Project

Tracking Room Occupancy

Course: COEN 313

Section: U

Instructor:

Dr. Sébastien Le Beux

Submitted On

08/04/2025

Completed by:

Fahed Waheed

"I certify that this submission is my original work and meets the Faculty's Expectations of Originality"

Fahed Waheed

April 8, 2025

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

1. ABSTRACT

This project requires developing a digital room occupancy counter design and simulation project using VHDL. Based on edge-triggered sensors, the system counts how people move into and out of a designated area and keeps track of the count at a defined maximum limit. It also ensures that accurate, glitch-free counting is performed, that overflow or underflow conditions are avoided, and the full_flag is set when the room reaches maximum limit. Emphasis was put on preparing a testbench with 15 test cases, covering normal values along with edge cases for all the features in the design. Simulation results verified that the design was indeed correct and bound within the required parameters, thus making it appropriate for real-time occupancy monitoring in smart buildings, labs, and security systems.

2. INTRODUCTION

This report explains the design and implementation of a room occupancy counter using VHDL. VHDL stands for Very High-Speed Integrated Circuit Hardware Description Language and it is used for describing the hardware behaviour and structure. It allows to verify digital system correctness. The system in question, in this project, tracks the entry and the exit of people in a room. To keep track of the program correctness, two software are being used. First program that will be used is ModelSim. ModelSim is a very powerful tool for VHDL. It allows to simulate test and debug a VHDL code. This will be done with the addition of a testbench to test the system functionality. Using a testbench with ModelSim will allow to generate waveforms to then analyse the clock signals, the inputs and the outputs and how the work together. Analysing means to check for signal changes and design behaves as expected in real-time. For synthesis and design implementation purposes, the second software, Vivado will be used. This tool will allow to simulate the RTL elaborated schematic and the implemented schematic. Moreover, it will allow to generate a synthesized design.

In sum, this report will help get a better understanding of the design, VHDL implementation, testing and synthesis process of the room occupancy counter.

3. DESIGN

3.1 Inputs and Outputs

In order to understand the inputs of the VHDL design, it is necessary to first understand the objective of the design. The design requires tracking and monitoring the occupancy of a room, hence the inputs **entry_sensor** and **exit_sensor** are simulating two doors, one for the entry and the other for the exit. Every single time someone enter or exit the room, the output **occupancy_count** (room occupancy) should increment or decrement accordingly. In other words, anytime the **entry_sensor** is set to high, **occupancy_count** should be incremented. On the contrary, anytime the **exit_sensor** is set to high, **occupancy_count** should be decremented. Next, it is essential to make the circuit design synchronous. Therefore, the **clock** input is being used. In fact, in logic circuits, they are two types of logic: combinational and sequential. In this design, sequential logic circuit is being used. This will allow to make a circuit react in steps (per clock tick). Indeed, making the circuit combinational is not the best option because it will make the circuit react instantly. Without a clock, many errors may occur, such as counting an entry or an exit more than once. Moreover, it will prevent glitches and unstable behaviour. The maximum occupancy of the room can also be changed dynamically, adjustable through an 8-bit input signal **capacity**. Whenever the max capacity is reached the output **full_flag** should be high and low otherwise. Finaly, the user must also be able to reset the **occupancy_count** to 0, thus the output **reset** is also required. A summary of the **entity project** is given bellow for a better understanding.

Table 1: Summary of entity project

Port Name	Mode	Type	Description
clock	in	std_logic	System clock for synchronization
reset	in	std_logic	reset to clear internal states
entry_sensor	in	std_logic	Detects someone entering the room
exit_sensor	in	std_logic	Detects someone exiting the room
capacity	in	unsigned (7 downto 0)	Sets the maximum allowed occupancy

full_flag	out	std_logic	Becomes '1' when room is full
occupancy_count	out	unsigned (7 downto 0)	Current number of people inside the room

