Faculty of Engineering and Computer Science
Expectations of Originality

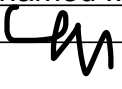
This form sets out the requirements for originality for work submitted by students in the Faculty of Engineering and Computer Science. Submissions such as assignments, lab reports, project reports, computer programs and take-home exams must conform to the requirements stated on this form and to the Academic Code of Conduct. The course outline may stipulate additional requirements for the course.

1. Your submissions must be your own original work. Group submissions must be the original work of the students in the group.
2. Direct quotations must not exceed 5% of the content of a report, must be enclosed in quotation marks, and must be attributed to the source by a numerical reference citation[1]. Note that engineering reports rarely contain direct quotations.
3. Material paraphrased or taken from a source must be attributed to the source by a numerical reference citation.
4. Text that is inserted from a web site must be enclosed in quotation marks and attributed to the web site by numerical reference citation.
5. Drawings, diagrams, photos, maps or other visual material taken from a source must be attributed to that source by a numerical reference citation.
6. No part of any assignment, lab report or project report submitted for this course can be submitted for any other course.
7. In preparing your submissions, the work of other past or present students cannot be consulted, used, copied, paraphrased or relied upon in any manner whatsoever.
8. Your submissions must consist entirely of your own or your group's ideas, observations, calculations, information and conclusions, except for statements attributed to sources by numerical citation.
9. Your submissions cannot be edited or revised by any other student.
10. For lab reports, the data must be obtained from your own or your lab group's experimental work.
11. For software, the code must be composed by you or by the group submitting the work, except for code that is attributed to its sources by numerical reference.

You must write one of the following statements on each piece of work that you submit:
For individual work: **"I certify that this submission is my original work and meets the Faculty's Expectations of Originality",** with your signature, I.D. #, and the date.
For group work: **"We certify that this submission is the original work of members of the group and meets the Faculty's Expectations of Originality",** with the signatures and I.D. #s of all the team members and the date.

A signed copy of this form must be submitted to the instructor at the beginning of the semester in each course.

I certify that I have read the requirements set out on this form, and that I am aware of these requirements. I certify that all the work I will submit for this course will comply with these requirements and with additional requirements stated in the course outline.

Course Number: COEN 313 Instructor: Otmane Ait Mohamed
Name: Mohamed Mrizek I.D. # 40234343
Signature:  Date: 17/06/2024

[1] Rules for reference citation can be found in "Form and Style" by Patrich MacDonagh and Jack Bordan, fourth edition, May, 2000, available at http://www.encs.concordia.ca/scs/Forms/Form&Style.pdf.
Approved by the ENCS Faculty Council February 10, 2012

PROJECT REPORT

Digital Systems Design II

Course: COEN 313 **Section:** AA

Project Title: Design and Implementation of a Digital Room Occupancy Monitoring System Using VHDL

Due Date: 2024 – 06 – 17

Name: Mohamed Mrizek **ID No.:** 40234343

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

Signature:  **Date:** 2024 – 06 –17

YYYY – MM – DD

Introduction:

This project was conducted to design a sophisticated digital system capable of accurately tracking and monitoring the occupancy of a room. Within this system, the primary entrance door is equipped with a highly responsive photocell, which, upon the obstruction of light, induces a change in a binary signal denoted as X. Similarly, individuals exiting the room pass through a second door featuring a photocell, triggering a binary signal Y when the light is interrupted. The system has a predefined maximum occupancy threshold, adjustable through an input signal known as max_occupancy, allowing for values up to a maximum limit of 63. This value should be stored in a register. When the room's occupancy reaches this defined maximum, the system must promptly respond by illuminating a conspicuous red indicator light, signifying that the room has reached its capacity. This state change is reflected in the binary signal Z. Furthermore, the system is equipped with a reset mechanism, invoked through a reset signal. This reset functionality enables the system to return to an initial state, allowing for accurate occupancy tracking to commence anew.

