

Automated Reception Desk for Hospitals

Team Members:

1. 221CS251, Srivatsan Suresh, srivatsansuresh.221cs251@nitk.edu.in, 9449820314
2. 221CS258, Vaibhav Anurag, anuragvaibhav.221cs258@nitk.edu.in, 79799 15189
3. 221CS264, Himaneesh Yadala, himaneeshyadala.221cs264@nitk.edu.in, 97399 55000

Abstract:

Background:

An automated system in hospitals for managing appointments can improve the efficiency and accuracy of the scheduling process. It can also reduce the workload of receptionists, allowing them to focus on other tasks, such as managing patient information and assisting patients. One of the key benefits of automating the appointment scheduling process is the ability to match patients with the most appropriate doctor based on their medical needs. This ensures that patients receive the appropriate care from specialized practitioners, ultimately leading to better treatment outcomes. Another benefit of an automated system is the ability to notify patients of their appointment details in a timely manner. This contributes to the overall enhancement of the patient experience, as it fosters a sense of transparency and reliability in the healthcare system. This model can be made using just logic gates, Flip Flops and Counters. This simplistic design also makes the model economical.

Motivation: We wanted to combine two different domains Healthcare and Technology and decided to make a project, which can make the experience of patients and hospital staff better in many ways. With our project we aim to address global health challenges by providing innovative solutions to streamline the process.

Introduction:

The proposed automated appointment management system for hospitals utilizes a four-bit integer input mechanism, aimed at optimizing the scheduling process. In this compact format, the first two bits are allocated to express the patient's specific query number, covering a range from 0 to 3. Simultaneously, the remaining two bits are harnessed for generating an automated indicator of consultant availability. This approach enhances the precision and efficiency of appointment scheduling, ensuring that patients receive tailored care from the right consultant promptly.

The workflow is as follows-

Upon receiving the input, the system performs the following actions:

- Query 0: If the query is '0,' the system promptly checks the availability of Doctor A. If Doctor A is ready, the system assigns them to the patient and provides a two-bit output indicating Doctor A's availability for consultation. In case Doctor A is occupied, the system notifies the patient about a brief wait.
- Query 3: For query '3,' the system checks if Doctor B is available. If so, Doctor B is assigned to the patient, and a two-bit output confirms Doctor B's readiness for consultation. If Doctor B is unavailable, the patient is informed about a short waiting period.
- Queries 1 and 2: When the query is '1' or '2,' the system assesses the availability of both Doctor A and Doctor B. If both doctors are free, the system efficiently assigns the patient to the first available doctor and provides a two-bit output, specifying which doctor is ready for consultation. If both doctors are occupied, the patient is courteously informed about a brief wait.

The waiting time is flexible, allowing the hospital to adapt based on current circumstances, patient queues, or hospital policies. After the waiting period, the system kindly requests input again.

In summary, this proposed automated system uses a four-bit input to represent the patient's query and consultant availability and provides a two-bit output indicating consultant availability or a waiting signal. The system is designed to match patients with the most suitable doctor based on their medical needs and the doctor's availability, enhancing the scheduling process's efficiency and accuracy.

This system aims to ensure that patients receive appropriate and timely medical attention. By using the patient's query as input, it directs them to doctors best suited

to address their specific needs, improving the overall quality of care at the hospital. In this scenario, the hospital has two doctors, S1 and S2, and can handle four query types, with specific doctors assigned to each query. Patients are automatically directed to the doctor best equipped to handle their medical needs.

However, in cases where the required doctors are unavailable, patients are required to wait, which is a common practice in hospitals. To address this, the system could enhance wait times by updating the availability of doctors regularly and providing estimated wait times to patients.

The implementation of this system marks a significant advancement in hospital operations by automating the allocation of doctors to patients in accordance with their specific medical needs. The overarching goal is to enhance overall operational efficiency, diminish patient wait times, and guarantee timely access to the appropriate medical care. This automation proves particularly pivotal in the context of patients with urgent medical requirements, as it directly contributes to an improved and more seamless patient experience while also bolstering the effectiveness of healthcare delivery throughout the facility. By aligning the right medical expertise with patients promptly, the system becomes an invaluable tool in optimizing the healthcare ecosystem and prioritizing the well-being of those seeking medical attention.

