**Київський національний університет імені Тараса Шевченка**

**факультет радіофізики, електроніки і комп’ютерних систем**

Курсова робота з курсу

**Прикладна теорія цифрових автоматів**

Варіант: ***7511***

Роботу виконав:

студент 3 курсу КІ-СА

Мургашов Гліб Едуардович

Науковий керівник:

кандидат фізико-математичних наук, доцент

Баужа Олександр Стасісович

**Київ 2020**

**Зміст**

**Вступ2**

1. **Синтез комбінаційних схем3**
   1. Синтез функції f14
   2. Синтез функції f26
   3. Синтез функції f38
2. Синтез операційного і керуючого автомата**10**

Синтез керуючого автомата**11**

Операційний автомат**19**

Спільна схема**28**

**Висновок30**

**Вступ**

В *теорії цифрових пристроїв* ***комбінаційною логікою*** (комбінаційною схемою) називають логіку функціонування пристроїв комбінаційного типу. *У комбінаційних пристроїв стан виходу однозначно визначається набором вхідних сигналів.*

***Скінченний автомат*** – *це пристрій, який складається з пам’яті та комбінаційної схеми.* Від звичайної комбінаційної схеми він відрізняється тим, що його виходи залежать не лише від поданих входів, а й від *входів в попередні моменти часу.* При поданні однакових вхідних сигналів в різні моменти часу роботи автомата, вихідні сигнали можуть *відрізнятися.*

Так, говорять про те, що автомат має *стани*. В один момент, автомат перебуває в одному стані, в залежності від теперішнього стану і поданих вхідних сигналів, переходить в новий *стан.*

Є різні типи скінченних автоматів:

Автомат Мура – автомат, у якого вихідні сигнали залежать лише від стану автомата.

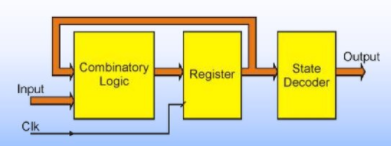


Рисунок 1. Автомат Мура

Автомат Мілі – автомат, у якого вихідні сигнали залежать від стану автомата та поданих вхідних сигналів.

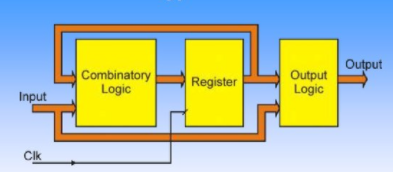


Рисунок 2. Автомат Мілі

У даній роботі для побудови структурних схем перемикальних функцій та автомату Мура було використано Proteus 8 Professional (у подальшому — Proteus) — пакет програм для автоматизованого проектування електронних схем, що розробляється компанією Labcenter Electronics.

**ПРАКТИЧНА ЧАСТИНА**

*Варіант:*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
|  |  |  |  |  |  |  |  |  |  |

1. **Синтез комбінаційних схем**

ТАБЛИЦЯ ІСТИННОСТІ ПЕРЕМИКАЛЬНИХ ФУНКЦІЙ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *х4* | *х3* | *х2* | *х1* | *f1* | *f2* | *f3* |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | *h7=1* | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | *h7=1* | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | x | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | *h8=0* | x |
| 0 | 1 | 1 | 1 | x | 1 | x |
| 1 | 0 | 0 | 0 | 1 | x | *h4=0* |
| 1 | 0 | 0 | 1 | 0 | x | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | *h1=1* | *h9=1* | 0 |
| 1 | 1 | 0 | 0 | *1* | 0 | x |
| 1 | 1 | 0 | 1 | *h2=1* | 1 | *h5=1* |
| 1 | 1 | 1 | 0 | *h3=1* | 1 | *h6=0* |
| 1 | 1 | 1 | 1 | 1 | *h10=0* | 1 |

ЕЛЕМЕНТНИЙ БАЗИС

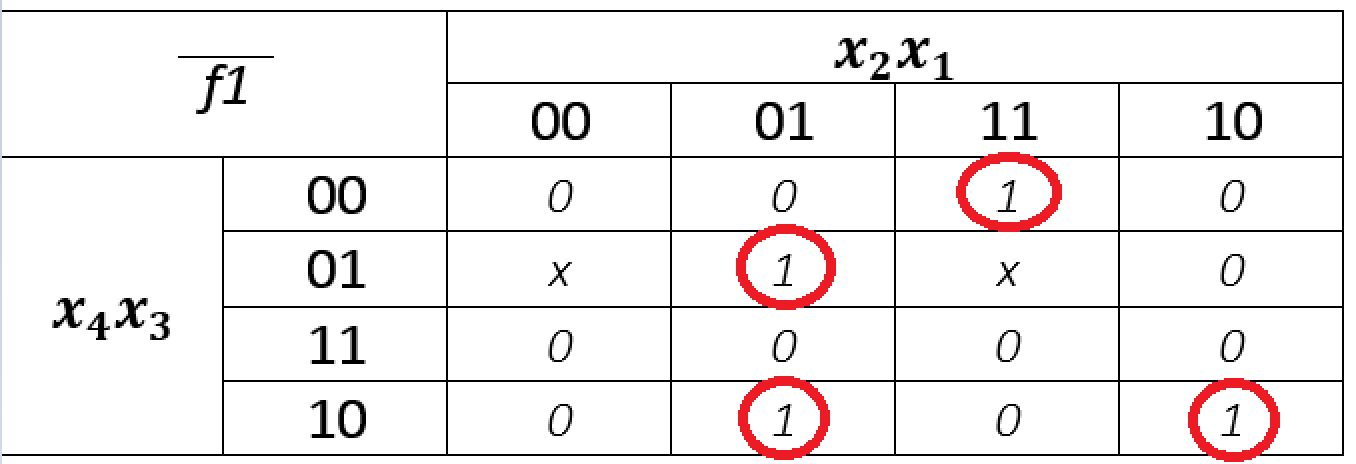
|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 0 | 3АБО, 4І, НЕ |

* 1. ***Синтез функції f1***

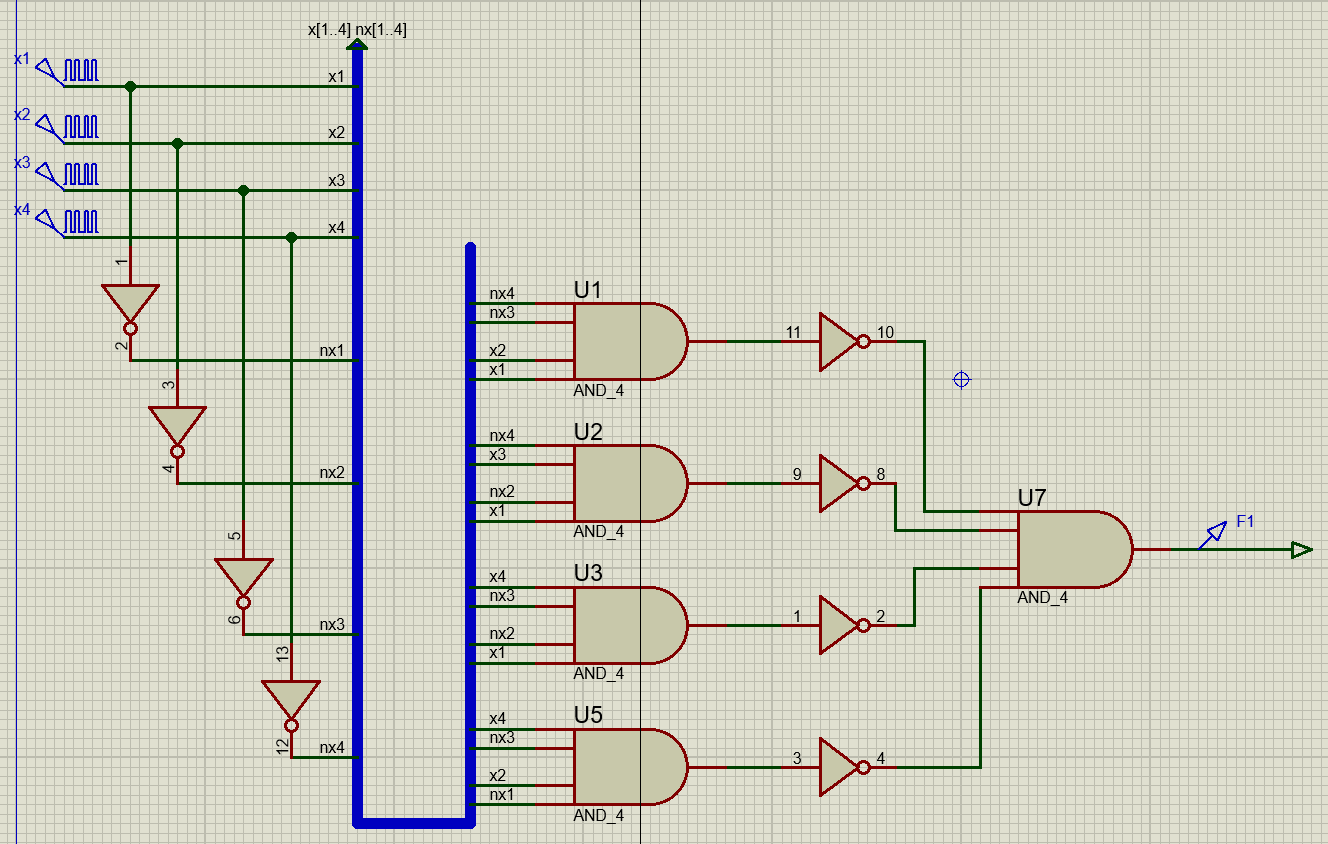
**Карта Карно функції :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***f1*** | |  | | | |
| **00** | **01** | **11** | **10** |
|  | **00** | **1** | **1** | **0** | **1** |
| **01** | **x** | **0** | **x** | **1** |
| **11** | **1** | **1** | **1** | **1** |
| **10** | **1** | **0** | **1** | **0** |