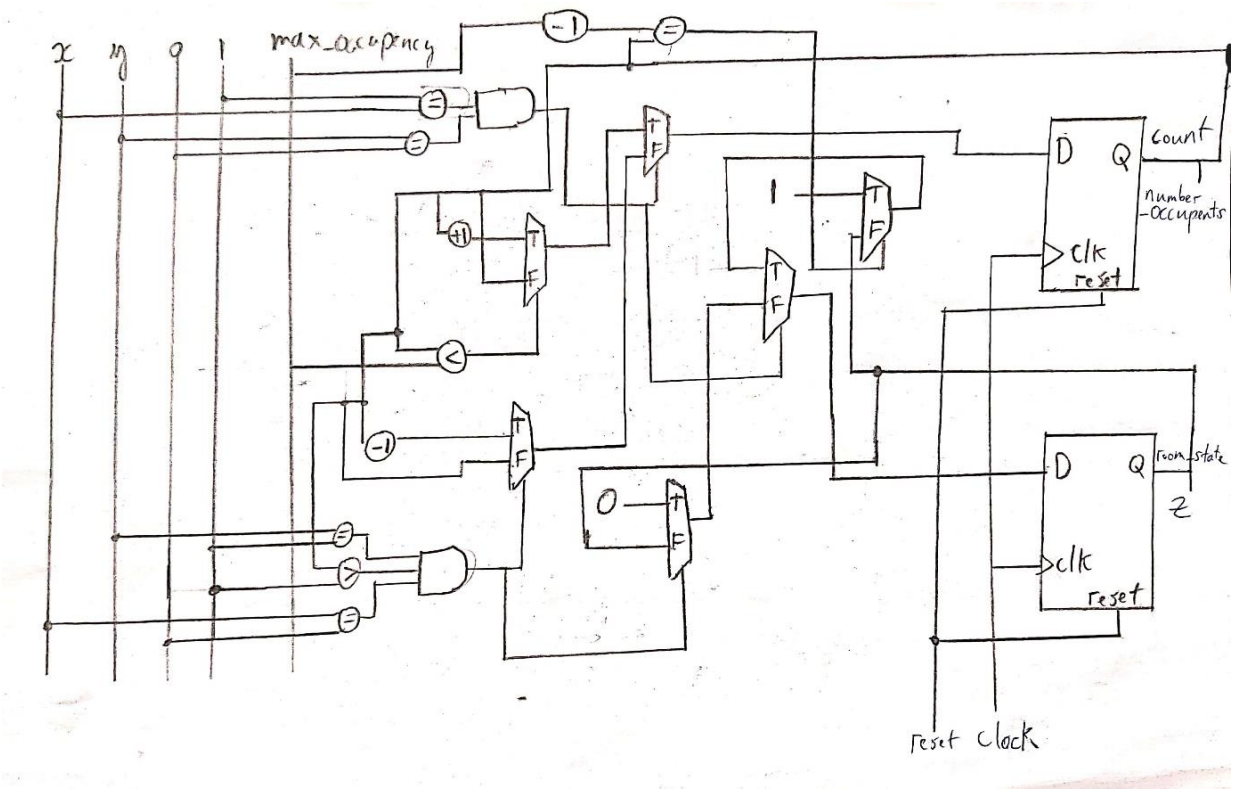
Procedure:

The following tasks led to the completion of the project:

1. Provide a conceptual diagram for the digital system. Identify the different blocks, such as multiplexers, flip-flops, incrementors, etc., and their interconnection.
2. Model your circuit using VHDL.
3. Provide a relevant test bench. Make sure to cover enough scenarios to verify your design.
4. Provide simulation and synthesis results for a Xilinx Nexys A7 FPGA development board (note: no actual FPGA implementation is required). Include the Vivado log file in the report.
5. Based on the results, comment on the quality of the design in terms of speed and FPGA resource utilization.

Results and discussion:

1) The following figure is a conceptual diagram of the digital system. Two D flip flops (one for each output) and five multiplexers were used to model the different conditional statements, such as increasing the count if X = 1 and Y = 0 and count < max_occupation. The diagram also contains basic arithmetic components such as incrementors, decrementors, and comparators.



