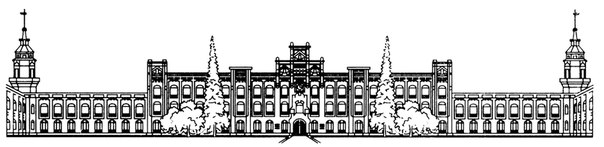
МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КПІ»

Кафедра Автоматизованих систем обробки інформації та управління

Звіт з лабораторної роботи № \_\_

«**Шифратори да дешифратори»**

# З дисципліни «Архітектура комп’ютера»

|  |  |  |
| --- | --- | --- |
| Перевірила:  доц. Коган А.В. |  | Виконав: |
|  |  | Студент групи |
|  |  |  |
|  | Київ 2021 |  |

**Лабораторна робота № 2**

**Шифратори да дешифратори**

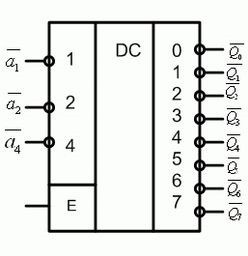
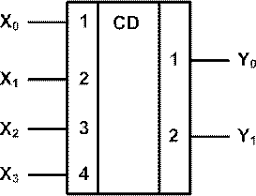
Цифровi системи здiйснюють обробку iнформацiї у двiйковому кодi. До основних операцiй належать кодування i декодування, мультиплексування, де- мультиплексування, порiвняння, перетворення кодiв.

Дешифратор – це комбiнацiйний пристрiй, який перетворює комбiнацiю вхiдних змiнних в активний сигнал тiльки на одному iз виходiв. Дешифратор перетворює двiйковий код числа на входi в активний сигнал на тому виходi, номер якого вiдповiдає десятковому еквiваленту двiйкового коду на входi.

Повним дешифратором називають дешифратор, який має n входiв i *m* = 2*n* виходiв. Активним сигналом може бути 0 або 1 в залежностi вiд типу входiв чи виходiв. Вiдповiдно до цього дешифратори бувають з прямими та iнверсними входами та виходами. Деякi дешифратори мають один або декiлька входiв дозволу, якi використовуються для управлiння роботою дешифратора.

При наявностi активного сигналу на входi дозволу, робота дешифратора дозволена (на одному з виходiв є активний сигнал), iнакше на всiх виходах дешифратора присутнi неактивнi сигнали. На р[ис.1](#_bookmark0) зображений дешифратор 3x8 з iнверсними входами та виходами та одним входом управлiння.

Шифратор - це комбiнацiйний пристрiй, який виконує функцiю, обернену до функцiї дешифратора. Зазвичай, допускається подавання тiльки одного активного сигналу на вхiд. Шифратор перетворює число, яке вiдповiдає номеру входу на який подається активний сигнал, на двiйкове. Кiлькiсть входiв n i виходiв m пов’язана спiввiдношенням *m* = 2*n*. На [рис.1](#_bookmark0) зображений шифратор 4x2 з прямим входами та виходами.

a) b)

Рис. 1: Дешифратор (a) та шифратор(b)

**Завдання:**

Побудувати дешифратор та шифратор на логічних елементах.

Побудувати дешифратор 4х16 за допомогою 2-ох дешифраторів 3х8.

Побудувати логічну функцію за допомогою дешифратора.

На [рис.2](#_bookmark1) зображена схема 3-входового дешифратора побудованого на елементах 3-I (тут X2 та Y7 - cтаршi розряди, X0 та Y0 - молодшi). Перевiряючи таблицю iстинностi та часову дiаграму, помiтно, що дешифратор працює правильно: на виходi, номер якого вiдповiдає десятковому еквiваленту двiйкового коду на входi, з’являється 1, на всiх iнших 0.

Отже, активний сигнал побудованого дешифратора - 1.

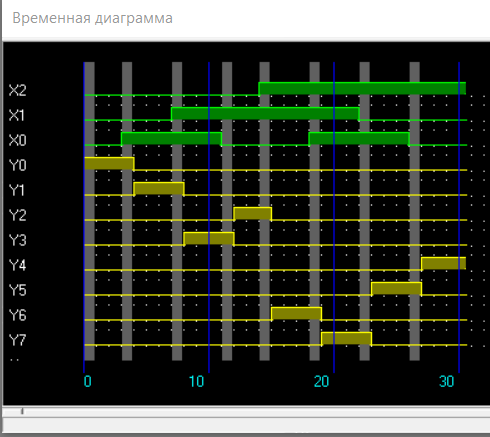
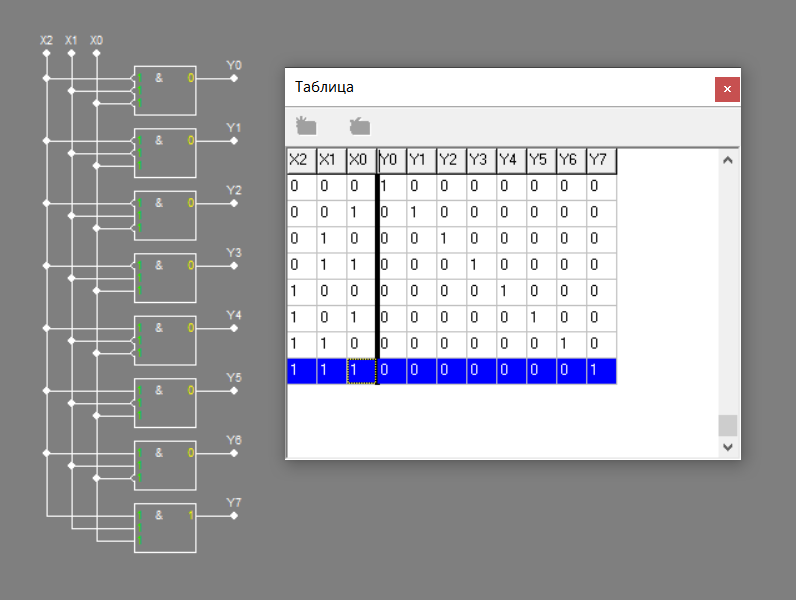


Рис. 2: Схема дешифратора на логiчних елементах 3-I, його таблиця iстинностi та часова дiаграма входiв та виходiв

Окрім цього, був побудований шифратор на логiчних елементах 4-АБО. Його схема i таблиця iстинностi зображенi на р[ис.3,](#_bookmark2) а часова дiаграма входiв та виходiв на [рис.4.](#_bookmark4)

