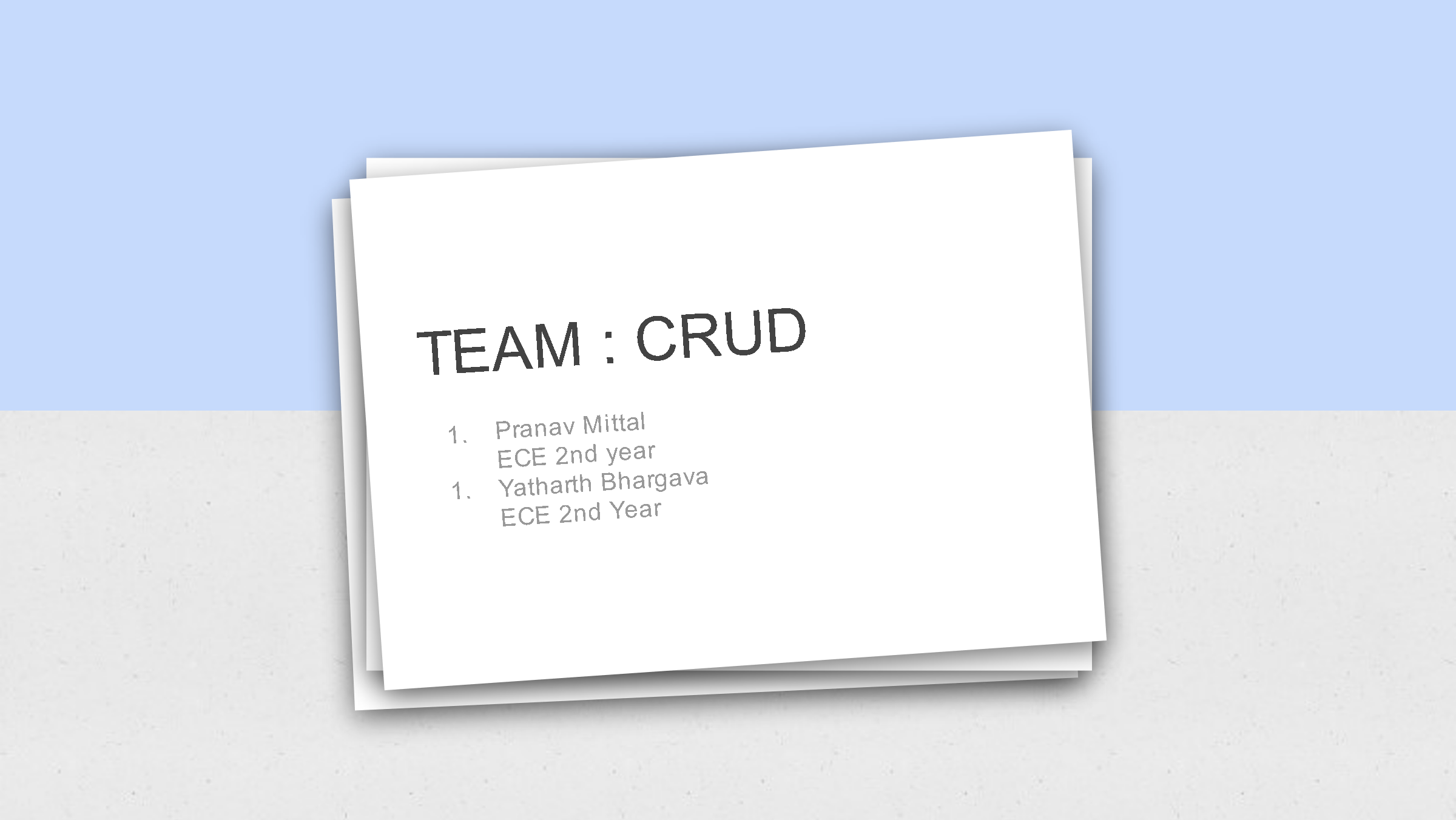
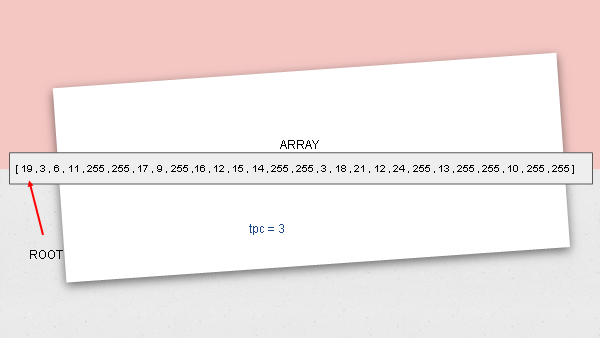
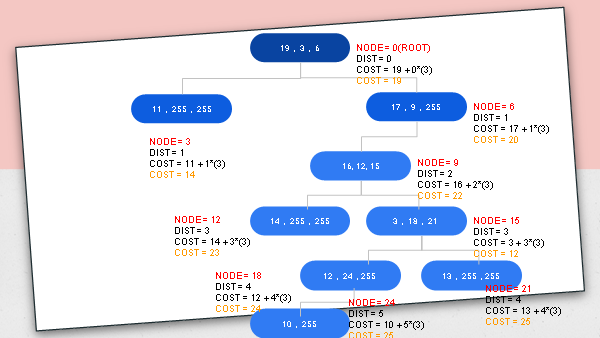
Digisim PS2