2) The following VHDL code is used to model the circuit:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity room_occupancy is
  port (
    clk, reset : in std_logic;
    X, Y : in std_logic;
    max_occupancy : in unsigned(6 downto 0);
    number_occupants : out unsigned(6 downto 0);
    Z : out std_logic
  );
end room_occupancy;
architecture occupancy_arch of room_occupancy is
  signal count : unsigned(6 downto 0) := (others => '0');
  signal room_state : std_logic := '0';
begin
  process (clk)
  begin
    if rising_edge(clk) then
      if (reset = '1') then
        count <= (others => '0');
        room_state <= '0'; -- Room not full, light off
      elsif (X = '1' and Y = '0') then
        if count < max_occupancy then
```

```
            count <= count + 1;
          end if;
          if count = max_occupancy - 1 then
            room_state <= '1'; -- Room full, light on
          end if;
        elsif (Y = '1' and X = '0' and count > 0) then
          count <= count - 1;
          room_state <= '0'; -- Room not full, light off
        end if;
      end if;
   end process;
        Z <= room_state;
        number_occupants <= count;
end occupancy_arch;
```

The VHDL code simulates the tracking and monitoring of the occupancy of a room, with signals to track the number of occupants, handle room entry and exit, and indicate when the room is full. The code uses a clocked process to ensure changes occur synchronously with the clock signal and a reset mechanism to initialize the system.

3) I created the following testbench that covers enough scenarios to verify my design:

```
library IEEE;
use IEEE.Std_logic_1164.all;
use IEEE.Numeric_Std.all;
entity room_occupancy_tb is
end;
architecture bench of room_occupancy_tb is
 component room_occupancy
    port (
      clk, reset : in std_logic;
      X, Y : in std_logic;
      max_occupancy : in unsigned(6 downto 0);
      number_occupants : out unsigned(6 downto 0);
      Z : out std_logic
    );
 end component;
 signal clk, reset, X, Y: std_logic;
 signal max_occupancy: unsigned(6 downto 0);
 signal number_occupants: unsigned(6 downto 0);
 signal Z: std_logic;
begin
 uut: room_occupancy port map ( clk => clk,
                        reset => reset,
                        X => X,
                        Y => Y,
                        max_occupancy => max_occupancy,
                        number_occupants => number_occupants,
                        Z => Z );
 -- Clock generation
```