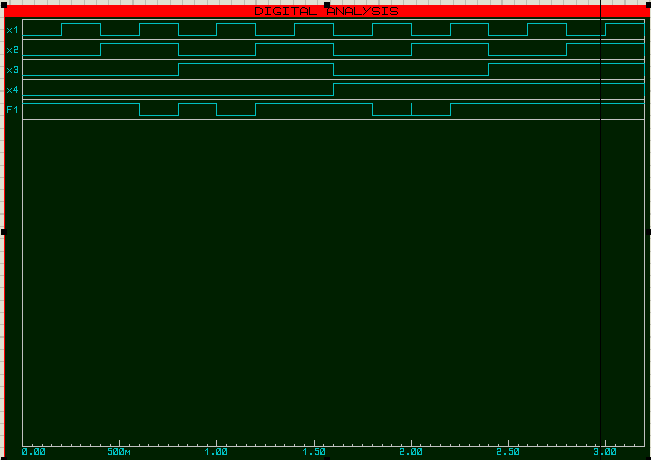
**А для функції :**

****

**Схема в ПЗ Proteus:**

****

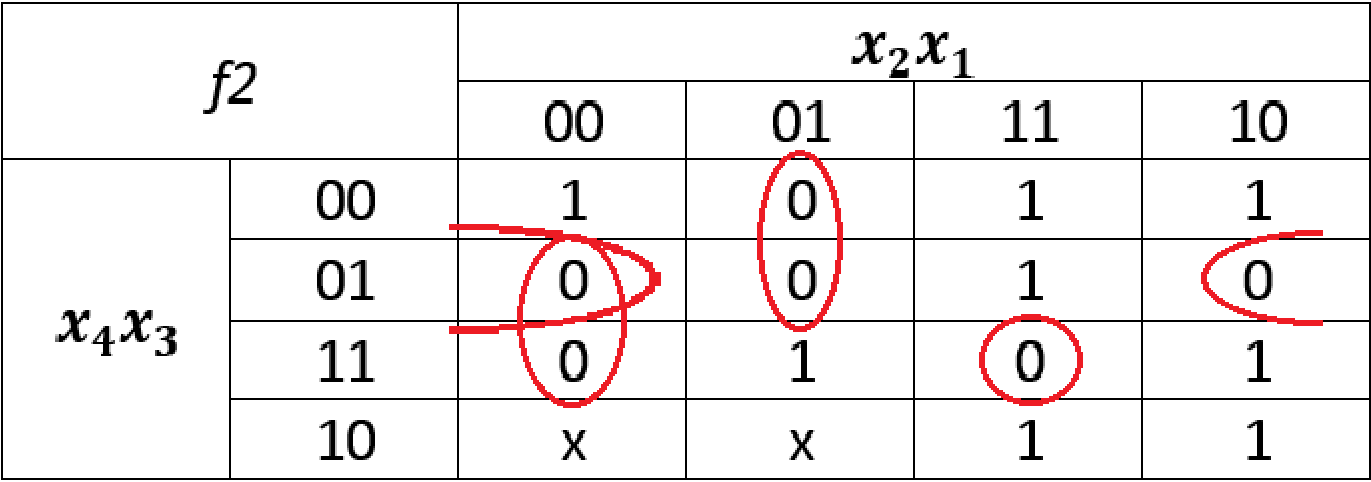
**Входи / вихідний сигнал:**

****

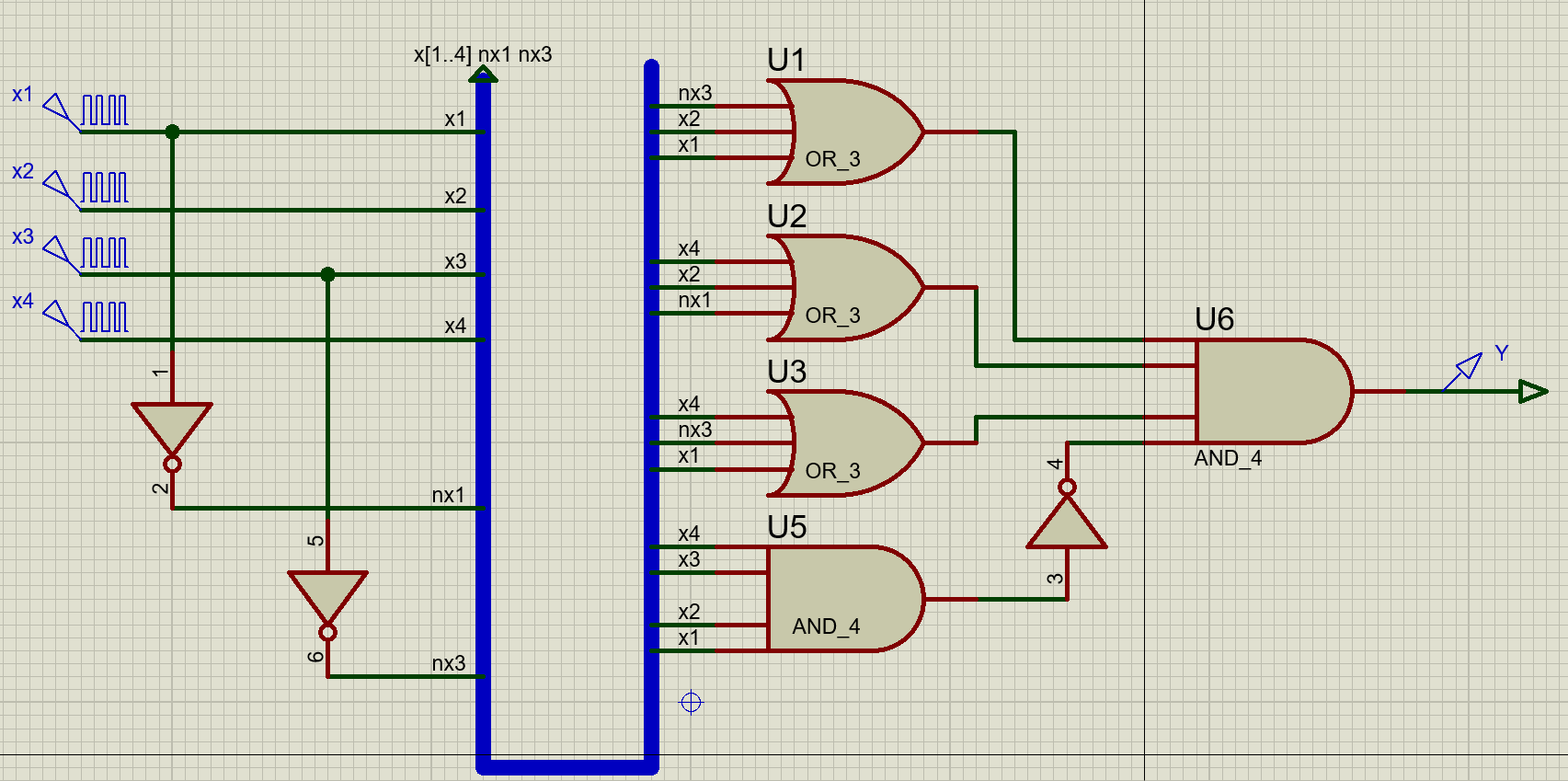
* 1. ***Синтез функції f2***

**Карта Карно функції :**

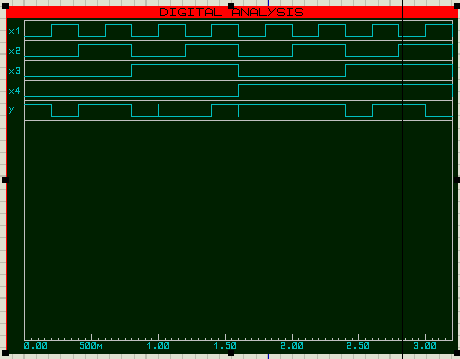
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***f2*** | |  | | | |
| **00** | **01** | **11** | **10** |
|  | **00** | **1** | **0** | **1** | **1** |
| **01** | **0** | **0** | **1** | **0** |
| **11** | **0** | **1** | **0** | **1** |
| **10** | **x** | **x** | **1** | **1** |