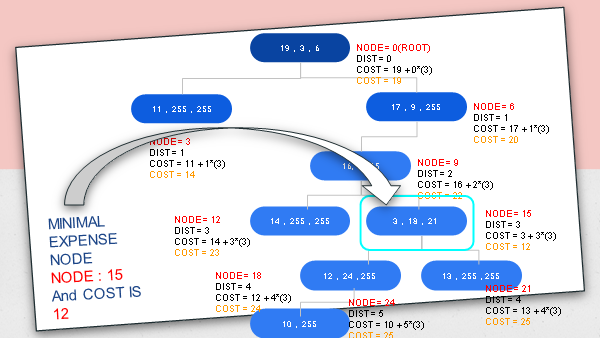
Test Case:





Start:

|  |  |  |
| --- | --- | --- |
| Queue | Address | Front |
| - | 0 | 0 |
| 255 | 1 | 0 |
| 3,255 | 2 | 1 |
| 6,3 | 255 | 1 |
| 255,6,3 | 3 | 2 |
| 255,6 | 4 | 1 |
| 255,6 | 5 | 1 |
| 255,6 | 6 | 1 |
| 255 | 7 | 0 |
| 9,255 | 8 | 1 |
| 9 | 255 | 0 |
| 255,9 | 9 | 1 |
| 255 | 10 | 0 |
| 12,255 | 11 | 1 |
| 15,12,255 | 255 | 2 |
| 255,15,12 | 12 | 2 |
| 255,15 | 13 | 1 |
| 255,15 | 14 | 1 |
| 255,15 | 15 | 1 |
| 255 | 16 | 0 |
| 18,255 | 17 | 1 |
| 21,18,255 | 255 | 2 |
| 255,21,18 | 18 | 2 |
| 255,21 | 19 | 1 |
| 24,255,21 | 20 | 2 |
| 24,255,21 | 21 | 2 |
| 24,255 | 22 | 1 |
| 24,255 | 23 | 1 |
| 24 | 255 | 0 |
| 255,24 | 24 | 1 |
| 255 | 25 | 0 |
| 255 | 26 | 0 **(END)** |



Circuit Explanation:

psuedocode

// BFS 255 wla concept

#include <bits\stdc++.h>

using namespace std;

int main(){

int info[]= {116,10,3,100,7,26,55,97,14,255,111,19,255,68,84,255,255,177,81,90,255,255,2,3,4,5,93,255,255,9,9};

int ans = 1e5,tpc;

cin>>tpc;

queue<int> q;

q.push(0);

q.push(255);

int dist = 0;

while(!q.empty()){

int curr = q.front();

q.pop();

if(curr == 255){

dist++;

if(!q.empty())

q.push(255);

continue;

}

cout<<curr<<"= curr\n";

cout<<dist<<" = its dist \n";

ans = min(ans,info[curr] + dist\*tpc);

if(info[curr+1]!=255)

q.push(info[curr+1]);

if(info[curr+2]!=255)

q.push(info[curr+2]);