Working:

For Query 0, if the input is '0,' the system promptly checks if Doctor A is available. If Doctor A is free, the system assigns the patient to Doctor A and provides a two-bit response to indicate Doctor A's readiness for consultation. However, if Doctor A is currently occupied, the system politely informs the patient about a short wait.

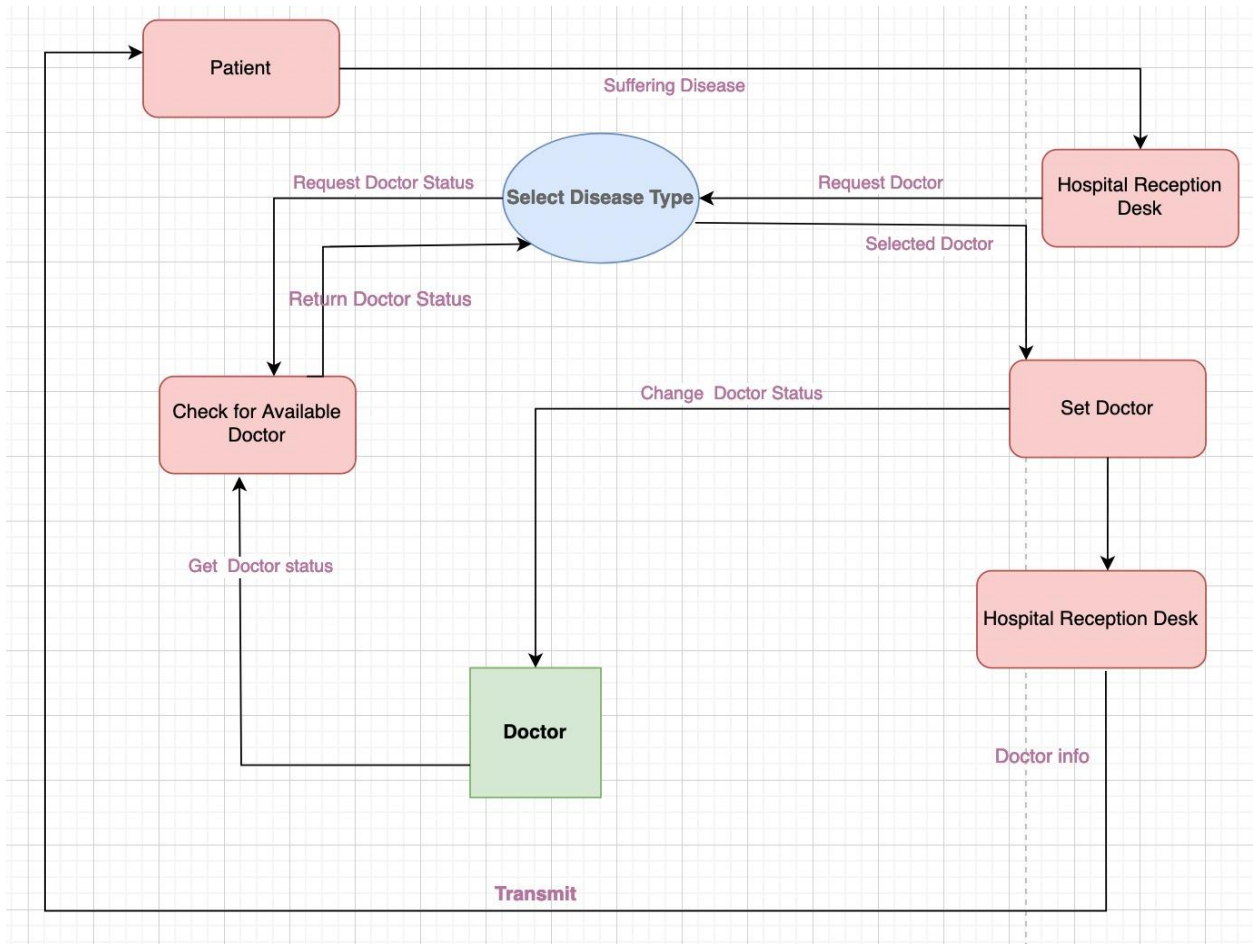
For Query 3, when the input is '3,' the system checks the availability of Doctor B. If Doctor B is ready, the system assigns the patient to Doctor B and provides a two-bit output to confirm Doctor B's readiness for consultation. If Doctor B is not available, the patient is informed about a brief waiting period.

