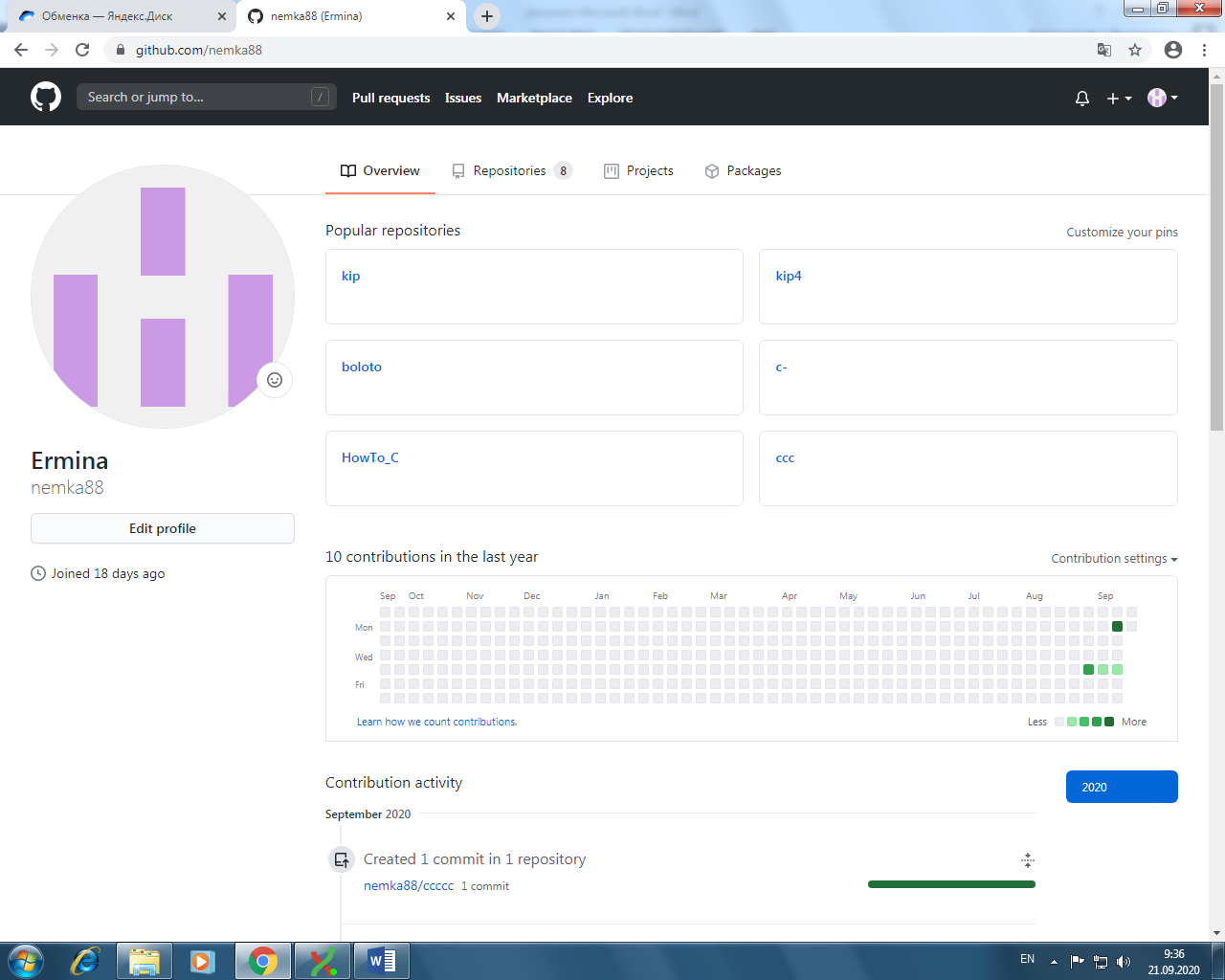
***Ермина Софья***

***2ИСИП-319***

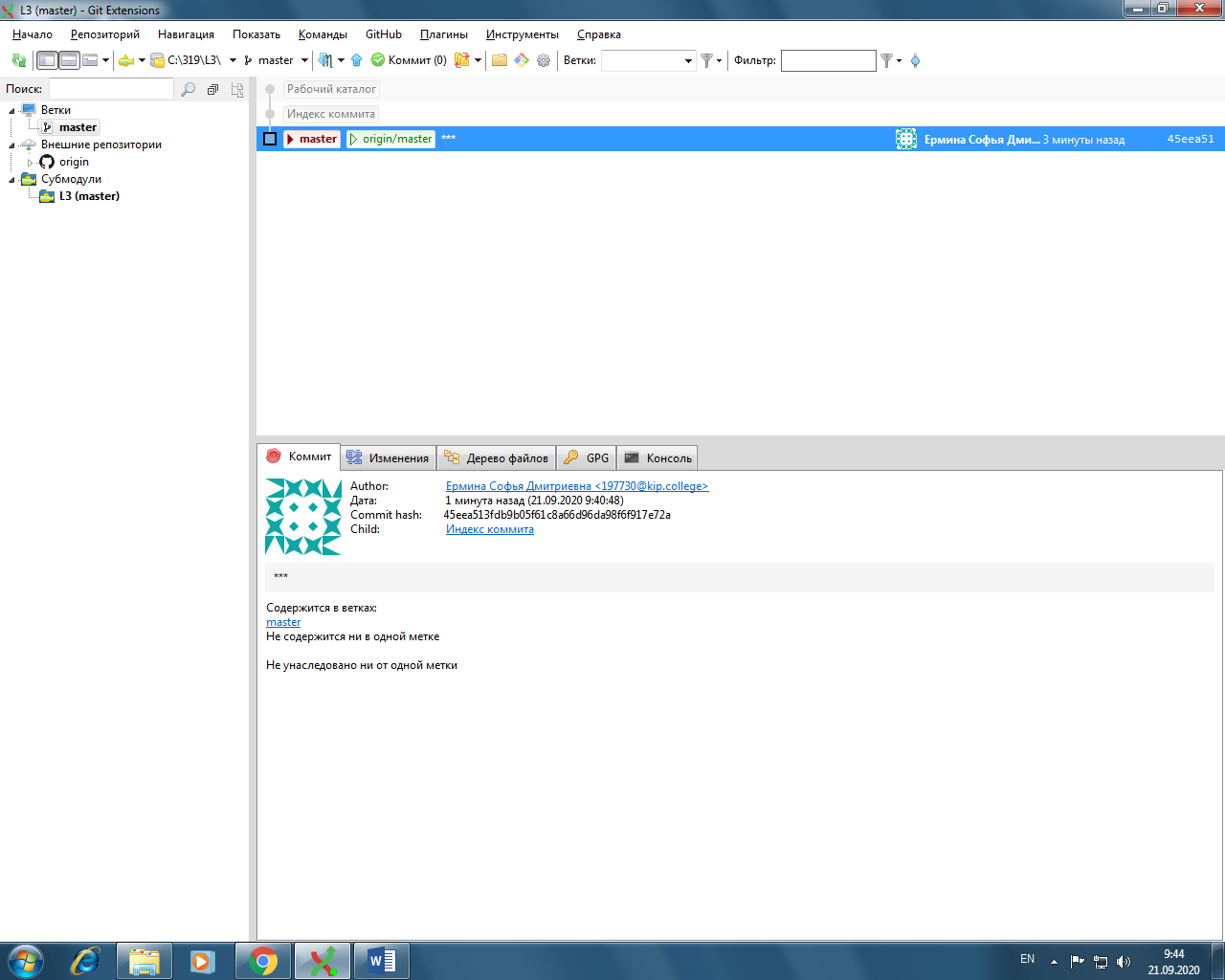
***Основы алгоритмизации и программирования***

**Задание L1.0 Зарегистрироваться на GitHub, опробовать работу с гит репозиторием через гит 2.8 GitExstension.**

***Ход выполнения задания***



* Зарегистрировалась на https://github.com/



* Заглянула в папку яндекс диск/обменка/Git
* Получила копию репозитория к себе на PC.
* Затем перешла в папку репозитория и создала текстовый файл.
* Текст файла:

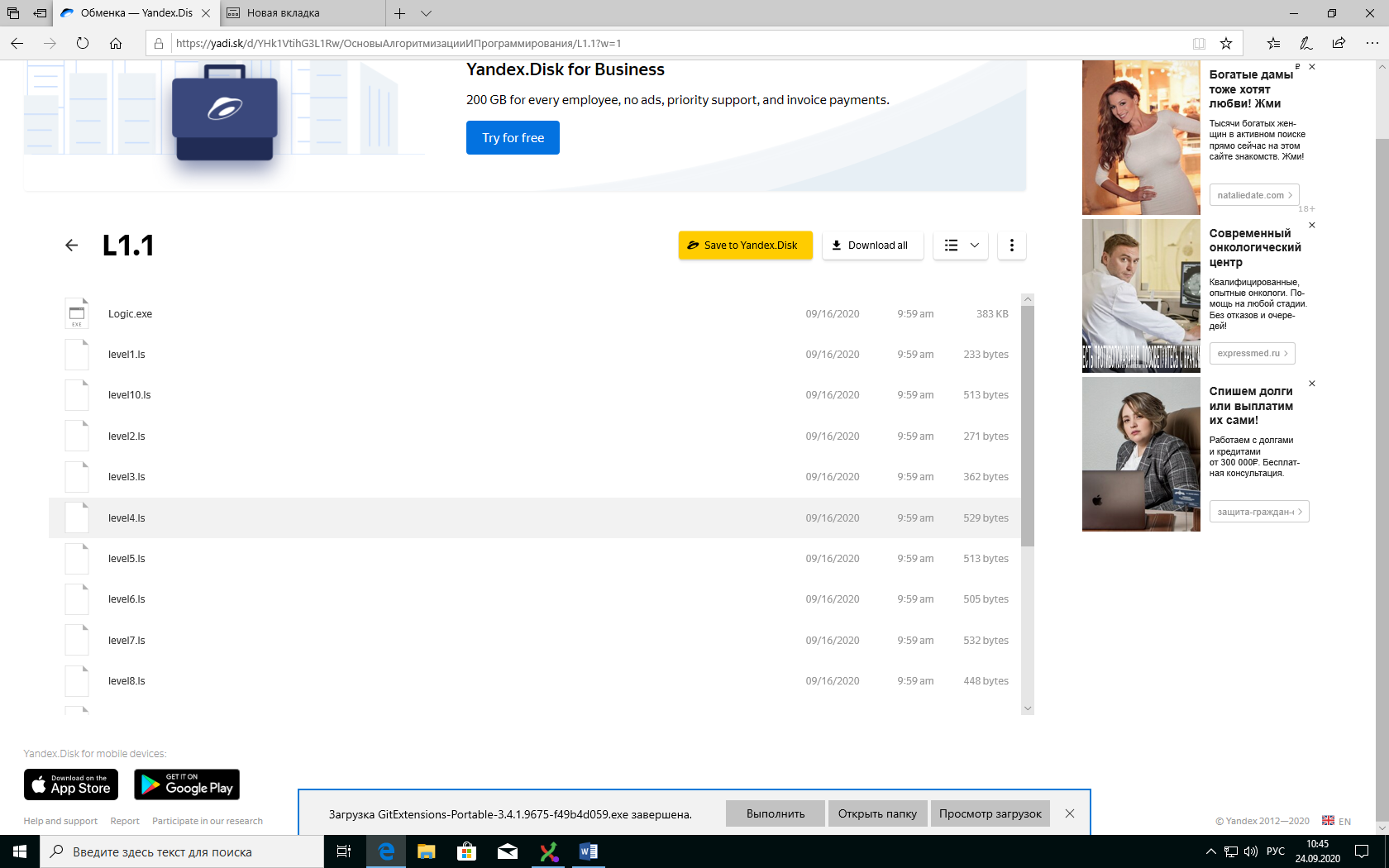
"ФИО

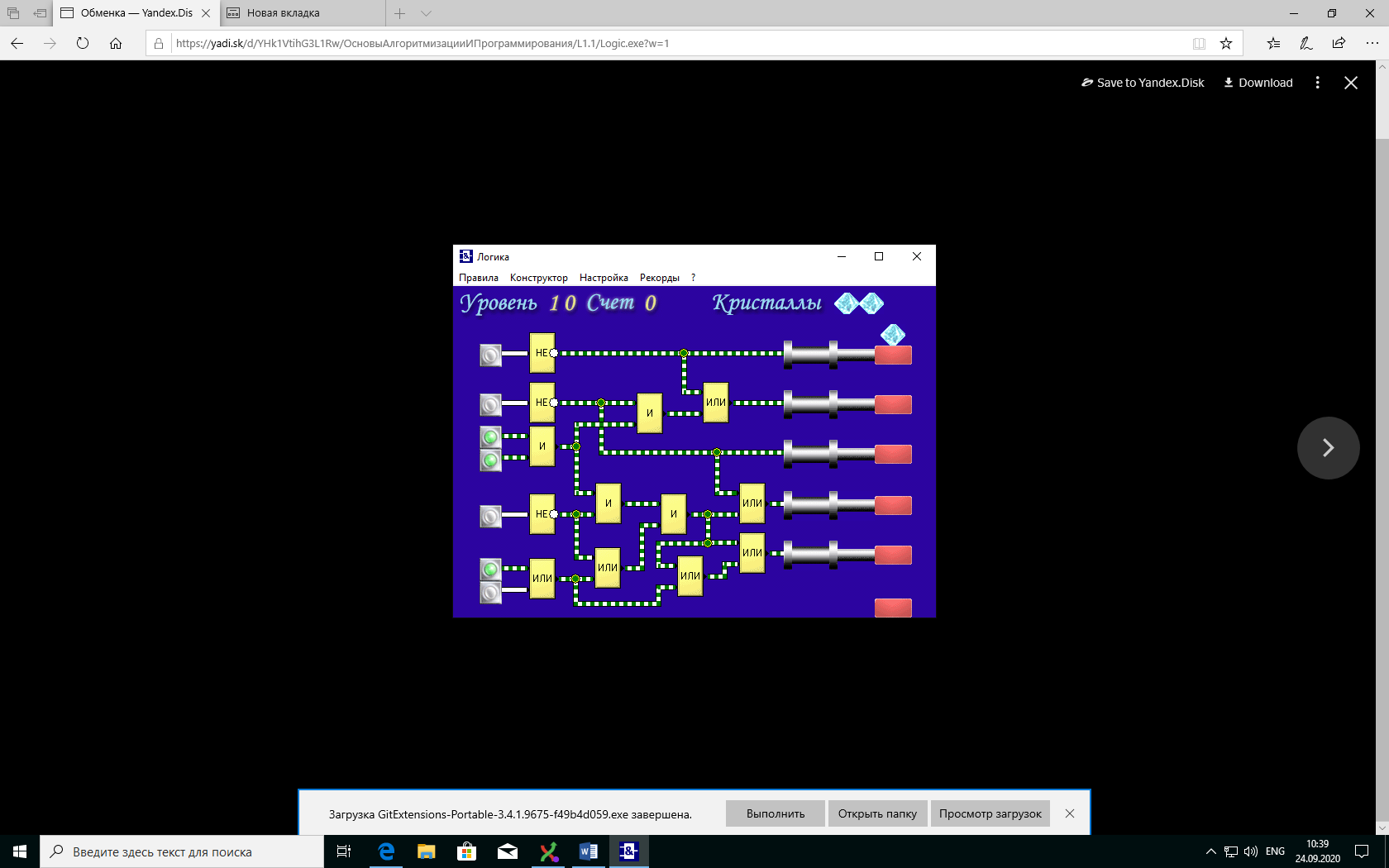
Привет мир"

* Сохранила файл.
* Проиндексировала файл через GitExtensions.
* Сделала коммит через GitExtensions с коментарием "\*\*\*"
* Отправила изменения на GitHub.