}

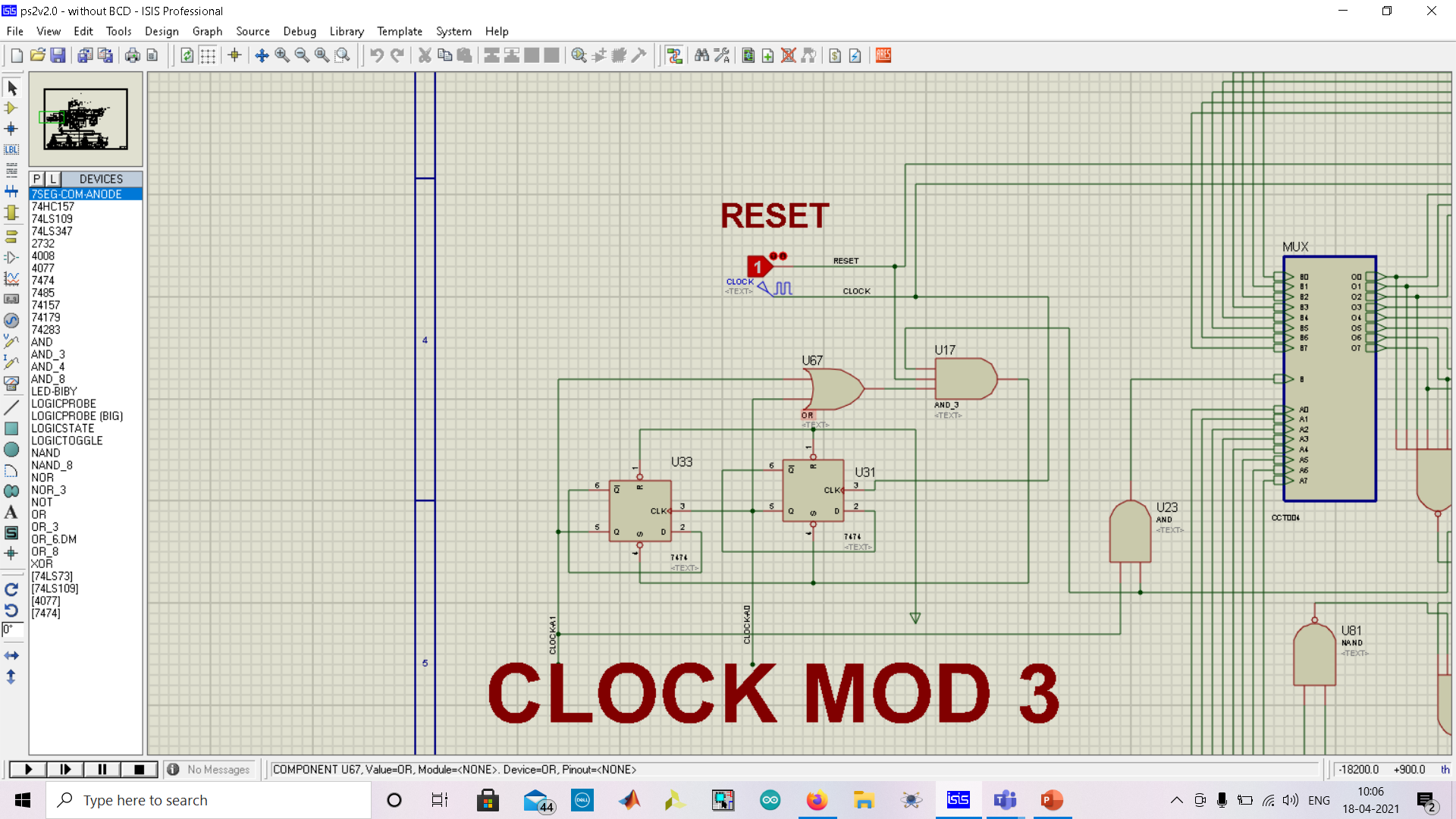
cout<<ans<<"=ans\n";

return 0;