Queries 1 and 2 are slightly different; when the input is '1' or '2,' the system assesses the availability of both Doctor A and Doctor B. If both doctors are free, the system efficiently assigns the patient to the first available doctor and provides a two-bit output specifying which doctor is ready for consultation. If both doctors are currently occupied, the system courteously informs the patient about a short waiting time.

Truth Table:

A	B	S1(t)	S2(t)	X	Y	S1(t+1)	S2(t+1)
0	0	0	0	1	0	1	0
0	0	0	1	1	0	1	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	1	0	1	0
0	1	0	1	1	0	1	1
0	1	1	0	0	1	1	1
0	1	1	1	0	0	1	1
1	0	0	0	1	0	1	0
1	0	0	1	1	0	1	1
1	0	1	0	0	1	1	1
1	0	1	1	0	0	1	1
1	1	0	0	0	1	0	1
1	1	0	1	0	0	0	1
1	1	1	0	0	1	1	1
1	1	1	1	0	0	1	1

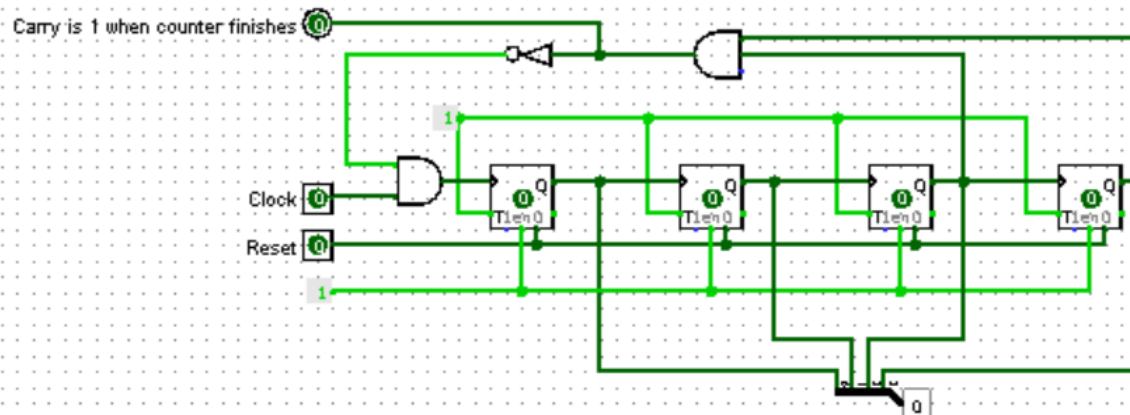
Flowchart:



Logisim Circuit Diagram:

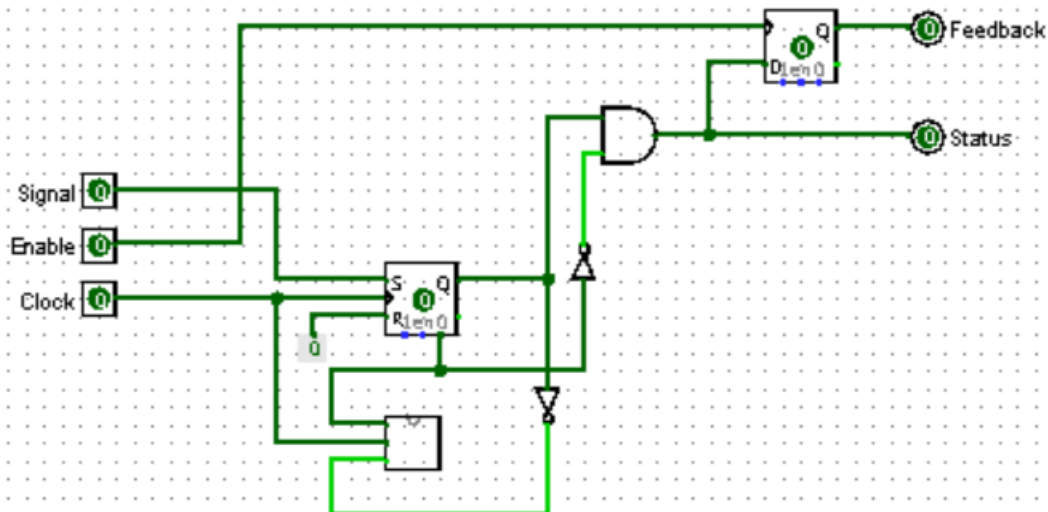
4 bit up counter

A counter circuit that returns 1 and stops counting as soon as it reaches the 0x0

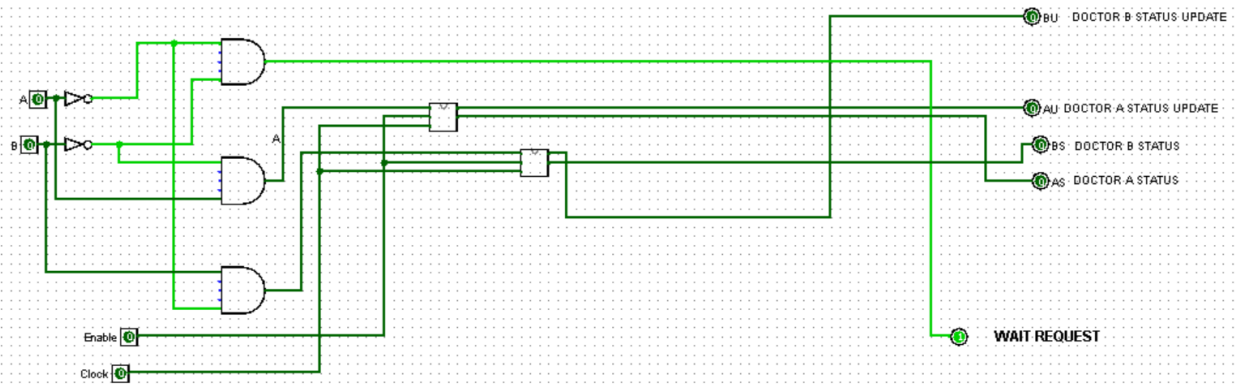


Timer

A circuit that combines the use of flip flop and counter to give status and feedback of doctors



DOCTOR AVAILABILITY STATUS UPDATE



During the consultation , the status of doctor availability is kept occupied for 15 seconds
Any enable is turned on, BU/AU checks and updates the status of respective doctors

INPUT EXPLANATION:

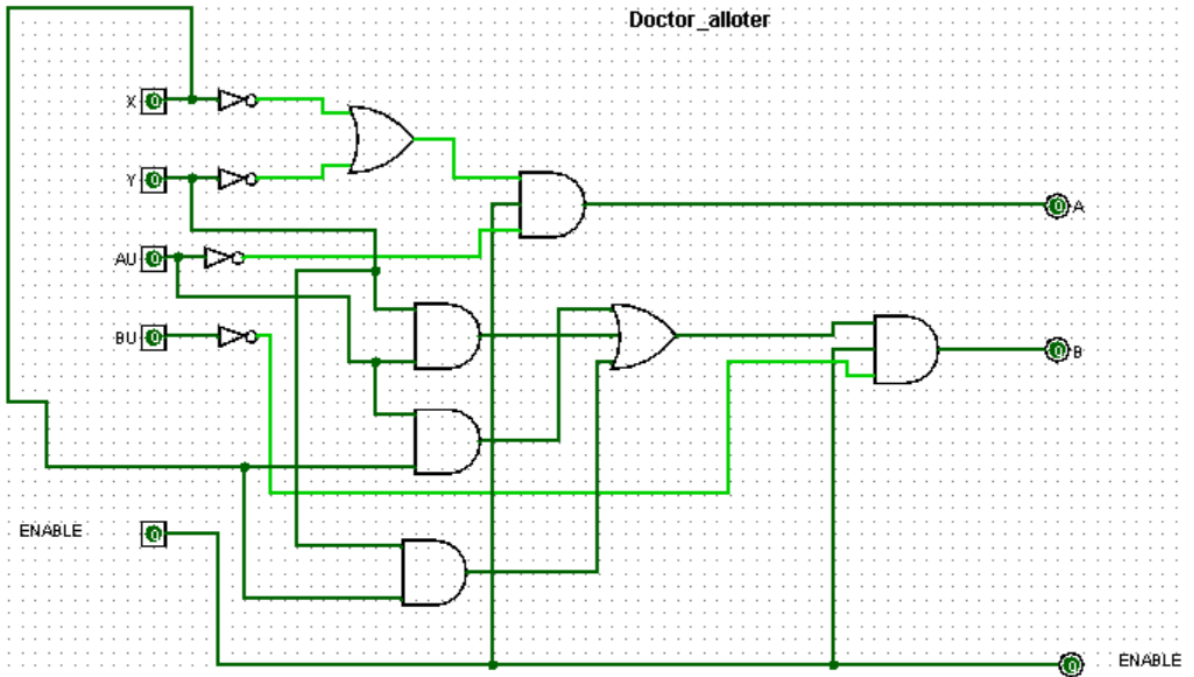
A and B is input which represents the doctor allotted by doctor_alloter