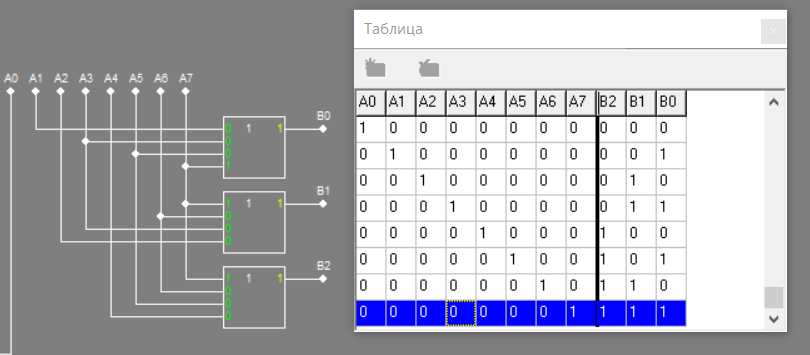


Рис. 3: Схема шифратора та його таблиця iстинностi

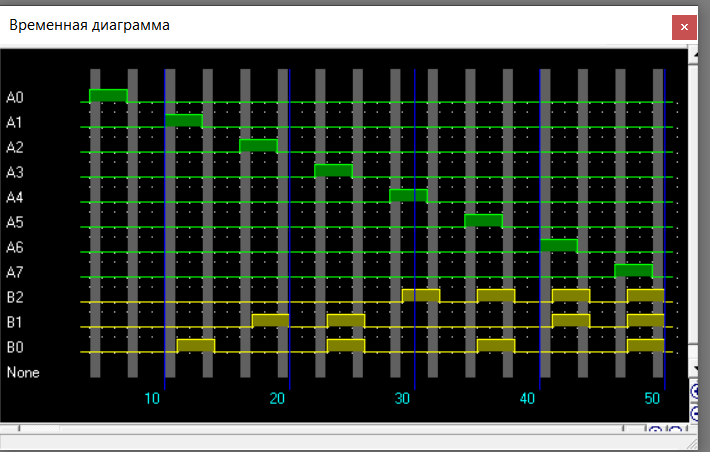


Рис. 4: Часова дiаграма входiв та виходiв шифратора

**Побудова дешифратора 4x16 за допомогою 2-ох дешифраторiв 3x8**

Часто у дешифраторiв є хоча б один вхiд дозволу. Саме завдяки такому

входу можна побудувати схему дешифратора бiльшого розряду з двох даних дешифраторiв.

В цьому завданнi було побудовано схему дешифратора розряд якого на 1 бiльший вiд розрядiв 2-ох даних дешифраторiв. Нехай A3 - найстарший розряд. Схема була побудована наступним чином: входи A0, A1, A2 подаються на iнформацiйнi входи обох дешифраторiв, а A3 - на вхiд дозволу.

Таке з’єднання дешифраторiв ще називають каскадним. Тодi 2 дешифратора будуть працювати однаково, лише один буде вмикатися при A3=0, а iнший при A3=1. Це еквiвалентно роботi 4-розрядного дешифратора. Отримана схема, її таблиця iстинностi зображена на [рис.5,](#_bookmark5) а часова дiаграма входiв та виходiв на [рис.6.](#_bookmark6)

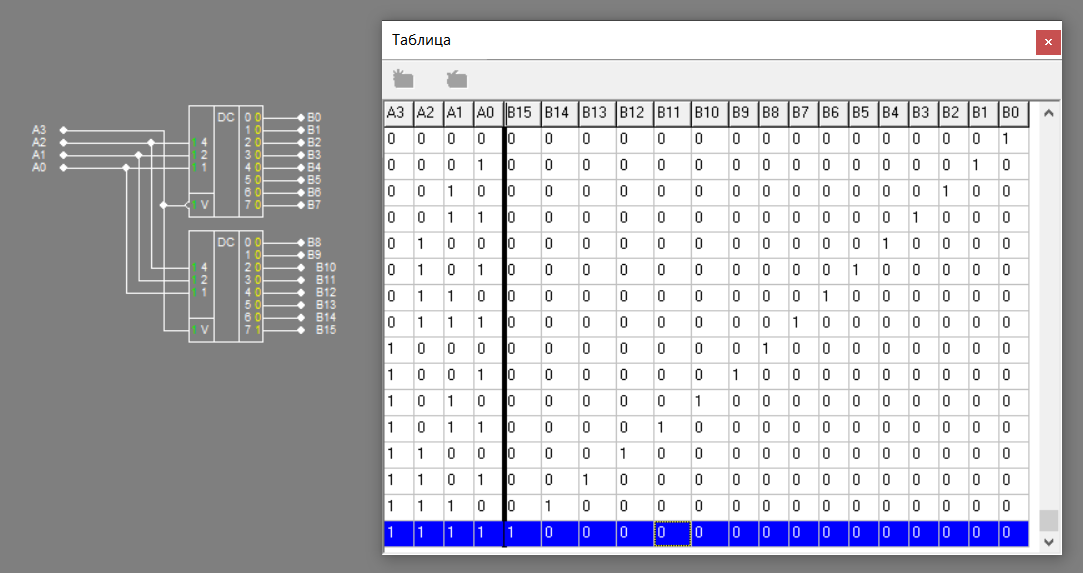


Рис. 5: Схема дешифратора 4x16 на 2-ох дешифраторах 3х8

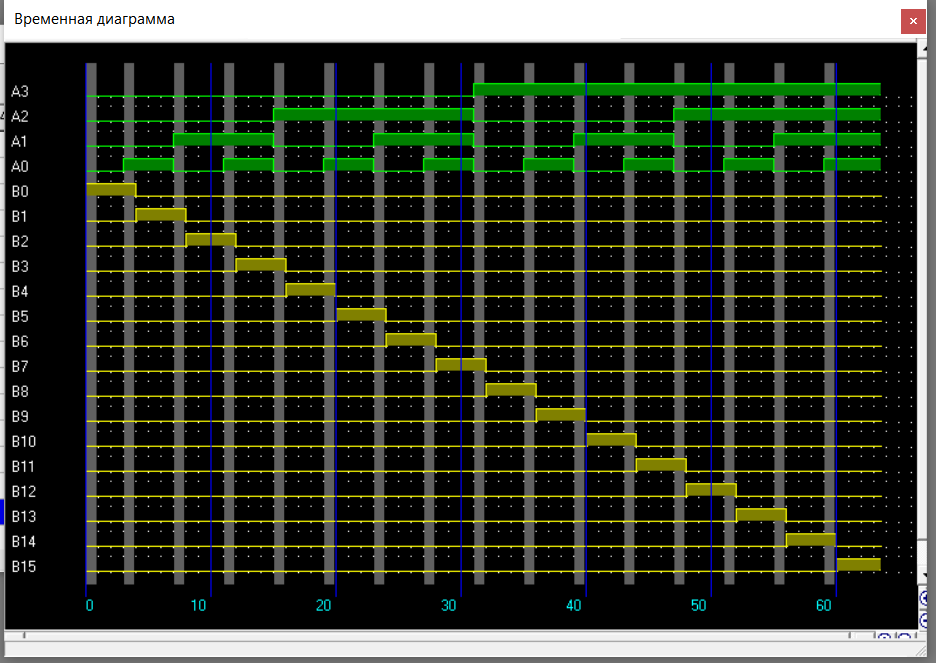


Рис. 6: Часова дiаграма входiв та виходiв дешифратора 4x16

**Побудова логiчної функцiї за допомогою дешифратора**

Дешифратори можуть використовуватися для спрощення побудови логiчних функцiй. Нехай є довiльна функцiя 4-ох змiнних, задана таблицею iстинностi [1.](#_bookmark3)