3.2 Conceptual Diagram

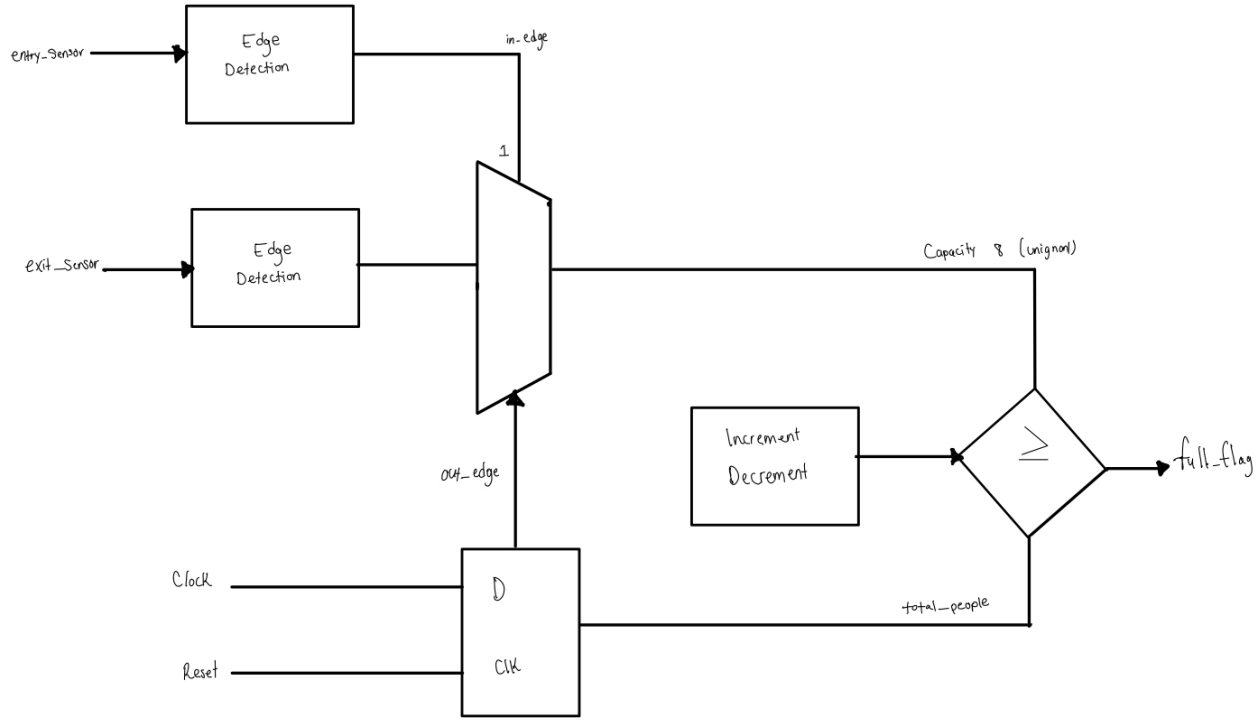

Figure 1: Conceptual Diagram

3.3 VHDL Implementation

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity project is
port (
clock : in std_logic;
reset : in std_logic;
entry_sensor : in std_logic;
exit_sensor : in std_logic;
capacity : in unsigned(7 downto 0); -- Max capacity input
full_flag : out std_logic;
occupancy_count : out unsigned(7 downto 0)
);
end project;

architecture arch_project of project is

signal total_people : unsigned(7 downto 0) := (others => '0');
signal full_internal: std_logic := '0';

signal entry_in, exit_out : std_logic := '0';
signal in_edge, out_edge : std_logic;

begin

process(clock)
begin
if rising_edge(clock) then
if reset = '1' then
total_people <= (others => '0');
full_internal <= '0';
entry_in <= '0';
exit_out <= '0';
else
-- Edge detection
in_edge <= entry_sensor and not entry_in;
out_edge <= exit_sensor and not exit_out;

-- Update sensor memory
entry_in <= entry_sensor;
exit_out <= exit_sensor;

-- Entry/Exit Logic
if in_edge = '1' and out_edge = '1' then
-- Ignore simultaneous
null;
elsif in_edge = '1' then
if total_people < capacity then
total_people <= total_people + 1;
end if;
elsif out_edge = '1' then
if total_people > 0 then
total_people <= total_people - 1;
end if;
end if;

-- Full flag logic
if total_people >= capacity then
full_internal <= '1';
else
full_internal <= '0';
end if;
end if;
end if;
end process;

-- Outputs
full_flag <= full_internal;
occupancy_count <= total_people;

