# Abstract

This project entails the design of a sophisticated digital system for precise room occupancy tracking and monitoring using VHDL code. The system incorporates highly responsive photocells at the primary entrance and exit doors, generating binary signals (X and Y) upon light obstruction. A configurable maximum occupancy threshold, denoted as max\_occupancy, is established to a maximum limit of 63, stored in a register. Upon reaching this limit, a red indicator light is activated (signal Z), indicating full occupancy.

The conceptual diagram of the digital system is presented, highlighting key components such as multiplexers, flip-flops, and incrementors, along with their interconnections. VHDL is employed for circuit modeling, and a comprehensive test bench is provided to validate the design under various scenarios. Simulation and synthesis results for a Xilinx Nexys A7 FPGA development board are included in the report, accompanied by the Vivado log file.

Through rigorous analysis of the results, the quality of the design is assessed in terms of speed and FPGA resource utilization. This project combines theoretical understanding with practical implementation, showcasing an innovative approach to digital systems for room occupancy tracking.

**Room occupancy design using**

**VHDL**

**By**

**Usman Shehu Usman**

**ID: 40174744**

**A project report for COEN 313**

**Concordia University**

**24th November 2023**

Table of Contents

[Abstract 1](#_Toc152011090)

[Preface 4](#_Toc152011091)

[Introduction 5](#_Toc152011092)

[Conceptual Diagram 5](#_Toc152011093)

[Design Blocks 6](#_Toc152011094)

[Implementation 7](#_Toc152011095)

[VHDL code 7](#_Toc152011096)

[Test bench 9](#_Toc152011097)

[XDC file 12](#_Toc152011098)

[Simulation and synthesis results 12](#_Toc152011099)

[Results 13](#_Toc152011100)

# Preface

Copyright 2022 Usman Usman.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ”Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sub-license, and/or sell copies of the software, and to permit persons to whom the Software is furnished to do so, is subjected to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# Introduction

In response to the demand for sophisticated room occupancy monitoring, this project presents a cutting-edge digital system. Leveraging responsive photocells and dynamic threshold adjustment, the system accurately tracks occupancy levels. Configurable to a maximum of 63 occupants, it triggers a red indicator light when the limit is reached. The ensuing sections detail the system's architecture, VHDL modeling, comprehensive testing, and evaluation on a Xilinx Nexys A7 FPGA board, emphasizing its contribution to intelligent environmental monitoring systems.

# Conceptual Diagram

To design the VHDL code a conceptual diagram containing combinational and sequential circuit, see below a hand drawn conceptual circuit.

A shadow of a person holding a diagram

Description automatically generated

Figure 1: Conceptual diagram of the circuit

## Design Blocks

From the diagram in figure 1, I listed the design blocks and their description below.

* Multiplexers: I used four multiplexers in the conceptual design, the first and second were for the entry door sensor and exit door sensor respectively and the last two were for the increment and decrement signals.
* Incrementor: To increase the occupancy count when a person enters.
* Decrementor: To decrease the occupancy count when a person exits.
* Flip flops: My conceptual design had three flip flops to store the states of the X, Y and Z signals.
* Comparator: To compare the maximum occupancy to the occupancy count.
* Register: To store the value of the occupancy count.

# Implementation

## VHDL code

In this section I show the VHDL code written to implement the conceptual circuit, see below.

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity room\_occupancy is

Port ( clk : in STD\_LOGIC;

reset : in STD\_LOGIC;

entrance\_sensor : in STD\_LOGIC;

exit\_sensor : in STD\_LOGIC;

max\_occupancy : in UNSIGNED(5 downto 0);

red\_indicator : out STD\_LOGIC);

end room\_occupancy;

architecture code\_arch of room\_occupancy is

signal X, Y, Z,max\_count,min\_count : STD\_LOGIC := '0';

signal occupancy\_count : INTEGER range 0 to 63 := 0;

signal max\_occupancy\_integer : INTEGER range 0 to 63 := 0;

begin

process(clk, reset)

begin

max\_occupancy\_integer <= to\_integer(max\_occupancy);-- input load

if reset = '1' then

-- Reset the system

X <= '0';

Y <= '0';

Z <= '0';

clk <= '0';

occupancy\_count <= 0;

max\_occupancy\_integer <= 0;

elsif (clk'event and clk = '1') then

if entrance\_sensor = '1' and occupancy\_count < max\_occupancy\_integer then --Sensor for the entrance

X <= '1';

occupancy\_count <= occupancy\_count + 1;

occupancy\_count >= max\_occupancy then

else

Z <= '1';

X <= '0';

end if;

if exit\_sensor = '1' then --Sensor for the exit

Y <= '1';

-- occupancy\_count <= occupancy\_count - 1;

if occupancy\_count = 0 then

min\_count <= '1';

else

min\_count <='0';

occupancy\_count <= occupancy\_count - 1;

end if;

else

Y <= '0';

end if;

end if;

end process;

red\_indicator <= Z; --red led light for max occupancy

end code\_arch;

## Test bench

I wrote a test bench code to test to cover the different scenarios to verify my design see below.