******

**Схема в ПЗ Proteus:**

****

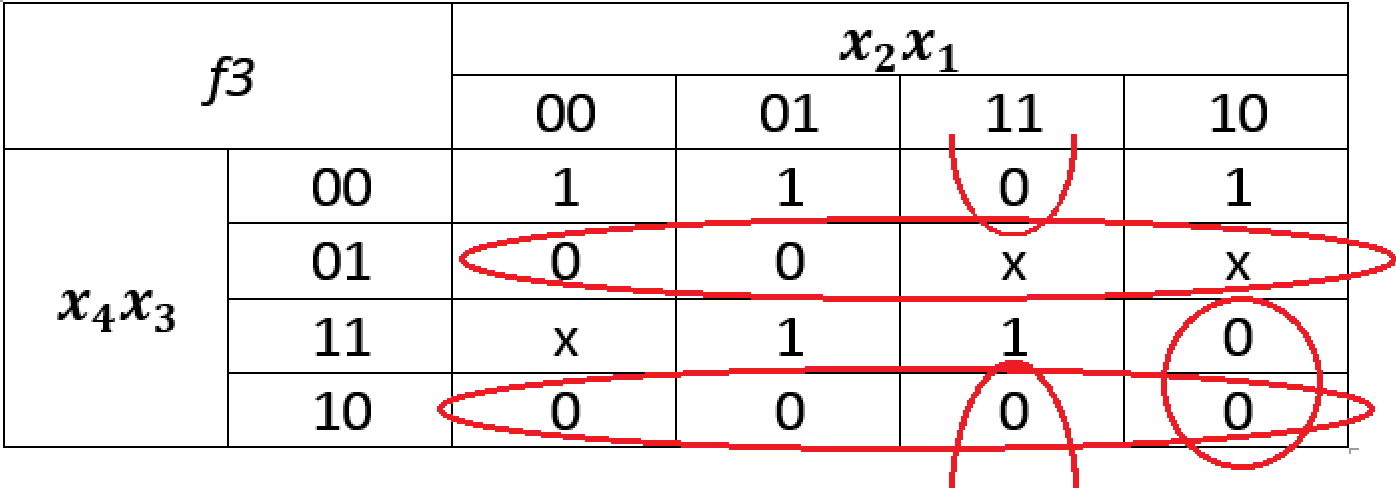
**Входи / вихідний сигнал:**

****

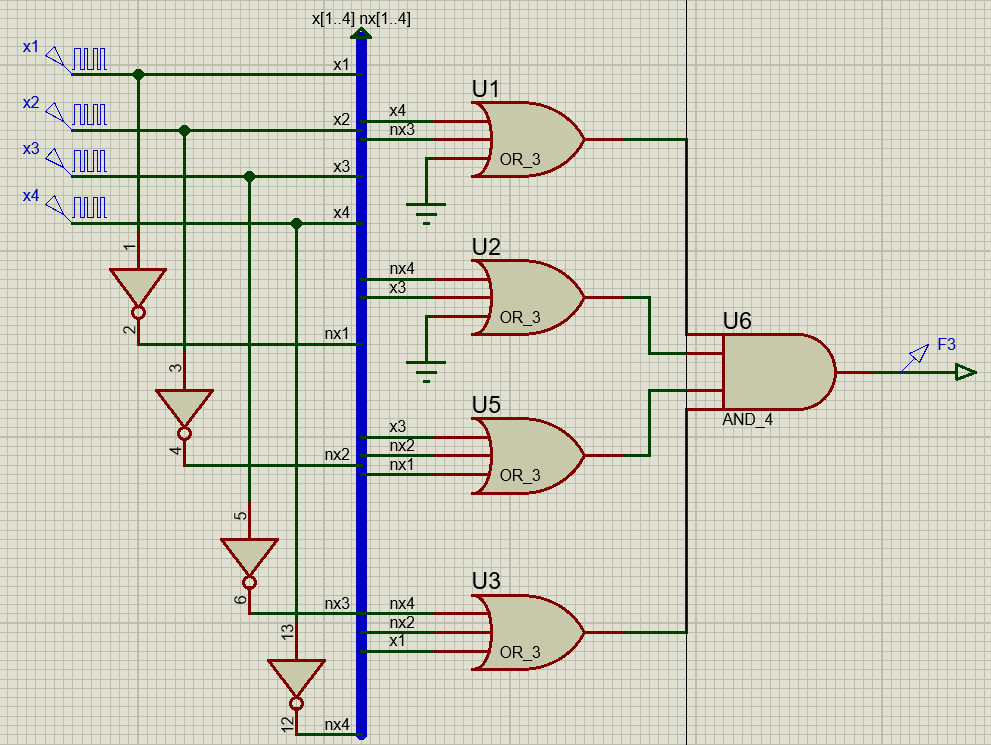
* 1. ***Синтез функції f3***

**Карта Карно функції :**

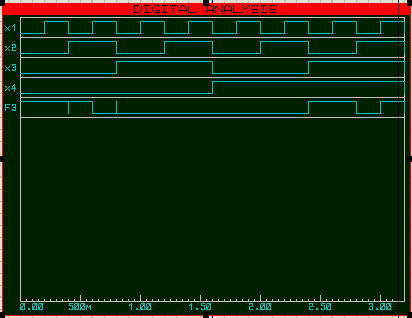
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***f3*** | |  | | | |
| **00** | **01** | **11** | **10** |
|  | **00** | **1** | **1** | **0** | **1** |
| **01** | **0** | **0** | **x** | **x** |
| **11** | **x** | **1** | **1** | **0** |
| **10** | **0** | **0** | **0** | **0** |

******

**Схема в ПЗ Proteus:**

****

**Входи / вихідний сигнал:**

****

1. **Синтез операційного та керуючого автомату**

ВИБІР ТИПУ КЕРУЮЧОГО АВТОМАТУ

|  |  |
| --- | --- |
| *h7* | Тип автомату |
| 1 | Мура |

ВИБІР АРИФМЕТИЧНОЇ ОПЕРАЦІЇ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *h4* | *h3* | *h2* | *h1* | Обчислення функції |
| 0 | 1 | 1 | 1 | D=A(B–1)+2C |

ВИБІР ТИПУ ТРИГЕРУ

|  |  |  |
| --- | --- | --- |
| *h6* | *h5* | Тип тригеру |
| 0 | 1 | D |

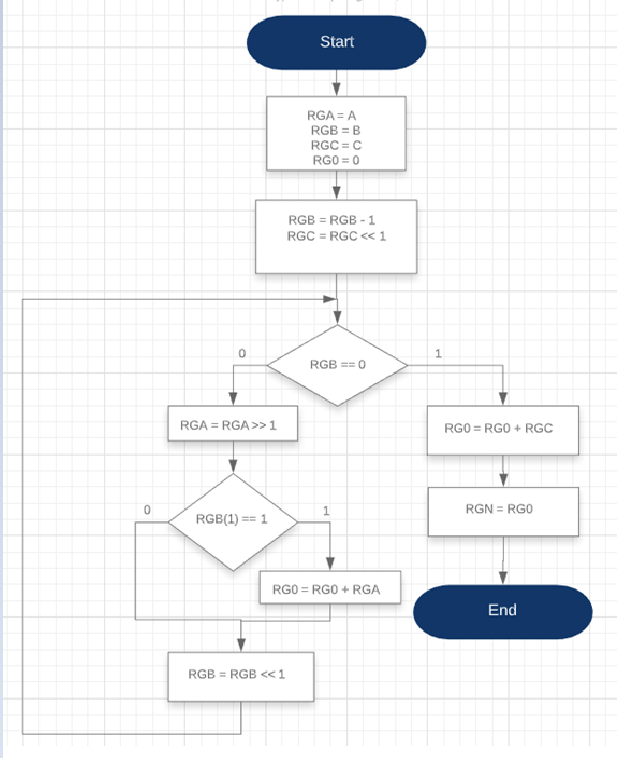
ВИБІР Алгоритму

|  |  |  |
| --- | --- | --- |
| *h 3* | *h 7* | Тип алгоритму |
| 1 | 1 | Алгоритм №4 |

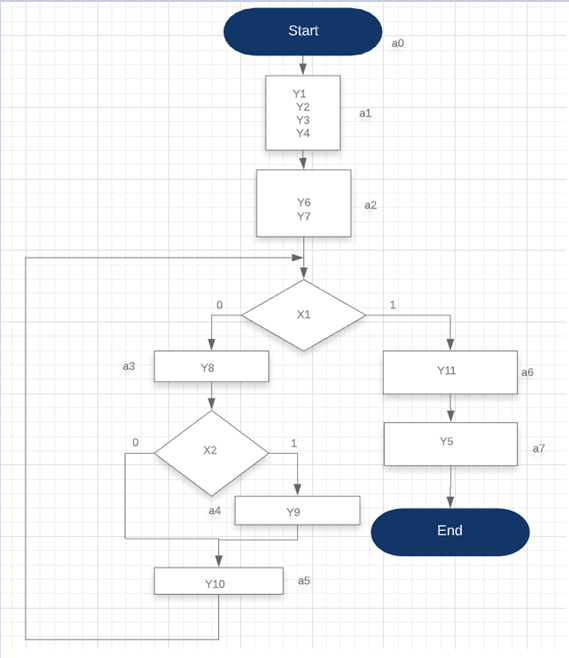
***Алгоритм №4:*** *Множення з старших розрядів множника та зсувом множеного вправо при чому сума часткових добутків залишається нерухомою.*