Таблиця 1 – Таблиця iстинностi функцiї

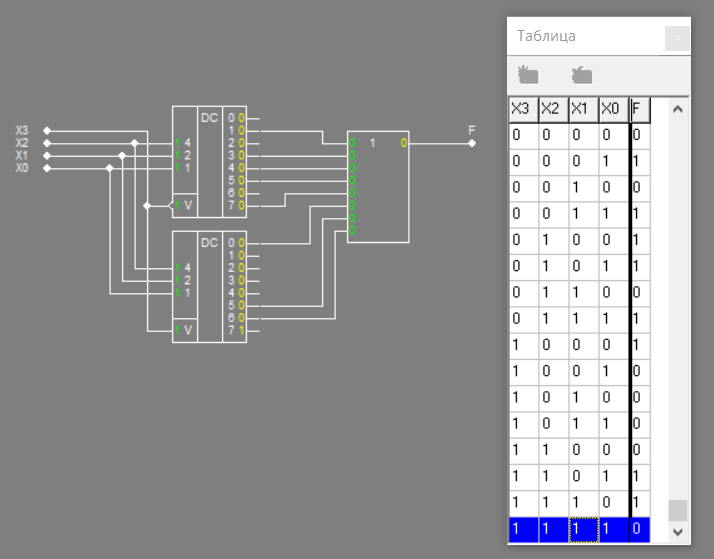
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *X*3 | *X*2 | *X*1 | *X*0 | F |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |

Продовження таблицi 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *X*3 | *X*2 | *X*1 | *X*0 | F |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

Дану функцiю можна записати у виглядi *F* = *{*1*,* 3*,* 4*,* 5*,* 7*,* 8*,* 13*,* 14*}* - набори на яких функцiя набуває логiчної 1. Тобто, якщо на входi подано один з цих наборiв, то F=1.

За такою логiкою була побудована схема, зображена на [рис.7,](#_bookmark7) а її часова дiаграма на [рис.8.](#_bookmark8) Можна помiтити, що отримана таблиця iстинностi збiгається з початковою, отже, схема була побудована правильно.



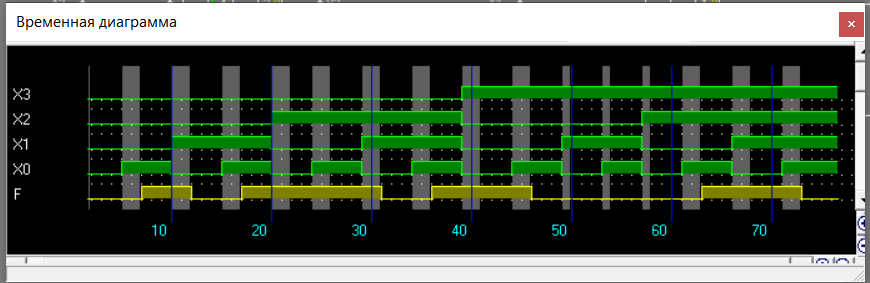
Рис. 7: Схема логiчної функцiї та її таблиця iстинностi

Рис. 8: Часова дiаграма входiв та виходiв логiчної функцiї

**Лабораторна робота № 3**

Лічильник

*Лічильник* - послідовнісна схема, яка перетворює поступаючі на вхід імпульси в паралельний двійковий код, відповідний їх кількості.

Лічильники виконуються на тригерах з рахунковим входом. Лічильник, який складається із n тригерів може порахувати 2n імпульсів. Число n визначає кількість розрядів двійкового числа, яке може бути записане в лічильник.

Лічильники характеризуються модулем (коефіцієнтом) підрахунку.

Один Т-тригер забезпечує коефіцієнт перерахунку М = 2, а n тригерів - М = 2n. При підрахунку імпульсів необхідно формувати перенесення з i-го в (i+1)-ий розряд за наступними правилами:

* Перенос СR з i-го в (i+1)-ий розряд формується, якщо у всіх розрядах з i-го по 0-й записана максимальна для даної системи числення цифра, при цьому розряди молодші (i+1)- го обнулюються. На прямих виходах тригерів цих розрядів Qi формується негативний перепад, а на інверсних - позитивний.
* Якщо в лічильнику використовуються тригери з прямим динамічним входом, то сигнал переносу в підсумовуючому лічильнику знімається з інверсних виходів попередніх тригерів, а якщо тригери з інверсним динамічним входом, то сигнал переносу береться з прямих виходів.

**Завдання:** Побудувати

* схему чотиророзрядного лічильника на JK тригерах.
* схему для чотиророзрядного лічильника із
* схему для чотиророзрядного лічильника із
* схему для чотиророзрядного лічильника із

1. Побудувати схему чотиророзрядного лічильника на JK тригерах:

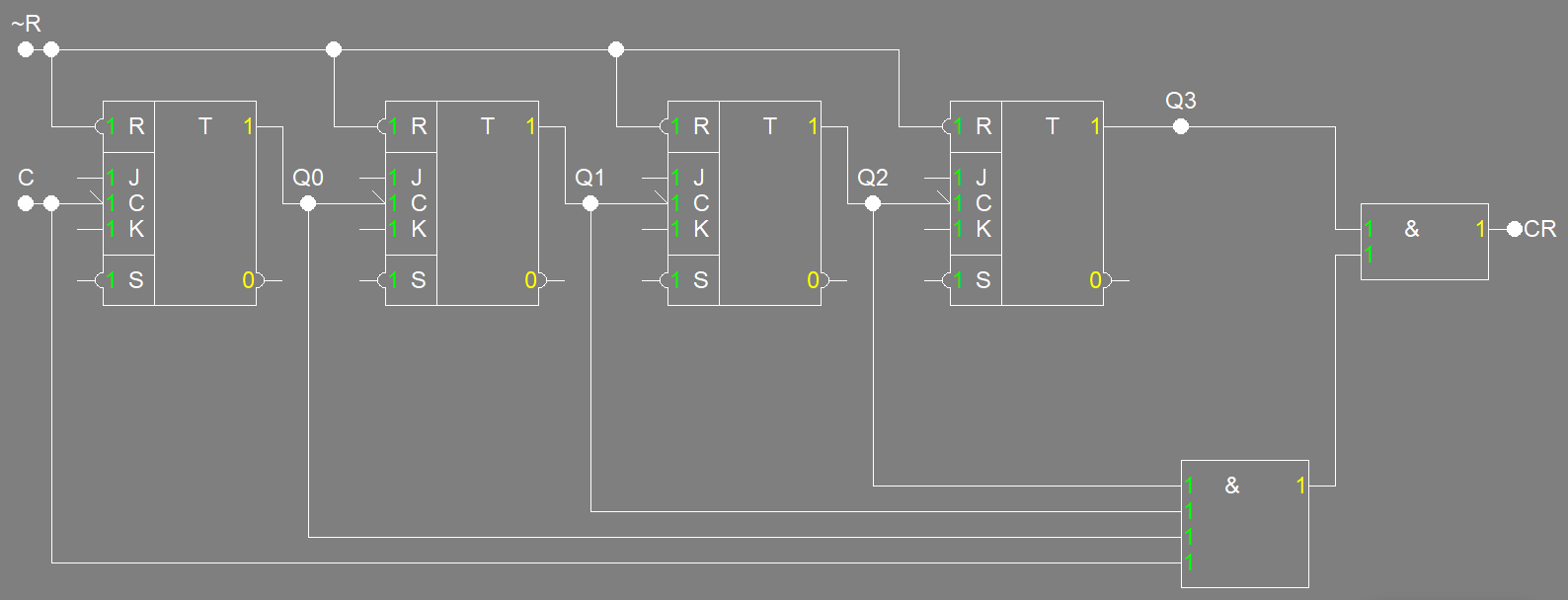


Рис 1: Схема чотиророзрядного лічильника на JK тригерах

Варто зауважити, що даний лічильних є асинхронний та складаться із 4 тригерів, тому може мати різних виходів, тобто може зберігати число від 0 до 16;

На наступних рисунках показано таблицю із максимальним та мінімальним значенням лічильника:

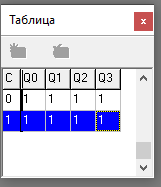
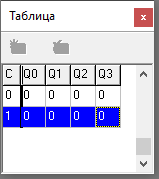
 

Рис 2: Макс. значення лічильника Рис 3: Мін. значення лічильника

2. Побудувати схему згідно та змоделювати роботу лічильника з коефіцієнтом підрахунку :

Для того щоб лічильних містити хоча б 10 чисел, він повинен мати 4 тригера.

У такому випадку ми зможемо зберігати чисел, але нам потрібно тільки 10.

Наш лічильник повинен досягати числа 10 та обнулятися при досягненні імпульсу 11. У двійковій системі числення .

Звідси, у момент часу коли лічильник міститиме число 11, на виходах 1, 3 та 4 встановиться висока напруга.

Ці три виходи ми під’єднуємо до логічного І. Тепер при досягненні 11, наш 3-І подасть сигнал для обнулення лічильника;

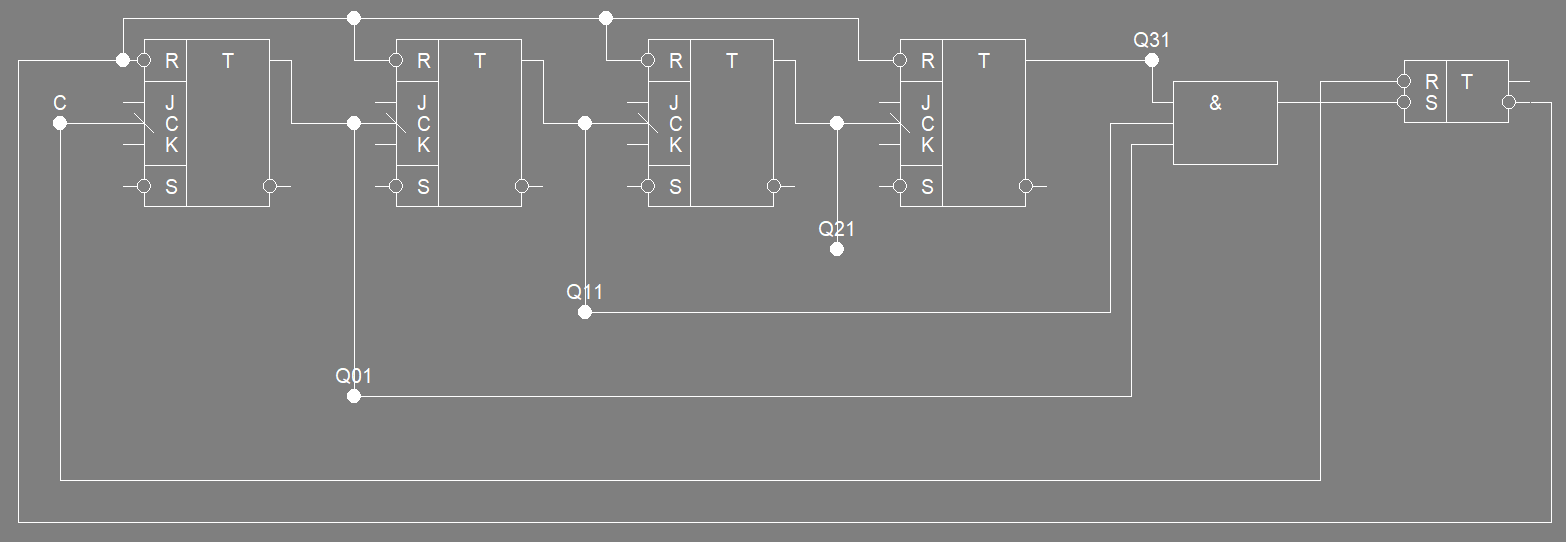


Рис 4: Схема для чотиророзрядного лічильника із

3. Побудувати схему згідно та змоделювати роботу лічильника з коефіцієнтом підрахунку :

Для того, щоб наший лічильний міг зберігати як мінумум 24 числа, ми повинні використати 5 тригерів.

Аналогічно попередньому завданню: для того, щоб дізнатися, виходи яких тригерів потрібно під’єднувати до І, перетворимо число 25 в двійкову систему числення ;

Будуємо схему для чотиророзрядного лічильника із :

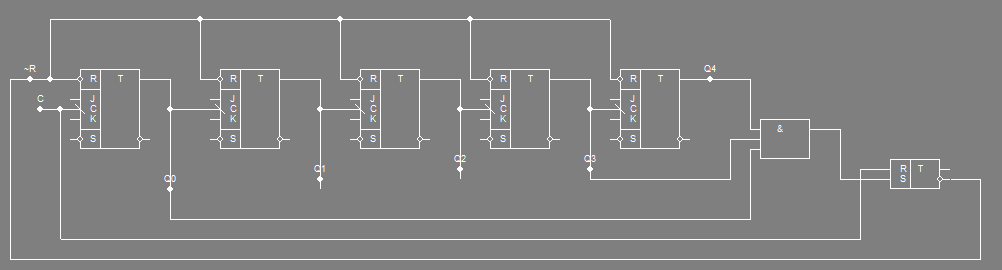


Рис 5: Схема для чотиророзрядного лічильника із

4. Побудувати схему згідно та змоделювати роботу лічильника з коефіцієнтом підрахунку :

Для того, щоб наший лічильний міг зберігати як мінумум 43 числа, ми повинні використати 6 тригерів.