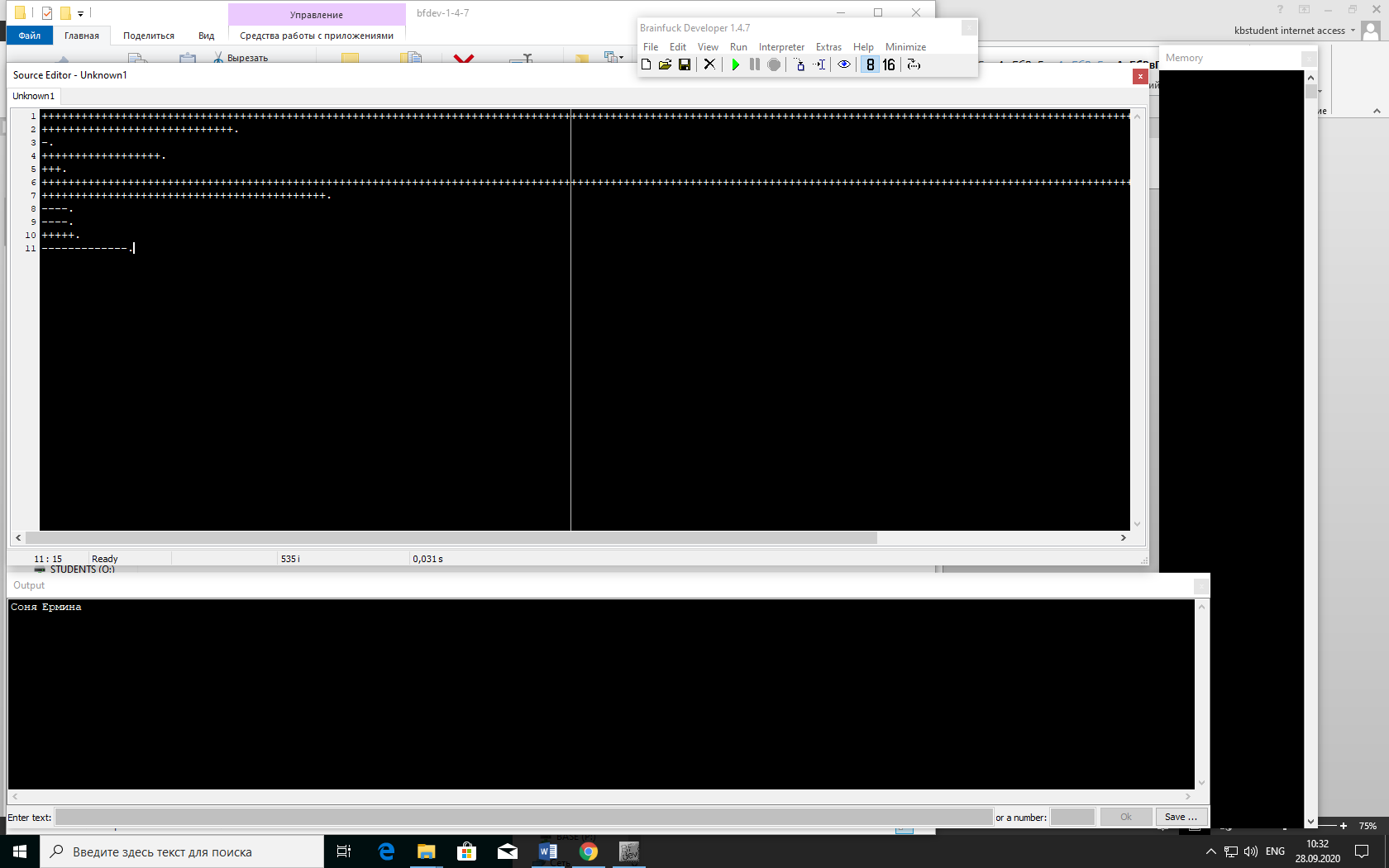
**Задание L1.1. Повторить алгебру логики с помощью игры**

1. Зашла на Яндекс диск в папку Архитектура Аппаратных Средств
2. Скачала программу Logic.exe и начала играть



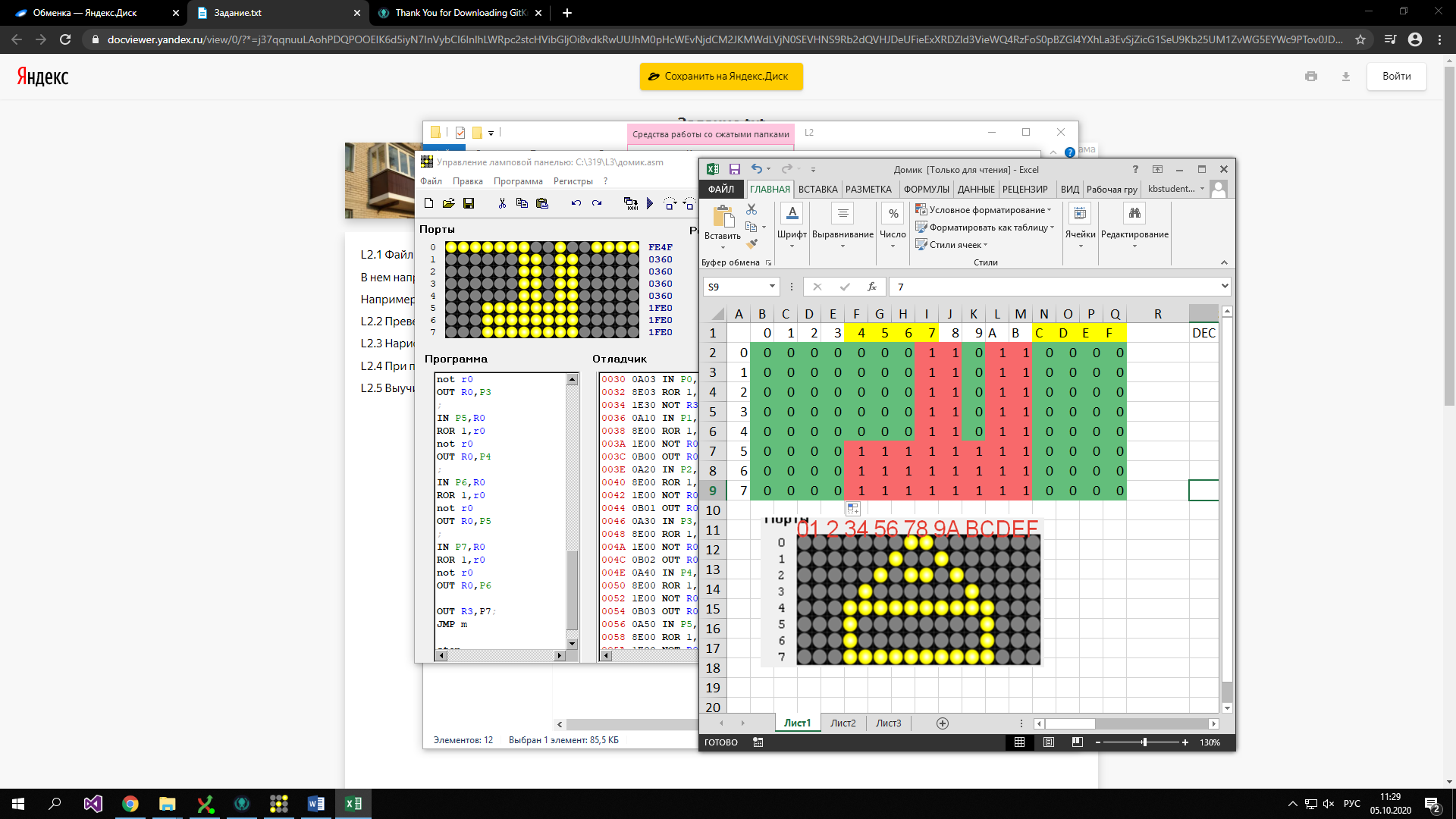
1. 
2. Дошла до 10 уровня и увидел картинку Шрека с ослом.

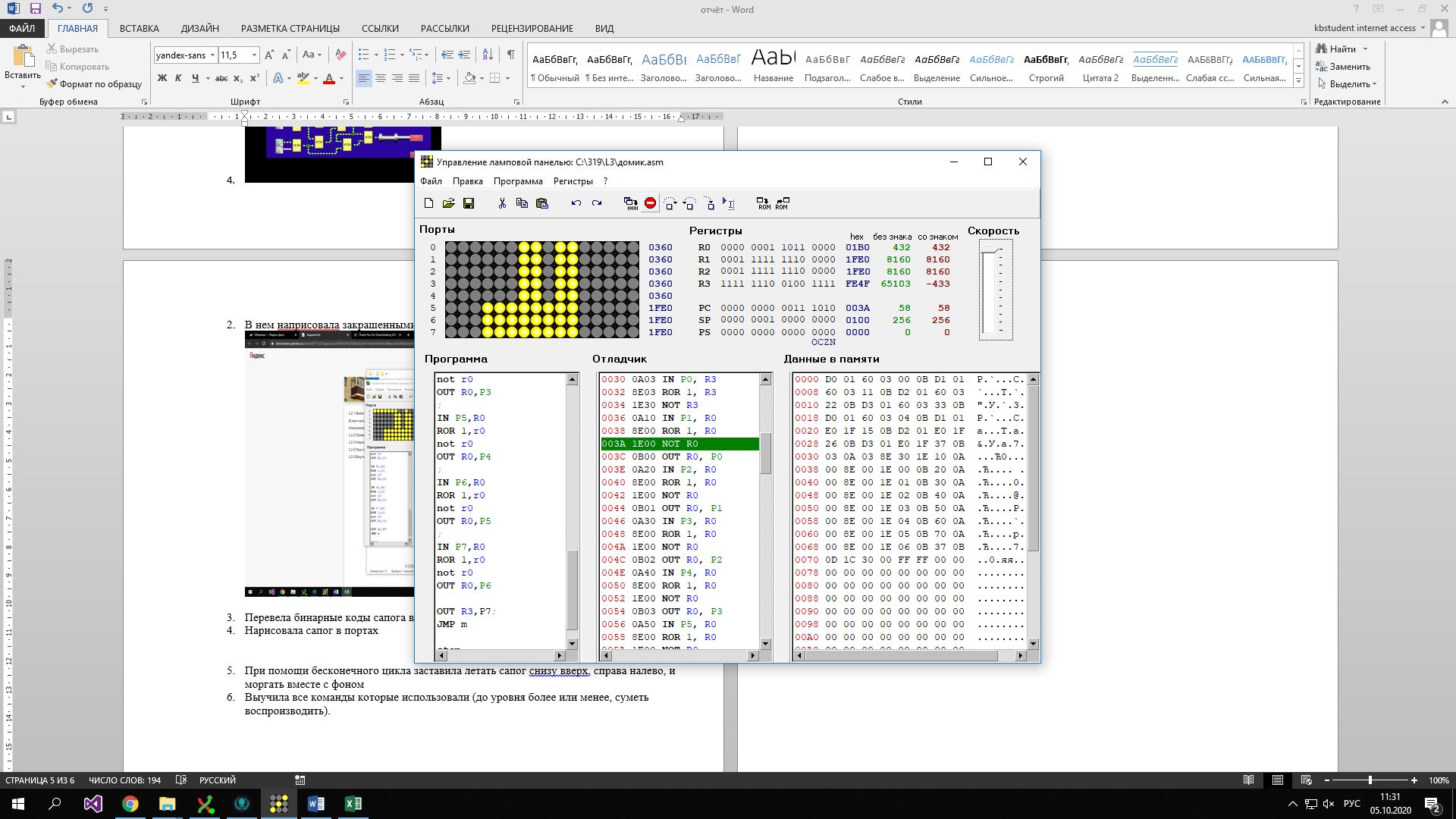
**Задание L1.2 Написать своё имя и фамилию в Brainfuck**

1. Зашла на Яндекс диск в папку Основы Алгоритмизации и Программирования
2. Скачала программу Brainfuck и начала кодить.

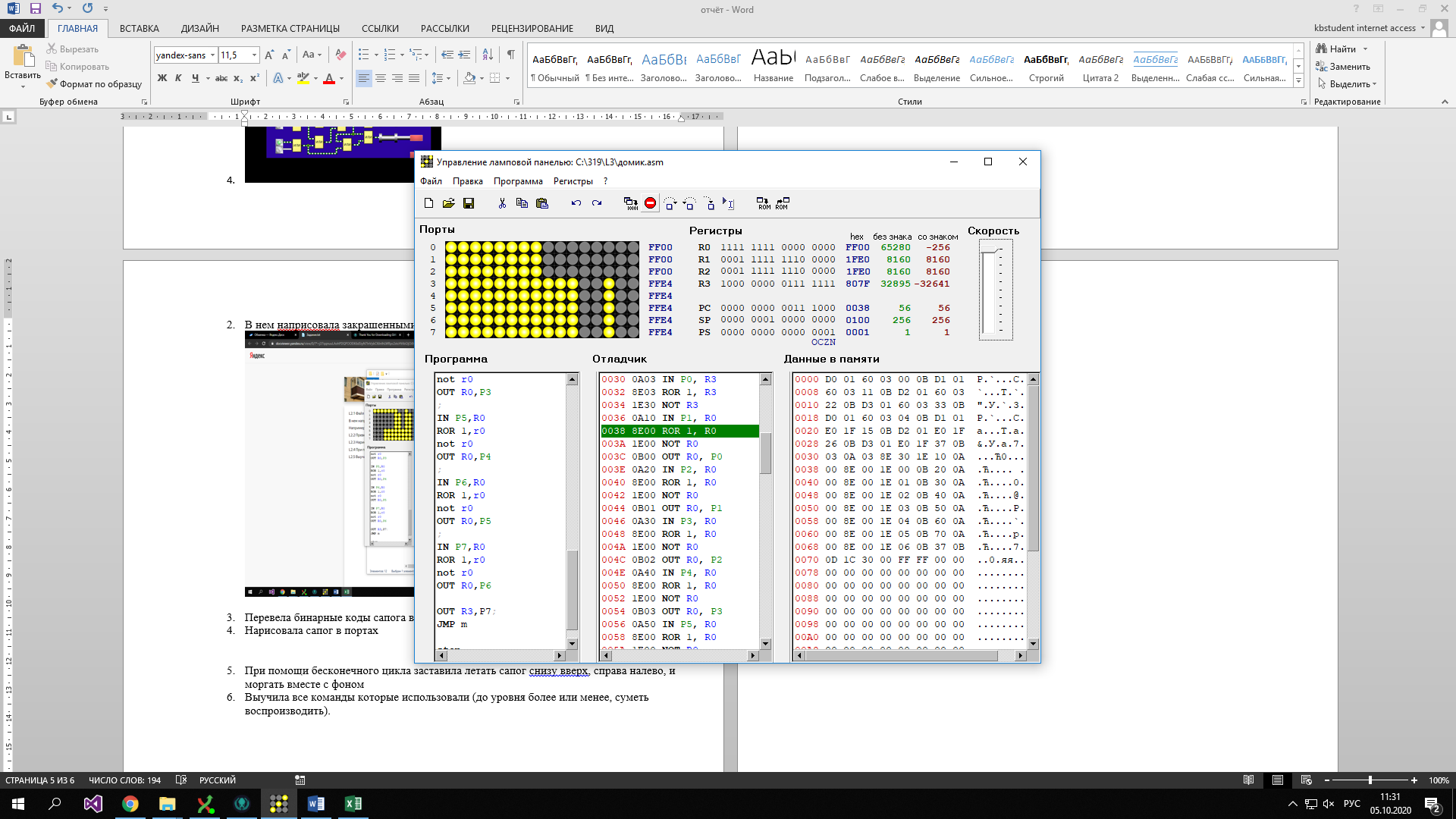
1. Получила результат.

**Задание L2**

1. Файл эксель открыла
2. В нем нарисовала закрашенными квадратиками сапог
3. Перевела бинарные коды сапога в шеснадцатеричные коды.
4. Нарисовала сапог в портах



1. При помощи бесконечного цикла заставила летать сапог снизу вверх, справа налево, и моргать вместе с фоном