**2.1*) Cинтез керуючого автомата***

***Змістовна схема алгоритму для керуючого автомата:***



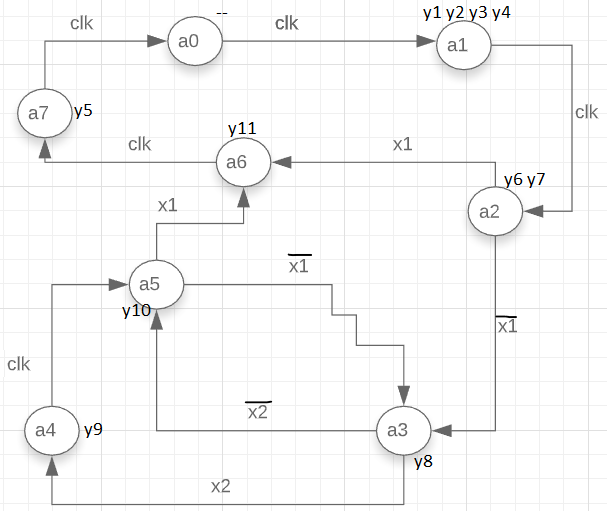
**Синтез керуючого автомата:**



Змістовна таблиця кодування:

|  |  |  |
| --- | --- | --- |
| *Код* | *Зміст* | *Примітки* |
| Y1 | RG3 = A | Запис числа A в старші розряди регістру RGА |
| Y2 | RGB = B | Запис числа A в регістр RGB |
| Y3 | RGC = C | Запис числа C в регістр RGC |
| Y4 | RG0 = 0 | Заповнення RG0 нулями |
| Y5 | RGN = RG0 | Запис RGO в RGN |
| Y6 | RGB = RGB - 1 | Зберігання в RGB значення «B -1» |
| Y7 | RGC = RGC << 1 | Зберігання в RGC значення «2\*C» |
| Y8 | RGA = RGA >> 1 | Зсув RGA вправо |
| Y9 | RG0 = RG0 + RGA |  |
| Y10 | RGB = RGB << 1 | Зсув RGB вліво |
| Y11 | RG0 = RG0 + RGC | RG0 = A(B-1) + 2C |
| X1 | RGB == 0 | Регістр RGB порожній? 1 – ТАК, 0 – НІ |
| X2 | RGB(1) == 1 | Старший біт RGB є 1? 1 – ТАК, 0 - НІ |

*Граф схема переходів:*

****

**Пряма таблиця переходів-виходів автомата Мура**

|  |  |  |  |
| --- | --- | --- | --- |
| *Початковий стан*  *Sm* | Y (вихідний сигнал,що виробляється при переході | *Стан переходу Sk* | *Умова переходу* |
|  | *-* |  | *1* |
|  | *y1,y2,y3,y4* |  | *1* |
|  | *y5,y6,y7* |  |  |
|  |  |
|  | *y8* |  |  |
|  |  |
|  | *y9* |  | *1* |
|  | *y10* |  |  |
|  |  |
|  | *y11* |  | *1* |
|  | *y5* |  | *1* |

Кількість тригерів для кодування станів m = ] log2 8 [ = 3.

Потрібно рівно 3 D-тригера.

**Кодування станів автомата:**

Кодуємо стани автомата алгоритмом для D-тригерів:

*кількість входжень в вершину-стан .*

*Сортуємо числа по спаданню, вершину з найбільшим кодуємо «0000»;*

*наступні кодом з однією одиницею: «001», «010»… «100»;*

*потім з двома: «110», «011» і т.д.*

|  |
| --- |
|  |
|  |
|  |

= 1,

= 1,

= 1,

= 2,

= 1,

= 2,

= 2,

= 1.

***Структурна таблиця переходів-виходів автомата Мура***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Початковий стан*  *Sm* | K(*Sm*) | Y (вихідний сигнал,що виробляється при переході | *Стан переходу Sk* | *K(Sk)* | *Умова переходу* | *ФЗ* |
|  | *100* | *-* |  | *110* | *1* |  |
|  | *110* | *y1,y2,y3,y4* |  | *011* | *1* |  |
|  | *011* | *y5,y6,y7* |  | 000 |  |  |
|  | *010* |  |  |
|  | *000* | *y8* |  | 101 |  |  |
|  | 001 |  |  |
|  | *101* | *y9* |  | *001* | *1* |  |
|  | *001* | *y10* |  | 000 |  |  |
|  | 010 |  |  |
|  | *010* | *y11* |  | *111* | *1* |  |
|  | *111* | *y5* |  | *100* | *1* |  |

