Бюджетное учреждение высшего образования

Ханты-Мансийского автономного округа

«Сургутский государственный университет»

Политехнический институт

Кафедра автоматики и компьютерных систем

Отчет

Модель Светофора

Выполнил:

Бельтюков Михаил Олегович студент группы 609-22

Проверил: Брагинский М.Я.

Сургут

2025 г.

**Задание:**

1. описать работу ЦА (управляющего двухсекционным светофором) диаграммой переходов, таблицей переходов
2. создать программную модель ЦА (на любом ЯП) с отображением состояний, входных и выходных сигналов (таймер и лампы)

**Ход Работы:**

1. Таблица переходов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Время | Состояния | | | | |
| S0 | S1 | S2 | S3 | S4 |
| T0=0 | S1 |  |  |  |  |
| T1=5 |  | S2 |  |  |  |
| T2=8 |  |  | S3 |  |  |
| T3=18 |  |  |  | S4 |  |
| T4=21 |  |  |  |  | S1 |

Легенда

S1:

Автомобильный – красный

Пешеходный – зелёный

S2:

Автомобильный – красный + желтый

Пешеходный – красный

S3:

Автомобильный – зелёный

Пешеходный – красный

S4:

Автомобильный – желтый

Пешеходный – красный

S0:

Автомобильный – нет цвета

Пешеходный - нет цвета

1. Диаграмма переходов

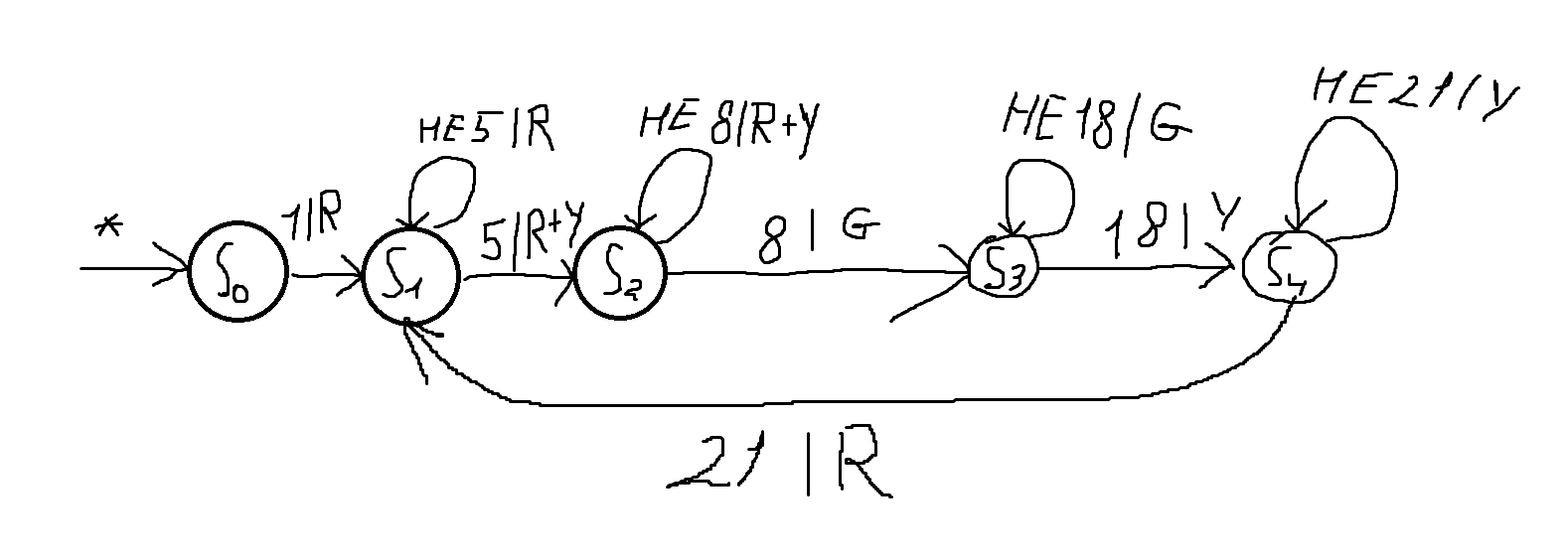


Рисунок 1 – диаграмма переходов светофора

Легенда совпадает с пунктом 1.

1. программная модель ЦА

#include <unistd.h>

#include <chrono>

#include <iostream>

#include <thread>

using namespace std::chrono\_literals;

enum colors { RED = 31, GREEN = 32, YELLOW = 33, NO\_COLOR = 0 };

void set\_color(int textColor) { std::cout << "\033[" << textColor << "m"; }

void out() { std::cout << "0" << std::endl; }

void change\_state\_pedestrian(bool is\_cur\_red, int seconds) {

system("clear");

set\_color(NO\_COLOR);

std::cout << "pedestrian:\ntime: " << seconds << std::endl;

if (is\_cur\_red) {

set\_color(NO\_COLOR);

out();

set\_color(GREEN);

out();

} else {

set\_color(RED);

out();

set\_color(NO\_COLOR);

out();

}

}

/\*\*

\*@param cur\_color переменная отвеччает за цвет светофора 1 - red, 2 - yellow,

\*3 - red+yellow, 0 - green

\*

\*/

void change\_state\_auto(int cur\_color, int seconds) {

set\_color(NO\_COLOR);

std::cout << "auto:\ntime: " << seconds << std::endl;

if (cur\_color == 1) {

set\_color(RED);

out();

set\_color(NO\_COLOR);

out();

out();

} else if (cur\_color == 2) {

set\_color(NO\_COLOR);

out();

set\_color(YELLOW);

out();

set\_color(NO\_COLOR);

out();

} else if (cur\_color == 3) {

set\_color(RED);

out();

set\_color(YELLOW);

out();

set\_color(NO\_COLOR);

out();

} else if (cur\_color == 0) {

set\_color(NO\_COLOR);

out();

set\_color(NO\_COLOR);

out();

set\_color(GREEN);

out();

}

set\_color(NO\_COLOR);

}

bool is\_cur\_red(int var) { return var % 2; }

void timer(int seconds, int state) {

for (int i = 0; i < seconds; i++) {

change\_state\_pedestrian(state, i);

change\_state\_auto(state, i);

std::this\_thread::sleep\_for(1000ms);

}

state += 2;

change\_state\_pedestrian(0, seconds + 1);

change\_state\_auto(state, 1);

std::this\_thread::sleep\_for(1000ms);

change\_state\_pedestrian(0, seconds + 2);

change\_state\_auto(state, 2);

std::this\_thread::sleep\_for(1000ms);

change\_state\_pedestrian(0, seconds + 3);

change\_state\_auto(state, 3);

std::this\_thread::sleep\_for(1000ms);

}

int main() {

long long helper = 0; // разное время на зелёный и красный

while (1) {

if (is\_cur\_red(helper)) {

timer(5, 1);

} else {

timer(10, 0);

}

helper++;

}

return 0;

}

Листинг 1 – программная модель ЦА.