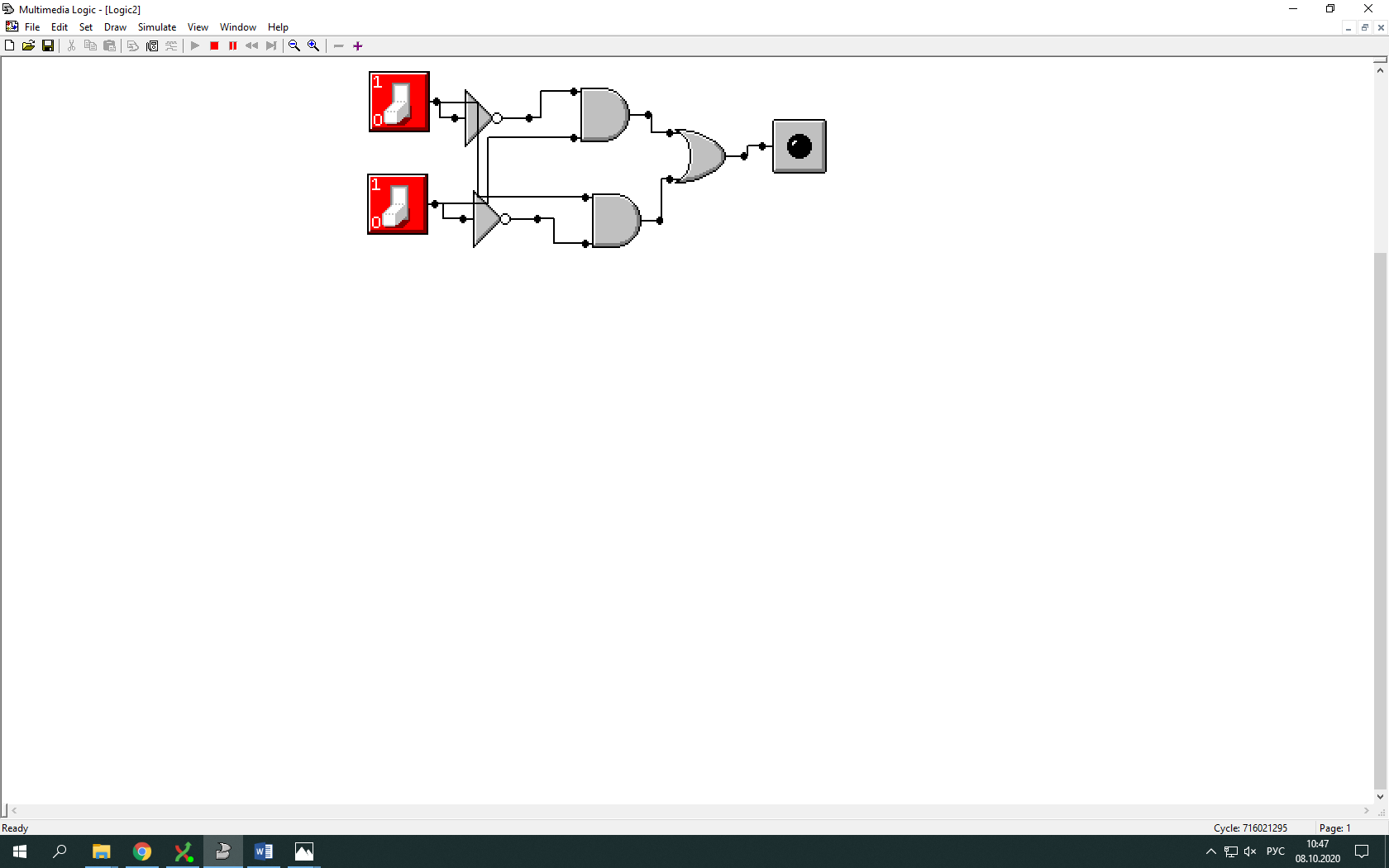


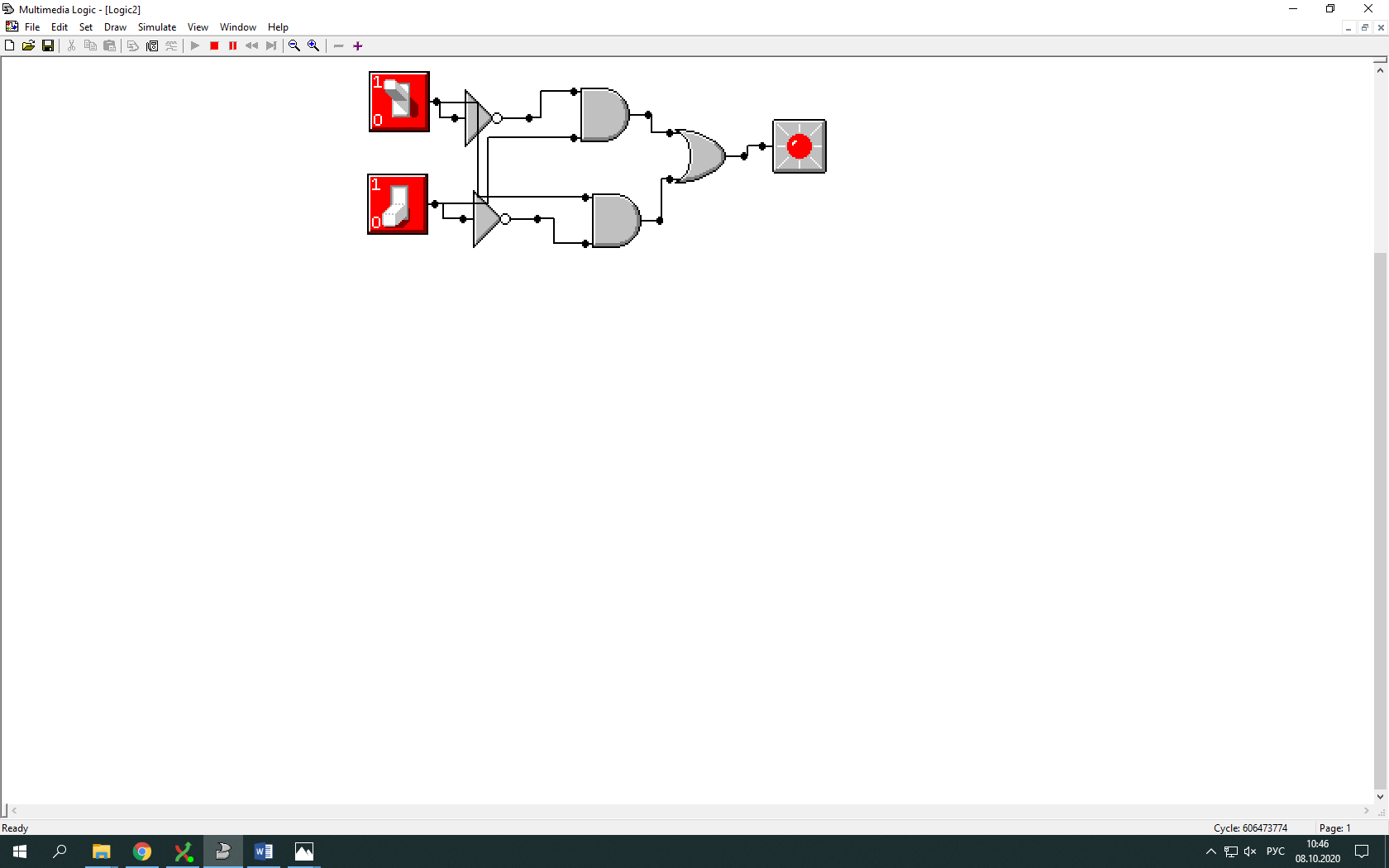
1. Выучила все команды которые использовали (до уровня более или менее, суметь воспроизводить).

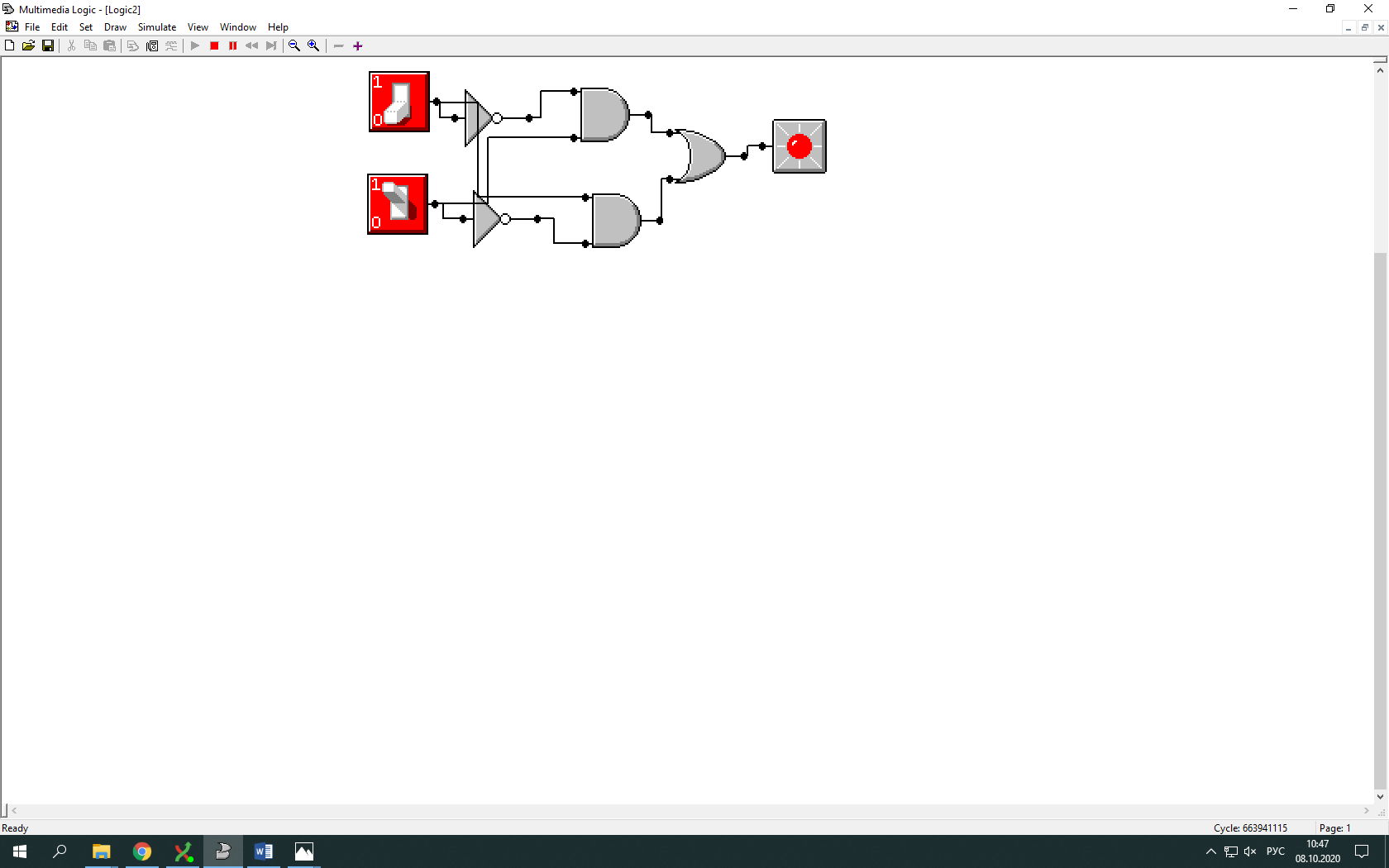
* NOT Rx - логические «НЕ» (инверсия)
* ROR n,Rx - циклический сдвиг вправо
* IN Py,Rx - ввод из порта Py в регистр Rx
* OUT Rx,Py - вывод из регистра Rx в порт Py
* MOV SP,Rx - пересылка Rx:=SP
* MOV Rx,SP - пересылка SP:=Rx

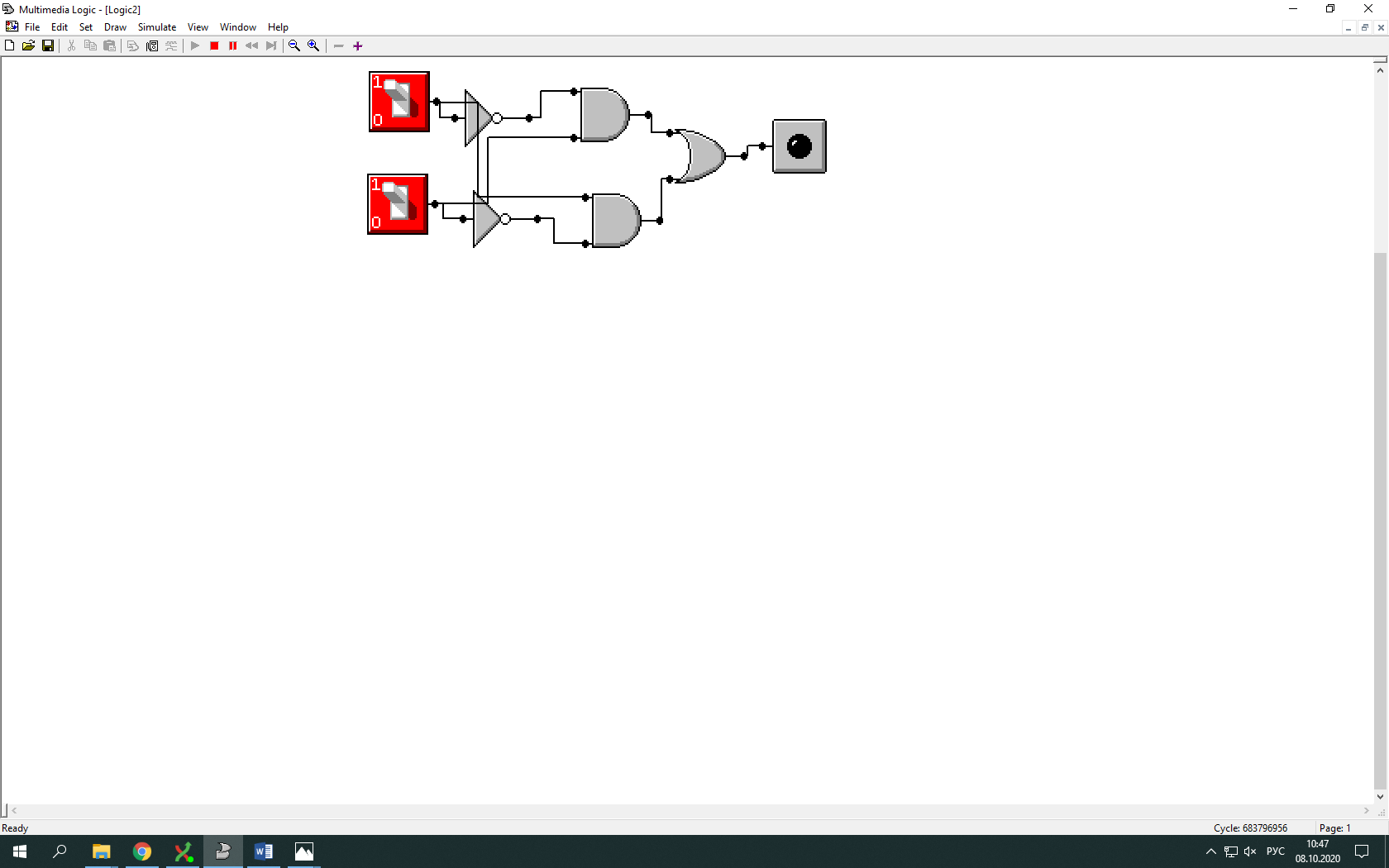
**Задание L3**

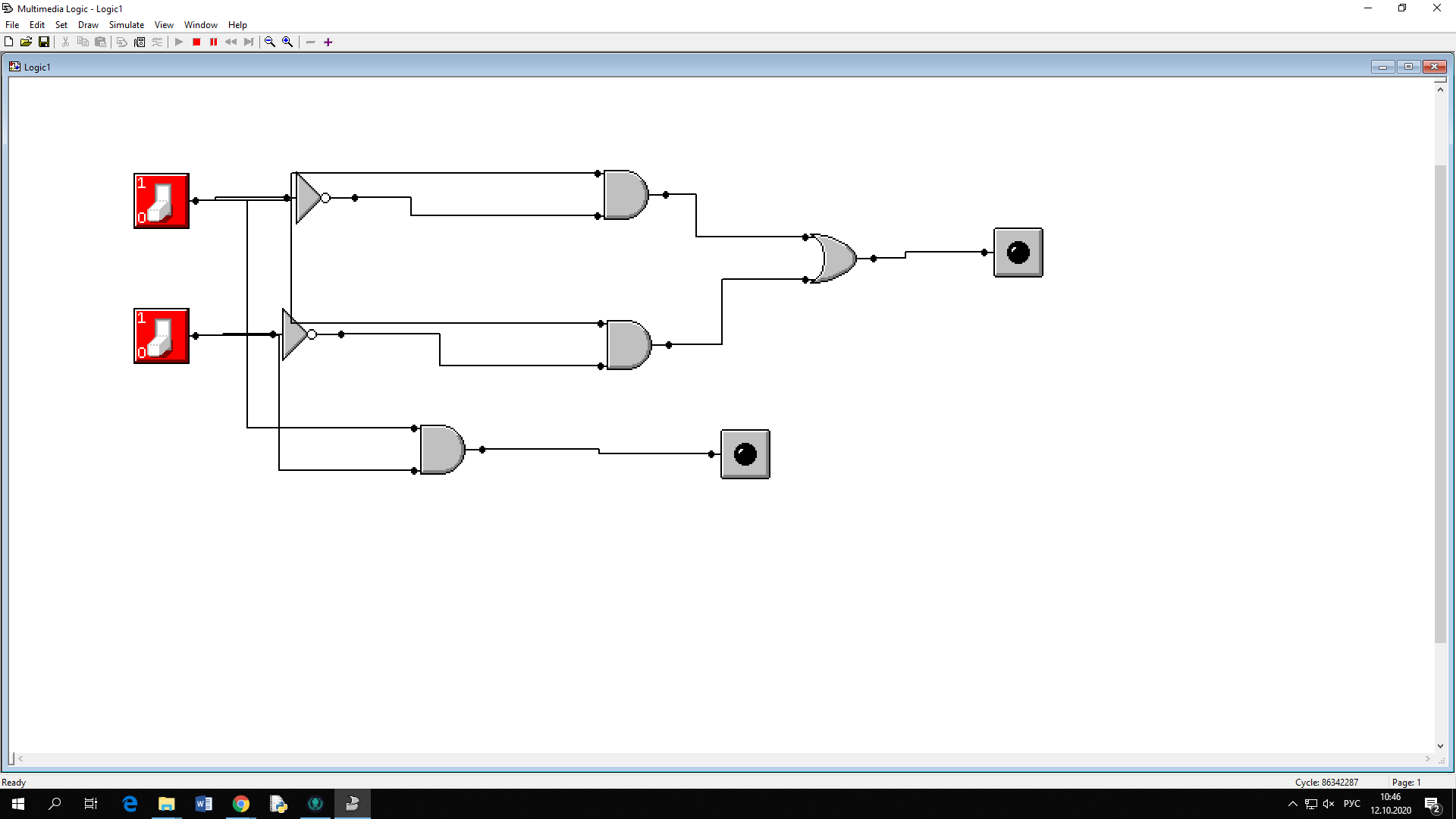
1. Скачала Mulimedia Logic.
2. Сделала ламповую схему.