***Система рівнянь переходів:***

***Система рівнянь виходів:***

*y1 = y2 = y3 = y4 = ,*

*y6 = y7 = ,*

*y8 = ,*

*y9 = ,*

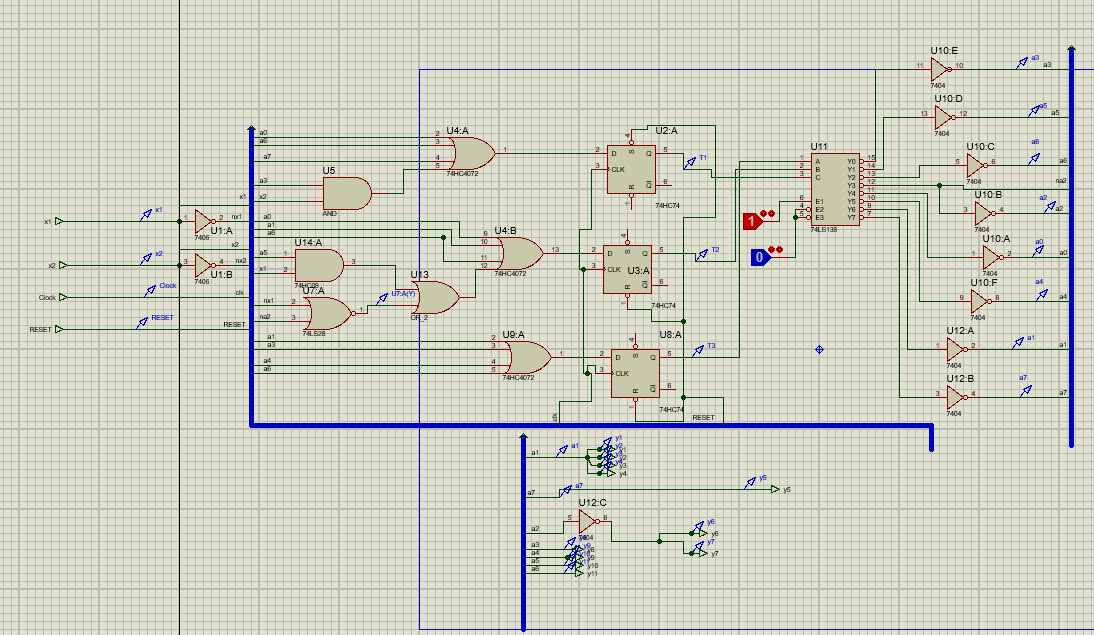
*y10 = ,*

*y11 = .*

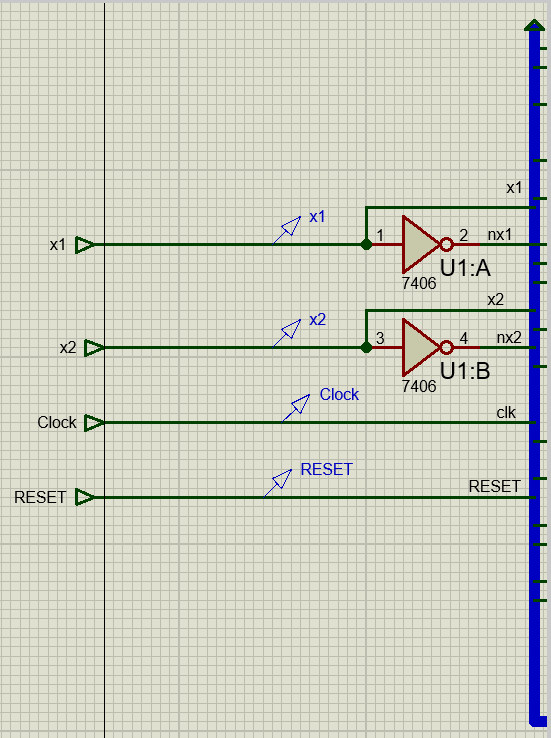
*y5 = ,*

**Схема керуючого автомату**

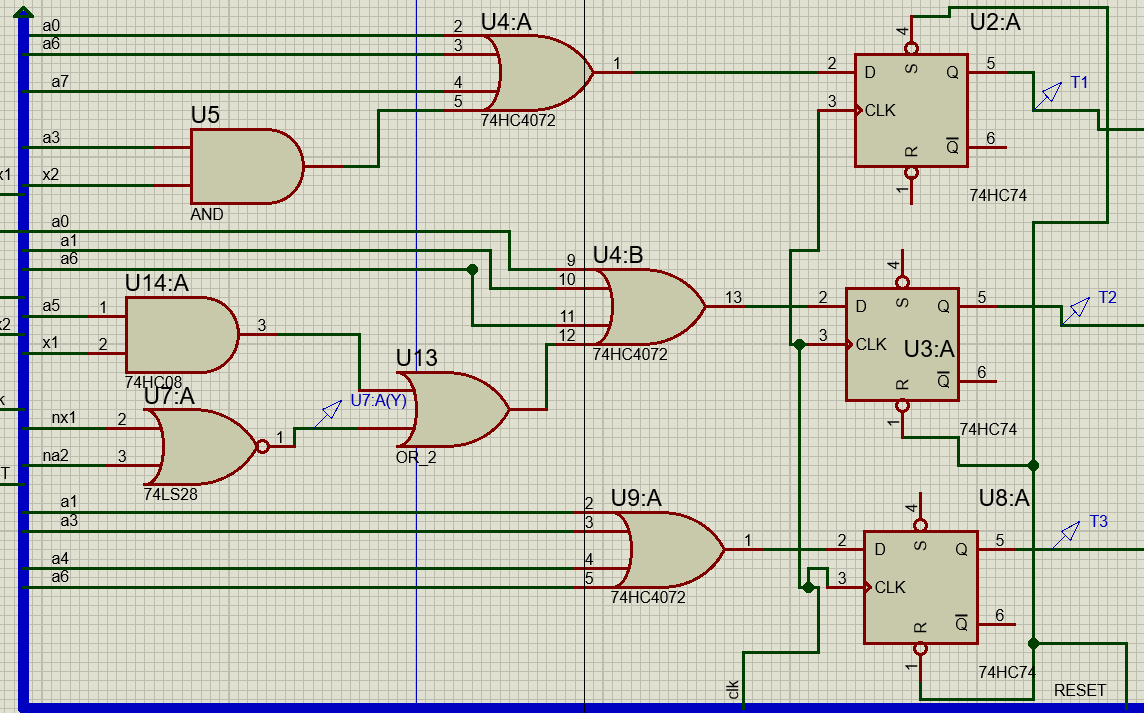
**Схема загалом:**

****

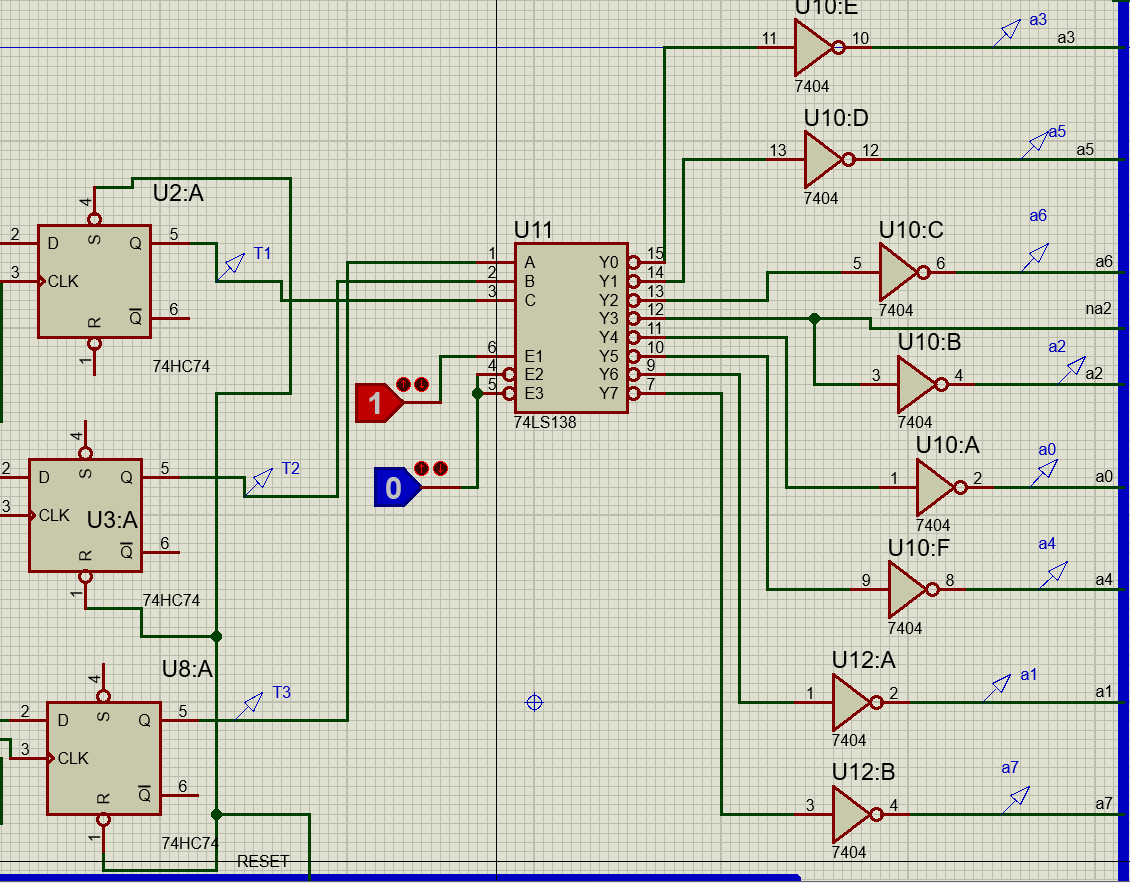
**Входи:**

****

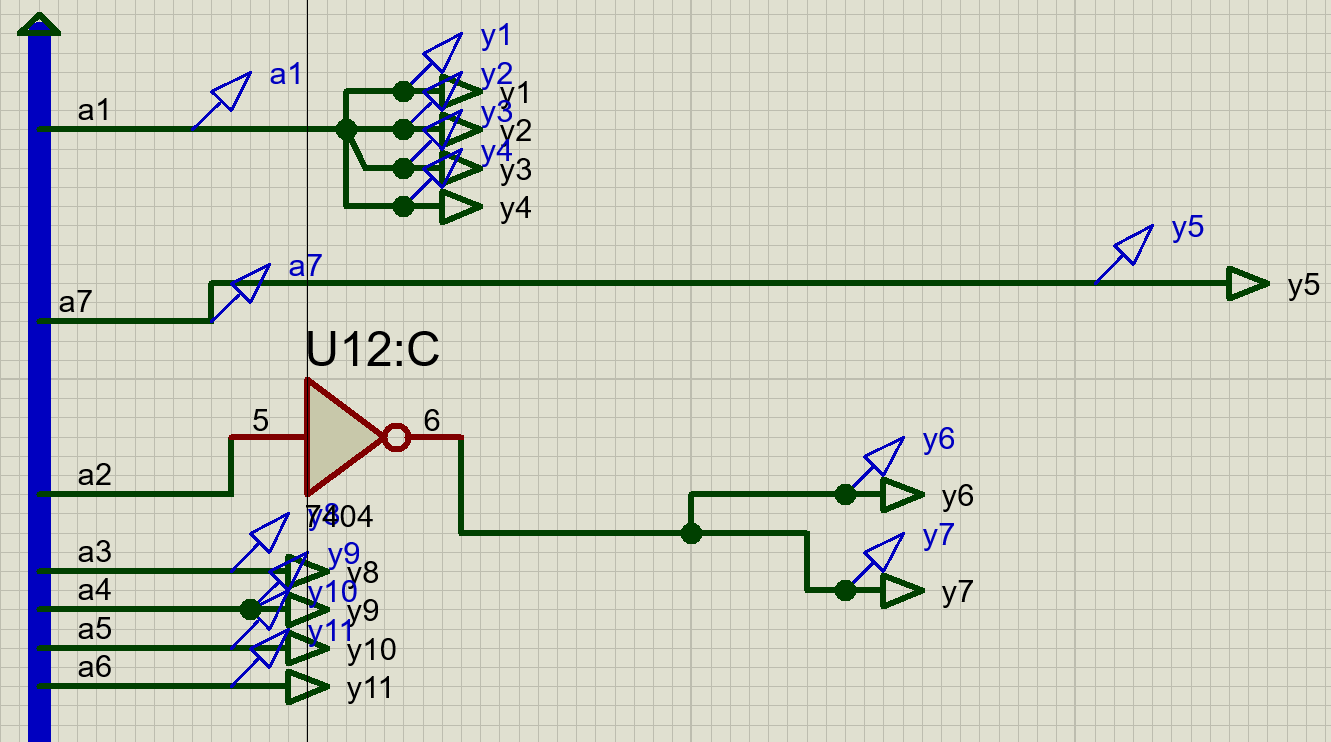
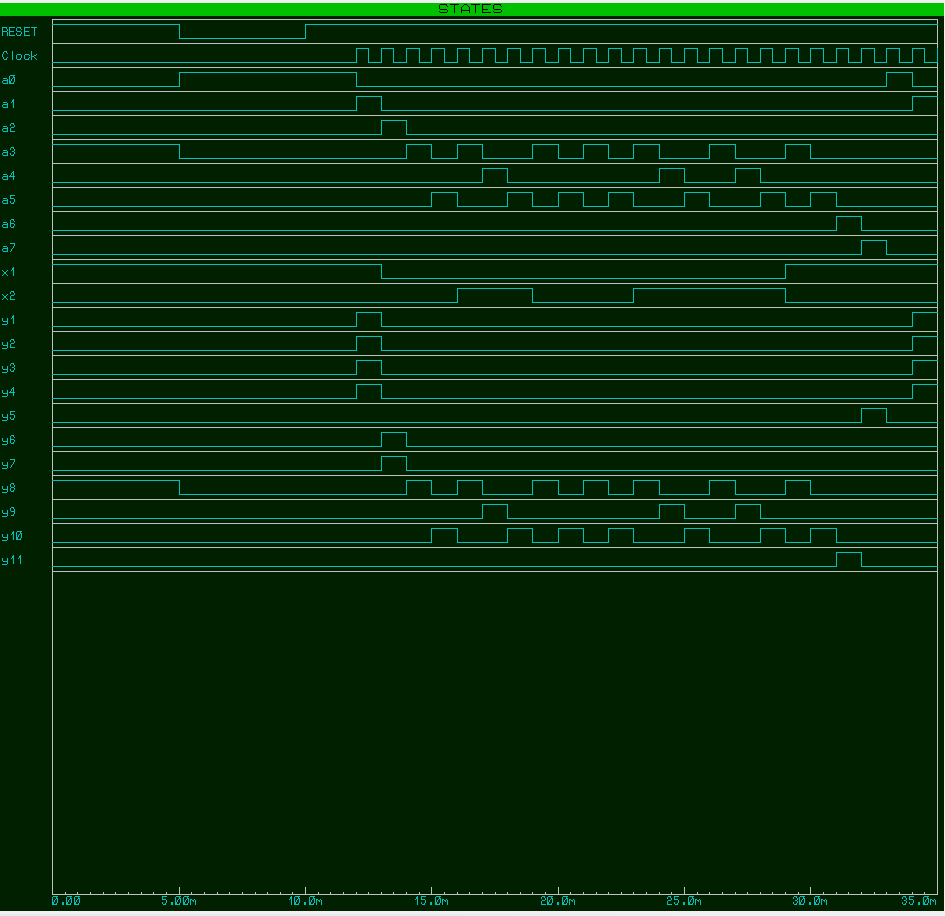
**Функції збудження:**

