PROJECT REPORT

COEN 313

Digital Systems Design II

Project: Room Counter

Adam Benhamou

ID: 40304141

Instructor: Dr. Sébastien Le Beux

Date: 2025-06-21

"I certify that this submission is my original work and meets

the Faculty's Expectations of Originality"

Adam Benhamou

Table of Contents

1. INTRODUCTION

This report documents the design, implementation, and testing of a room occupancy monitoring system. The project simulates the use of two photocell sensors to detect people entering and exiting a room and maintains a running count of current occupancy. An 8-bit input sets the maximum allowable number of occupants. When this threshold is reached, a max_capacity output is asserted to prevent further entries unless a person exits or if the system performs a reset. The design was implemented in VHDL, simulated in ModelSim, and synthesized targeting a Xilinx Nexys A7 FPGA board.

2. SYSTEM SPECIFICATIONS AND REQUIREMENTS

2.1 Functional Requirements

- Detect binary pulses on enter_sensor and exit_sensor
- Update current_occupancy accordingly
- Cap the occupancy at a programmable max_occupancy
- Assert max_capacity when room is full
- Reset system to 0 occupancy on reset
- 8-bit wire bus

2.2 Hardware Blocks

- Comparators (greater than, less than, equal to)
- Unsigned increment/decrement logic
- Multiplexers for sequential logic
- DFF for state retention

2.3 Inputs and Outputs Table

Signal	Direction	Description
Clk	IN	Clock signal
Reset	IN	Asynchronous reset
Enter_sensor	IN	Pulse input for entry detection
Exit_sensor	IN	Pulse input for exit detection
Max_occupancy [7:0]	IN	Maximum allowable occupancy
Current_occupancy [7:0]	OUT	Current occupancy count
Max_capacity	OUT	Detect when room is full

3. DESIGN METHODOLOGY

3.1 Top-Level Structure

The top-level entity, RoomCounter, instantiates a submodule “enter_sensor_logic” which determines if a person can enter. Internally, signals track whether a rising edge occurs on enter_sensor or exit_sensor, enabling detection of distinct pulses.

3.2 Internal Logic

- **Occupancy_next** calculates the next value based on whether an extry or exit occurred, this part of the design is known as next-state logic
- **Occupancy_count** stores the current state and is updated on every rising edge of the clock, this part of the design is known as current-state logic
- **Max_capacity** is asserted when **occupancy_count** is equal to **max_occupancy**

3.3 Specifics

Due to the fact that the photocell sensors have their own clock signals, possibly going I reject the possibility of someone leaving and entering the room at the exact same time due to the conflict it can create with the max_capacity signal.

4. TESTBENCH AND SIMULATION RESULTS

4.1 Test Cases

Initial conditions: enter and exit sensor = 0, clk = 0, reset = 0, max_occupancy = 3

Test#	Action	Expectation
1	1 person exits	Occupancy stays 0 (no decrement)
2	1 person enters	Occupancy = 1
3	2 more enter	Occupancy = 3, max_capacity = 1
4	1 more enters	Ignored, door is locked. Occupancy = 3
5	1 exits	Occupancy = 2, max_capacity = 0
6	1 enters	Occupancy = 3, max_capacity = 1

4.2 Simulation Output Confirmations:

- Correct edge detection
- Accurate occupancy tracking
- Assertion of max_capacity once limit is reached
- Prevention of over-counting entries beyond the limit

4.3 Edge Cases:

- The system does check if occupancy is greater than zero before decrementing. This senario is counter-intuitive due to the fact that no one can exit the room if no one is inside the room but because we are using photocell sensors, the light in the room controls the sensor just as much as the people walking in/out the doorway. (Only avoids going negative but isnt a permenant solution)
- I assume that upon installation of the system the max_occupancy is fixed, that is, the room cannot shrink or expand. In the event that we want to increase/decrease the max_occupancy, the system must be reconfigured with the new max_occupation as an initial condition (reset required).
- In the event that someone leaves at the exact same time as someone who enters and both the person that enters and exits spend the exact same amount of time (down to picoseconds) passing through the sensors, the system will only decrement due to the decrementing multiplexer having priority in this design. Because this event is nearly impossible to occur in the real world, I ignored this senario.

4.4 DO File Strategy:

The “dofile.do” file automates test cases using force and run commands. Clock is toggled between 1 and 0 every 2ns using force “-repeat 4ns clk 0 0ns, 1 2ns”. All waveforms and signals are monitored to verify behavior.

5. SYTHESIS RESULTS

5.1 Logic Blocks Used:

The design was synthesized in Vivado targeting the Xilinx Nexys A7 board. The RTL schematic and post-synthesis schematic confirm logic resource usage is minimal.

Block Name	Count
D Flip-Flop	10
Multiplexer	2
Comparator	3

5.2: Detailed Implementation

Overall, the architecture is well-suited for real-time systems. It uses straightforward pipelined logic and is fully synthesizable, achieving low latency and high clock frequency of 107MHz on the Xilinx Artix-7 FPGA.

6. CONCLUSION

This project successfully demonstrated a VHDL-based room occupancy counter system, with modular design, pulse stable logic, and efficient implementation. The logic prevents overflow and meets all design requirements. Future improvements may include debouncing with timers or FSM-based entry control for scalability.

7. REFERENCES

I. COEN313 Project Guidelines, Concordia University, Summer 2025
II. Form and Style: The Guide to Writing a Technical Report, Concordia University
III. ModelSim – INTEL FPGA STARTER EDITION
IV. Xilinx Vivado Design Suite Documentation