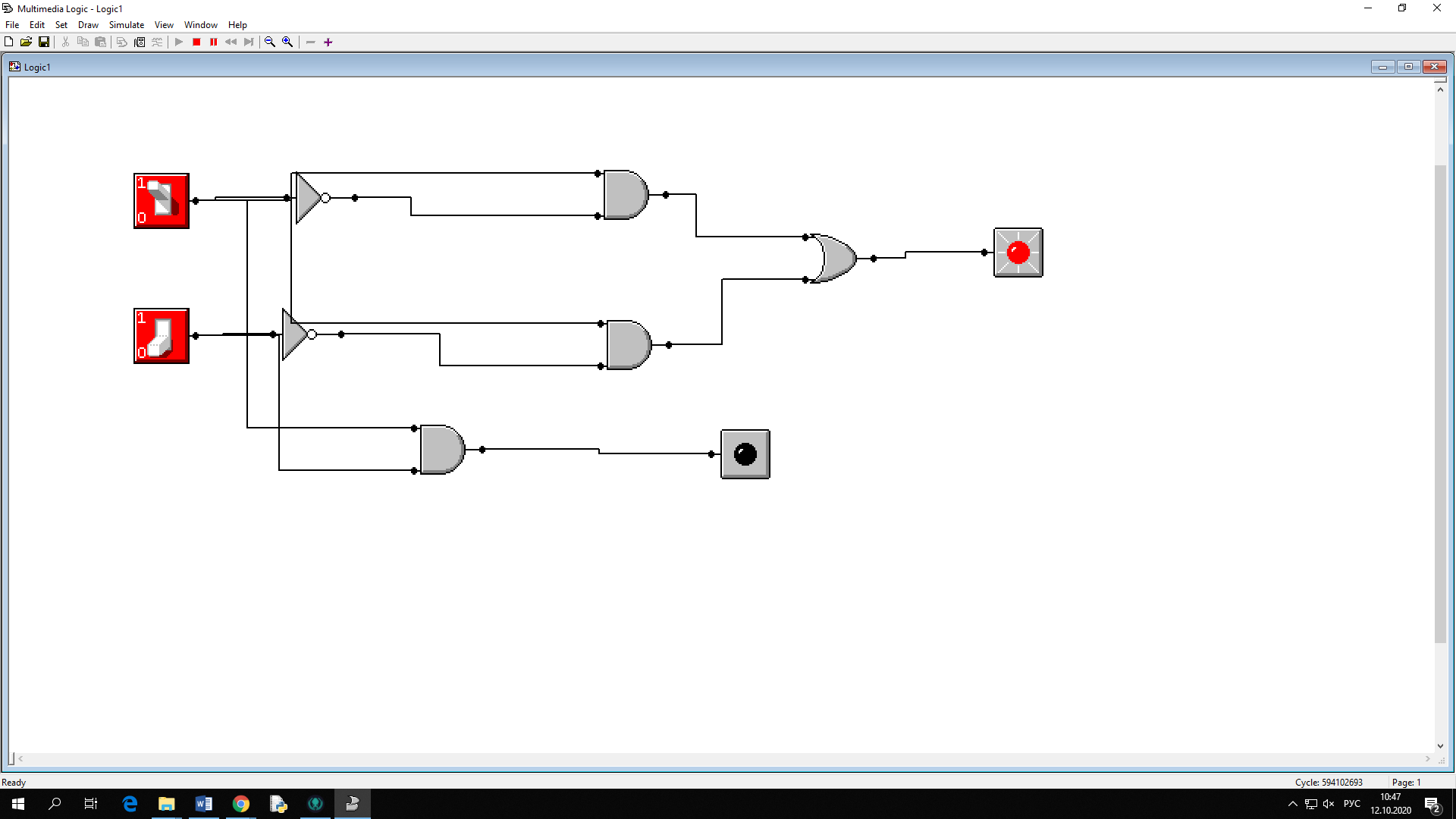


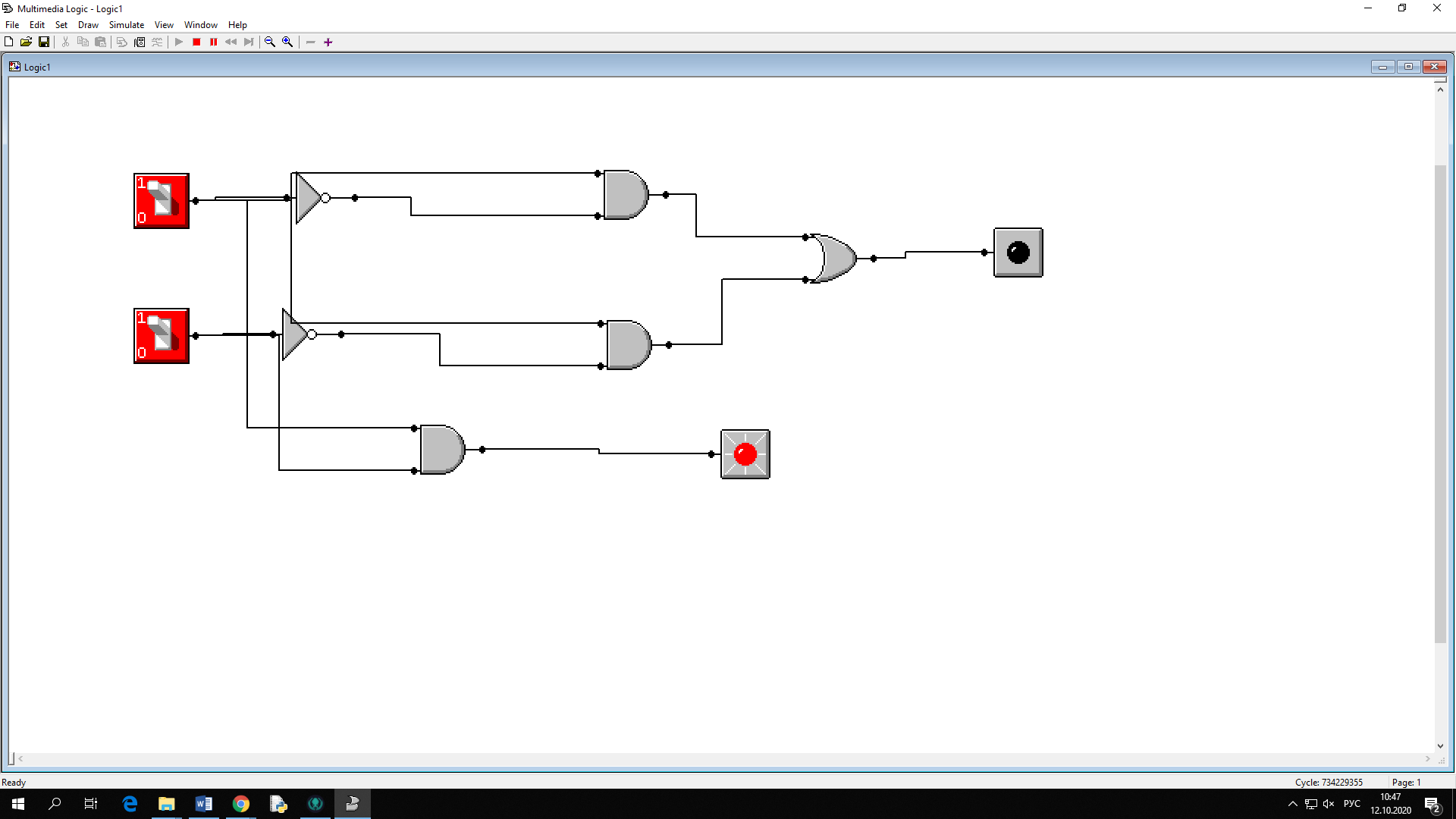


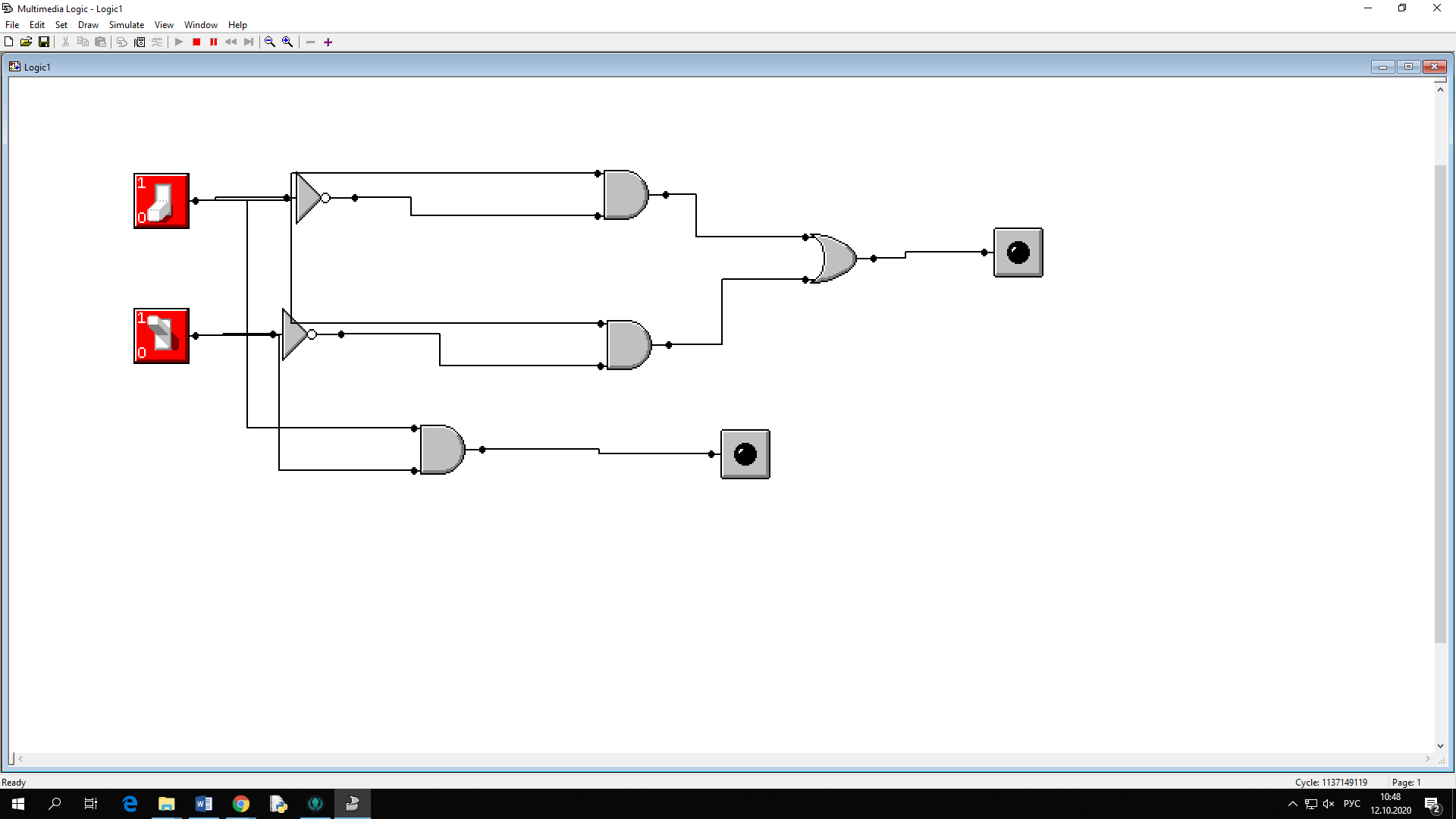


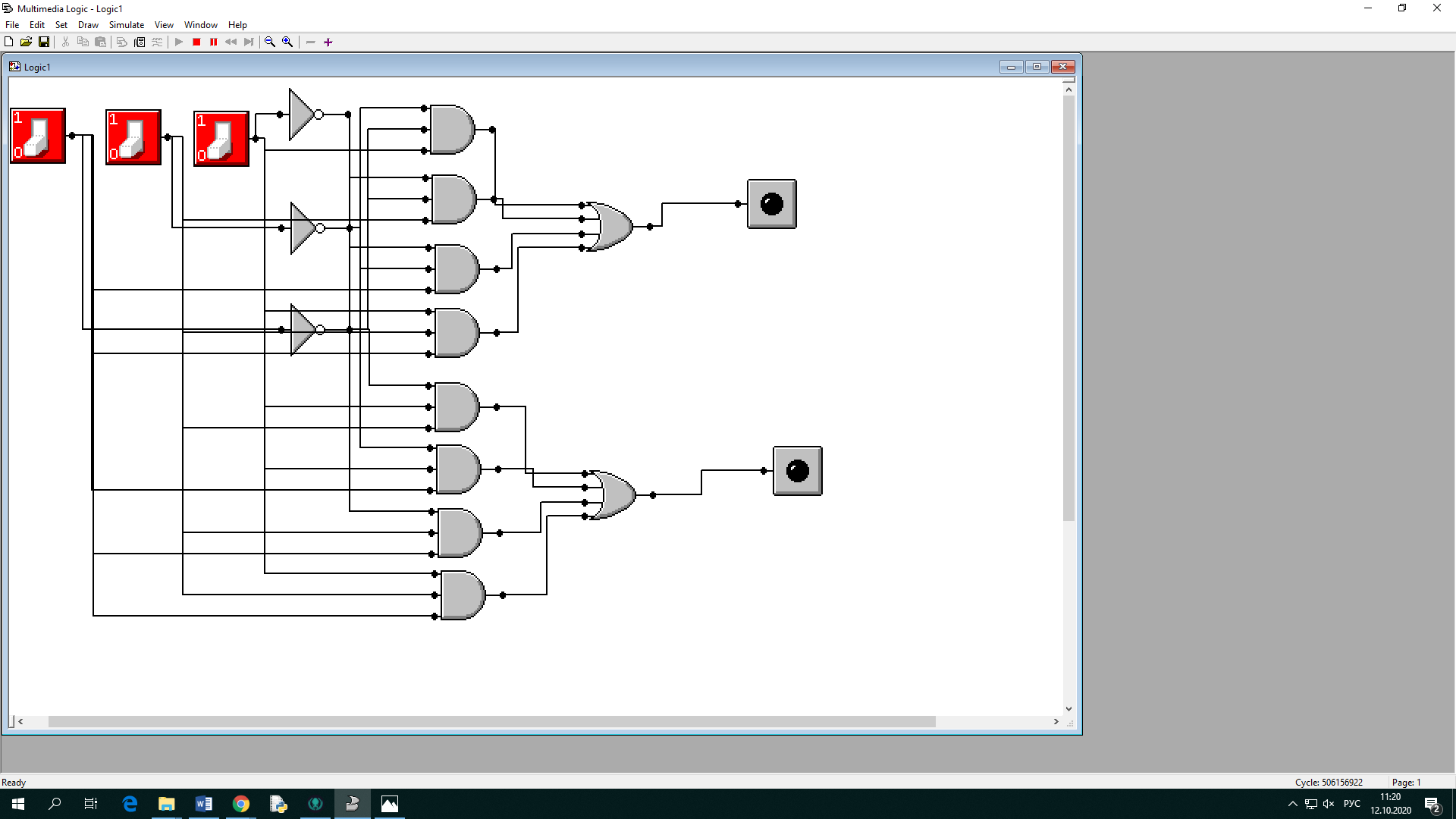


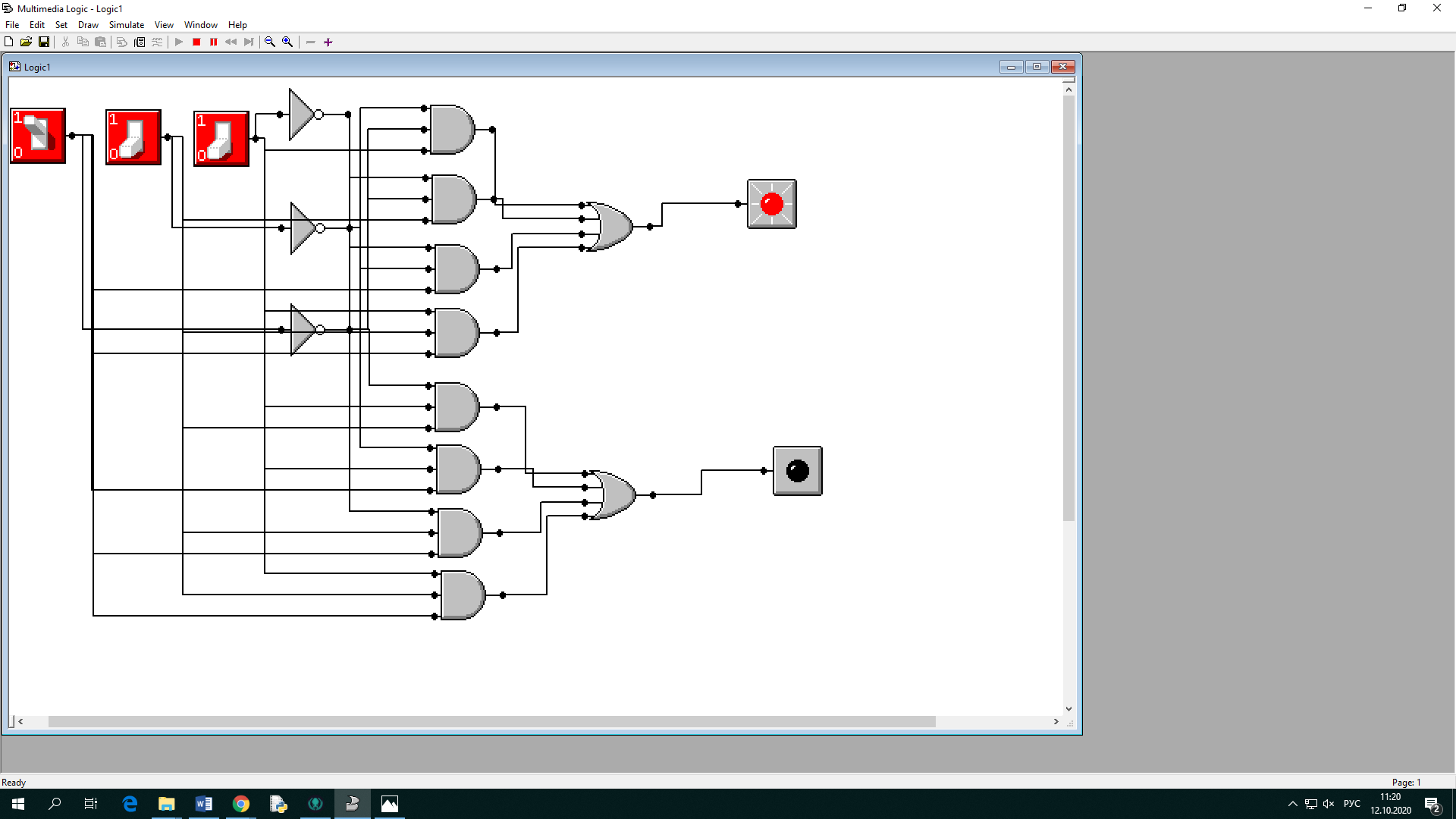