end arch_project;
```

4. RESULTS

The following figures are the results of the VHDL Implementation using ModelSim and Vivado.

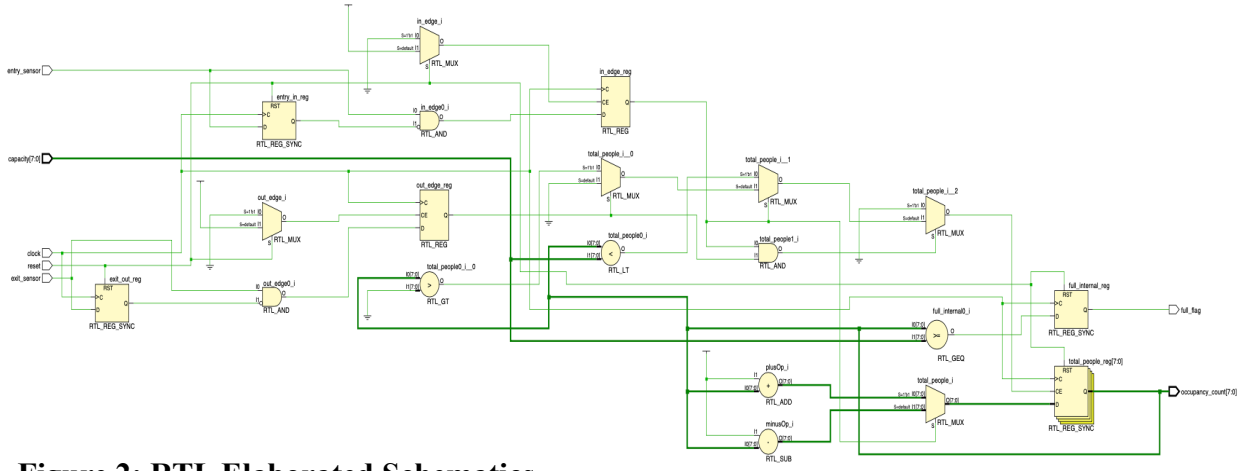

Figure 2: RTL Elaborated Schematics

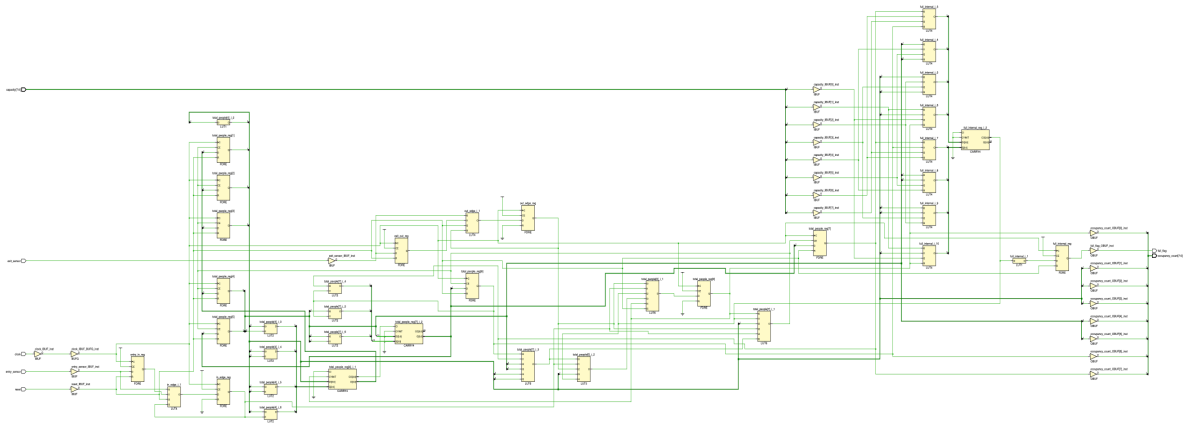

Figure 3: Implemented Schematics

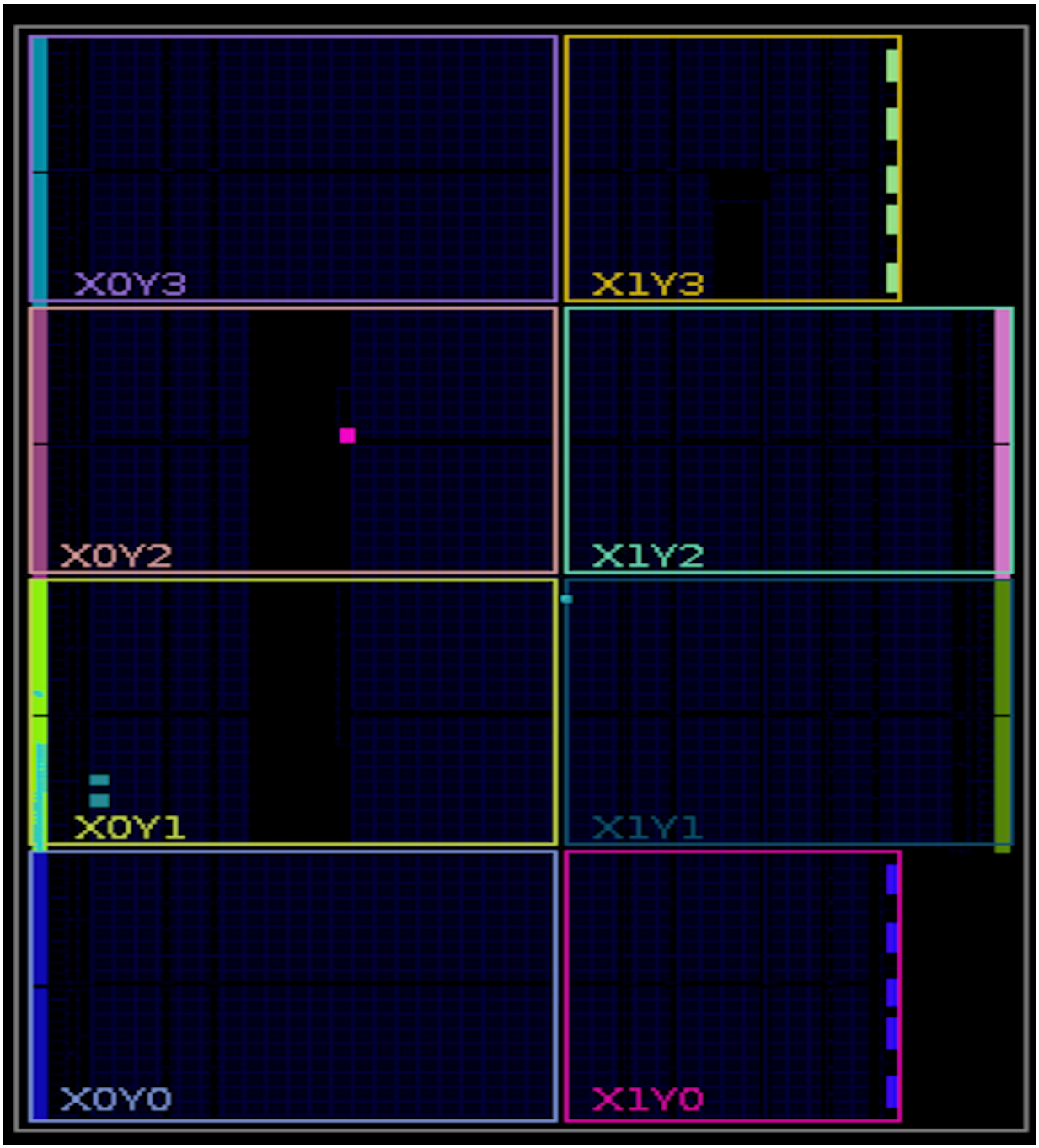

Figure 4: Synthesized Design (Implemented Design)

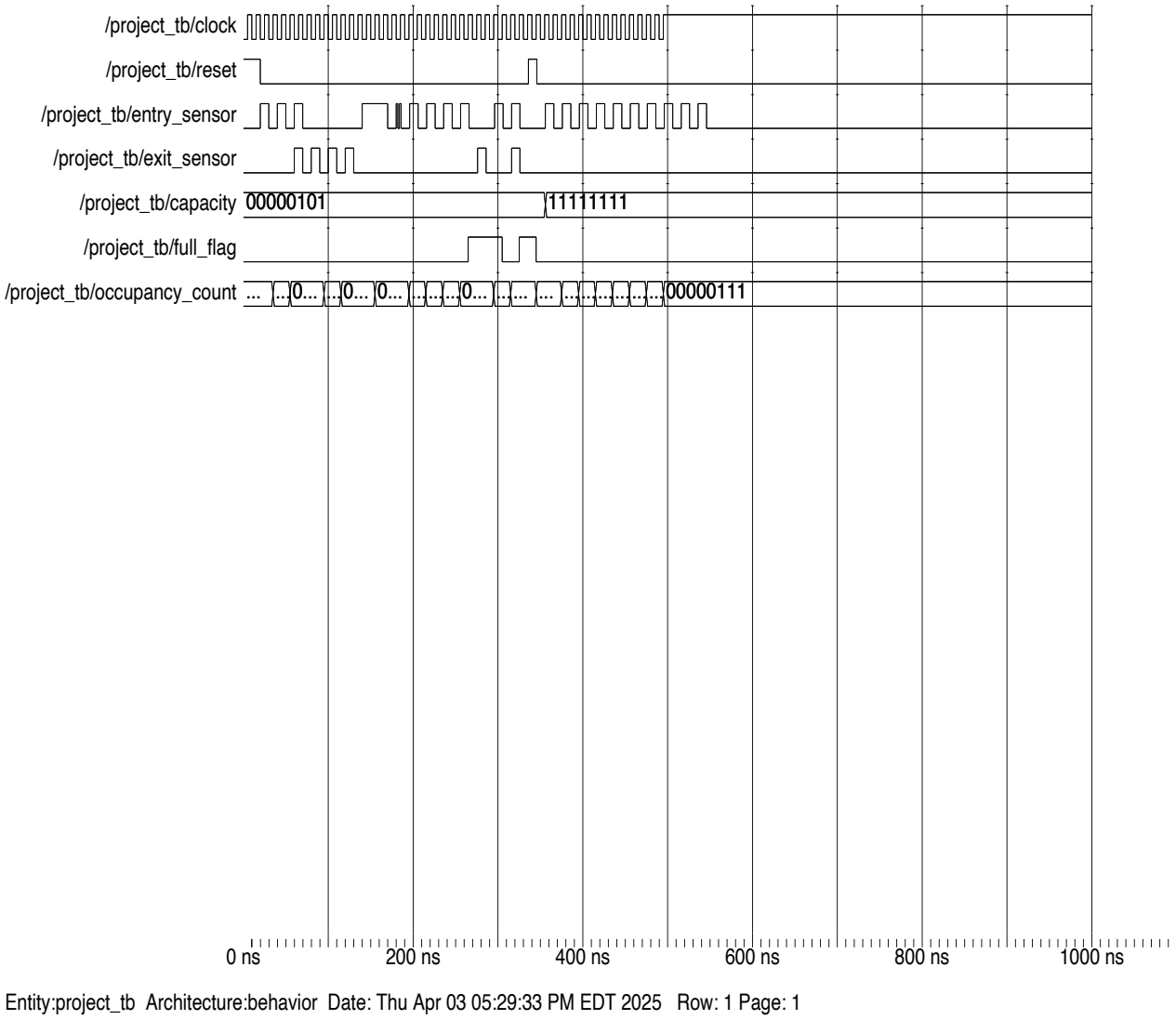

Figure 5: ModelSim Simulation Result

5. TESTING AND DISCUSION

5.1 Testbench

In Digital System Design, not only it is important to design and implemented circuits according to the requirements but also it is important to test many to all possibility of inputs and scenarios to ensure proper functionality of the design. For this project, the VHDL code has been implemented previously. It is now essential to test the functionality of the code. In any HDL language, it is possible to write down a testbench. A testbench will henceforth allow to test the behaviour of the design. It is a different VHDL file that does not describe the hardware, but it simulates the design according to the inputs. Please refer to the second submission .zip file for the testbench file (**project_tb.vhd)** and **Figure 4** for the ModelSim simulation result. The testbench tests 15 cases, each one conforming the proper functionality of the design. Two different **capacity** are used. From cases 1 through 14, the **capacity** is set to '00000101' or 5 in decimal. In the last case the **capacity** is set to '11111111' or 255 in decimal, which is the max capacity for a 8 bit-input signal.

To start simple, the first two cases are putting the inputs **entry_sensor** as high values. In this stage, the **occupancy_count** should be equal to '00000010' or 2 in decimal. Looking at **Figure 4**, the **occupancy_count** is '00000010'. Moreover, **full_flag** is low.

Case 3 is testing entry and exit simultaneously. The inputs **entry_sensor** and **exit_sensor** are set to high at the same time. The **occupancy_count** is still '00000010' and the **full_flag** is still low.