```
 clk_process: process
 begin
  clk <= '0';
  wait for 10 ns;
  clk <= '1';
  wait for 10 ns;
 end process;
 stimulus: process
 begin
  -- Reset the system
  reset <= '1';
  wait for 20 ns;
  reset <= '0';
  -- Set max_occupancy
  max_occupancy <= to_unsigned(63, max_occupancy'length);
  -- Add occupants using a loop
  for i in 1 to 65 loop
   X <= '1'; Y <= '0';
   wait for 20 ns; -- Ensure sufficient time for clock edge
   X <= '0';
   wait for 20 ns; -- Ensure sufficient time for clock edge
  end loop;
  -- Remove occupants using a loop
  for i in 1 to 40 loop
   X <= '0'; Y <= '1';
   wait for 20 ns; -- Ensure sufficient time for clock edge
   Y <= '0';
   wait for 20 ns; -- Ensure sufficient time for clock edge
  end loop;
  for i in 1 to 10 loop
   X <= '1'; Y <= '1';
   wait for 20 ns; -- Ensure sufficient time for clock edge
end loop;
  wait;
 end process;
end;
```

The testbench verifies 4 different functionalities using appropriate scenarios:

- First, it verifies the reset functionality. It ensures that the count and room_state are reset to 0 when the reset input is 1.

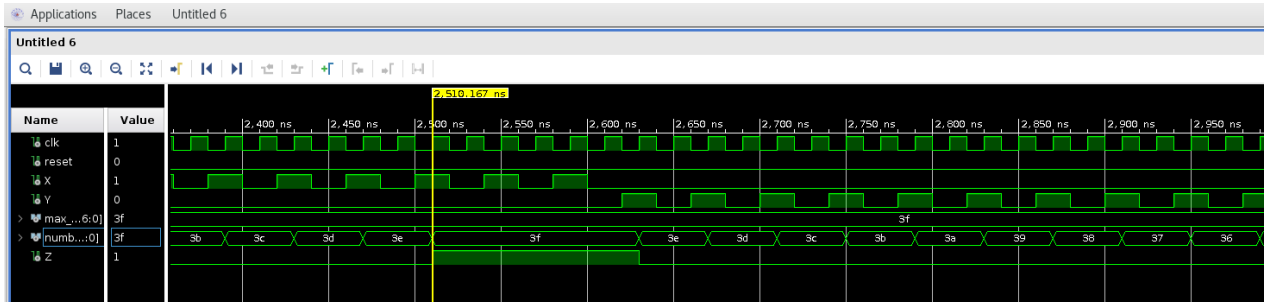
- Second, it verifies the adding occupants' functionality. It ensures that the number of occupants continues to increase when people enter until the maximum occupancy has occurred testing whether room_state will turn to 1.

- Third, it verifies the remove occupants' functionality. It ensures that the number of occupants continues to decrease when people exit until the occupancy hits 0 and also tests if room_state will turn to 0 when a person exits to ensure that the room is not cited as full.

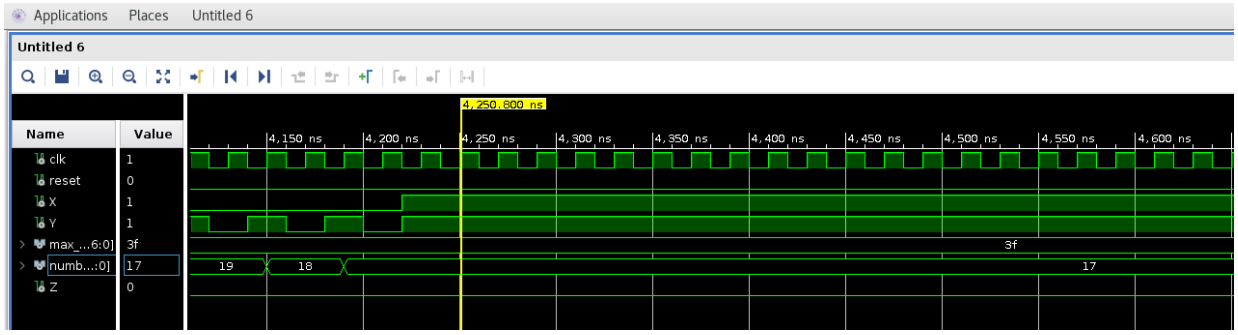
-Fourth, it verifies simultaneous entry and exit, it ensures that the system handles simultaneous entry and exit signals correctly (both X and Y are '1') meaning a no change in the value of count.

4) Simulating the previous VHDL code with the presented testbench led to the following results:

First, we activated the reset button so that number_of_occupants and Z are initialized to 0, meaning that the room is empty. We can notice then that when X = 1 and Y = 0, the number_of_occupants signal is incremented at the rising edge of the clock, with Z still being 0 since the room is not full yet.

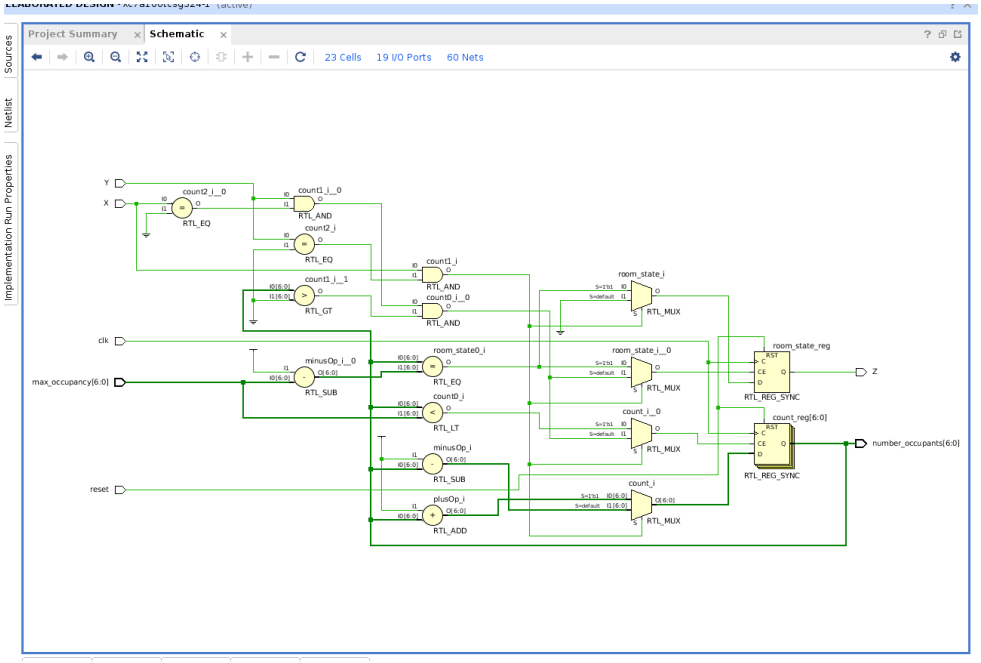


We continue incrementing the number of occupants, and we notice that when it reaches 63 (3f in hexadecimal), the Z value turns 1, meaning that the room is full. The Z value stays at 1 until we decrease the number of occupants. We notice that when Y = 1 and X = 0, the number of occupants starts to decrease by 1.

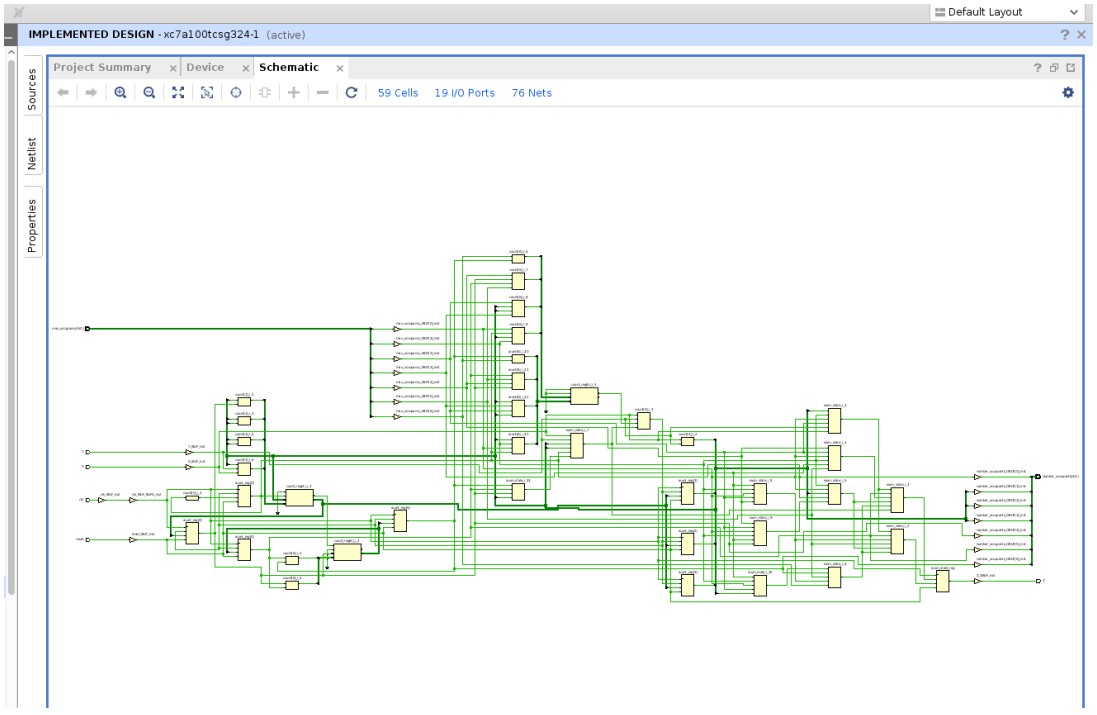