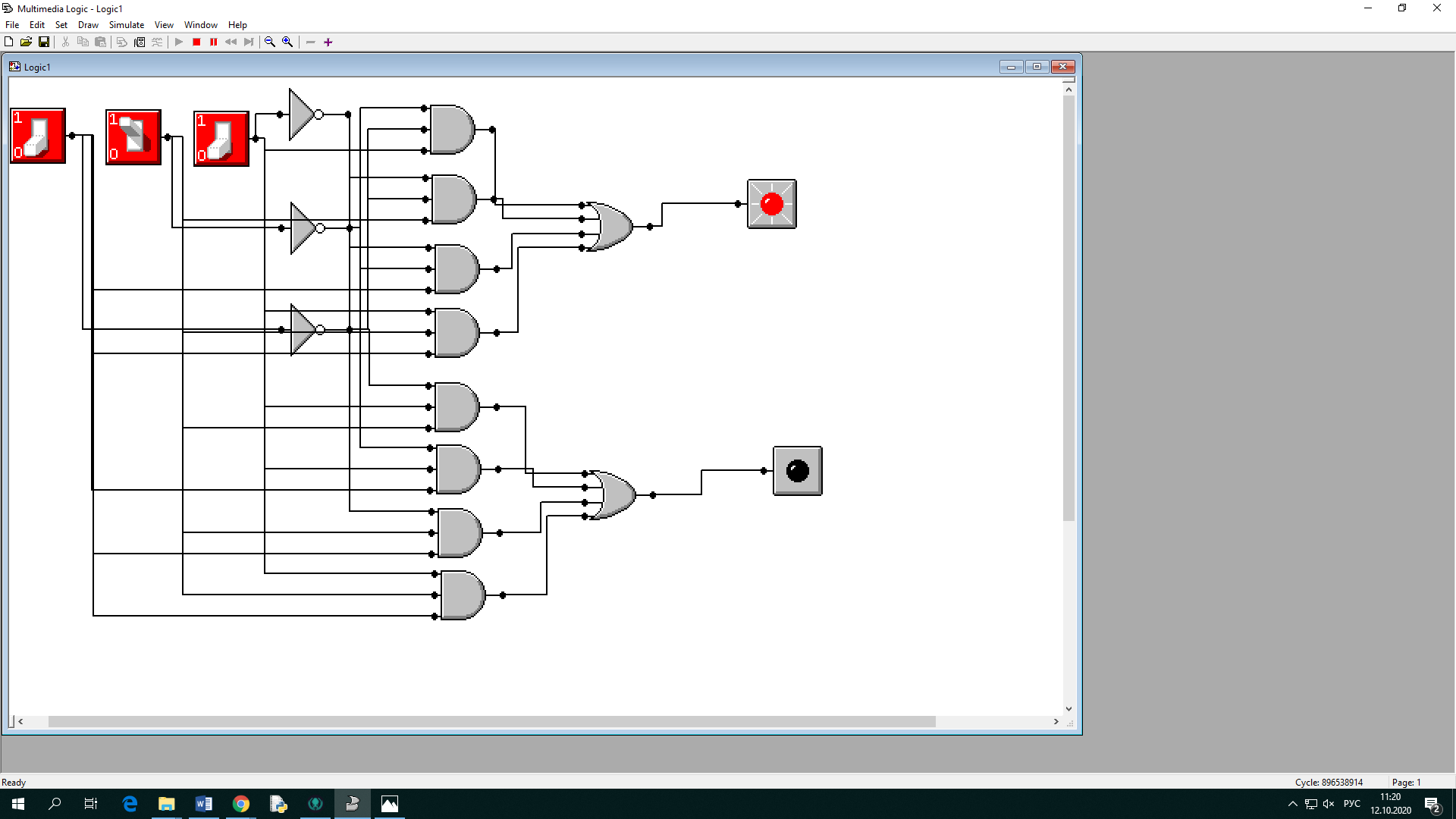


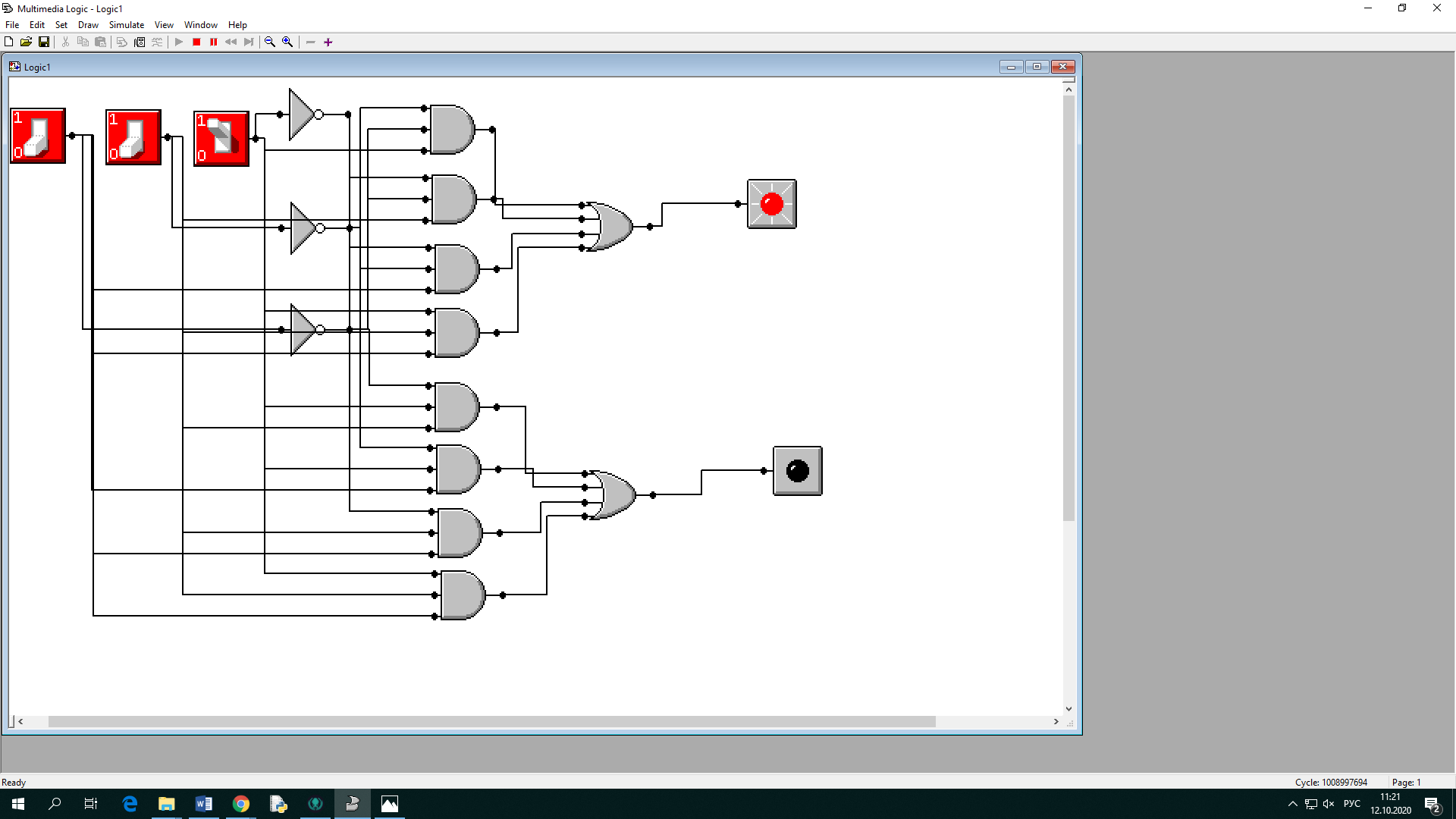


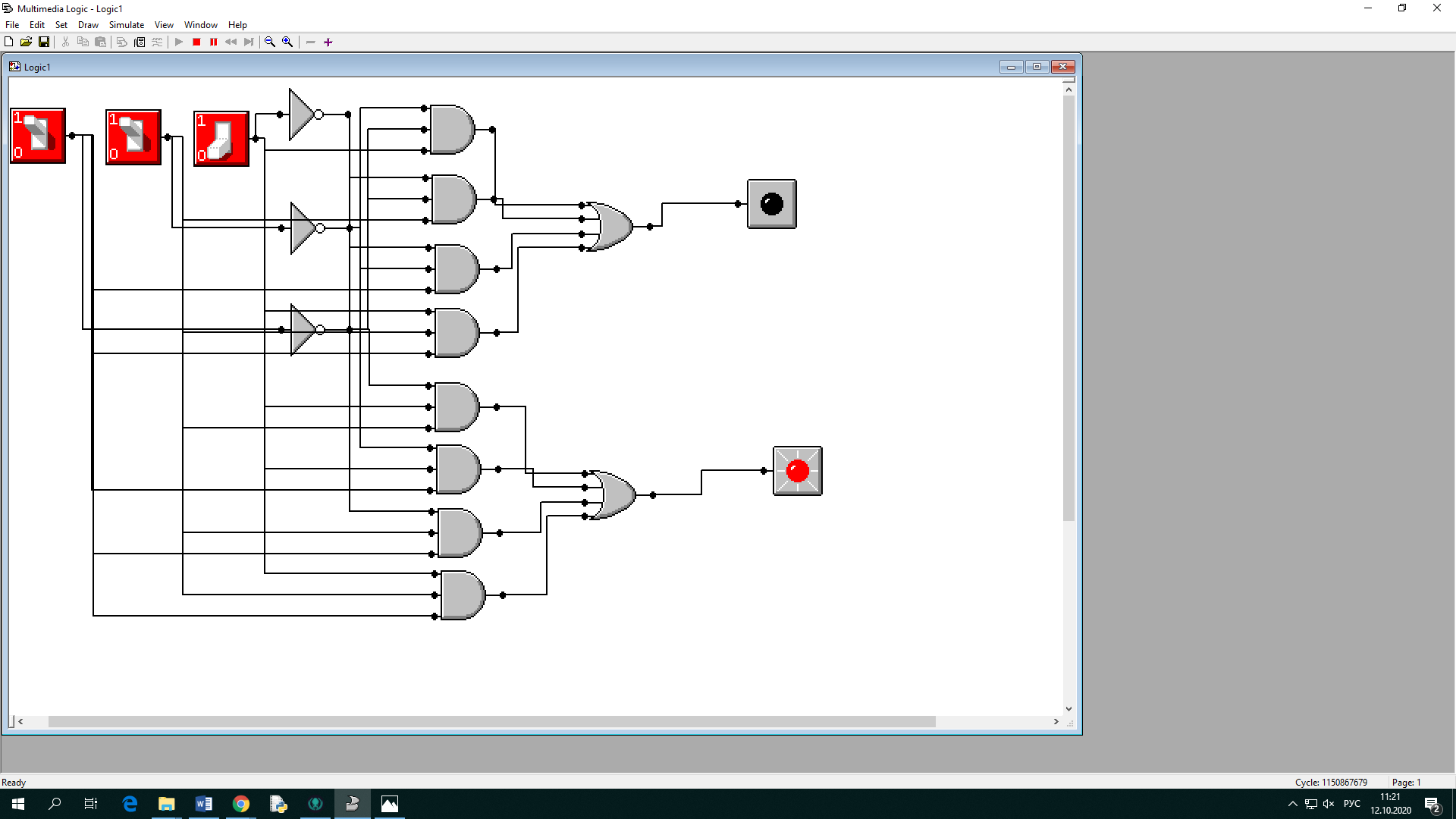


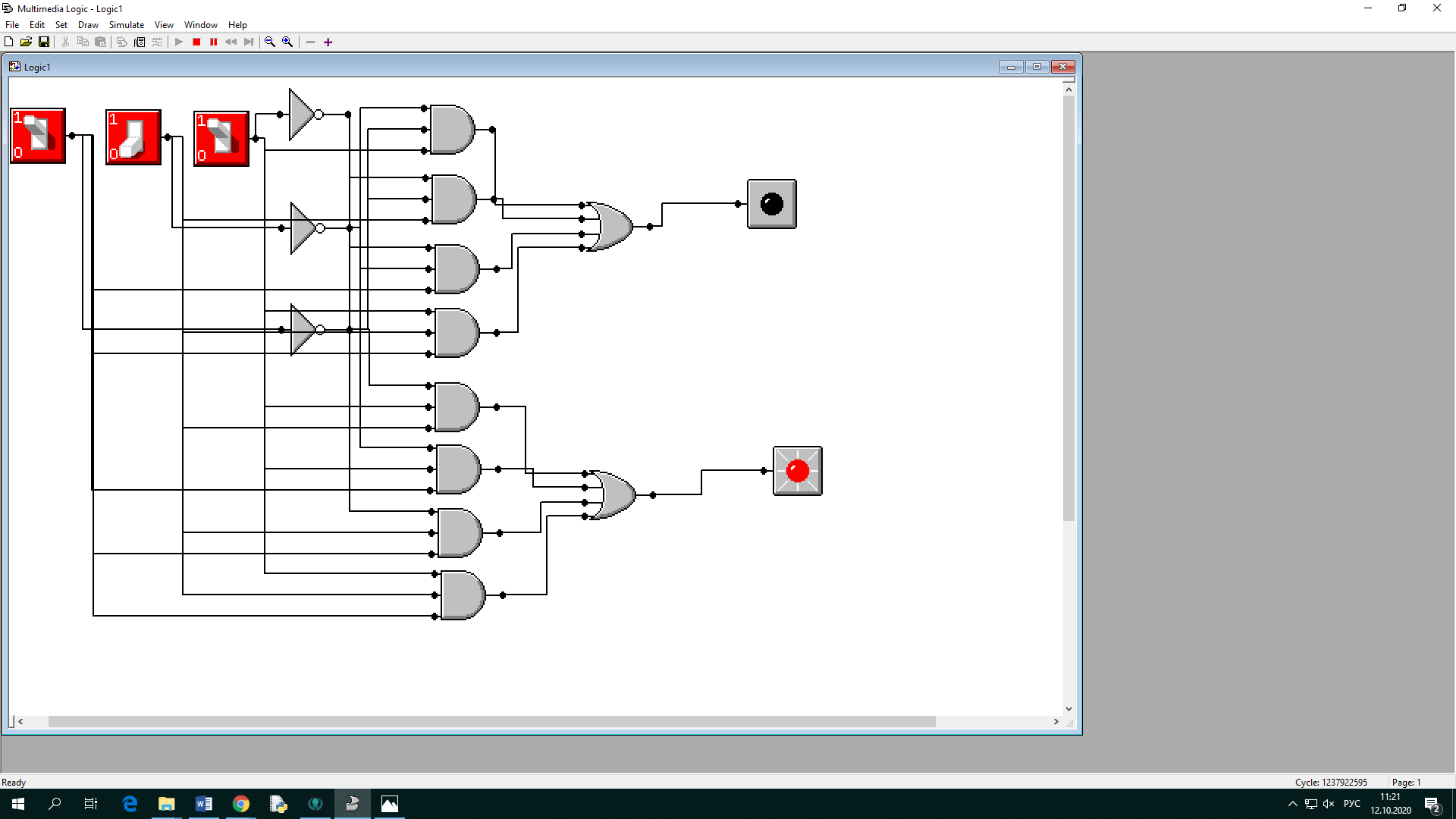


