**2CS507CC23 - Digital Electronics**

***Assignment***

***On***

**Digital Discussion Room Allocation System for NIMA Learning Center**

***Submitted By:***

**HEER KASTURI (24BTM013)**

**DIYA SHAH (24BTM011)**

****

**INSTITUTE OF TECHNOLOGY**

**NIRMA UNIVERSITY**

**AHMEDABAD – 382481**

**October 2025**

***ABSTRACT***

*NIMA Learning Center’s growing interest in discussion opportunities has underscored a continuing concern with the procedure by which students reserve their discussion spaces: booking a room involves the inefficient process of checking room availability by submitting multiple Google Forms. This paper introduces the Digital Discussion Room Booking System, a fully automated circuit designed using Logisim, thus facilitating the reservation of a discussion space. The Digital Discussion Room Booking System allocates one of the ten rooms available for discussion in real time, based on the occupancy of each of the appropriate rooms. The state of each discussion room is stored via flip-flops, and combinational logic allocates the first available discussion room to the requester for each request. If there are no meeting spaces available, the system displays an unavailability message. The room allocation automatically resets back to available after a specified length of time has passed, to fairly allocate and distribute the discussion rooms, in a continuous manner over a 4-8 hour period of operation. This project afforded examples demonstrating how to apply some principles of digital electronics (the combination of sequential and combinational logic circuits) with respect to administrative and logistical challenges faced each day in an academic environment (and beyond), and how it can save time, provide transparency, and convenience.*

***PROBLEM STATEMENT***

Students at NIMA Learning Center are experiencing difficulties booking discussion rooms due to multiple Google Forms and room availability not being updated in real-time. The current process is manual and inefficient. To eliminate this challenge, an alternative room allocation system based on digital logic is proposed. This system would independently and automatically book and assign a room from a selection of ten available options, updating the room occupancy in real-time, and automatically resetting the room after a predetermined amount of time. Together, these features would make for faster and more efficient room management.

***ALGORITHM***

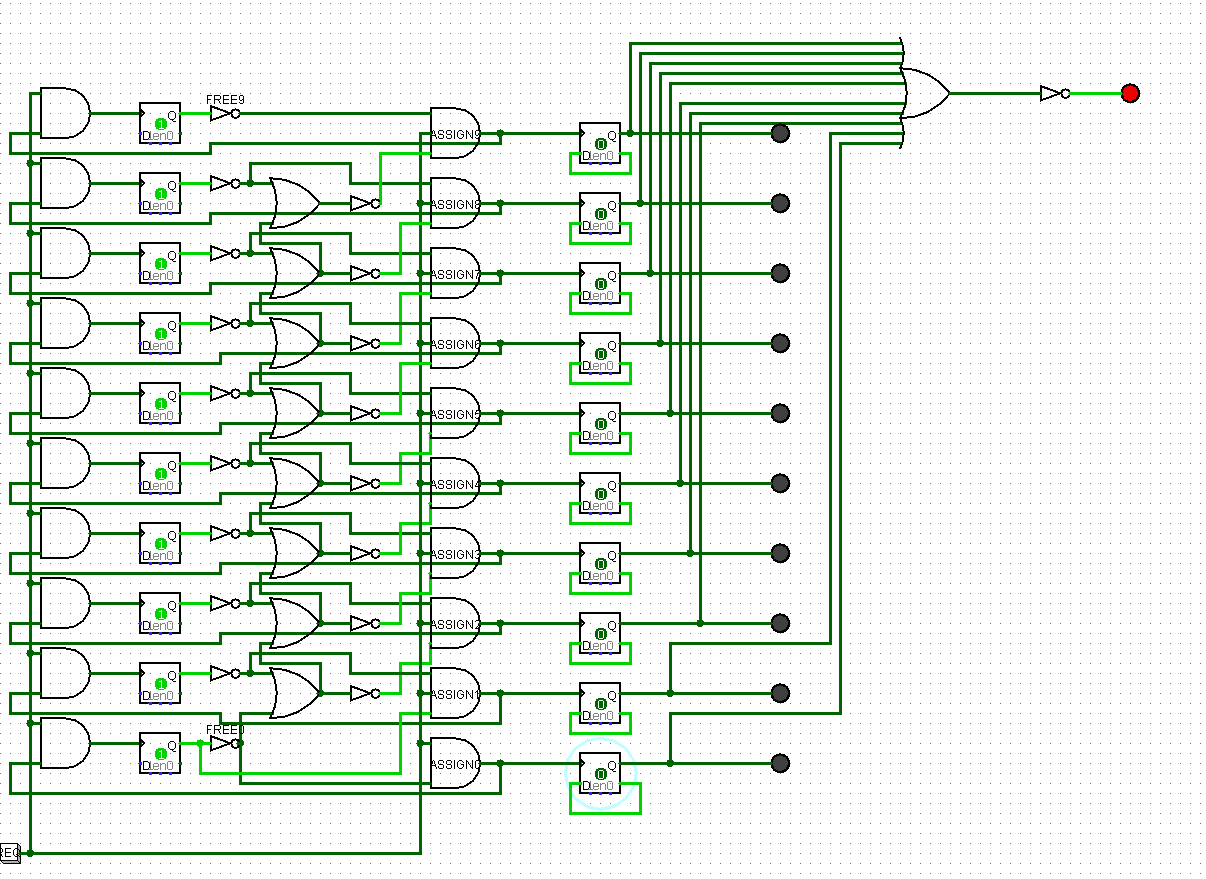
Start the system.  
  