Аналогічно попередньому завданню: для того, щоб дізнатися, виходи яких тригерів потрібно під’єднувати до І, перетворимо число 44 в двійкову систему числення ;

Будуємо схему для чотиророзрядного лічильника із :

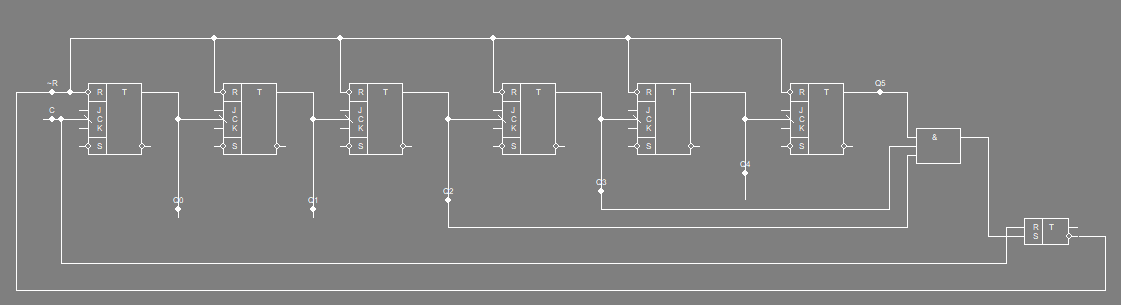


Рис 6: Схема для чотиророзрядного лічильника із

**Лабораторна робота № 4**

Напівсуматори. Суматори.

Напівсуматор — логічна схема з двома входами та двома виходами, яка виконує операцію арифметичного додавання двох однорозрядних чисел A та B з формуванням виходу переносу.

У напiвсуматора повинно бути 2 входи (A, B) i 2 виходи (S, *Сout*). А i В – це власне числа, якi додаються, S - результат додавання, який залишається в

поточному розрядi, *Сout* – перенесенння в наступний розряд.

Таблиця 1 – Таблиця iстинностi напівсуматора.

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | *Сout* | S |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Суматор

Суматор – це логічна схема, яка може додавати два однорозрядних двійкових числа з переносом з попереднього розряду.

У напiвсуматора немає входу, який би вiдповiдав за перенесення з

молодшого розряду. Це в свою чергу унеможливлює його використання для побудови суматорiв бiльших розрядностей, тому для побудови суматорiв бiльших розрядностей використовується суматор.

У суматора повинно бути 2 входи (A, B) i 2 виходи (S, *Сout*). А i В – це власне числа, якi додаються, S - результат додавання, який залишається в

поточному розрядi, *Сout* - перенесення в наступний розряд.

Також повинен бути додатковий вхiд, який вiдповiдає за перенесення з молодшого розряду *Cin*.

Таблиця істинності суматора виглядає наступним чином.

Таблиця 2 – Таблиця iстинностi суматора

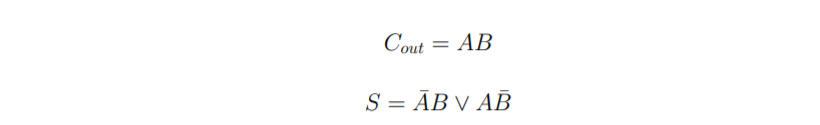
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | *Cin* | *Сout* | S |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

**Завдання:** Побудувати

* напівсуматор
* повний суматор для обрахунку мінімізованих функцій
* 2-ох розрядний суматор

1. Побудувати напівсуматор.

S i *Сout* будуть задаватися наступними логiчними виразами:

Для побудови схеми напiвсуматора був обраний базис I-НI:

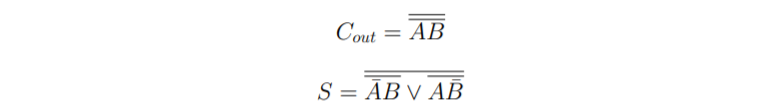


Схема напiвсуматора та її таблиця iстинностi зображенi на [рис.1,](#_bookmark2) а часова дiаграма її входiв та виходiв на [рис.2.](#_bookmark3)

Помітно, що кінцева таблиця істинності співпадає з початковою, отже, схема була побудана коректно.

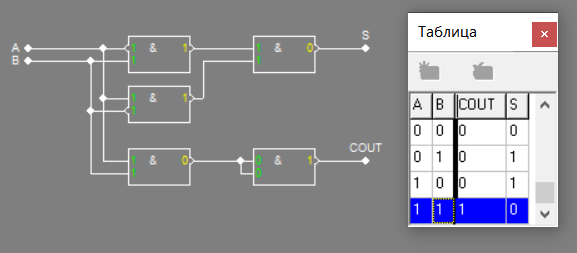


Рис. 1: Схема i таблиця iстинностi напiвсуматора

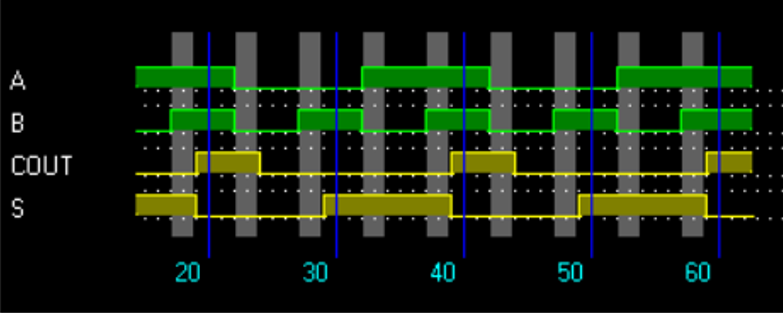


Рис. 2: Часова дiаграма входiв та виходiв напiвсуматора

2. Побудувати повний суматор для обрахунку мінімізованих функцій

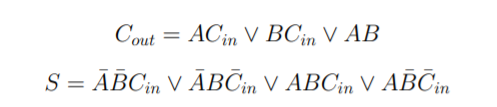
Для отримання мiнiмiзованих функцiй для *Сout* та S були використанi карти Карно.

Карти Карно були побудовані на основі таблиці істинності суматора.

Таблиця 3 – Карта Карно для *Сout* Таблиця 4 – Карта Карно для S

|  |  |  |
| --- | --- | --- |
| *AB∖Cin* | 0 | 1 |
| 00 | 0 | 0 |
| 01 | 0 | 1 |
| 11 | 1 | 1 |
| 10 | 0 | 1 |

|  |  |  |
| --- | --- | --- |
| *AB∖S* | 0 | 1 |
| 00 | 0 | 1 |
| 01 | 1 | 0 |
| 11 | 0 | 1 |
| 10 | 1 | 0 |



Для суматора був вибраний базис I-НI. Застосувавши правило де Моргана, отриманi наступнi вирази.