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity tb\_RoomOccupancy is

end tb\_RoomOccupancy;

architecture Behavioral of tb\_RoomOccupancy is

component code

Port ( clk : in STD\_LOGIC;

reset : in STD\_LOGIC;

entrance\_sensor : in STD\_LOGIC;

exit\_sensor : in STD\_LOGIC;

max\_occupancy : in UNSIGNED(5 downto 0);

red\_indicator : out STD\_LOGIC);

end component;

signal entry, ex, red, reset: STD\_LOGIC;

signal clk : std\_logic := '0';

signal max\_occupancy : UNSIGNED(5 downto 0);

for code1:code use entity work.room\_occupancy(code\_arch);

begin

code1: code port map (entrance\_sensor => entry, exit\_sensor => ex, max\_occupancy => max\_occupancy,            reset => reset, red\_indicator => red, clk => clk);

clk <= not clk after 1 ns;

stimulus: process

begin

-- Test case 1: Reset the system

reset <= '1'; wait for 10 ns;

reset <= '0'; wait for 10 ns;

-- Test case 2: Person enters the room

Entry <= '1'; wait for 10 ns;

Entry <= '0'; wait for 10 ns;

-- Test case 3: Person exits the room

ex <= '1'; wait for 10 ns;

ex <= '0'; wait for 10 ns;

-- Test case 4: Room reaches maximum occupancy

max\_occupancy <= "000101"; -- Set max occupancy to 5

entry <= '1'; wait for 10 ns; -- Person enters

entry <= '0'; wait for 10 ns;

wait;

end process stimulus;

end Behavioral;

## XDC file

To synthesise and simulate the code an XDC file is required to initialize the ports of the FPGA development board, see below.

# Simulation and synthesis results

Using Modelsim I simulated the VHDL code shown below.

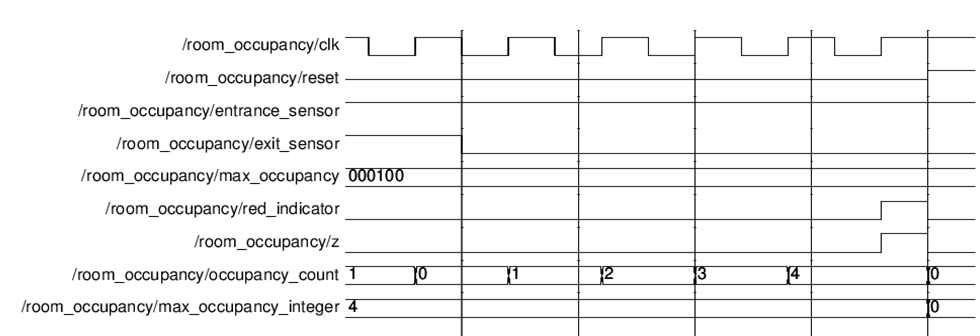


Fig 2: Modelsim Simulation

Using Vivado simulation software I synthesized the code, see below.

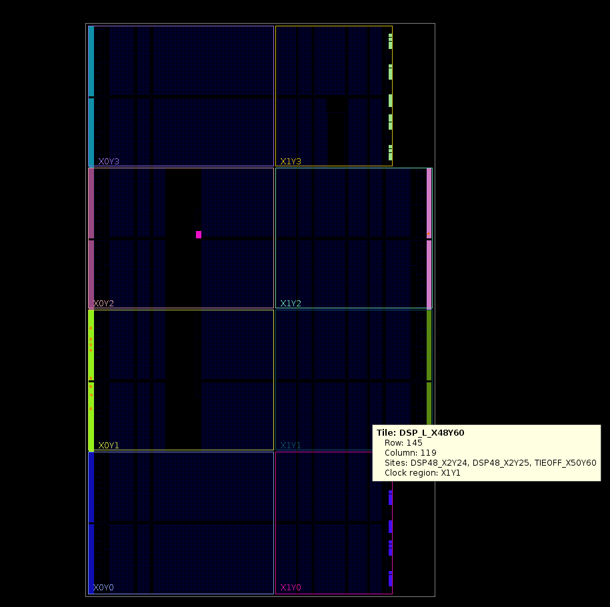


Fig 3: Vivado synthesis

# Results

After synthesis of the code using vhdl file and XDC file I obtained the Vivado logs and from the logs I noticed that the memory in MB was 7255.301 which is very high from the free physical which is 9685 MB it almost uses half of the available memory of the system. Therefore, the speed of the implementation is high, and the resource utilization is also high.