Cases 4 through 6 are testing exits. In fact, **exit_sensor** is set to high twice, making the **occupancy_count** set to '00000000'. Then, **exit_sensor** is set to high once more to test the **occupancy_count** behaviour after it has already outputted '00000000'. Indeed, **occupancy_count** remain at '00000000' and the **full_flag** is still low.

Case 7 is testing a long entry pulse. In fact, **entry_sensor** is set to high for 3 cycles. In this case, the entry should be counted only once. The **occupancy_count** outputted '00000001', which means it only incremented once. The **full_flag** is still low.

Case 8 is testing rapid toggling. Looking at **Figure 4**, it is possible to notice only one valid rising edge is detected, making the **occupancy_count** set to '00000010'. The **full_flag** is still low.

Cases 9 and 10 are testing the **capacity** input. In fact, **entry_sensor** is set to high for 3 entries, making **occupancy_count** set to '00000101' or 5 in decimal, which is the maximum capacity. Now, the **full_flag** is set to high. Then, **entry_sensor** is set to high for another entry, however the **occupancy_count** is still '00000101', since the **occupancy_count** is equal to **capacity**. The **full_flag** is still high.

Case 11 is testing exit after the **occupancy_count** has reached its maximum. The **exit_sensor** is set to high, so the **occupancy_count** goes to '00000100' or 4 in decimal and **full_flag** goes back to low.

Case 12 is testing the entry after the recent exit. The **entry_sensor** is set to high. This results the **occupancy_count** reach its maximum again. Now, the **full_flag** is set to high.

Case 14 is testing the **reset** input. In fact, **reset** is set to high, resulting the **occupancy_count** dropping to '00000000' and the **full_flag** to low.

Finaly, the **capacity** is set to '11111111' or 255 in decimal. The **entry_sensor** is set to high for 10 entries, making **occupancy_count** set to '00001010' or 10 in decimal. The **full_flag** is set to low.

5.2 Discussion