****

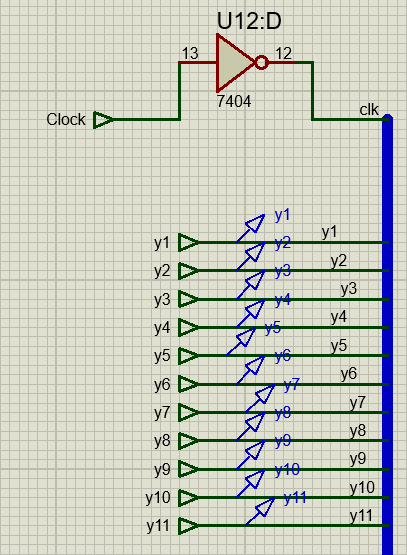
**Дешифратор станів:**

****

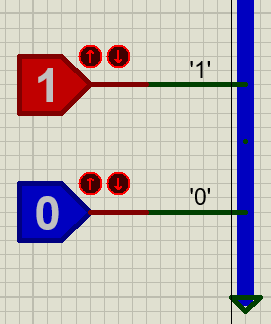
**Вихідні сигнали і графік:**

****

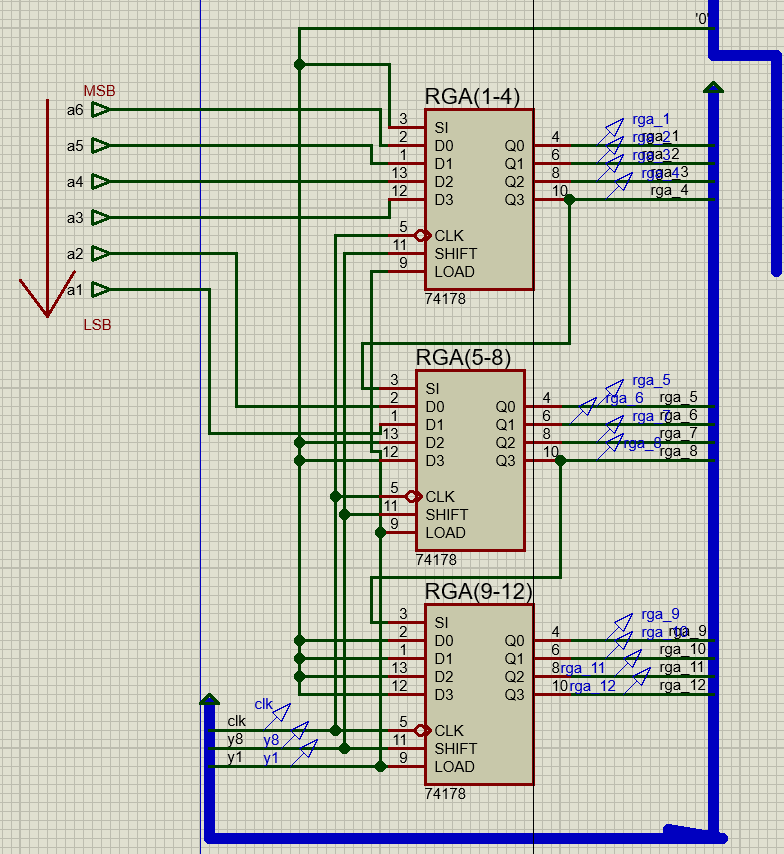
**2.2) Операційний автомат**

**Входи:**

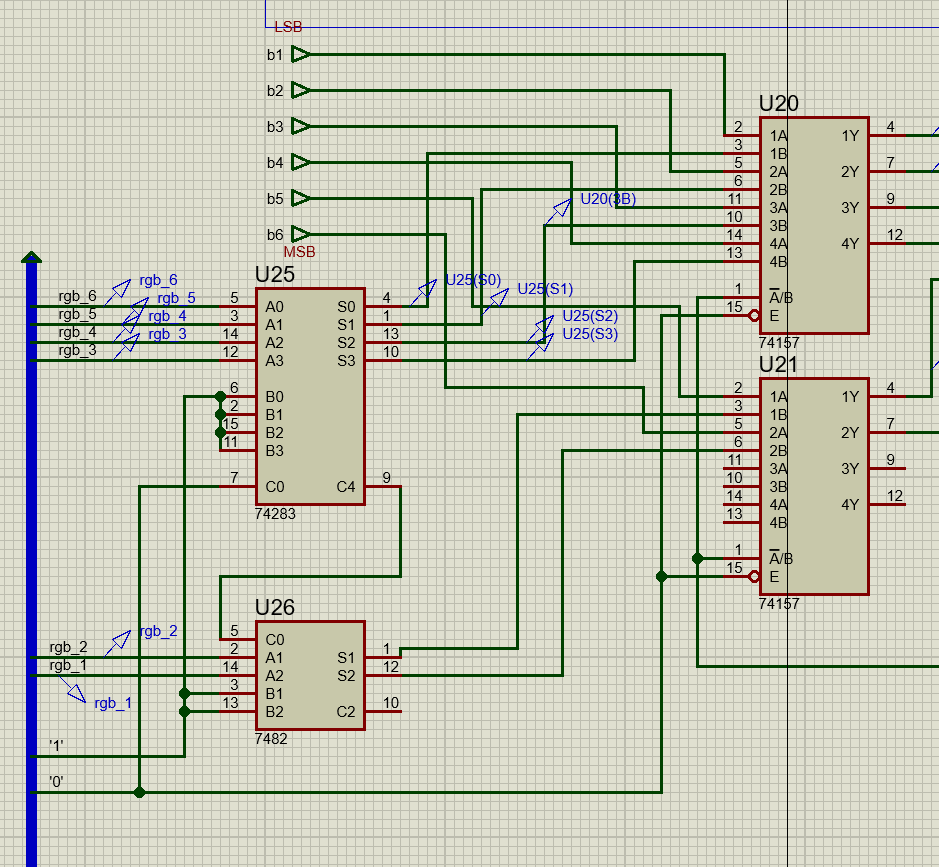
**Регістри *74178*, які я використовував для збереження чисел, проміжних данних і відповіді, мають вхід *CLK*, по задньому фронту, тому поставлено інвертор U12:D, бо D-тригери, що використовувались в керуючому автоматі мають вхід CLK по передньому фронту.**



Регістр RGA(), його заповнення при Y1 = 1 та зсув при Y8 = 1



“B” подається на мультиплексори U20, U21, які заповняють регістр RGB початковим значення «B» (при Y2 = 1), або значенням “RGB - 1” (при Y6 = 1)



Виходи мультиплексора *74157* приєднані до входів регістру RGB(1-6);

Також тут перевіряються умови “X1” i “X2”;

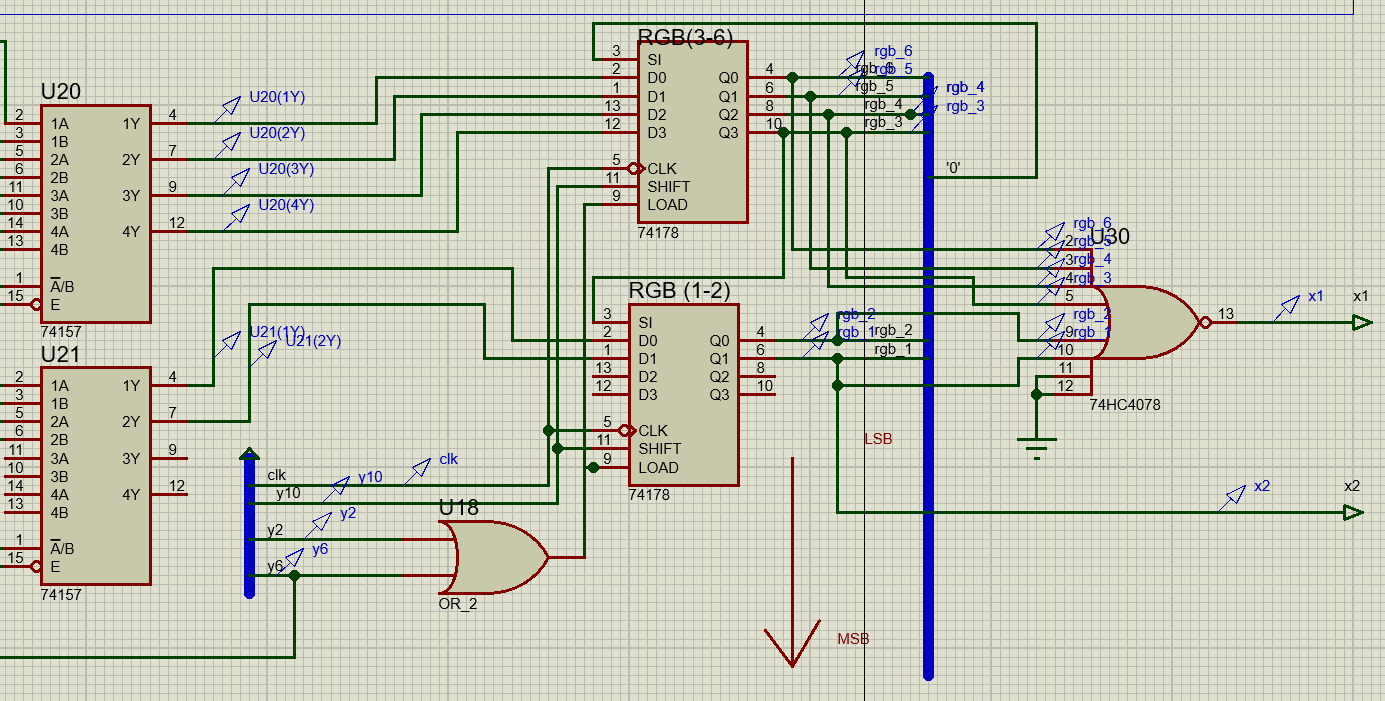
Y2 = 1: “RGB = B”

