system is armed automatically after driver turns off the ignition, exits the car (i.e., the driver's door has opened and closed) and T_ARM_DELAY has passed. If there is a passenger and both the driver's door and passenger's doors are open, the system arms itself after all the doors have been closed and T_ARM_DELAY has passed; that delay is restarted if a door is opened and reclosed before the alarm 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. When the passenger door is opened first, a separate, presumably longer, delay (T_PASSENGER_DELAY) is used for the countdown interval, giving some 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 the system is in the countdown waiting for the ignition to turn on or if the siren is on. The LED is off is the system is disarmed.

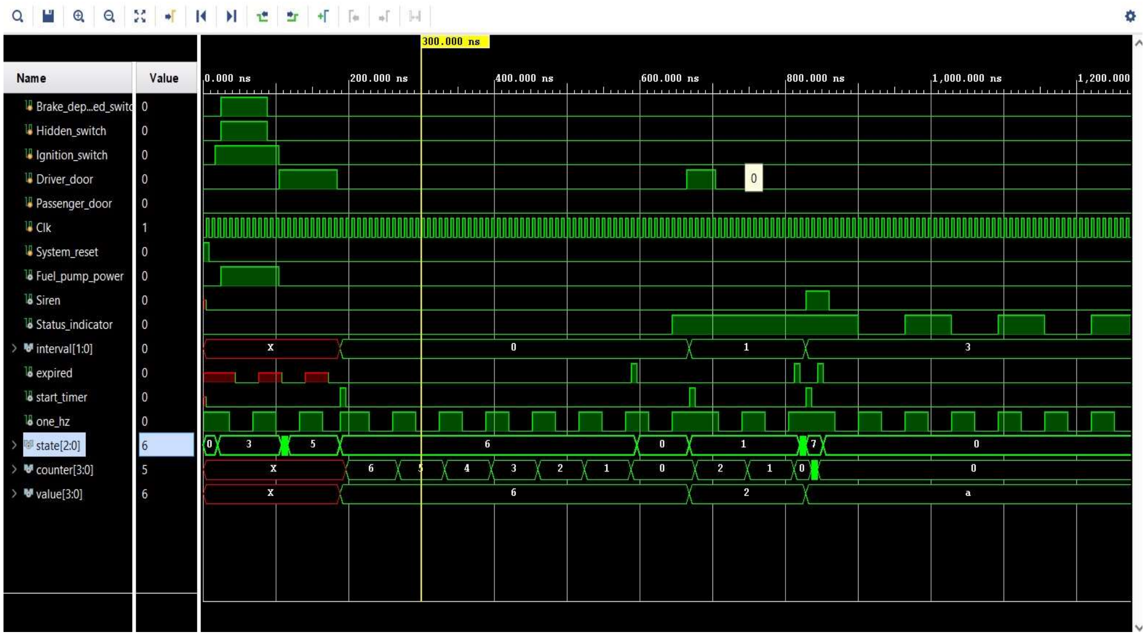
So far this all is ordinary alarm functionality. But a knowledgable thief might disable the siren and then just drive off with the car. So we've added an additional secret deterrent - control of power to the fuel pump. When the ignition is off power to fuel pump is cut off. Power is only restored when 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.



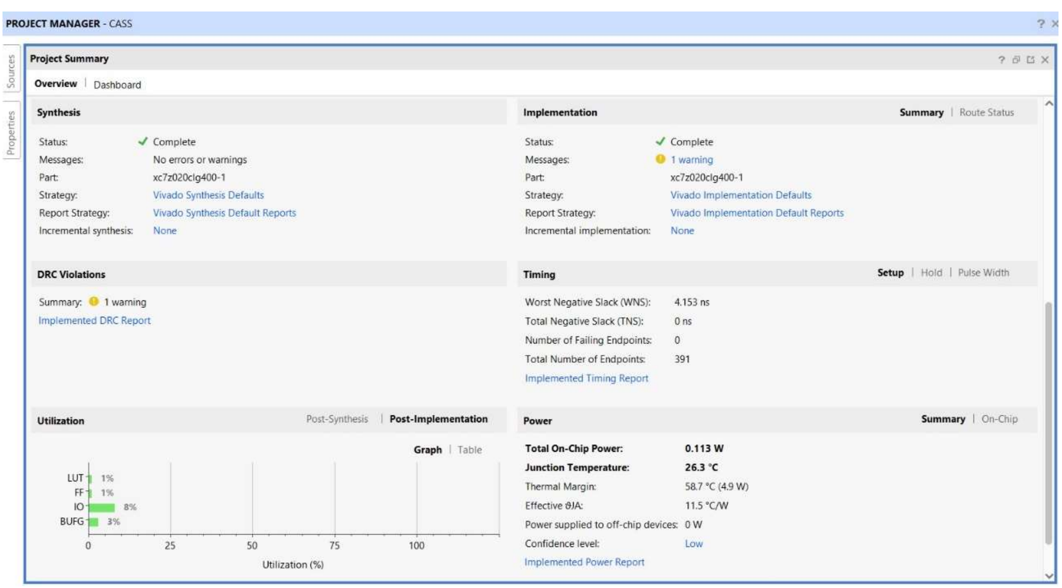
Results

Inputs :

t = 0 : system_reset = 1 , All doors are closed , Brake_depressed and hidden switch = 0;
#8 : system_reset = 0;
#8 : Ignition is on ; Brake_depressed and hidden switch = 1;
#16 : Ignition is off , Driver_door is opened
#80 : Driver_door is closed
#480 : Driver_door is opened
#40 : Driver_door is closed



After simulation we performed synthesis , implementation and finally bitstream generation. Here's the timing and power summary :



*At last, we loaded it on FPGA. We used **Zybo-Z7-20** here. The Zybo Z7 surrounds the Zynq with a set of multimedia and connectivity peripherals to create a formidable single-board computer. The Zybo platforms are video-capable and include MIPI CSI-2 compatible Pcam connector, HDMI input, HDMI output, and high DDR3L bandwidth.*

Features:

- Xilinx XC7Z010-1CLG400C / XC7Z020-1CLG400C
- *1GB DDR3L with 32-bit bus at 1066MHz*
- *16MB Quad-SPI flash*
- *Gigabit Ethernet PHY*
- *USB OTG PHY with host and device support*
- *Pcam camera connector with MIPI CSI-2 support*
- *Pmod connectors*

Specifications :

- 1 MSPS on-chip ADC
- 53,200 look-up tables
- 106,400 flip-flops
- 630KB RAM
- 4 clock management tiles
- 6 Pmod ports
- TX and RX ports HDMI CEC support
- 2 RGB LED

Switches, push-buttons & LEDs :

- *6 push-buttons (2 processor connected)*
- *4 slide switches*
- *5 LEDs (1 processor connected)*
- *2 RGB LEDs*

According to the block description of the system we need 5 push buttons and 8 slide switches but we have only 4 slide switches and 4 push-buttons on Zybo-Z7-20. Hence to check the functionality easily on FPGA we removed the reprogram feature of the system. Now we don't need to give time_parameter_selector and time_value as an input. Hence we need only 3 push-buttons (system_reset , brake_depressed_switch and hidden_switch) ; 3 slide switches (Driver-door , passenger-door and Ignition-switch) and 3 LEDs (Siren , Status_indicator and Fuel_pump_power).

Reference : http://web.mit.edu/6.111/volume2/www/f2019/handouts/labs/lab4_19/index.html