Finally, we test the case when X and Y are equal to 1 at the same time. We notice that the number of occupants stays the same when X and Y are equal to 1, which is in this case 17 in hexadecimal corresponding to 23 in decimal.

Using Vivado, we can generate the following elaborated diagram:



We were also able to synthesize the VHDL code, and we've got the following synthetized design:



5) In terms of speed:

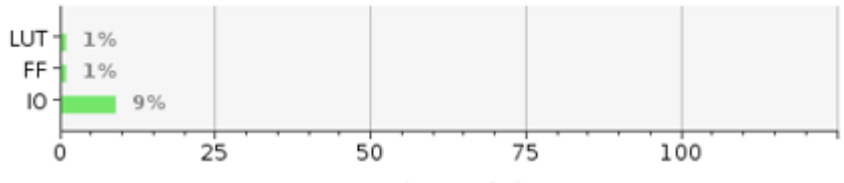
The design operates based on a clock signal with a period of 20 ns and a frequency of 50 MHZ. We can say that the frequency is modest and should be achievable on the FPGA device. The signals used in the design are 7-bit wide signals, so the arithmetic operations and comparisons are simple and fast.

In terms of FPGA resource utilization:

The design uses simple combinational logic elements (LUTs) for arithmetic and comparison operations and flip-flops for storing the count and room-state signals. Since these signals are of 7-bit length, the resource consumption is minimal. As for the I/O pins, we have a clock, reset, entry (‘X’), exit (‘Y’), max occupancy (‘max_occupancy’), and room full indicator (‘Z’). It’s efficient and minimal.

The following Vivado table shows how minimal our design is:

Resource	Utilization	Available	Utilization %
LUT	22	63400	0.03
FF	8	126800	0.01
IO	19	210	9.05



We can notice that LUT utilization is 0.03%, and FF utilization is 0.01%, and IO utilization is 9%. As we stated before, our design is efficient and minimal and can easily be handled by the FPGA.

Conclusion:

We successfully designed and implemented a digital system capable of accurately tracking and monitoring the occupancy of a room using VHDL. The system utilizes a combination of multiplexers, flip-flops, incrementors, and decrementors to model various conditions related to room entry and exit. A maximum occupancy threshold is set via an input signal, and a reset mechanism allows the system to return to an initial state for accurate tracking. Finally, It meets the requirements for accurate occupancy tracking, and minimal FPGA resource utilization.

Appendix:

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 Jun 13 12:43:02 2024

Process ID: 28678

Current directory:
/nfs/home/m/m_mrizek/COEN313/project/final_proj/final_proj.runs/synth_1

Command line: vivado -log room_occupancy.vds -product Vivado -mode batch -messageDb vivado.pb -notrace -source room_occupancy.tcl

Log file: /nfs/home/m/m_mrizek/COEN313/project/final_proj/final_proj.runs/synth_1/room_occupancy.vds

Journal file: /nfs/home/m/m_mrizek/COEN313/project/final_proj/final_proj.runs/synth_1/vivado.jou

#---

source room_occupancy.tcl -notrace

Command: synth_design -top room_occupancy -part xc7a100tcsg324-1

Starting synth_design

Attempting to get a license for feature 'Synthesis' and/or device 'xc7a100t'