}

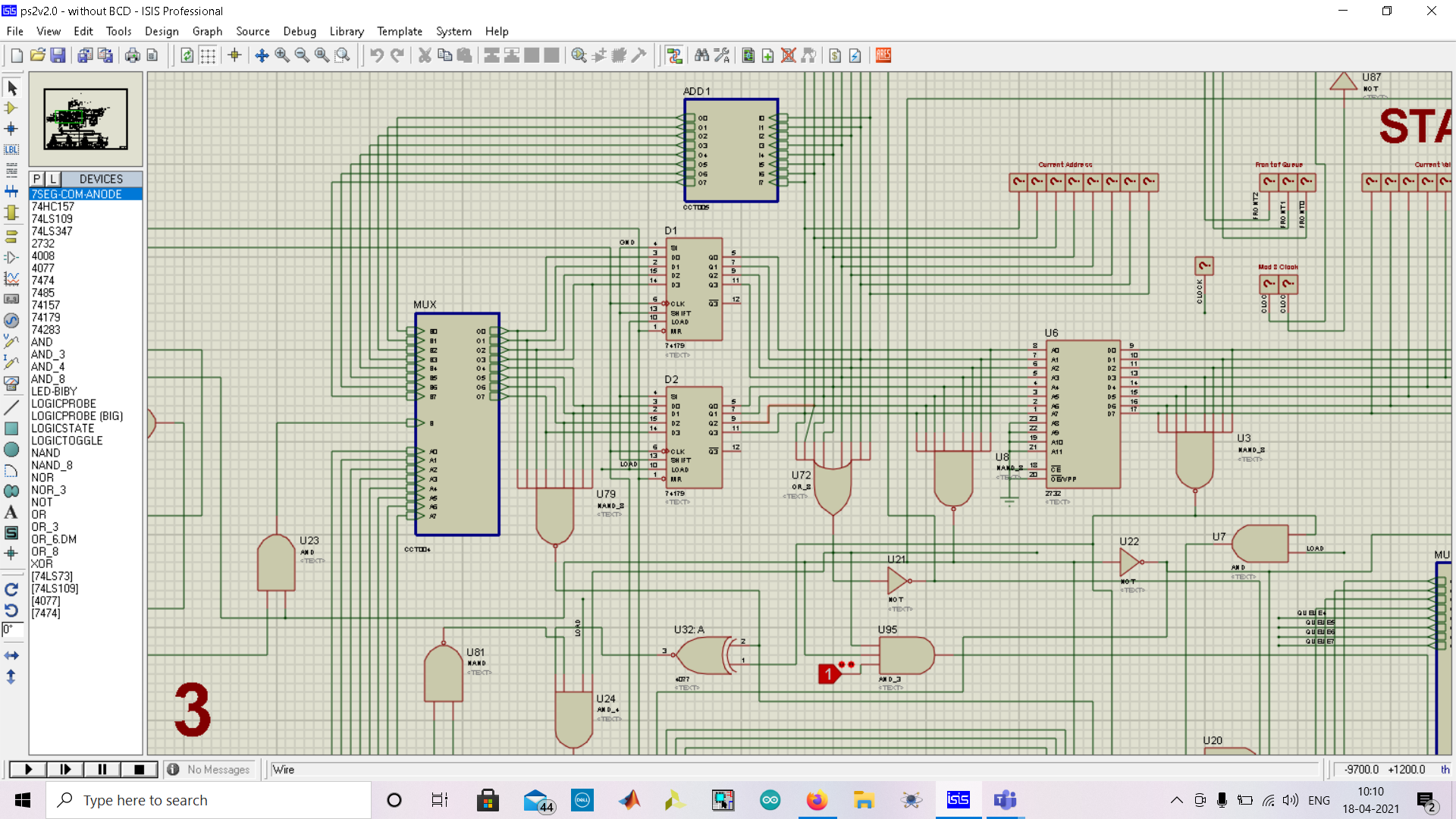
Explanation : Here we push 255 to queue at end of each level , when the popped element from queue == 255 , then we increment the value of TPC , to match the value dist\*tpc.