After confirming the validation of the design through 15 different cases, it is now possible to address the correctness and the efficiency of the design. The VHDL implemented design has a correct functional behaviour. All 15 cases passed with expected behaviour. To go more in depth, **occupancy_count** has been incremented and decremented based on its inputs. There has been no overflow detection and no underflow detection. The **full_flag** is high only when the **capacity** is reached. In other words, whenever the **occupancy** was equal to the **capacity**,

the **full_flag** was triggered. All cases confirm the correct implementation of the VHDL design, especially the cases 3, 6, 10 and 15. In fact, they test the most important cases, such as simultaneous entry and exit, exit at 0 **occupancy_count**, entry when full and the maximum 8 bit logic input. The VHDL design is also efficient. Indeed, only one synchronous process is used to control all logic. It supports the capacity from 0 to 255 in decimal with dynamically allocated inputs. It also considers long pulses and rapid toggling for both **entry_sensor** and **exit_sensor**. This prevents multiple counts from one pulse. Although, the design was simulated and validated without the use of a FPGA board, it is possible to confirm that the logic circuit and the VHDL implementation is accurate and well-organized.

6. Conclusion

In conclusion, the system's requirements were met because the room occupancy counter was fully functional and implemented using VHDL in a systematic manner. Every component including edge detection, reset, sensor state retention, overflow and underflow protection, and capacity control were simulated and verified. The testbench executed 15 unique specified test cases encompassing normal functionality, boundary conditions, and exceptional situations. Each case was accurately modeled and has been verified, ensuring the design is ready for practical use. With use of ModelSim and Vivado, two powerful software, it was possible to reenforce the design correctness.

7. Appendix

7.1 Vivado Log file