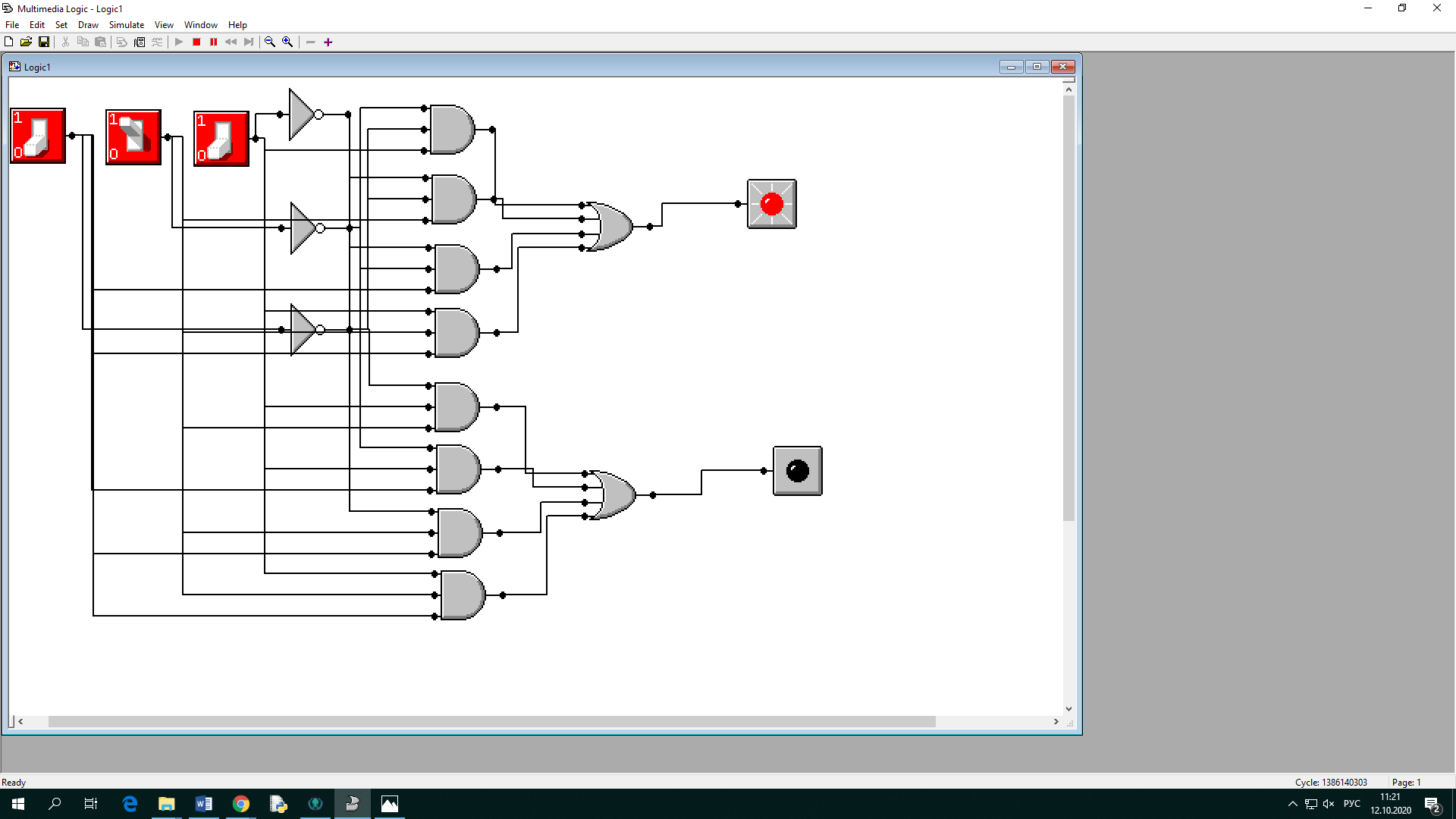


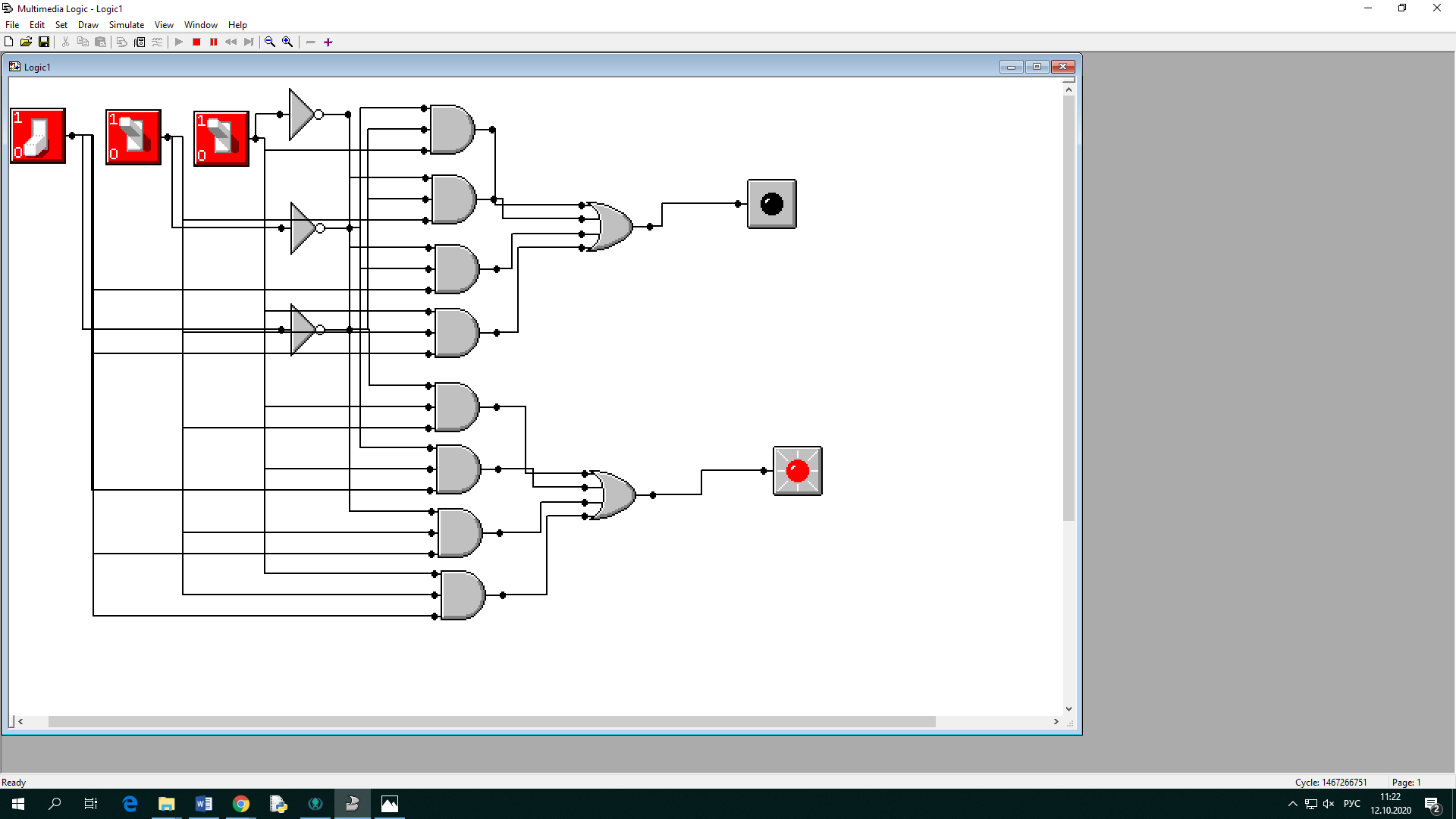


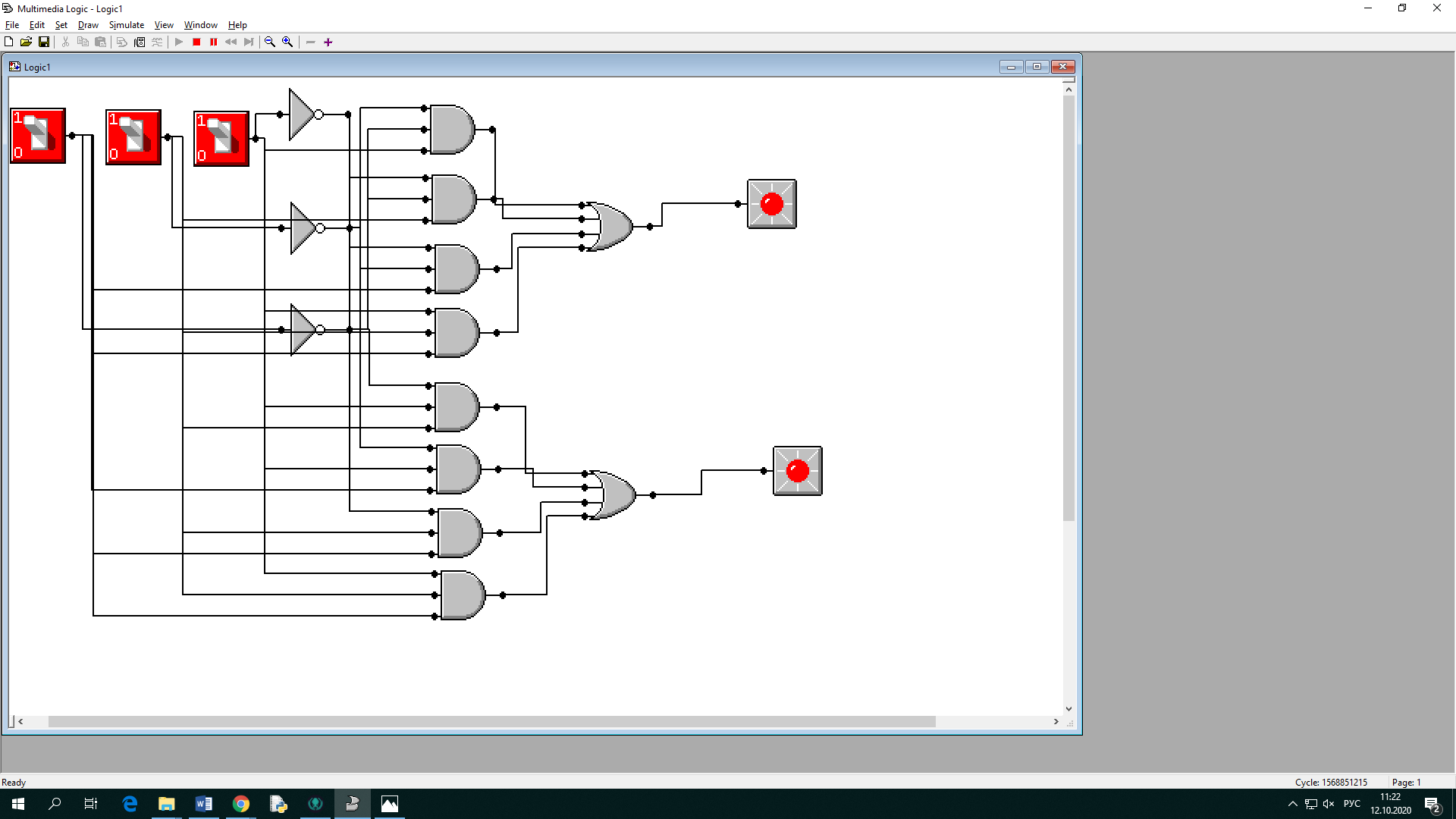
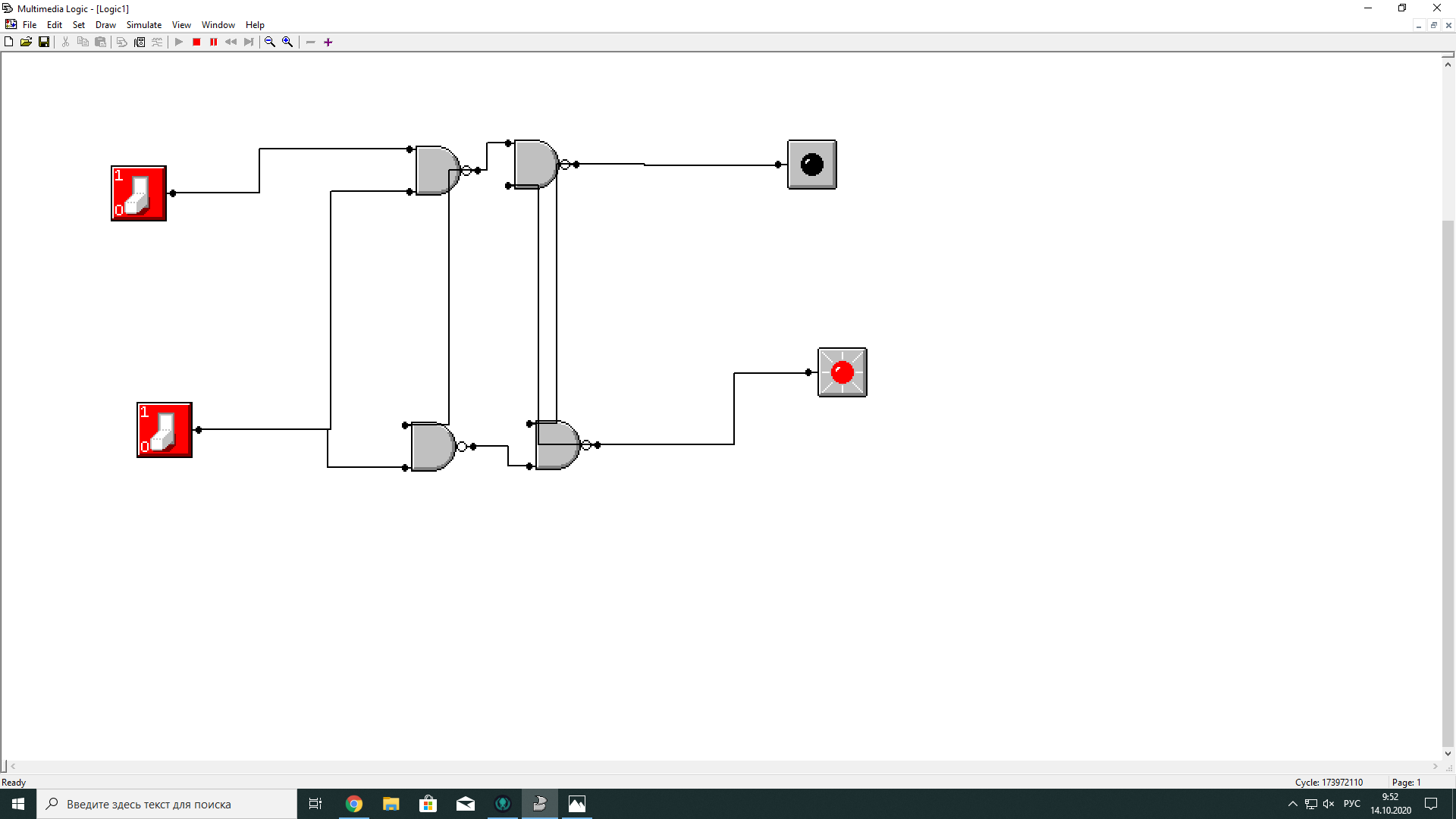
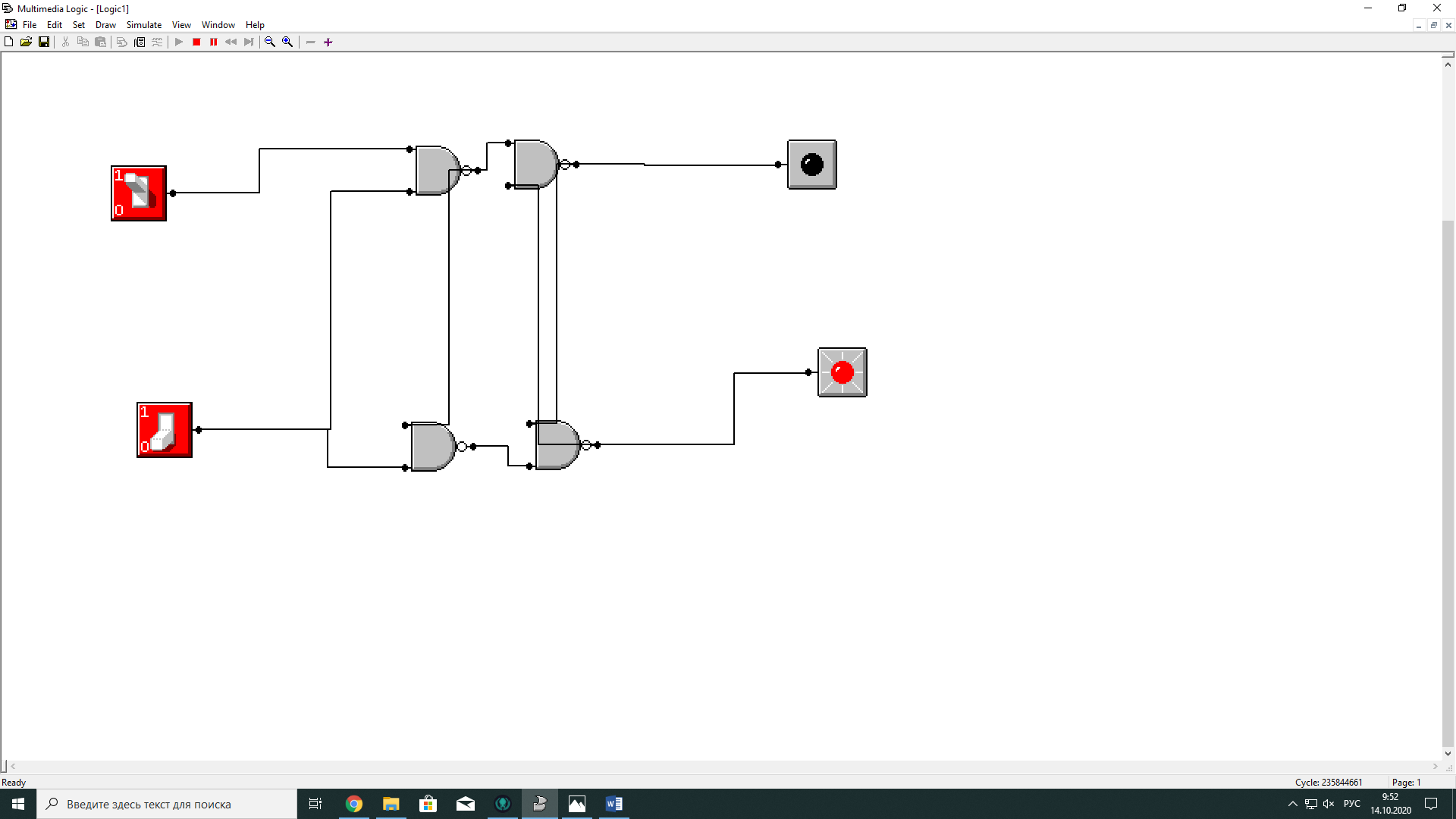