CLOCK:



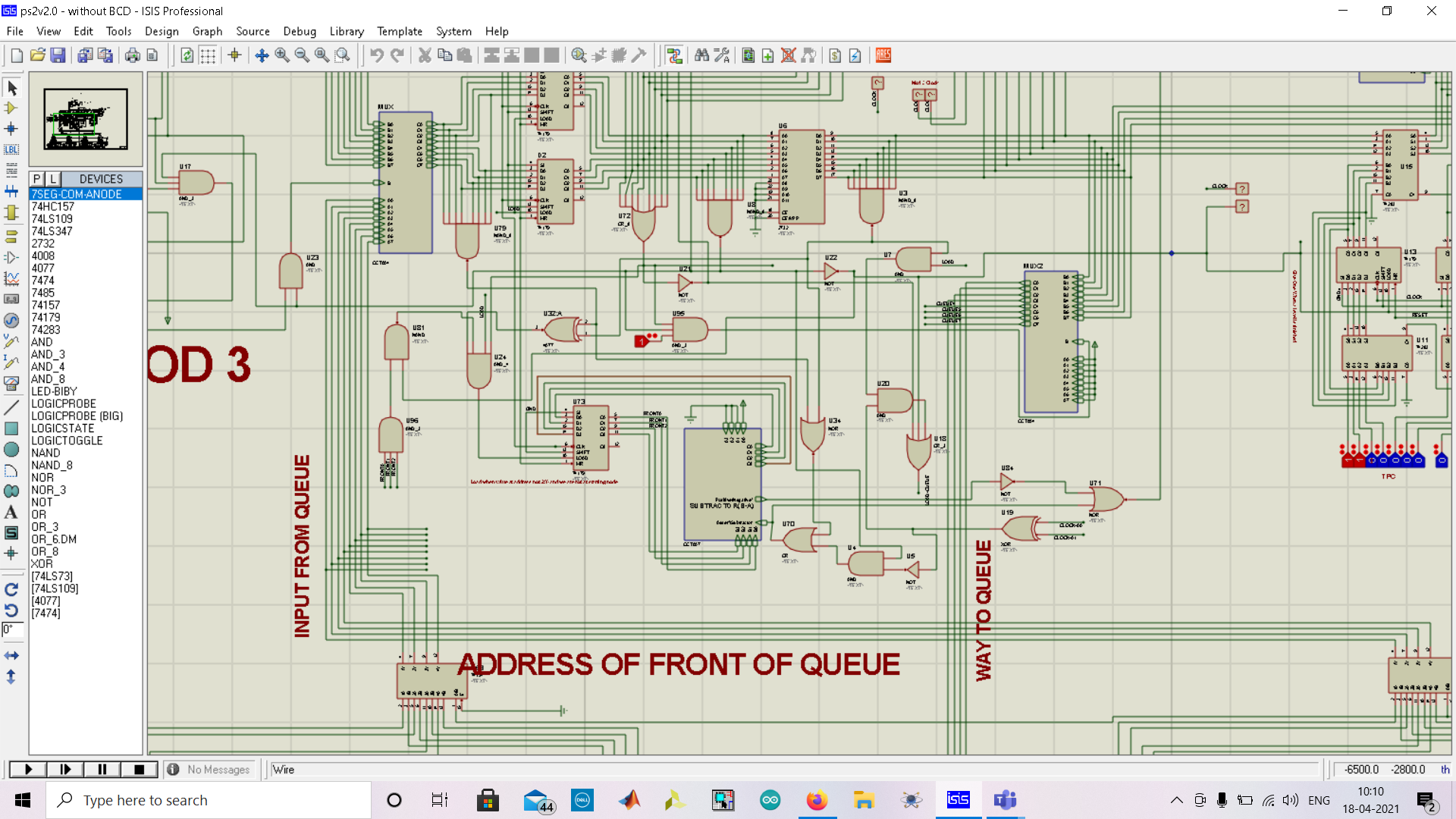
Its a down counter i.e the values shown are 11 , 10, 01 in this order.