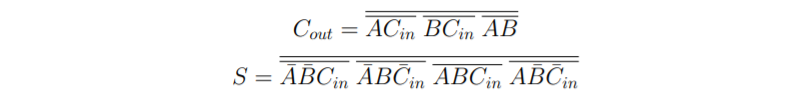


Схема суматора та її таблиця iстинностi зображенi на [рис.3,](#_bookmark4) а часова дiаграма її входiв та виходiв на [рис.4.](#_bookmark5)

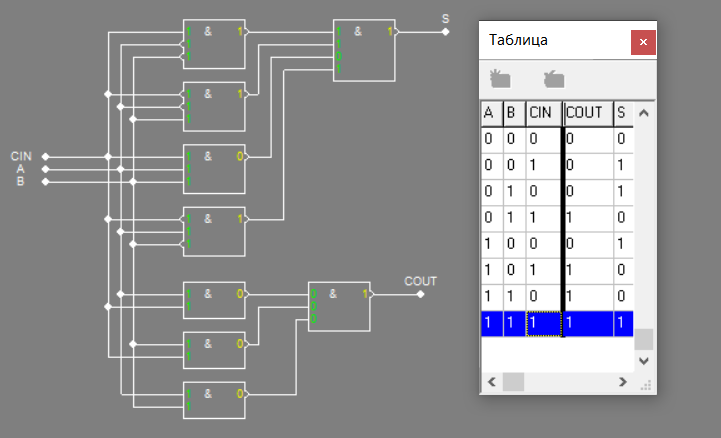
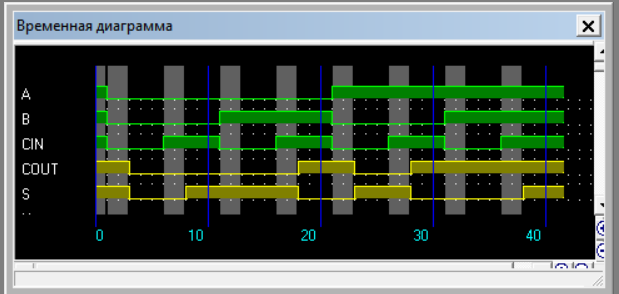


Рис. 3: Схема i таблиця iстинностi суматора

 Рис. 4: Часова дiаграма входiв та виходiв суматора

3. Побудова 2-ох розрядного суматора

Для побудови 2-ох розрядного суматора, можна використати 2 однорозрядних повних суматори, з’єднавши їх послiдовно. Тобто, на вхiд *𝐶𝑖𝑛* одного суматора потрiбно подати результат виходу *𝐶𝑜𝑢𝑡* iншого, що i було реалiзовано у схемi (рис. [5](#_bookmark6) ). У схемi вважається, що було перенесення з молодшого розряду (на вхiд *𝐶𝑖𝑛* першого суматора подається 1). Часова дiаграма схеми зображена на [рис.6.](#_bookmark7)

Також, був побудований 2-ох розрядний суматор на основi складеного в попередньому завданнi повного суматора. Його схема та часова дiаграма зображенi на [рис.](#_bookmark8)7 та [рис.](#_bookmark9)8 вiдповiдно.

Таблицi iстинностi обох складених суматорiв iдентичнi.