Output explanation:

The doctor status turns to 1

Doctor status is further given as feedback and becomes input for doctor_alloter

Doctor_alloter



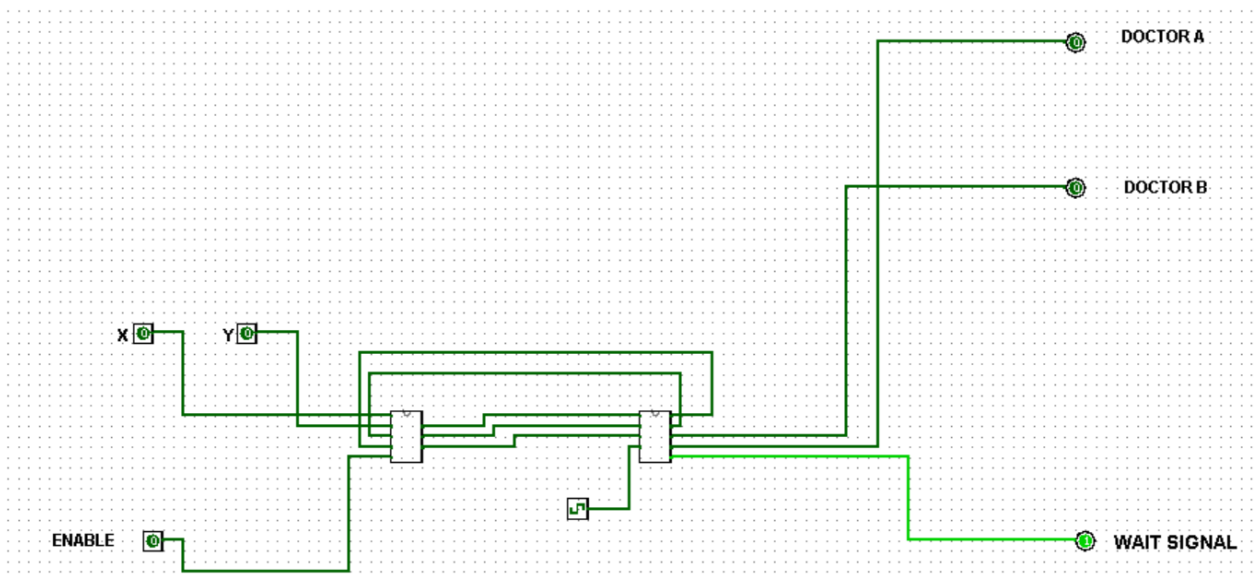
Input Information

X and Y are the inputs by user

AU and BU are status inputs given by doctor_availability

Output Information

A and B return which doctor is allotted



Verilog Code:

File: Reception_Desk.v

```

module reception_desk(query,A,B,clk,start,message);
    input [1:0] query;
    input A,B,clk,start;
    output reg [1:0] message;
    reg a = 1'b0;
    reg b = 1'b0;
    integer c_a = 0;
    integer c_b = 0;
    always@(negedge start)
    begin

```

```
if(query == 2'b00)
    if(a == 0)
        begin
            message = 2'b01;
            a = 1'b1;
        end
    else
        message = 2'b11;
if(query == 2'b01)
    if(a == 0)
        begin
            message = 2'b01;
            a = 1'b1;
        end
    else if(b == 0)
        begin
            message = 2'b10;
            b = 1'b1;
        end
    else
        message = 2'b11;
if(query == 2'b10)
    if(a == 0)
        begin
            message = 2'b01;
            a = 1'b1;
```

```

        end
        else if(b == 0)
        begin
            message = 2'b10;
            b = 1'b1;
        end
        else
            message = 2'b11;
    if(query == 2'b11)
        if(b == 0)
        begin
            message = 2'b10;
            b = 1'b1;
        end
        else
            message = 2'b11;
    end
always@(posedge clk)
begin
if(a == 1)
    c_a = c_a + 1;
if(b == 1)
    c_b = c_b + 1;
if(c_a == 15)
    a = 0;
if(c_b == 15)

```

```
        b = 0;
    end
endmodule
```

File: Reception_Desk_tb.v

```
`timescale 1ns/10ps
module testbench;
    reg start,clk;
    reg [1:0] query;
    wire [1:0] message;

    reception_desk doctor(query,A,B,clk,start,message);    //Create an object of
allot_doctor

    initial

    begin

        $display("300 time units is equal to 15 seconds.\nOutput '1' indicates
Doctor A has been allotted.\nOutput '2' indicates Doctor B has been allotted.\nOutput
'3' indicates the patient to wait until their doctor is available.");

        clk = 1'b0;

        repeat (100)

            #10 clk = ~clk;    //Create clock

    end

    initial

    begin

        $dumpfile("reception_desk.vcd");

        $dumpvars(0,doctor);
```

#10

start = 1'b1;

query = 2'b00;

start = 1'b0;

\$display(" | Incoming Query | Output To Query | Time Of Query |");

\$display("-----");

\$monitor(" | %d | %d | %3d |", query, message, \$time);

#90

start = 1'b1;

query = 2'b11;

start = 1'b0;

\$monitor(" | %d | %d | %3d |", query, message, \$time);

#100

start = 1'b1;

query = 2'b01;

start = 1'b0;

\$monitor(" | %d | %d | %3d |", query, message, \$time);

#100

start = 1'b1;

query = 2'b10;

start = 1'b0;

\$monitor(" | %d | %d | %3d |", query, message, \$time);

#100

start = 1'b1;

query = 2'b01;

start = 1'b0;

\$monitor("| %d | %d | %3d |", query, message, \$time);

#20

start = 1'b1;

query = 2'b00;

start = 1'b0;

\$monitor("| %d | %d | %3d |", query, message, \$time);

#20

start = 1'b1;

query = 2'b11;

start = 1'b0;

\$monitor("| %d | %d | %3d |", query, message, \$time);

#400

start = 1'b1;

query = 2'b11;

start = 1'b0;

\$monitor("| %d | %d | %3d |", query, message, \$time);

end

endmodule

```

PS D:\Downloads\Setups & Miscellaneous> vvp test.vvp
300 time units is equal to 15 seconds.
Output '1' indicates Doctor A has been allotted.
Output '2' indicates Doctor B has been allotted.
Output '3' indicates the patient to wait until their doctor is available.
VCD info: dumpfile reception_desk.vcd opened for output.
| Incoming Query | Output To Query | Time Of Query |
-----
|      0        |      1          |      10        |
|      3        |      2          |     100        |
|      1        |      3          |     200        |
|      2        |      3          |     300        |
|      1        |      3          |     400        |
|      0        |      3          |     420        |
|      3        |      3          |     440        |
|      3        |      3          |     840        |
PS D:\Downloads\Setups & Miscellaneous> |

```

References:

1. Digital Design by Moris Mano
2. <https://ciruciteasy.com>
3. <http://researchgate.com>