ROM:



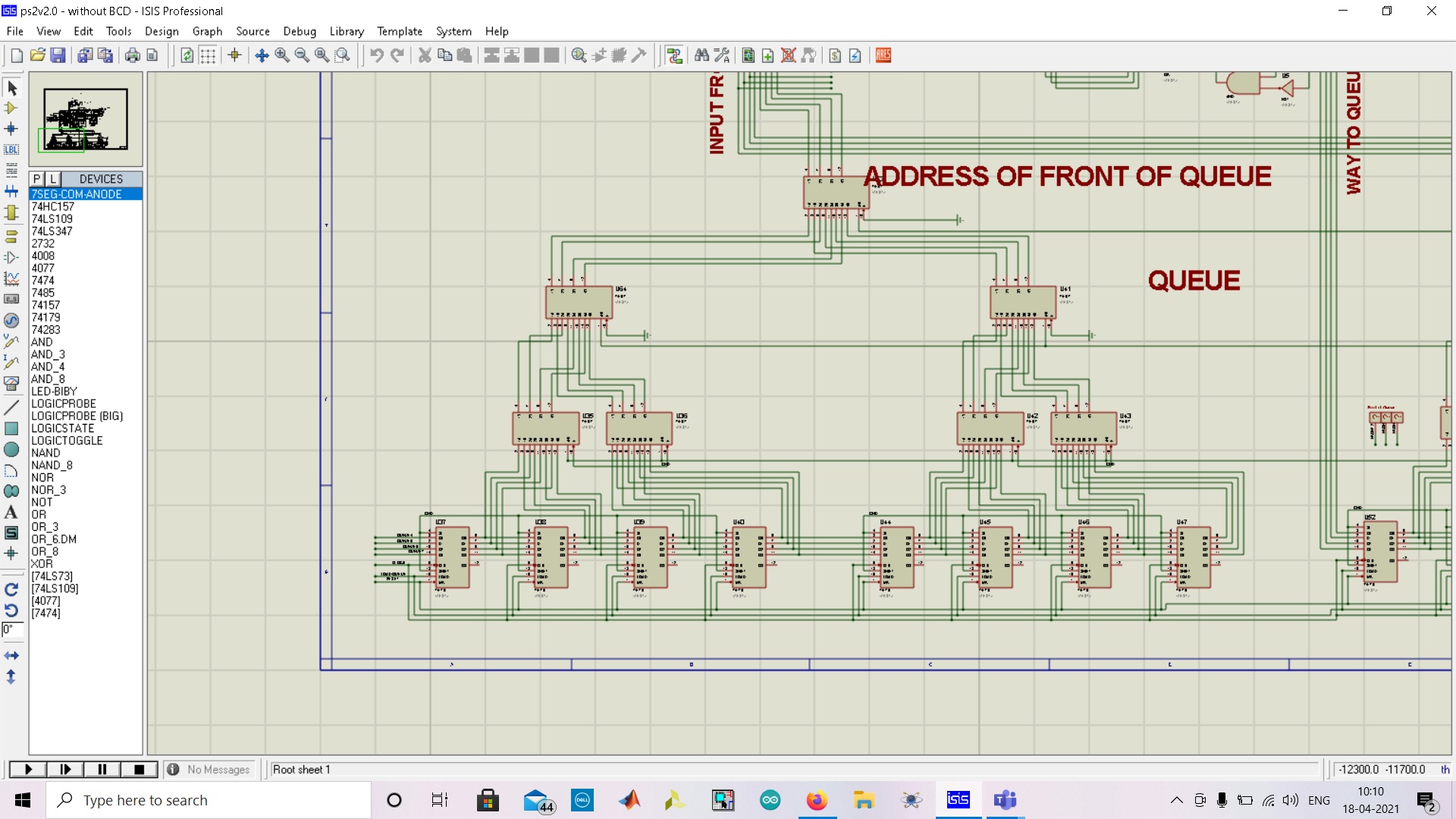
The address is taken at 11 we add 1 to address , at 10 we add 1 to address and at 01 we take the node at front of queue.