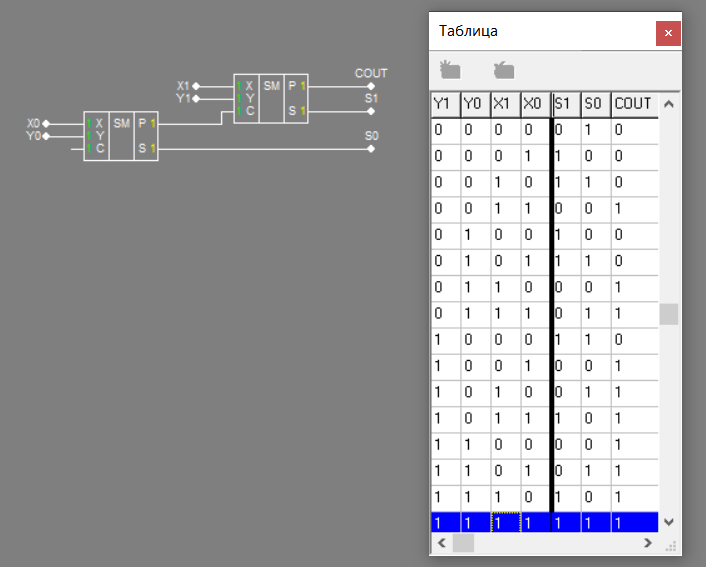


Рис. 5: Схема i таблиця iстинностi 2-ох розрядного суматора

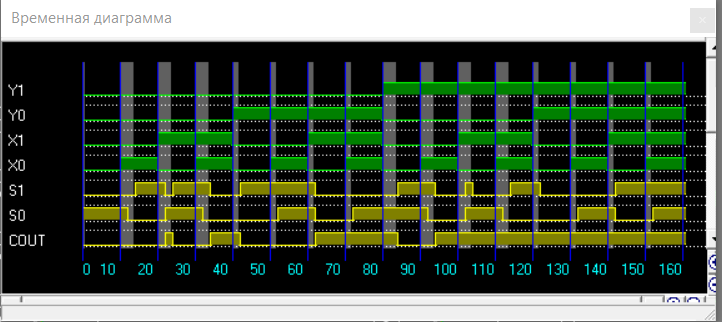


Рис. 6: Часова дiаграма входiв та виходiв 2-ох розрядного суматора

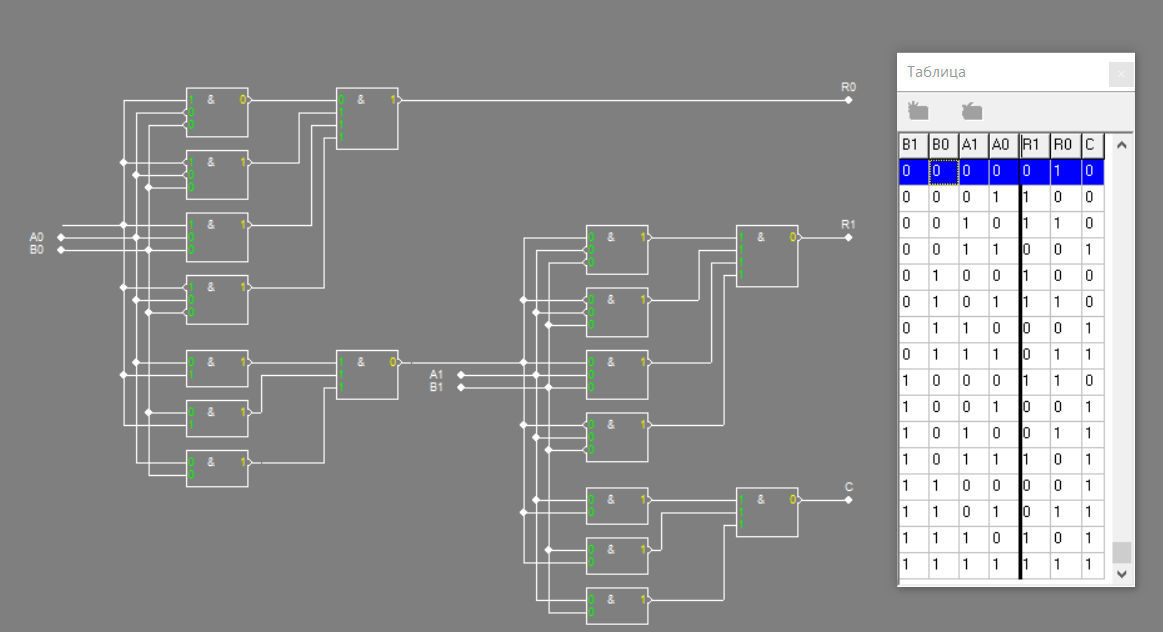


Рис. 7: Схема i таблиця iстинностi 2-ох розрядного суматора

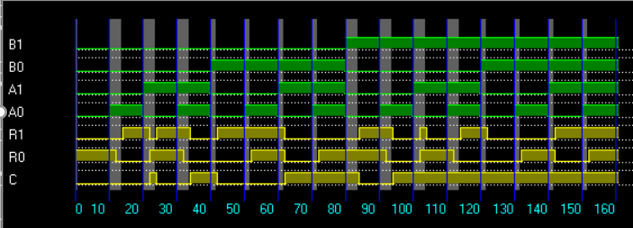


Рис. 8: Часова дiаграма входiв та виходiв 2-ох розрядного суматора

Висновок:

У результатi даної лабораторної роботи, було дослiджено роботу напiвсуматора та суматора. Побудовано їхнi схеми в базисi I-НI, таблицi iстинностi та часовi дiаграми.

Складено схему 2-ох розрядного суматора на основi двох мiкросхем однорозрядного суматора наявних в пакетi AFDK, використавши послiдовне з’єднання, а також на основi самостiйно складених повних однорозрядних суматорiв.

Правильнiсть роботи суматорiв перевiрено за допомогою таблиць iстинностi та часових дiаграм.

**Лабораторна робота № 5**

Множення чисел

При множенні чисел у прямих кодах знакові та основні розряди обробляються роздільно. Для визначення знака добутку здійснюють підсумовування за модулем 2 цифр, записаних в знакових розрядах співмножників. Будемо вважати, що множене Y і множник Х – правильні двійкові дроби виду , , де

двійкові розряди xi, yi ∈ {0,1}. Тоді добуток Z модулів чисел дорівнює

 (1.1)

Множення *Y* і *Х* може бути реалізоване шляхом виконання циклічного процесу, характер якого залежить від конкретної форми виразу (1.1). Один цикл множення складається з додавання чергового часткового добутку, що представляє собою добуток множеного на одну цифру множника, до суми часткових добутків. Розрізняють чотири способи множення.

*Перший спосіб множення*

Вираз (1.1) можна представити у вигляді

 .

Звідси випливає, що отримані суми часткових добутків в *i*-му циклі (*i*= ) зводяться до обчислення



з початковими значеннями *i*=1, *Z*0=0, причому *Zn=Z=YX.*

Множення здійснюється з молодших розрядів множника, сума часткових добутків зсувається вправо, а множене залишається нерухомим.

*Другий спосіб множення.*

Запишемо (1.1) у вигляді



Очевидно, що процес множення може бути зведений до n-кратного виконання циклу



з початковими значеннями *i*=1, *Y*0*=Y*2*-n* , *Z0*=0. Множення здійснюється з молодших розрядів, множене зсувається вліво, а сума часткових добутків залишається нерухомою.

*Третій спосіб множення.*



Суму часткових добутків у i-м циклі (i= ) можна одержати по формулі



Початковими значеннями є i=1, Z0=0. Множення здійснюється зі старших розрядів множника, сума часткових добутків зсувається вліво, а множене нерухоме.

*Четвертий спосіб множення.*



Процес множення може бути зведений до *n*-кратного виконання циклу



з початковими значеннями *i*=1, *Y*0=*Y*2-1, *Z*0=0.

Множення виконується зі старших розрядів множника, сума часткових добутків залишається нерухомою, а множене зсувається вправо.

Принцип побудови пристроїв, що реалізують різні способи множення, показаний на рис. 1.1, де *RG*3 – регістр множеного, *RG*1 – регістр добутку, *RG*2 – регістр множника. Цифрами зазначені номери розрядів *SM* і регістрів, а стрілками показаний напрямок зсуву кодів у регістрах.

*а б*



*в г*

Рис. 1.1: Операційні схеми пристроїв для множення чисел: *а* – перший спосіб; *б* – другий спосіб; *в* – третій спосіб; *г* – четвертий спосіб

Під час множення *першим способом* (рис. 1.1, *а*) в першому такті *i*-го циклу аналізується значення *RG*2[1] – молодшого (*n*-го) розряду регістру *RG*2, в якому знаходиться чергова цифра множника. Вміст *RG*3 додається до суми часткових добутків, що знаходяться в регістрі *RG*1, якщо *RG*2[1]=1, або не додається, якщо *RG*2[1]=0. В другому такті здійснюється правий зсув у регістрах *RG*1 і *RG*2, що еквівалентно множенню їхнього вмісту на 2–1. При зсуві цифра молодшого розряду регістру *RG*1 записується у вивільнюваний старший розряд регістру *RG*2. Після виконання *n* циклів молодші розряди 2*n*-розрядного добутку будуть записані в регістр *RG*2, а старші – у *RG*1.

Перед початком множення *другим способом* (рис. 1.1, *б*) множник *Х* записують в регістр *RG*2, а множене *Y* – в молодші розряди регістру *RG*3 (тобто в регістрі *RG*3 установлюють *Y*0 = *Y*2*–n*). В кожному *i*-му циклі множення додаванням кодів *RG*3 і *RG*1 управляє цифра *RG*2[1], а в регістрі *RG*3 здійснюється зсув вліво на один розряд, в результаті чого формується величина *Yi* = 2*Yi–*1. Оскільки сума часткових добутків в процесі множення нерухома, зсув в регістрі *RG3* можна виконати суміщення в часі з підсумовуванням (як правило, *t*п ≥ *t*з). В цьому випадку *t*м = *nt*п. Завершення операції множення визначається за нульовим вмістом регістру *RG*2, що також приводить до збільшення швидкодії, якщо множник ненормалізований.

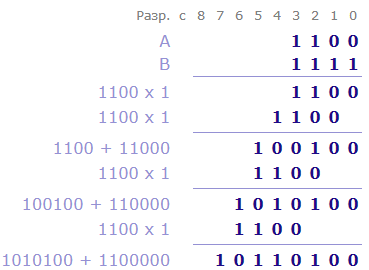
При множенні *третім способом* (рис. 1.1, в) множник *Х* записується в старші розряди *RG*2, при цьому *RG*2[1]=0. Вага молодшого розряду *RG*3 дорівнює 2–2*n*, тому код в регістрі *RG*3 являє собою значення *Y*2–*n*. В кожному циклі множення підсування виконується при *RG*2[*n+*1]=1. В регістрах *RG*1 і *RG*2 виконується лівий зсув. В результаті підсумовування вмісту *RG*3 і *RG*1 може виникнути перенос в молодший розряд регістру *RG*2, що реалізується на *SM.* Збільшення довжини *RG*2 на один розряд усуває можливість поширення переносу в розряди множника. Після виконання *n* циклів молодші розряди добутку будуть знаходитися в регістрі *RG*1, а старші – в регістрі *RG*2. Час множення третім способом визначається аналогічно першому способу.