```
#-----------------------------------------------------------
# Vivado v2018.2 (64-bit)
# SW Build 2258646 on Thu Jun 14 20:02:38 MDT 2018
# IP Build 2256618 on Thu Jun 14 22:10:49 MDT 2018
# Start of session at: Thu Apr  3 18:01:16 2025
# Process ID: 3248
# Current directory: /nfs/home/f/f_wah/COEN313/Project
# Command line: vivado
# Log file: /nfs/home/f/f_wah/COEN313/Project/vivado.log
# Journal file: /nfs/home/f/f_wah/COEN313/Project/vivado.jou
#-----------------------------------------------------------
start_gui
create_project Coen313_Project /nfs/home/f/f_wah/COEN313/Project/Coen313_Project -part
xc7a100tcsg324-1
INFO: [IP_Flow 19-234] Refreshing IP repositories
INFO: [IP_Flow 19-1704] No user IP repositories specified
INFO: [IP_Flow 19-2313] Loaded Vivado IP repository
'/CMC/tools/xilinx/Vivado_2018.2/Vivado/2018.2/data/ip'.
create_project: Time (s): cpu = 00:00:07 ; elapsed = 00:00:07 . Memory (MB): peak =
6280.184 ; gain = 16.309 ; free physical = 12743 ; free virtual = 23932
add_files -norecurse /nfs/home/f/f_wah/COEN313/Project/project.vhd
import_files -force -norecurse
INFO: [filemgmt 20-348] Importing the appropriate files for fileset: 'sources_1'
update_compile_order -fileset sources_1
launch_runs synth_1 -jobs 2
[Thu Apr  3 18:07:24 2025] Launched synth_1...
Run output will be captured here:
/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.runs/synth_1/runm
e.log
launch_runs impl_1 -jobs 2
[Thu Apr  3 18:08:10 2025] Launched impl_1...
Run output will be captured here:
/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.runs/impl_1/runme
.log
launch_runs impl_1 -to_step write_bitstream -jobs 2
[Thu Apr  3 18:10:31 2025] Launched impl_1...
Run output will be captured here:
/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.runs/impl_1/runme
.log
open_run impl_1
INFO: [Netlist 29-17] Analyzing 15 Unisim elements for replacement
INFO: [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
INFO: [Project 1-479] Netlist was created with Vivado 2018.2
INFO: [Device 21-403] Loading part xc7a100tcsg324-1
```

INFO: [Project 1-570] Preparing netlist for logic optimization
Reading XDEF placement.
Reading placer database...
Reading XDEF routing.
Read XDEF File: Time (s): cpu = 00:00:00.01 ; elapsed = 00:00:00.01 . Memory (MB): peak = 6576.246 ; gain = 0.000 ; free physical = 12374 ; free virtual = 23642
Restored from archive | CPU: 0.010000 secs | Memory: 0.066750 MB |
Finished XDEF File Restore: Time (s): cpu = 00:00:00.01 ; elapsed = 00:00:00.01 . Memory (MB): peak = 6576.246 ; gain = 0.000 ; free physical = 12374 ; free virtual = 23642
INFO: [Project 1-111] Unisim Transformation Summary:
No Unisim elements were transformed.

open_run: Time (s): cpu = 00:00:14 ; elapsed = 00:00:17 . Memory (MB): peak = 6778.918 ; gain = 394.820 ; free physical = 12239 ; free virtual = 23509
WARNING: [Timing 38-313] There are no user specified timing constraints. Timing constraints are needed for proper timing analysis.
launch_simulation
INFO: [Vivado 12-5682] Launching behavioral simulation in '/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.sim/sim_1/behav/xsim'
INFO: [SIM-utils-51] Simulation object is 'sim_1'
INFO: [SIM-utils-54] Inspecting design source files for 'project' in fileset 'sim_1'...
INFO: [USF-XSim-97] Finding global include files...
INFO: [USF-XSim-98] Fetching design files from 'sim_1'...
INFO: [USF-XSim-2] XSim::Compile design
INFO: [USF-XSim-61] Executing 'COMPILE and ANALYZE' step in '/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.sim/sim_1/behav/xsim'
xvhdl --incr --relax -prj project_vhdl.prj
INFO: [VRFC 10-163] Analyzing VHDL file "/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.srcs/sources_1/imports/Project/project.vhd" into library xil_defaultlib
INFO: [VRFC 10-307] analyzing entity project
INFO: [USF-XSim-69] 'compile' step finished in '5' seconds
INFO: [USF-XSim-3] XSim::Elaborate design
INFO: [USF-XSim-61] Executing 'ELABORATE' step in '/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.sim/sim_1/behav/xsim'
Vivado Simulator 2018.2
Copyright 1986-1999, 2001-2018 Xilinx, Inc. All Rights Reserved.
Running: /CMC/tools/xilinx/Vivado_2018.2/Vivado/2018.2/bin/unwrapped/lnx64.o/xelab -wto fc66b8f100ed4209b5857907f506247d --incr --debug typical --relax --mt 8 -L xil_defaultlib -L secureip --snapshot project_behav xil_defaultlib.project -log elaborate.log
Using 8 slave threads.
Starting static elaboration
Completed static elaboration