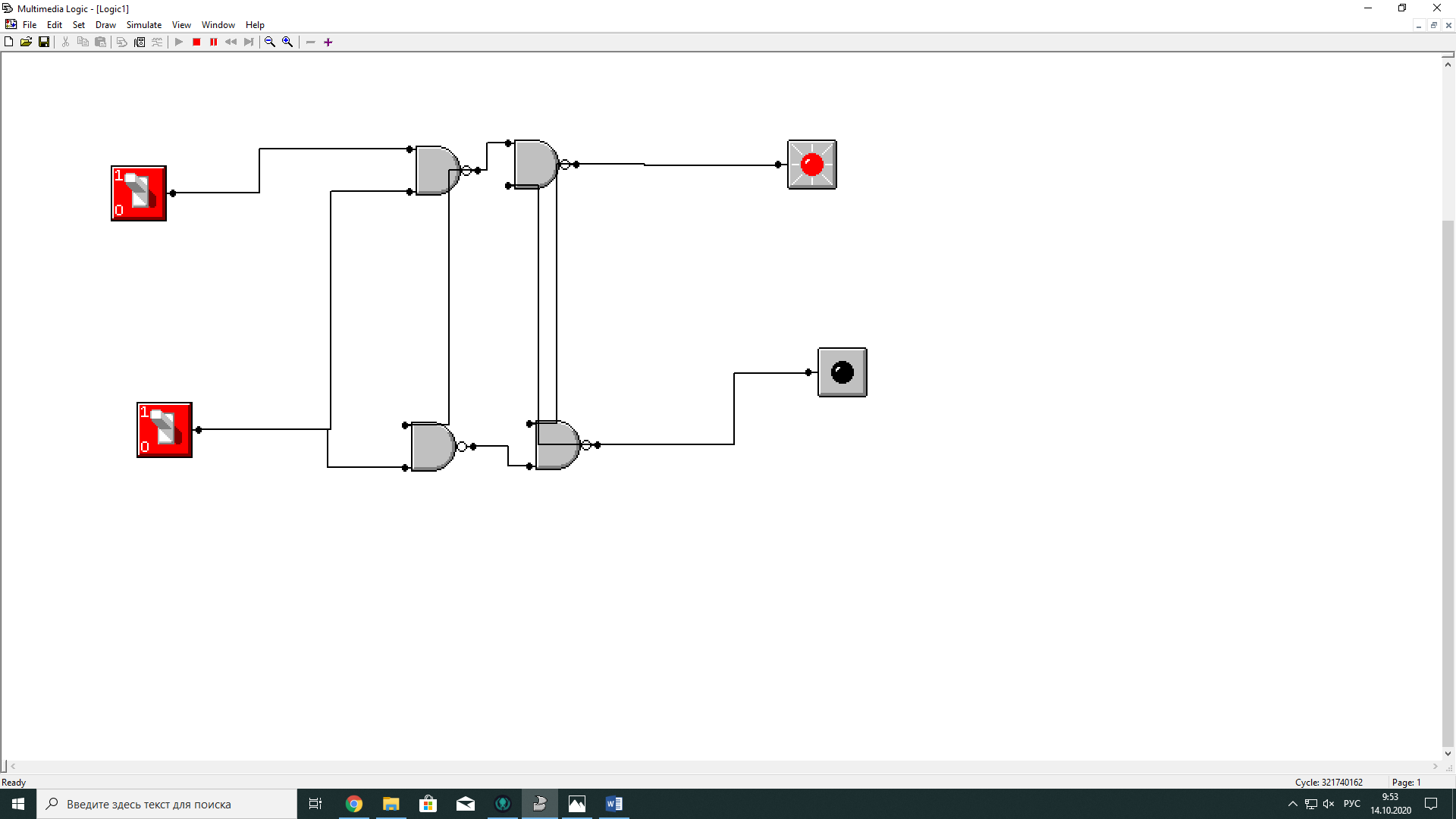


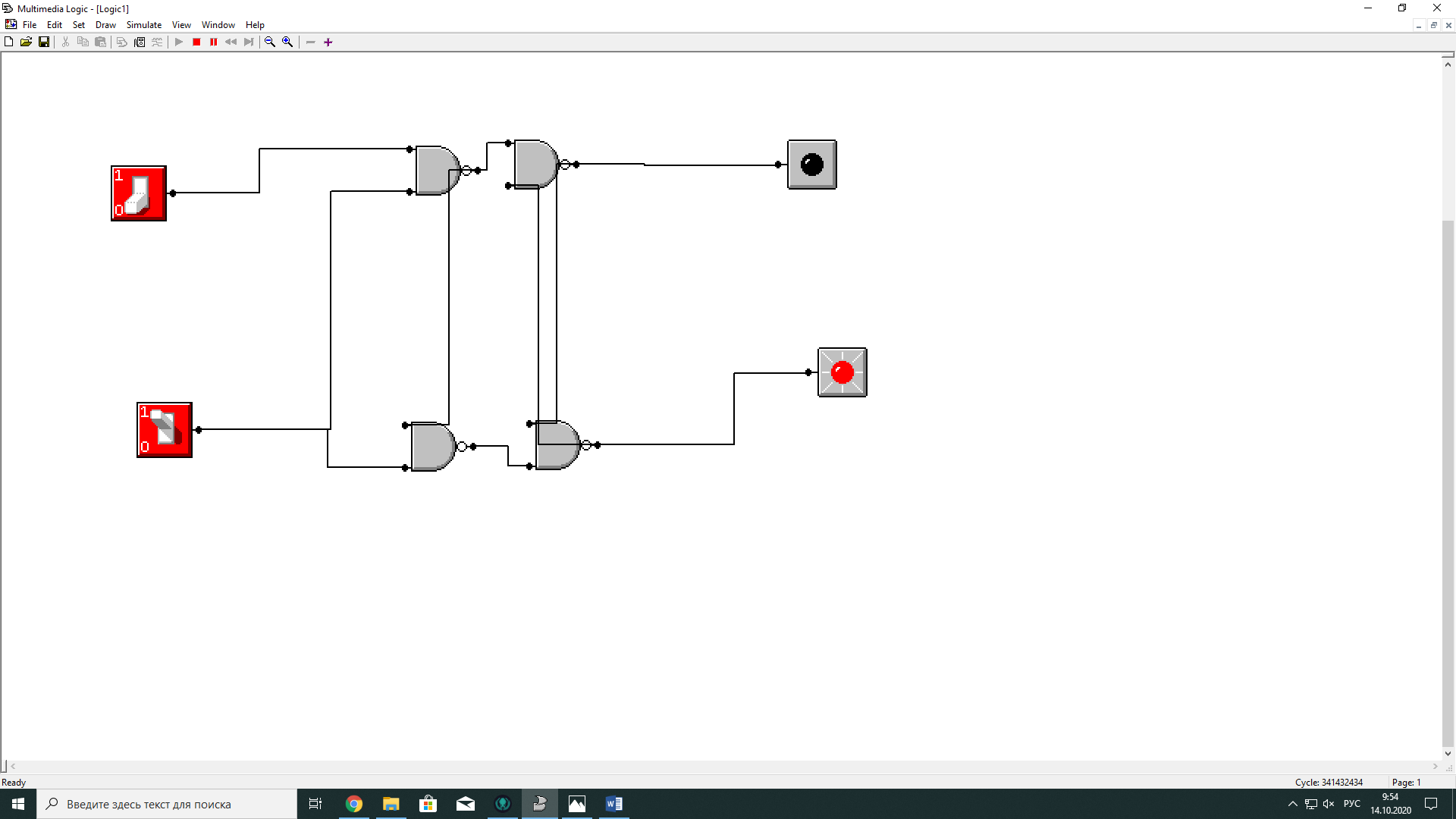


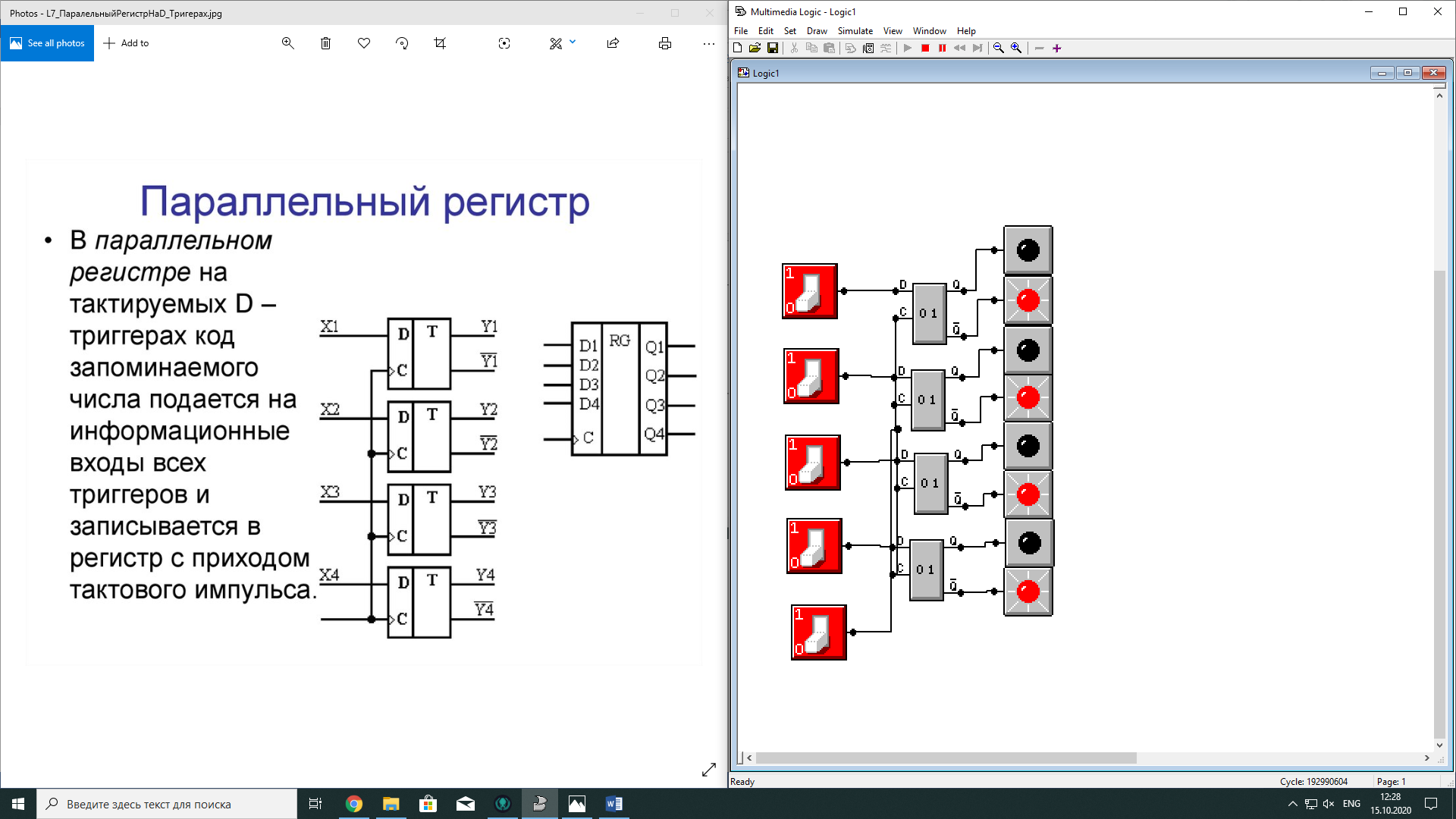


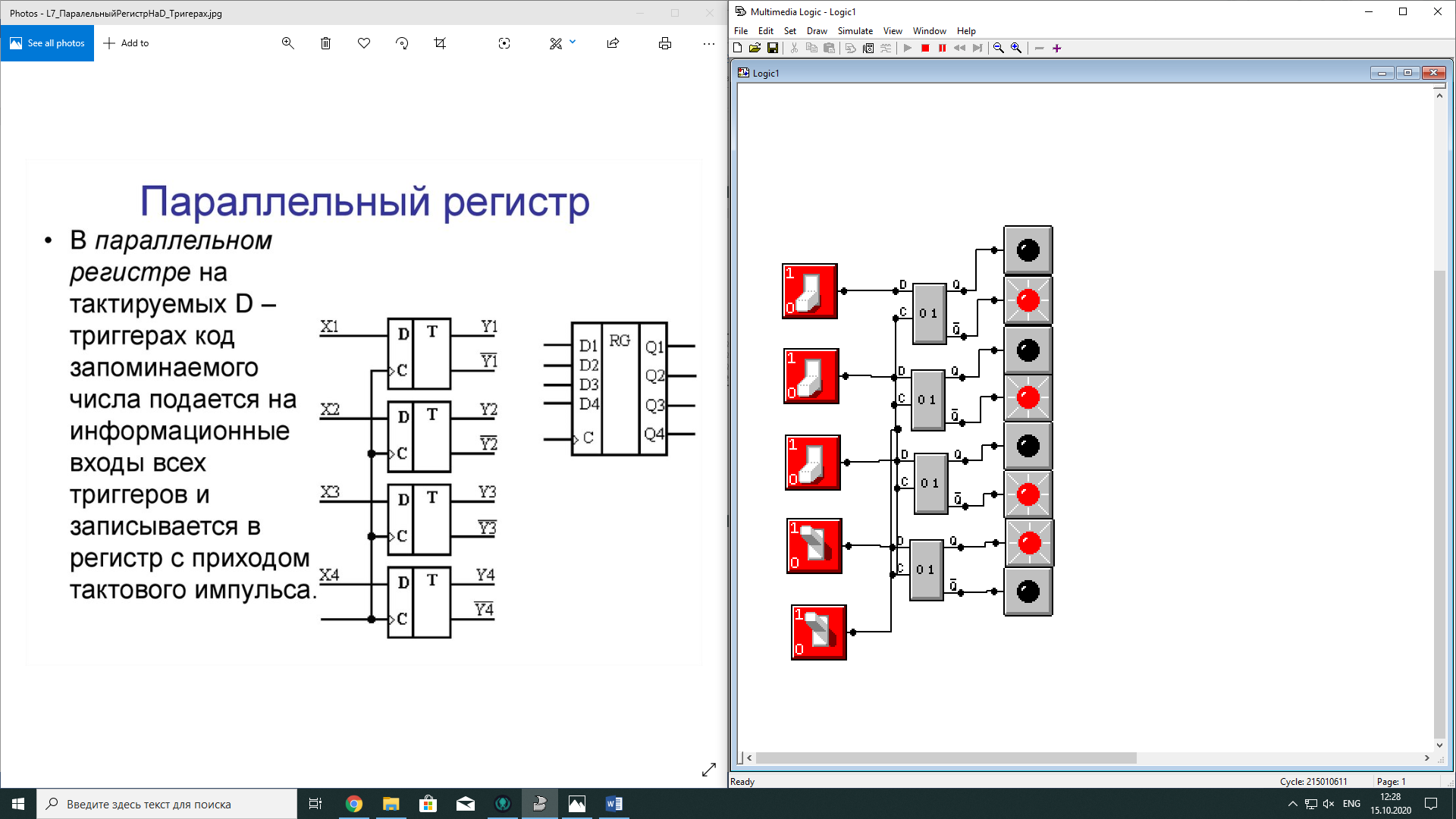


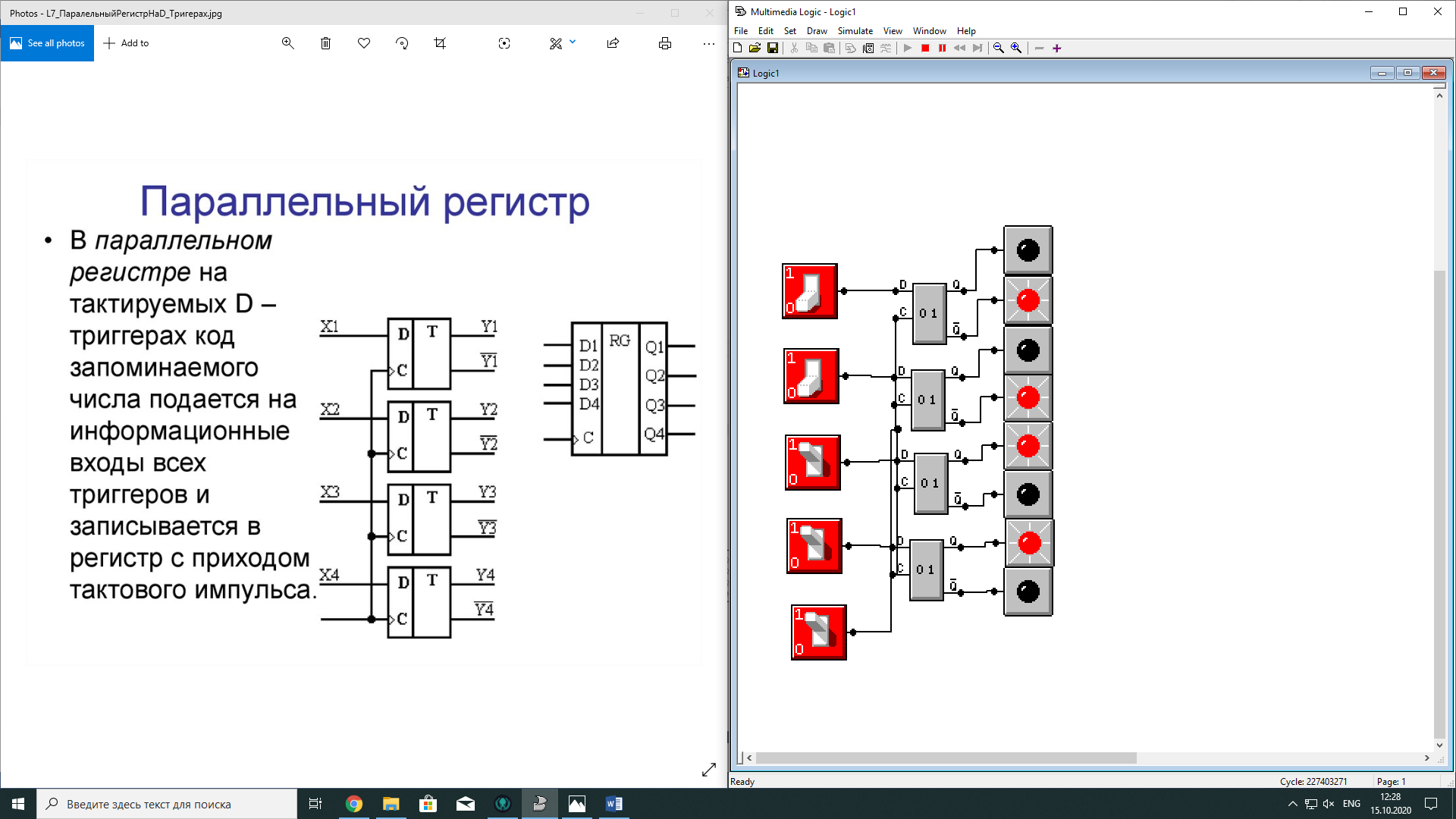
  


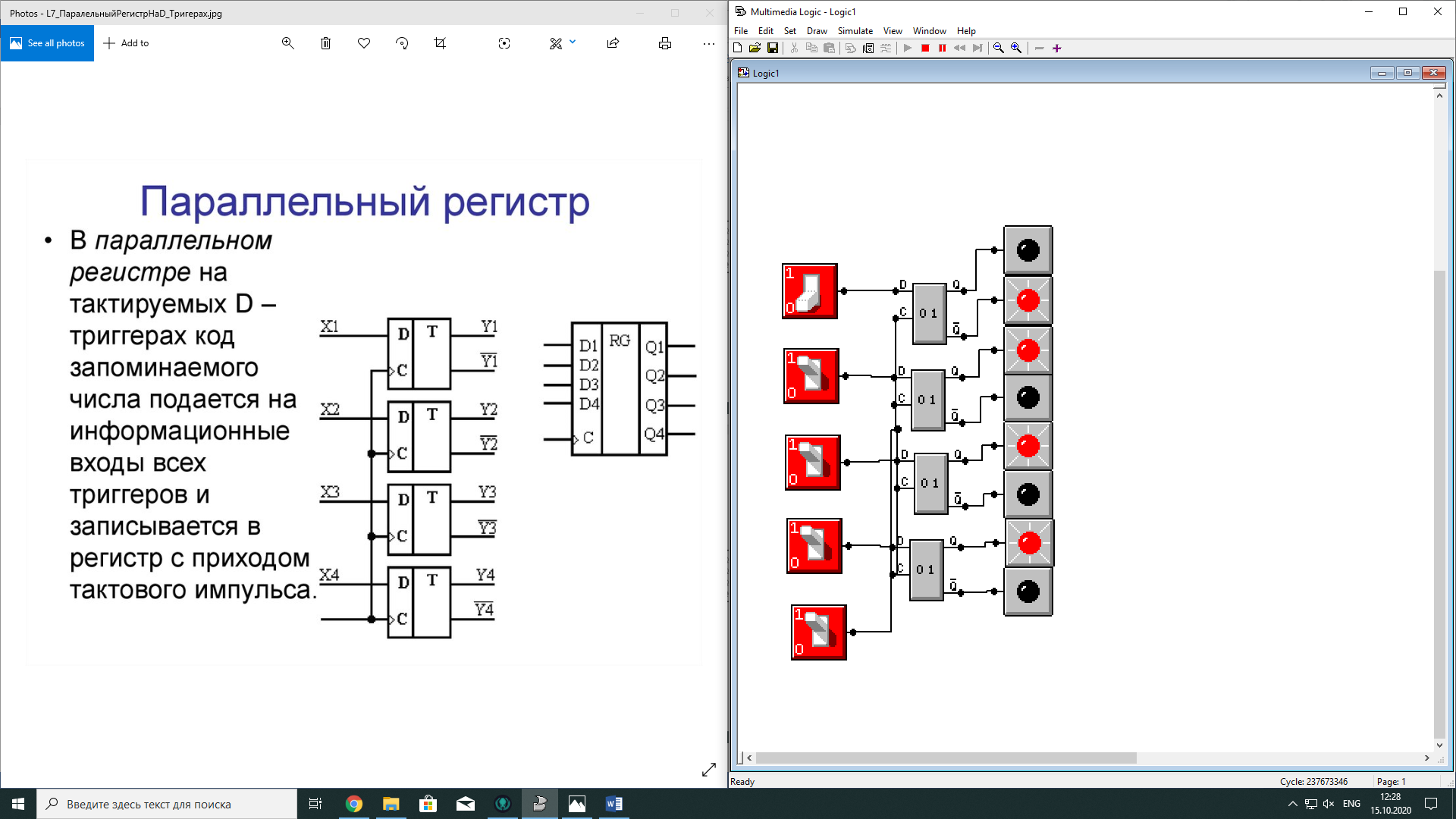




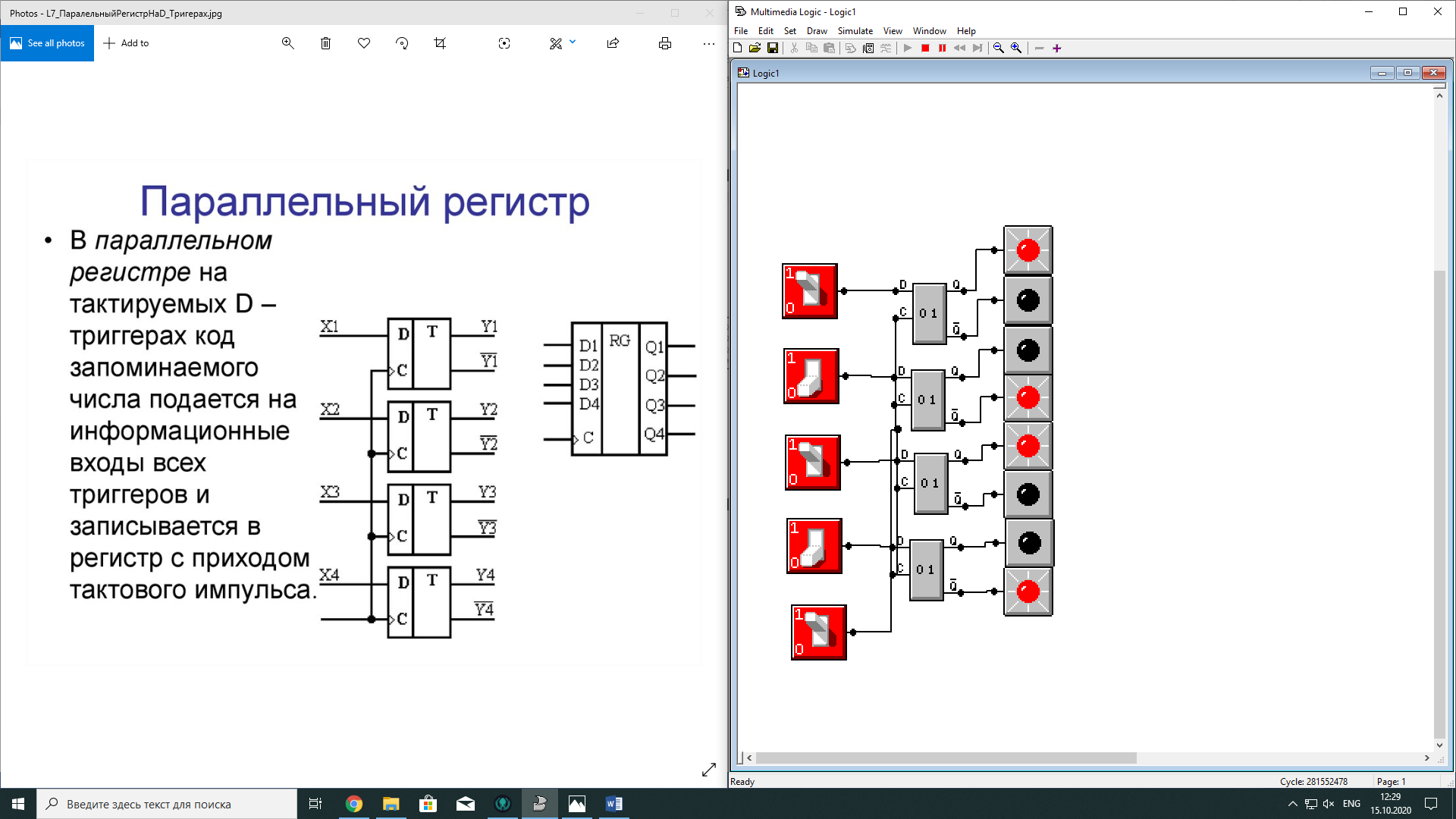












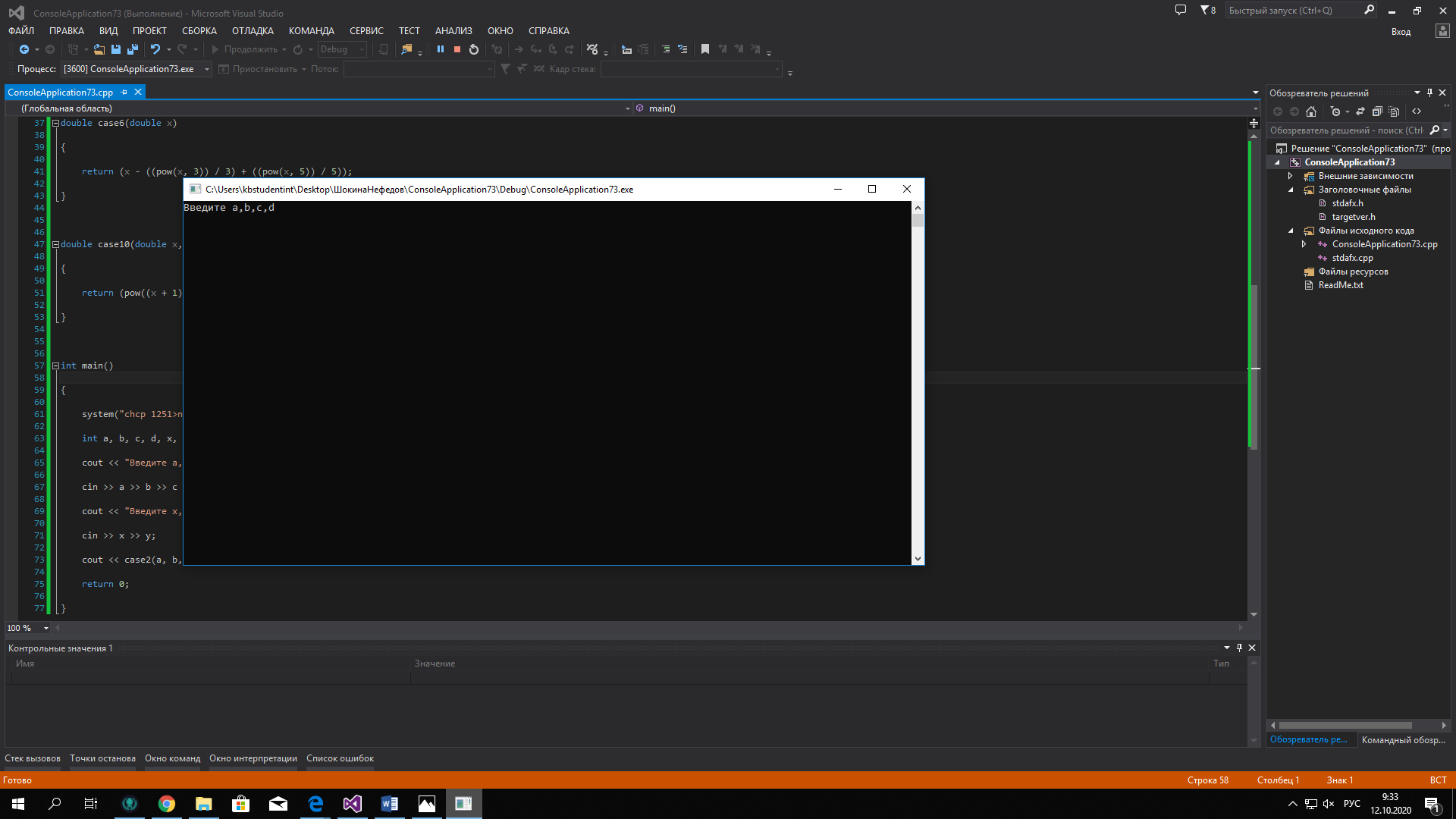
1. Закоммитила на гитхаб отчет и проект
2. Выгрузила на гитхаб

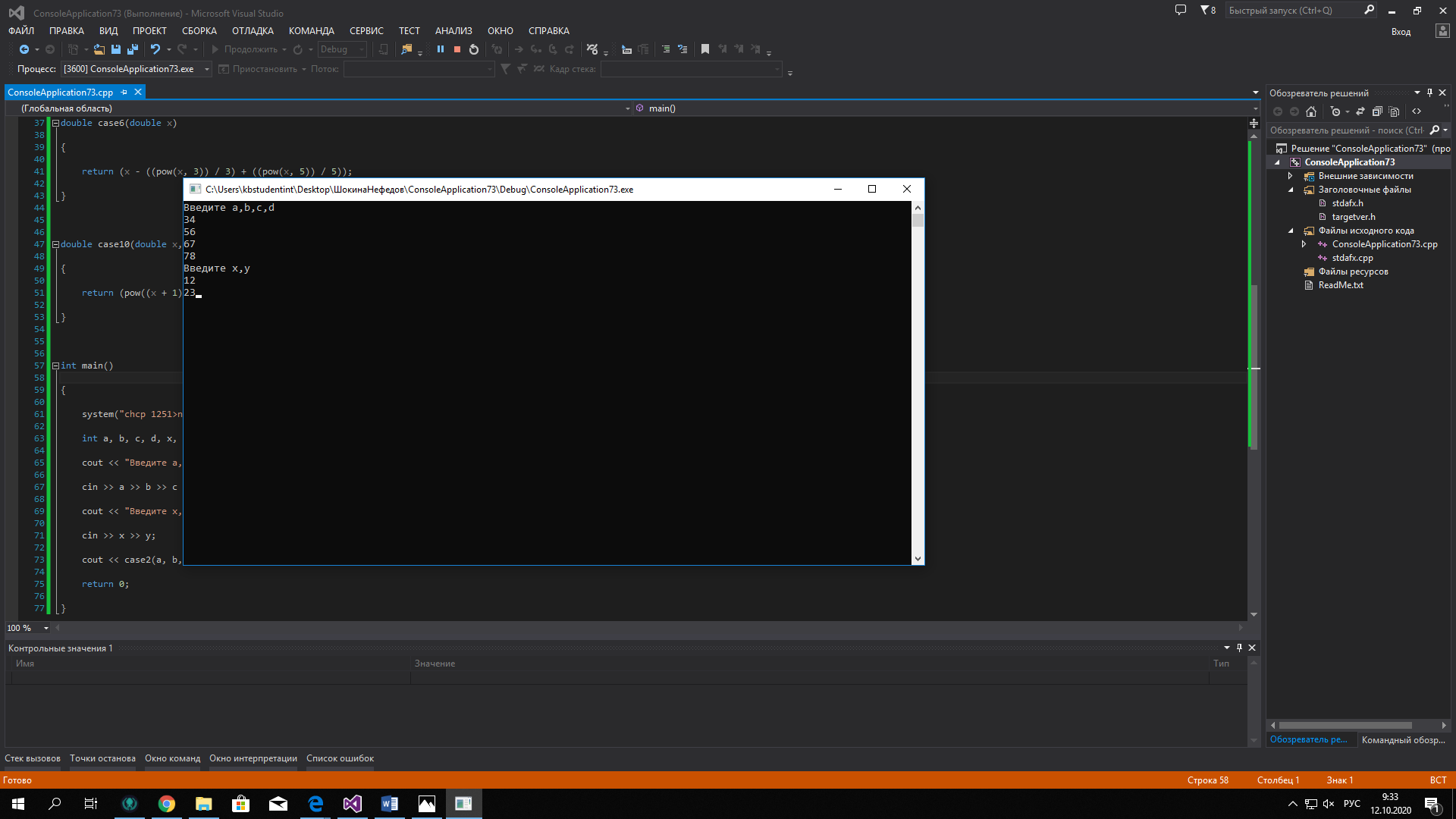
**Задание L3.**

1. Открыла Visual Studio
2. Зашла на Яндекс диск в папку Основы Алгоритмизации и Программирования.
3. Скачала папку L3.
4. Написала код для решения примеров.

|  |  |
| --- | --- |
| 1. #include "stdafx.h" 2. #include <cmath> 3. #include <cstdlib> 4. #include <iostream> 5. using namespace std; 6. double case2(double a, double b, double c, double d) 7. { 8. return ((a / c) \* (b / d)) - ((a \* b - c) / (c \* d)); 9. } | C:\Users\kbstudentint\Pictures\Screenshots\Снимок экрана (7).png |
| double case4(double x, double y)  {  return ((x + y) / (y + 1)) - ((x \* y - 12) / (34 + x));  } | C:\Users\kbstudentint\Pictures\Screenshots\Снимок экрана (8).png |
| double case6(double x)  {  return (x - ((pow(x, 3)) / 3) + ((pow(x, 5)) / 5));  } | C:\Users\kbstudentint\Pictures\Screenshots\Снимок экрана (9).png |
| double case10(double x, double y)  {  return (pow((x + 1) / (x - 1), x)) + (18 \* x \* (pow(y, 2)));  }  int main()  {  system("chcp 1251>nul");  int a, b, c, d, x, y;  cout << "Введите a,b,c,d";  cin >> a >> b >> c >> d;  cout << "Введите x,y";  cin >> x >> y;  cout << case2(a, b, c, d) << " " << case4(x, y) << " " << case6(x) << " " << case10(x, y) << " ";  return 0;  } | C:\Users\kbstudentint\Pictures\Screenshots\Снимок экрана (10).png |