INFO: [Common 17-349] Got license for feature 'Synthesis' and/or device 'xc7a100t'

INFO: Launching helper process for spawning children vivado processes

INFO: Helper process launched with PID 28694

Starting Synthesize : Time (s): cpu = 00:00:01 ; elapsed = 00:00:02 . Memory (MB): peak = 1401.586 ; gain = 85.801 ; free physical = 10289 ; free virtual = 22197

INFO: [Synth 8-638] synthesizing module 'room_occupancy' [/nfs/home/m/m_mrizek/COEN313/project/project.vhd:15]

INFO: [Synth 8-256] done synthesizing module 'room_occupancy' (1#1) [/nfs/home/m/m_mrizek/COEN313/project/project.vhd:15]

Finished Synthesize : Time (s): cpu = 00:00:01 ; elapsed = 00:00:02 . Memory (MB): peak = 1446.227 ; gain = 130.441 ; free physical = 10301 ; free virtual = 22210

Finished Constraint Validation : Time (s): cpu = 00:00:02 ; elapsed = 00:00:03 . Memory (MB): peak = 1446.227 ; gain = 130.441 ; free physical = 10301 ; free virtual = 22210

Start Loading Part and Timing Information

Loading part: xc7a100tcsg324-1

INFO: [Device 21-403] Loading part xc7a100tcsg324-1

Finished Loading Part and Timing Information : Time (s): cpu = 00:00:02 ; elapsed = 00:00:03 . Memory (MB): peak = 1454.223 ; gain = 138.438 ; free physical = 10300 ; free virtual = 22209

INFO: [Synth 8-5818] HDL ADVISOR - The operator resource <adder> is shared. To prevent sharing consider applying a KEEP on the output of the operator [/nfs/home/m/m_mrizek/COEN313/project/project.vhd:22]

Finished RTL Optimization Phase 2 : Time (s): cpu = 00:00:02 ; elapsed = 00:00:03 . Memory (MB): peak = 1462.227 ; gain = 146.441 ; free physical = 10292 ; free virtual = 22201

Report RTL Partitions:

+-+--------------+------------+----------+

| |RTL Partition |Replication |Instances |

+-+--------------+------------+----------+

+-+--------------+------------+----------+

No constraint files found.

Start RTL Component Statistics

Detailed RTL Component Info :

+---Adders :

2 Input 7 Bit Adders := 2

+---Registers :

7 Bit Registers := 1

1 Bit Registers := 1

+---Muxes :

2 Input 2 Bit Muxes := 1

2 Input 1 Bit Muxes := 2

Finished RTL Component Statistics

Start RTL Hierarchical Component Statistics

Hierarchical RTL Component report

Module room_occupancy

Detailed RTL Component Info :

+---Adders :

2 Input 7 Bit Adders := 2

+---Registers :

7 Bit Registers := 1

1 Bit Registers := 1

+---Muxes :

2 Input 2 Bit Muxes := 1

2 Input 1 Bit Muxes := 2

Finished RTL Hierarchical Component Statistics

Start Part Resource Summary

Part Resources:

DSPs: 240 (col length:80)

BRAMs: 270 (col length: RAMB18 80 RAMB36 40)

Finished Part Resource Summary

No constraint files found.

Start Cross Boundary and Area Optimization

Warning: Parallel synthesis criteria is not met