Y6 = 1: “RGB = RGB - 1”

Y10 = 1: “RGB << 1”

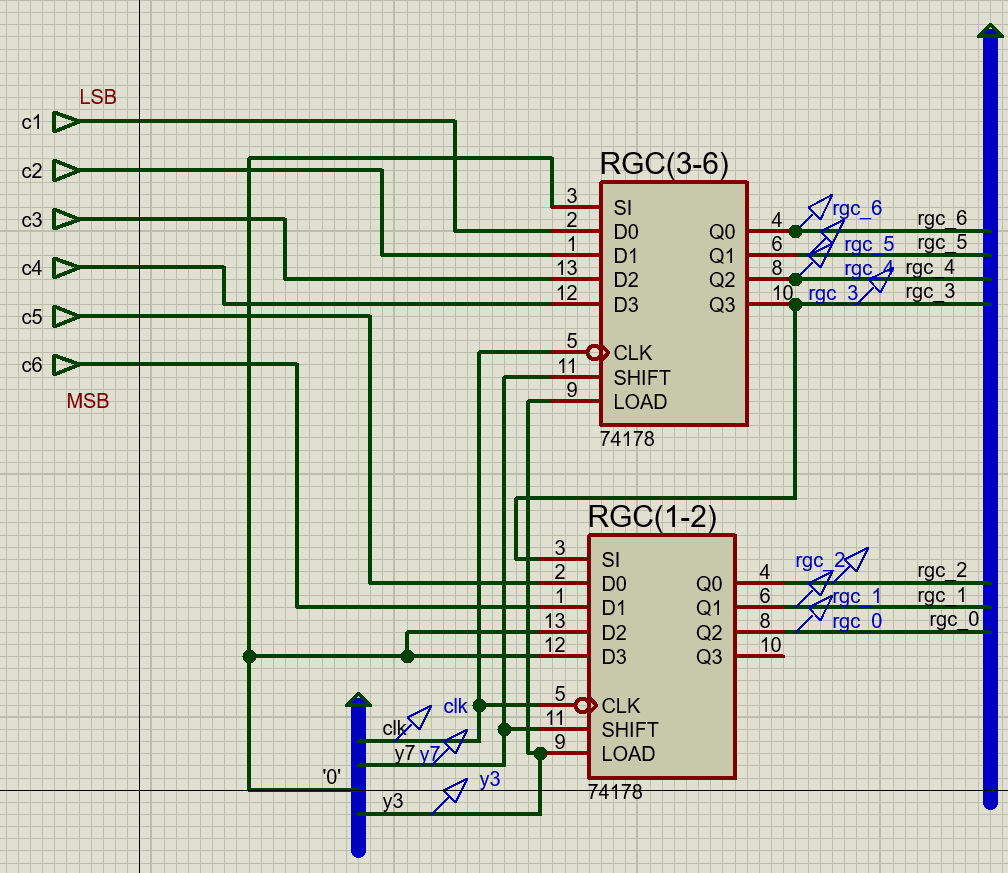
X1 = 1: “RGB == 0”

X2 = 1: “Старший біт RGB == 1”



Y3 = 1: “RGC = C”

Y7 = 1: “RGC << 1(RGC = RGC \* 2)”

****

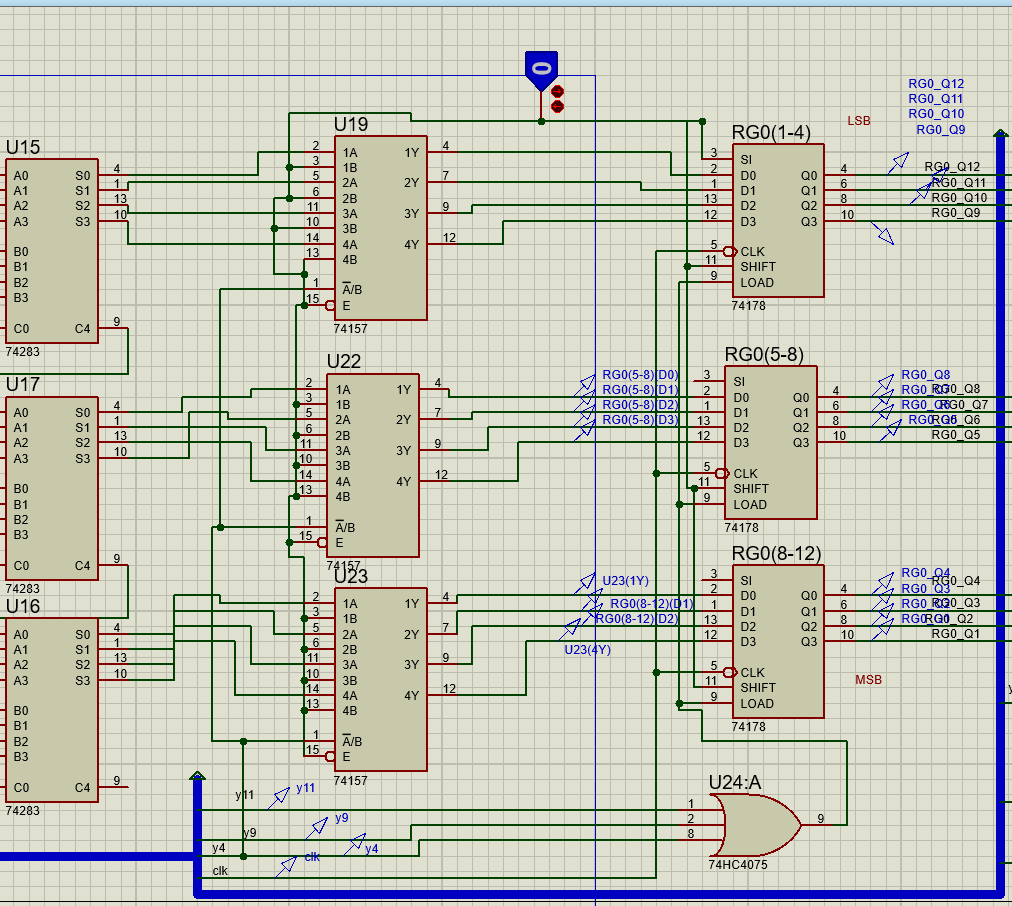
Виходи головних суматорів *74283* (U15, U17, U16), мультиплексори U19 і U22 і сам регістр RG0(1-12).

RGO = [ “0” АБО RG0 + (RGC/RGA) ]