Initialize all room status bits = 0 (0- room is free, 1-room occupied).  
  
To check the next available room sequentially from Room 1 to Room 10.  
  
If a room’s status bit = 0 (free):  
 a. Light the LED associated with it.  
 b. Change its status bit to 1.  
 c. Display the assigned room number.  
 d.(optional) Start a timer for that room’s booking duration.  
  
If all status bits = 1 (all rooms occupied):  
 Light the "no room available" LED.  
  
  
Whenever a room is occupied, the program shows the next available room by updating and displaying the number of the room that is next free.

***COMPONENTS USED:***

1. 20- D flip flops
2. 19 NOT gates
3. 9- 2 INPUT OR gates
4. 2- 4 INPUT OR gates
5. 1- 5 INPUT OR gate
6. 1- 10 INPUT OR gate
7. 19 AND gates
8. 11 LED

***CIRCUIT DESIGN***

The circuit executes the first-free room assignment for 10 discussion rooms (Room0..Room9). It is a synchronous design (employs D flip-flops for room state) with combinational logic to detect free rooms, and select the lowest-numbered free room when a booking request (REQ) is received.  
Main signals / buses used-  
REQ — booking request input (button)  
CLK — global clock (for D flip-flops)  
ROOMi— flip-flop output for room i (1 = occupied, 0 = free)  
FREEi = NOT(ROOMi) — 1 when room i is free  
ASSIGNi — one-hot signal indicating room i is the first free candidate (before gating by REQ)  
ASSIGNi\_pulse — ASSIGNi gated by REQ (final pulse used to set flip-flop)  
NO\_ROOM = NOT(ANY\_FREE) — indicates all rooms occupied.



BLOCK BY BLOCK DESCRIPTION

1. Room State Memory (D Flip-Flops)

Ten D flip-flops (ROOM0...ROOM9) keep track of room occupancy.

When ROOMi.Q = 1, that means room i is occupied.

LEDs are driven by ROOMi.Q to provide a visual indication of whether or not the room is occupied.

The flip-flops will need to be clocked (there is a CLK in the schematic, but it is not shown in the picture); whenever a clock edge happens, the assignments will latch and persist across clock edges.

2. Free Signal Generation

Each ROOMi.Q output is inverted using a NOT gate to generate FREEi = NOT(ROOMi.Q).

Every time that FREEi = 1, that means the corresponding room is free.

3. Priority (first-free) Logic → ASSIGN Signal.

The circuit creates a one-hot ASSIGN vector such that exactly the lower-index room which is free will have ASSIGN = 1.

The pattern of implementation (as seen in the schematic) is as follows:

ASSIGN0 = FREE0

ASSIGN1 = FREE1 AND NOT(FREE0)

ASSIGN2 = FREE2 AND NOT(FREE1 OR FREE0)

ASSIGN9 = FREE9 AND NOT(FREE8 OR ... OR FREE0)

This part will be built using some OR gates in succession to accumulate the lower index signals from FREE (keeping it an OR prefix), while the inverters and AND gates will create one hot encoding from ASSIGNi = 1.

4. Request Gating (REQ)

There will be a horizontal line where REQ is routed across all ASSIGN outputs.

Each ASSIGNi is combined with REQ (often in some case by a 3-input AND gate, or a second 2-input AND) to create ASSIGNi\_pulse = ASSIGNi AND REQ.