Front of Queue:



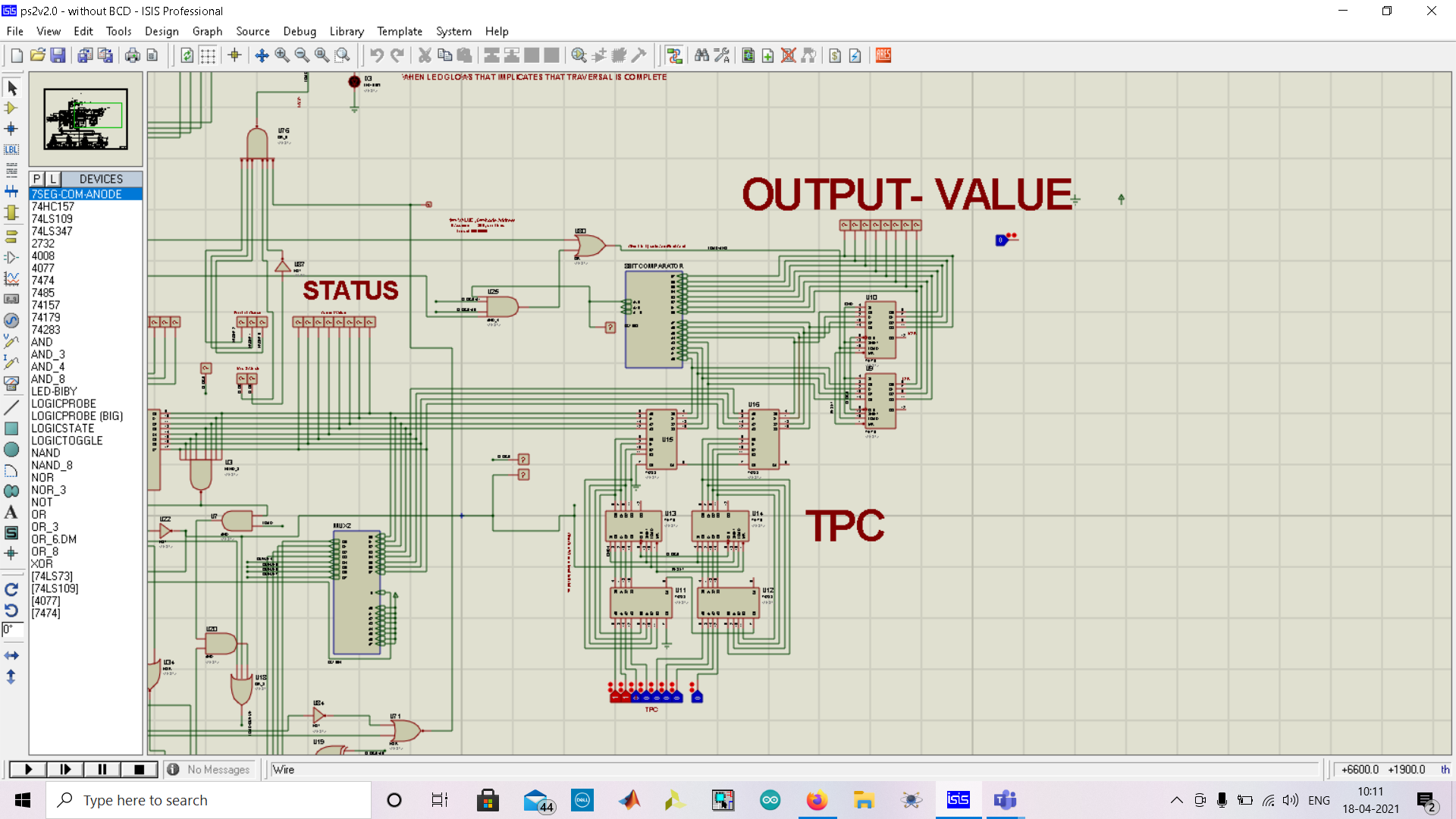
The front position of queue is stored in a 4 bit register and based on current address and mod 3 clock output it is incremented or decremented.

Queue:



The queue is basically a line of registers , with a load line for parallel right shift and MUX connected to select values via select lines FRONT0 ,FRONT 1, FRONT 2.

TPC:



Whenever we receive an address of 255 from queue , it means that a level has been traversed and TPC is incremented in the current value .

The stored value of TPC is then used to add to the value of node , that is used to calculate the expense.