Finished Cross Boundary and Area Optimization : Time (s): cpu = 00:00:06 ; elapsed = 00:00:16 . Memory (MB): peak = 1596.059 ; gain = 280.273 ; free physical = 10102 ; free virtual = 22013

Report RTL Partitions:

+-+--------------+------------+----------+

| |RTL Partition |Replication |Instances |

+-+--------------+------------+----------+

+-+--------------+------------+----------+

No constraint files found.

Start Timing Optimization

Finished Timing Optimization : Time (s): cpu = 00:00:06 ; elapsed = 00:00:16 . Memory (MB): peak = 1596.059 ; gain = 280.273 ; free physical = 10102 ; free virtual = 22012

Report RTL Partitions:

+-+--------------+------------+----------+

| |RTL Partition |Replication |Instances |

+-+--------------+------------+----------+

+-+--------------+------------+----------+

Start Technology Mapping

Finished Technology Mapping : Time (s): cpu = 00:00:06 ; elapsed = 00:00:16 . Memory (MB): peak = 1596.059 ; gain = 280.273 ; free physical = 10101 ; free virtual = 22011

Report RTL Partitions:

+-+--------------+------------+----------+

| |RTL Partition |Replication |Instances |

+-+--------------+------------+----------+

+-+--------------+------------+----------+

Start IO Insertion

Start Flattening Before IO Insertion

Finished Flattening Before IO Insertion

Start Final Netlist Cleanup

Finished Final Netlist Cleanup

Finished IO Insertion : Time (s): cpu = 00:00:06 ; elapsed = 00:00:16 . Memory (MB): peak = 1596.059 ; gain = 280.273 ; free physical = 10101 ; free virtual = 22011

Report Check Netlist:

+------+------------------+-------+---------+-------+------------------+

| |Item |Errors |Warnings |Status |Description |

+------+------------------+-------+---------+-------+------------------+

|1 |multi_driven_nets | 0| 0|Passed |Multi driven nets |

+------+-----------------+-------+---------+-------+-----------------+

Start Renaming Generated Instances

Finished Renaming Generated Instances : Time (s): cpu = 00:00:06 ; elapsed = 00:00:16 . Memory (MB): peak = 1596.059 ; gain = 280.273 ; free physical = 10101 ; free virtual = 22011

Report RTL Partitions:

+-+--------------+------------+----------+

| |RTL Partition |Replication |Instances |

+-+--------------+------------+----------+

+-+--------------+------------+----------+

Start Rebuilding User Hierarchy

Finished Rebuilding User Hierarchy : Time (s): cpu = 00:00:06 ; elapsed = 00:00:16 . Memory (MB): peak = 1596.059 ; gain = 280.273 ; free physical = 10101 ; free virtual = 22011

Start Renaming Generated Ports

Finished Renaming Generated Ports : Time (s): cpu = 00:00:06 ; elapsed = 00:00:16 . Memory (MB): peak = 1596.059 ; gain = 280.273 ; free physical = 10101 ; free virtual = 22011

Start Handling Custom Attributes