```
Starting simulation data flow analysis
Completed simulation data flow analysis
Time Resolution for simulation is 1ps
Compiling package std.standard
Compiling package std.textio
Compiling package ieee.std_logic_1164
Compiling package ieee.numeric_std
Compiling architecture arch_project of entity xil_defaultlib.project
Built simulation snapshot project_behav

****** Webtalk v2018.2 (64-bit)
  **** SW Build 2258646 on Thu Jun 14 20:02:38 MDT 2018
  **** IP Build 2256618 on Thu Jun 14 22:10:49 MDT 2018
    ** Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.

source
/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.sim/sim_1/behav/x
sim/xsim.dir/project_behav/webtalk/xsim_webtalk.tcl -notrace
INFO: [Common 17-206] Exiting Webtalk at Thu Apr  3 18:16:41 2025...
run_program: Time (s): cpu = 00:00:02 ; elapsed = 00:00:09 . Memory (MB): peak =
7082.195 ; gain = 0.000 ; free physical = 11751 ; free virtual = 23131
INFO: [USF-XSim-69] 'elaborate' step finished in '9' seconds
INFO: [USF-XSim-4] XSim::Simulate design
INFO: [USF-XSim-61] Executing 'SIMULATE' step in
'/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.sim/sim_1/behav/x
sim'
INFO: [USF-XSim-98] *** Running xsim
   with args "project_behav -key {Behavioral:sim_1:Functional:project} -tclbatch
{project.tcl} -log {simulate.log}"
INFO: [USF-XSim-8] Loading simulator feature
Vivado Simulator 2018.2
Time resolution is 1 ps
source project.tcl
# set curr_wave [current_wave_config]
# if { [string length $curr_wave] == 0 } {
#   if { [llength [get_objects]] > 0} {
#     add_wave /
#     set_property needs_save false [current_wave_config]
#   } else {
#      send_msg_id Add_Wave-1 WARNING "No top level signals found. Simulator will start
without a wave window. If you want to open a wave window go to 'File->New Waveform
Configuration' or type 'create_wave_config' in the TCL console."
#   }
# }
# run 1000ns
INFO: [USF-XSim-96] XSim completed. Design snapshot 'project_behav' loaded.
```

INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:06 ; elapsed = 00:00:16 . Memory (MB): peak = 7148.637 ; gain = 66.441 ; free physical = 11703 ; free virtual = 23090
reset_run synth_1
WARNING: [Vivado 12-1017] Problems encountered:
1. Failed to delete one or more files in run directory /nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.runs/synth_1

close_sim
INFO: [Simtcl 6-16] Simulation closed
launch_simulation
INFO: [Vivado 12-5682] Launching behavioral simulation in '/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.sim/sim_1/behav/xsim'
INFO: [SIM-utils-51] Simulation object is 'sim_1'
INFO: [SIM-utils-54] Inspecting design source files for 'project' in fileset 'sim_1'...
INFO: [USF-XSim-97] Finding global include files...
INFO: [USF-XSim-98] Fetching design files from 'sim_1'...
INFO: [USF-XSim-2] XSim::Compile design
INFO: [USF-XSim-61] Executing 'COMPILE and ANALYZE' step in '/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.sim/sim_1/behav/xsim'
xvhdl --incr --relax -prj project_vhdl.prj
INFO: [USF-XSim-69] 'compile' step finished in '1' seconds
INFO: [USF-XSim-3] XSim::Elaborate design
INFO: [USF-XSim-61] Executing 'ELABORATE' step in '/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.sim/sim_1/behav/xsim'
Vivado Simulator 2018.2
Copyright 1986-1999, 2001-2018 Xilinx, Inc. All Rights Reserved.
Running: /CMC/tools/xilinx/Vivado_2018.2/Vivado/2018.2/bin/unwrapped/lnx64.o/xelab -wto fc66b8f100ed4209b5857907f506247d --incr --debug typical --relax --mt 8 -L xil_defaultlib -L secureip --snapshot project_behav xil_defaultlib.project -log elaborate.log
Using 8 slave threads.
Starting static elaboration
Completed static elaboration
INFO: [XSIM 43-4323] No Change in HDL. Linking previously generated obj files to create kernel
INFO: [USF-XSim-69] 'elaborate' step finished in '1' seconds
INFO: [USF-XSim-4] XSim::Simulate design
INFO: [USF-XSim-61] Executing 'SIMULATE' step in '/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.sim/sim_1/behav/xsim'
INFO: [USF-XSim-98] *** Running xsim
  with args "project_behav -key {Behavioral:sim_1:Functional:project} -tclbatch {project.tcl} -log {simulate.log}"