1. Проверила код на работоспособность.





1. Написала код для решения задач.

|  |  |
| --- | --- |
| #include "stdafx.h"  #include <iostream>  #include <cmath>  using namespace std;  double plow(double a, double b)  {  return (a\*b) / 2;  }  double per(double a, double b)  {  double c = sqrt(a\*a - b\*b);  return (a + b + c);  }  int main() {  int a, b;  cout << "Введите a,b";  cin >> a >> b;  cout << "Площадь= " << plow(a, b) << "Периметр= " << per(a, b);  } | 1. Вычислить периметр и площадь прямоугольного треугольника по длинам двух катетов а и Ь. |
| #include "stdafx.h"  #include <iostream>  #include <cmath>  using namespace std;  double plow(double R)  {  return 3, 14 \* pow(R, 2);  }  double per(double R)  {  return (2 \* 3, 14 \* R);  }  int main() {  int R;  cout << "Введите R= ";  cin >> R;  cout << "Площадь= " << plow(R) << " " << "Периметр= " << per(R);  } | 2. Даны координаты трех вершин треугольника (х1( yj, (х2, у2), (х3, уз). Найти его периметр и площадь. |
| #include "stdafx.h"  #include <iostream>  #include <cmath>  using namespace std;  void summ\_sub\_mult\_div(double x, double y) {  cout << x + y;  cout << x - y;  cout << x\*y;  cout << x / y;  }  int main() {  int x, y;  cout << "Введите x,y ";  cin >> x >> y;  summ\_sub\_mult\_div(x, y);  } | 3. Вычислить длину окружности и площадь крута с заданным радиусом R. |
| #include "stdafx.h"  #include <iostream>  #include <cmath>  using namespace std;  void stepen(double P) {  cout << pow(P, 2) << endl;  cout << pow(P, 3) << endl;  cout << pow(P, 4) << endl;  cout << pow(P, 5) << endl;  }  int main() {  double p = 3.14;  stepen(p);  } | 4. Найти произведение цифр заданного четырехзначного числа. |

1. Закоммитила на гитхаб отчет и проект
2. Выгрузила на гитхаб