Digital System Design CSEN605

Smart Lecture Hall – Batch 2 – Project Report

The project “Smart Lecture Hall” revolves around counting the number of students inside a lecture hall at a certain point, and making sure the students entering the hall have their hands sanitized.

The idea we worked on revolved around using 3 IR sensors, each of which serves a critical function.
We had an enter sensor, which senses when a person enters the hall, incrementing the counter of the people inside the hall by one.
An exit sensor served the opposite function, decrementing the counter of the people inside the hall by one when someone exits the hall.
The final sensor, the sanitizing sensor, will turn on the motor which is responsible for dispensing hand sanitizer to the user when they put their hand in-front of it.

As previously mentioned, the concept revolved around the counter of the people inside the hall, which we displayed on the FPGA board using the 7 segment display BCD’s available.
The other output we are using is a motor that turns on to sanitize when a person places their hand by the sanitizer sensor.

We used a simplistic pin assignment that can be seen in the table below:

I/O	Direction	Pins
bcd0[1]	Output	PIN_B20
bcd0[2]	Output	PIN_A20
bcd0[3]	Output	PIN_B19
bcd0[4]	Output	PIN_A21
bcd0[5]	Output	PIN_B21
bcd0[6]	Output	PIN_C22
bcd0[7]	Output	PIN_B22
bcd1[1]	Output	PIN_F21
bcd1[2]	Output	PIN_E22
bcd1[3]	Output	PIN_E21
bcd1[4]	Output	PIN_C19
bcd1[5]	Output	PIN_C20
bcd1[6]	Output	PIN_D19
bcd1[7]	Output	PIN_E17
clock	Input	PIN_P11
enter_sensor	Input	PIN_W10
exit_sensor	Input	PIN_W5
motor	Output	PIN_AA15
sinitaize_sensor	Input	PIN_V10

The code is quite basic, using the built in clock in the FPGA, and the ticks that it generates due to its frequency. It checks that a second has passed, then proceeds to execute the internal main code. The built-in clock has a frequency of 50MHz. It generates ticks which we use to check that a second has passed by equating it to the frequency minus one. (In another words, every second, there are 50x10^6 ticks, where a tick represents a positive edge). Once a second does pass, we reset the ticks to 0, and execute the internal code.

This main code simply checks which sensors were triggered and which weren't. In the IF-conditions, we separate the cases of a person entering while no one else is exiting, and when a person exits while no one else is entering. Handling them this way ensures that in the case that someone enters while there's also exiting occurring, the counter does not change, since one entered and one exited. We also handled the cases where the least significant number would reach 9 while incrementing, it would go to 0 and the most significant number would increment by 1. The counter we use counts from 00 to 99 and resets if incremented at 99.

The code also checks for the sanitize sensor reading, meaning if a user places their hand by the sanitize sensor, it'll set the motor variable to 1, which is fed into the FPGA at a certain pin, which is connected to the motor VCC.

To constantly update the BCD value on the FPGA, we created a process on each counter (one for each digit from the two), ensuring that each time the counter changes, the BCD is updated.