Finished Handling Custom Attributes : Time (s): cpu = 00:00:06 ; elapsed = 00:00:16 . Memory (MB): peak = 1596.059 ; gain = 280.273 ; free physical = 10101 ; free virtual = 22011

Start Renaming Generated Nets

Finished Renaming Generated Nets : Time (s): cpu = 00:00:06 ; elapsed = 00:00:16 . Memory (MB): peak = 1596.059 ; gain = 280.273 ; free physical = 10101 ; free virtual = 22011

Start Writing Synthesis Report

Report BlackBoxes:

+-+--------------+----------+

| |BlackBox name |Instances |

+-+--------------+----------+

+-+--------------+----------+

Report Cell Usage:

+------+-------+------+

| |Cell |Count |

+------+-------+------+

|1 |BUFG | 1|

|2 |CARRY4 | 3|

|3 |LUT1 | 1|

|4 |LUT2 | 8|

|5 |LUT3 | 1|

|6 |LUT4 | 8|

|7 |LUT5 | 4|

|8 |LUT6 | 6|

|9 |FDRE | 8|

|10 |IBUF | 11|

|11 |OBUF | 8|

+------+-------+------+

Report Instance Areas:

+------+---------+-------+------+

| |Instance |Module |Cells |

+------+---------+-------+------+

|1 |top | | 59|

+------+---------+-------+------+

Finished Writing Synthesis Report : Time (s): cpu = 00:00:06 ; elapsed = 00:00:16 . Memory (MB): peak = 1596.059 ; gain = 280.273 ; free physical = 10101 ; free virtual = 22011

Synthesis finished with 0 errors, 0 critical warnings and 0 warnings.

Synthesis Optimization Runtime : Time (s): cpu = 00:00:06 ; elapsed = 00:00:16 . Memory (MB): peak = 1596.059 ; gain = 280.273 ; free physical = 10103 ; free virtual = 22013

Synthesis Optimization Complete : Time (s): cpu = 00:00:06 ; elapsed = 00:00:16 . Memory (MB): peak = 1596.059 ; gain = 280.273 ; free physical = 10112 ; free virtual = 22023

INFO: [Project 1-571] Translating synthesized netlist

INFO: [Netlist 29-17] Analyzing 14 Unisim elements for replacement

INFO: [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds

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

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.

INFO: [Common 17-83] Releasing license: Synthesis

12 Infos, 0 Warnings, 0 Critical Warnings and 0 Errors encountered.

synth_design completed successfully

synth_design: Time (s): cpu = 00:00:07 ; elapsed = 00:00:18 . Memory (MB): peak = 1748.027 ; gain = 444.895 ; free physical = 10073 ; free virtual = 21983

WARNING: [Constraints 18-5210] No constraint will be written out.

INFO: [Common 17-1381] The checkpoint '/nfs/home/m/m_mrizek/COEN313/project/final_proj/final_proj.runs/synth_1/room_occupancy.dcp' has been generated.

INFO: [runtcl-4] Executing : report_utilization -file room_occupancy_utilization_synth.rpt -pb room_occupancy_utilization_synth.pb

report_utilization: Time (s): cpu = 00:00:00.04 ; elapsed = 00:00:00.07 . Memory (MB): peak = 1772.051 ; gain = 0.000 ; free physical = 10074 ; free virtual = 21984

INFO: [Common 17-206] Exiting Vivado at Thu Jun 13 12:43:31 2024...