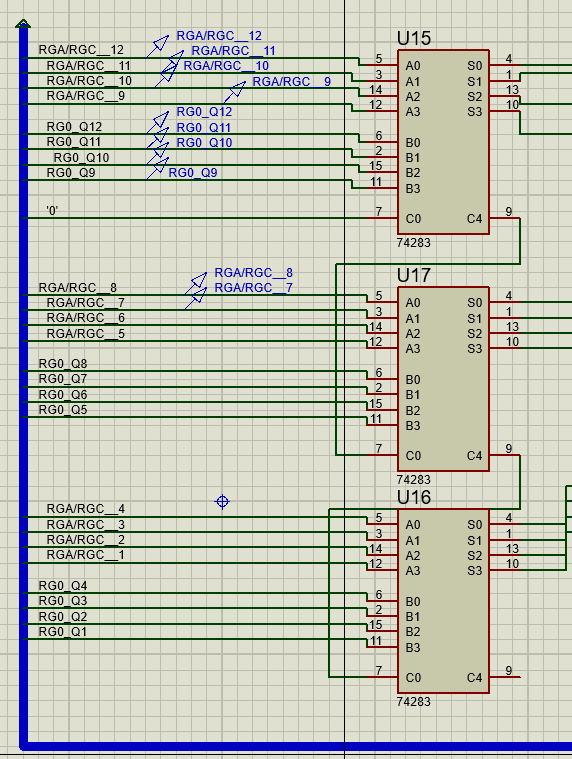
Y4 = 1: RGO = 0

Y9 = 1, Y11 = 0: RGO = RGO + RGA

Y11 = 1: RGO = RGO + RGC

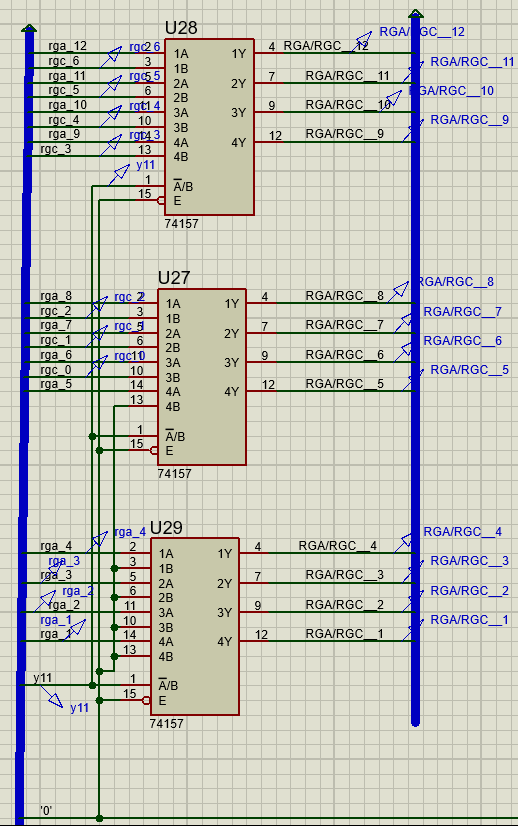
****

**Входи суматорів U15, U17, U16:**

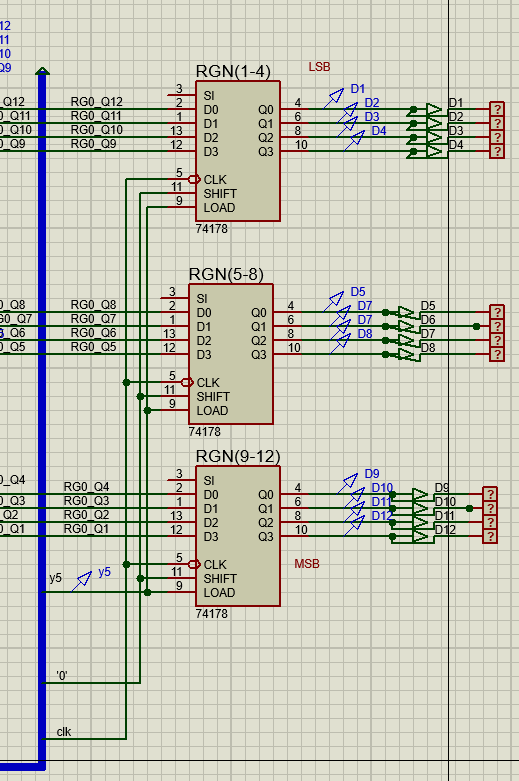
****

Мультиплексори U28(27,29), які при Y11 = 1: RGO = RGO + RGC

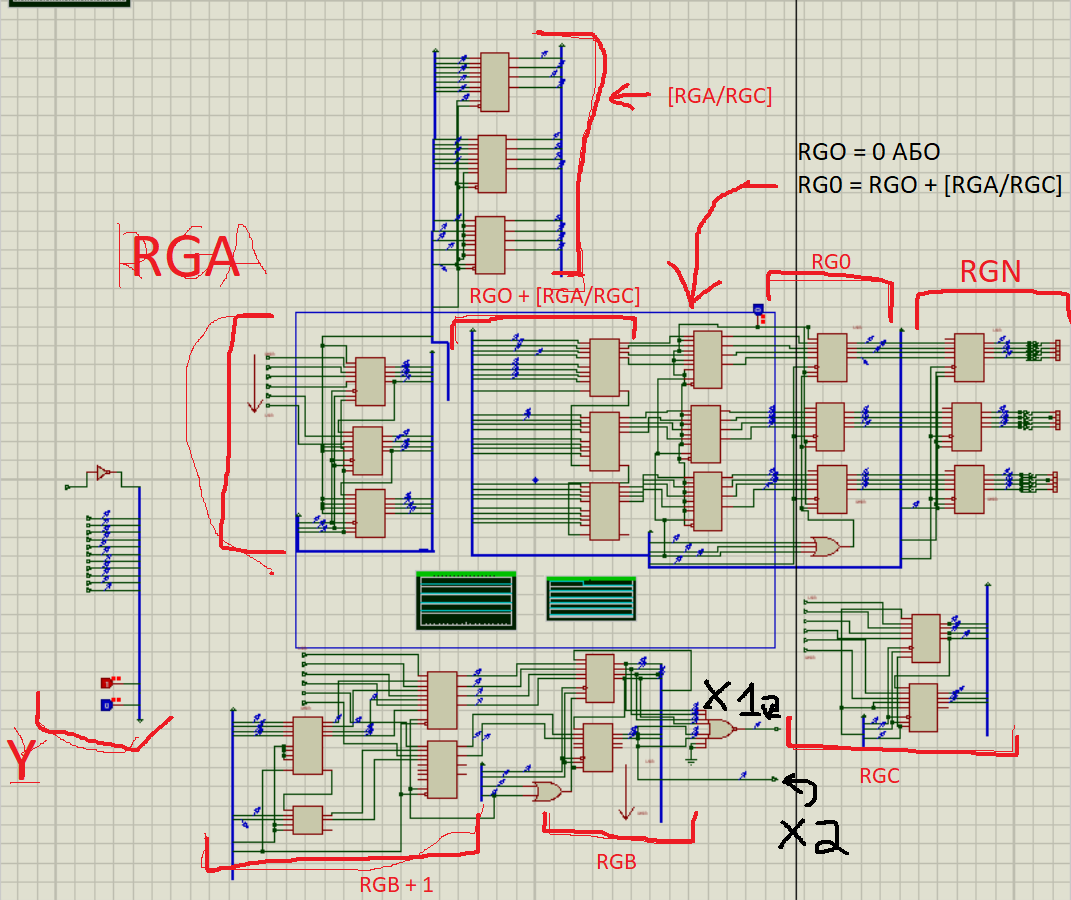
при Y11 = 0: RGO = RGO + RGA

****

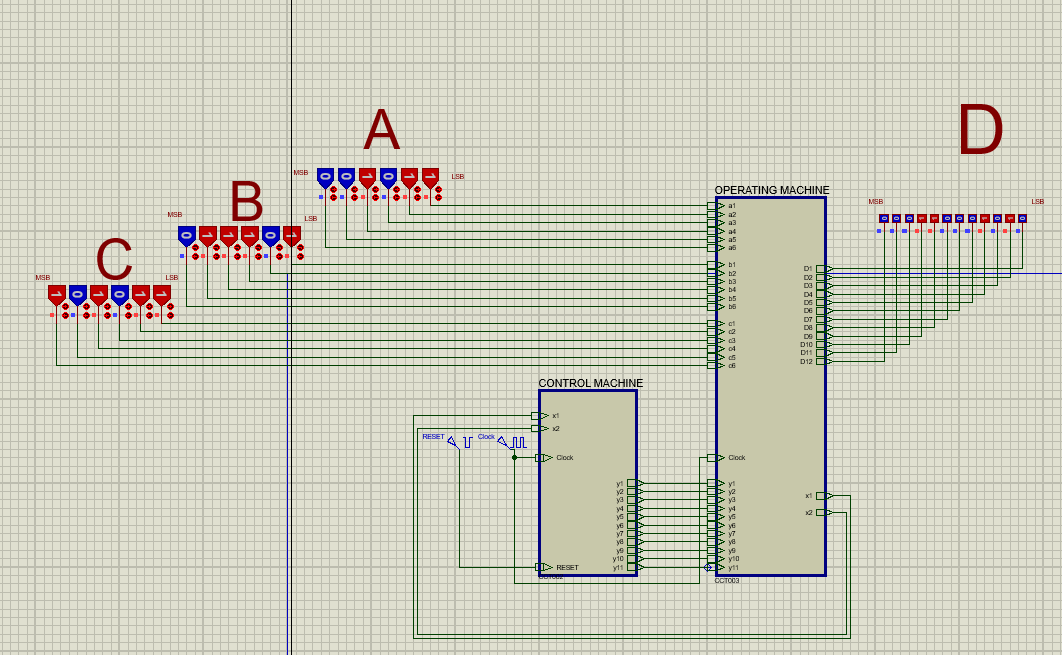
**Y5 = 1: RGN = RGO; Регістр RGN(1-12), який зберігає відповідь і виводить в «*D*»**

****

**Схема операційного автомата загалом:**

****

**2.3) Схема ОА і КА разом:**

****

**Перевірка:**

**A = ,**

**B = ,**

**C = ,**

**D = A (B - 1) + 2 C = 308 + 86**

**Висновки**

В курсовій роботі було побудовано комбінаційні схеми за данною таблицею істинності функцій. Схеми побудовано в заданому базисі. Для мінімізації та побудови було використано карти Карно та елементарні правила де-Моргана. Були виведені відповідні залежності виходів від входів та перевірено коректність їх побудови.

Було розроблено алгоритм для підрахунку числа, заданого певним шаблоном арифметичних дій. За цим алгоритмом побудовано блок-схему, граф-схему, структурну таблицю та синтезовано керуючий автомат Мура.

За розробленим алгоритмом було і побудовано відповідний операційний автомат для виконання заданих арифметичних дій. Автомати рознесено в два окремих блоки, які потім були з’єднанні разом.

Робота автоматів перевірено, обрахований результат співпадає з результатом роботи схеми.