5. Latching / D input logic (holding behavior)

In order to maintain the booked state until cleared, each D input must retain its value after it has been assigned. The implemented (recommended) D logic per room uses the following:

D\_i = (ROOMi.Q AND NOT CLEARi) OR ASSIGNi\_pulse

In the above, ROOMi.Q AND NOT CLEARi is true and holds the current 1 if not cleared. ASSIGNi\_pulse causes the D to be true and go to 1 whenever a new assignment takes place.

The schematic contains the (or is the next step intended) OR/AND pattern; use that exact wiring so that assigned LEDs will remain ON after releasing the REQ.

6. ANY\_FREE / NO\_ROOM indicator

All FREEi lines are connected to a multi-input OR (connected two-input ORs) to get to ANY\_FREE.

NO\_ROOM = NOT(ANY\_FREE) can be represented with just one LED that will light when there is no room available.

7. Encoder (FIRST\_FREE)

The one-hot ASSIGN vector is coded into FIRST\_FREE[3:0] with OR trees:

FIRST\_FREE[0] = ASSIGN1 OR ASSIGN3 OR ASSIGN5 OR ASSIGN7 OR ASSIGN9

FIRST\_FREE[1] = ASSIGN2 OR ASSIGN3 OR ASSIGN6 OR ASSIGN7

FIRST\_FREE[2] = ASSIGN4 OR ASSIGN5 OR ASSIGN6 OR ASSIGN7

FIRST\_FREE[3] = ASSIGN8 OR ASSIGN9

The FIRST\_FREE bus can be fed into an adder (+1) and 7-segment/hex display and can show the room numbers from 1..10.

***APPLICATION***

Automated Room Assignment System

Can be implemented for libraries, learning centers, or offices to automatically assign discussion or meeting rooms to users in real-time based on availability.

Such a system would remove the requirement for checking availability or multiple google forms for all the rooms.

Resource Management System

The same logic could be applied to assign additional limited resources (such as computers, parking spaces, or equipment) where availability also needs to be tracked in real-time.

Queue Management System

The same logic could be adapted for systematic and automatic counting, particularly in settings such as banks, hospitals, or ticket buying systems.

Time Based Booking System

The system could also be adapted with timing circuits that factor automatic release of the resource after a designated time interval, for hourly booking or shared lab resources.

IoT or Smart Building Integration

The design could also be extended utilizing microcontrollers or sensors, taking a smart room-booking system from in essence, real-time to smart real-time, in that hardware reports availability.

***ADVANTAGES***

Completely Automatic Functionality

The circuit autonomously identifies and assigns the first available room without any human involvement or additional search process.

Less Waiting Time

Because if a room is available (NO\_ROOM indication) users don't have to wait, complete forms, and find another room, they can just start over.

Efficient Use of Resouces

The system guarantees that rooms are used sequentially (Room 1 → Room 2 → ...) and that available rooms that receive requests are not being left vacant.

Scalable and Modular

The system can simply be built from 10 rooms to 20 rooms or more, you only need to add more D flip-flops and implement the priority logic again.

Data Consistency / Memory Retention

Each D flip-flop maintains the information that the room is occupied until the booking is over or reset, so it does not allow double bookings.

Synchronous and Predictable

Because the circuit is clock based, assignments are stable and predictable form, there are no glitches or race conditions when designed with edge detection, and stable when producing assignments.

Simple Hardware Implementation

The systems is built entirely from gates (AND, OR, NOT) and D flip-flops in all, although some parts can optionally be built from existing products (encoders, adders)...cost effective, quick to show how it works.

Clear Visual Indication

Simple LED or display provides immediate and verbal feedback to the users of room availability (occupied/free) and the room number assigned***.***

***CONCLUSION***

The Room Booking Logic Circuit for NIMA Learning Center effectively showcases the value of digital logic design in addressing practical management issues. By utilizing basic logic gates and D flip-flops, the circuit automates the administration of rooms for discussion based on room availability. This automation minimizes the time and confusion associated with manually or form-based room booking processes and contributes to efficient resource allocation.

The circuit offers a reliable, simple, and inexpensive hardware option that can be upgraded and/or extended to more sophisticated options like microcontrollers and IoT solutions for smart automation. In essence, the project illustrates the practicality of digital systems design to enhance our everyday administrative activities and user-experience in educational settings.