```
INFO: [USF-XSim-8] Loading simulator feature
Vivado Simulator 2018.2
Time resolution is 1 ps
source project.tcl
# set curr_wave [current_wave_config]
# if { [string length $curr_wave] == 0 } {
#   if { [llength [get_objects]] > 0} {
#     add_wave /
#     set_property needs_save false [current_wave_config]
#   } else {
#      send_msg_id Add_Wave-1 WARNING "No top level signals found. Simulator will start without a wave window. If you want to open a wave window go to 'File->New Waveform Configuration' or type 'create_wave_config' in the TCL console."
#   }
# }
# run 1000ns
INFO: [USF-XSim-96] XSim completed. Design snapshot 'project_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_runs impl_1 -to_step write_bitstream -jobs 2
[Thu Apr  3 18:18:27 2025] Launched synth_1...
Run output will be captured here: /nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.runs/synth_1/runme.log
[Thu Apr  3 18:18:27 2025] Launched impl_1...
Run output will be captured here: /nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.runs/impl_1/runme.log
synth_design -rtl -name rtl_1
Command: synth_design -rtl -name rtl_1
Starting synth_design
Using part: xc7a100tcsg324-1
Top: project
---------------------------------------------------------------------------------
Starting RTL Elaboration : Time (s): cpu = 00:00:01 ; elapsed = 00:00:02 . Memory (MB): peak = 7210.840 ; gain = 0.000 ; free physical = 11629 ; free virtual = 23016
---------------------------------------------------------------------------------
INFO: [Synth 8-638] synthesizing module 'project' [/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.srcs/sources_1/imports/Project/project.vhd:17]
INFO: [Synth 8-256] done synthesizing module 'project' (1#1) [/nfs/home/f/f_wah/COEN313/Project/Coen313_Project/Coen313_Project.srcs/sources_1/imports/Project/project.vhd:17]
---------------------------------------------------------------------------------
Finished RTL Elaboration : Time (s): cpu = 00:00:01 ; elapsed = 00:00:02 . Memory (MB): peak = 7223.848 ; gain = 13.008 ; free physical = 11644 ; free virtual = 23032
---------------------------------------------------------------------------------
```

Report Check Netlist:
+------+-----------------+-------+---------+-------+-----------------+
| |Item |Errors |Warnings |Status |Description |
+------+-----------------+-------+---------+-------+-----------------+
|1 |multi_driven_nets | 0| 0|Passed |Multi driven nets |
+------+-----------------+-------+---------+-------+-----------------+

Start Handling Custom Attributes

Finished Handling Custom Attributes : Time (s): cpu = 00:00:01 ; elapsed = 00:00:02 . Memory (MB): peak = 7223.848 ; gain = 13.008 ; free physical = 11644 ; free virtual = 23032

Finished RTL Optimization Phase 1 : Time (s): cpu = 00:00:01 ; elapsed = 00:00:02 . Memory (MB): peak = 7223.848 ; gain = 13.008 ; free physical = 11644 ; free virtual = 23032

INFO: [Project 1-570] Preparing netlist for logic optimization

Processing XDC Constraints
Initializing timing engine
Completed Processing XDC Constraints

INFO: [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
INFO: [Project 1-111] Unisim Transformation Summary:
No Unisim elements were transformed.

RTL Elaboration Complete: : Time (s): cpu = 00:00:05 ; elapsed = 00:00:04 . Memory (MB): peak = 7417.445 ; gain = 206.605 ; free physical = 11536 ; free virtual = 22926
5 Infos, 0 Warnings, 0 Critical Warnings and 0 Errors encountered.
synth_design completed successfully
write_schematic -format pdf -orientation portrait
/nfs/home/f/f_wah/COEN313/Project/schematic.pdf
/nfs/home/f/f_wah/COEN313/Project/schematic.pdf
current_design impl_1
current_design rtl_1
current_design impl_1
write_schematic -format pdf -orientation portrait -force
/nfs/home/f/f_wah/COEN313/Project/schematic.pdf
/nfs/home/f/f_wah/COEN313/Project/schematic.pdf
write_schematic -format pdf -orientation portrait
/nfs/home/f/f_wah/COEN313/Project/RTL_schematic.pdf
/nfs/home/f/f_wah/COEN313/Project/RTL_schematic.pdf

```
write_schematic -format pdf -orientation portrait
/nfs/home/f/f_wah/COEN313/Project/RTL_schematic.pdf
/nfs/home/f/f_wah/COEN313/Project/RTL_schematic.pdf
current_design rtl_1
write_schematic -format pdf -orientation portrait
/nfs/home/f/f_wah/COEN313/Project/RTL_schematic.pdf
/nfs/home/f/f_wah/COEN313/Project/RTL_schematic.pdf
current_design impl_1
write_schematic -format pdf -orientation portrait
/nfs/home/f/f_wah/COEN313/Project/schematic.pdf
/nfs/home/f/f_wah/COEN313/Project/schematic.pdf
close_sim
INFO: [Simtcl 6-16] Simulation closed
exit
INFO: [Common 17-206] Exiting Vivado at Thu Apr  3 18:55:50 2025...
```