Перед множенням *четвертим способом* (рис. 1.1, г) множник записують в регістр *RG*2, а множене – в старші розряди регістру *RG*3 (тобто в *RG*3 установлюють *Y*0=*Y*2–1). В кожнім циклі цифра *RG*2[*n*+1], що знаходиться в старшому розряді регістру *RG*2, управляє підсумовуванням, а в *RG*3 здійснюється правий зсув на один розряд, що еквівалентно множенню вмісту цього регістра на 2–1. Час виконання множення четвертим способом складає *tм*=*nt*п, визначається аналогічно другому способу.

**Завдання:** Виконати множення чисел першим способом.

Перемножимо числа 12 та 15. 12 \* 15= 180.

12 у двійковій системі 1100, 15 – 1111.



При перемноженні виходить 10110100, тобто 18.

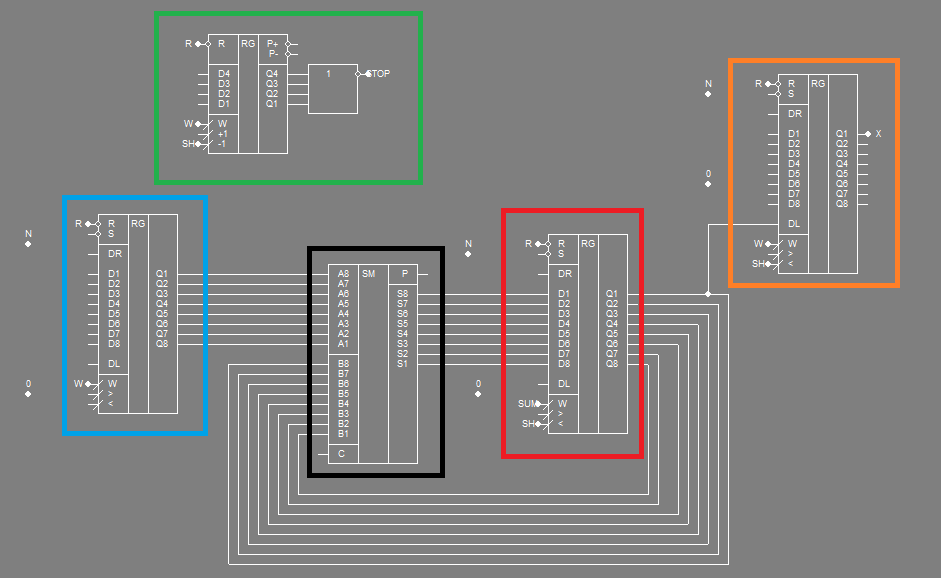


Рис 2.1: Схема множення першим способом

Розглянемо дану схему (рис 2.1) детальніше. Зеленим кольором обведено лічильник, який слугує для підрахунку етапів множення і повідомляє коли ми повинні припинити множення. Кількість етапів дорівнює максимальній розрядності із чисел А та В. У даному випадку це 5.

Синім та помаранчевим кольором обведено регістри в які ми записуємо наші числа. Чорним кольором обведено суматор. Червоним кольором обведено регістр, у який записується результат.

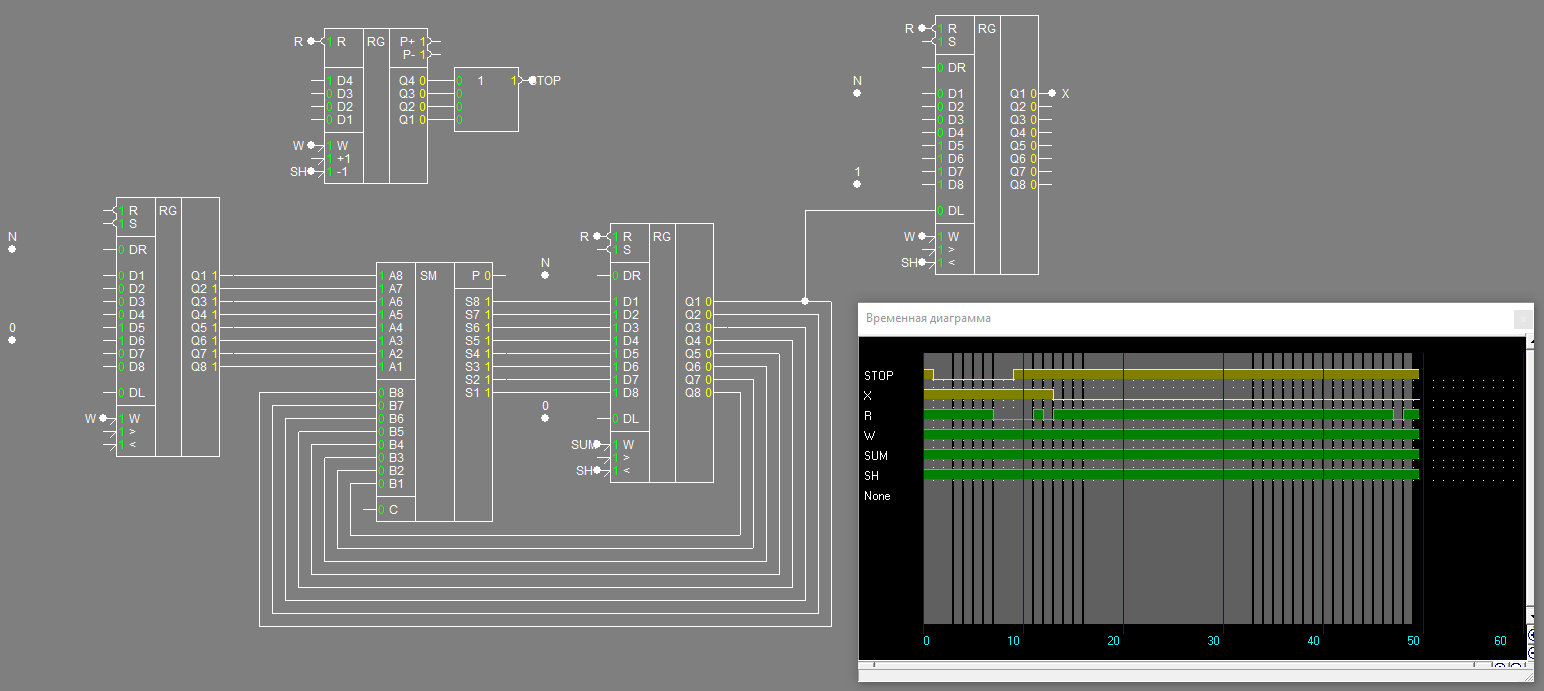


Рис 2.2: Початковий стан схеми

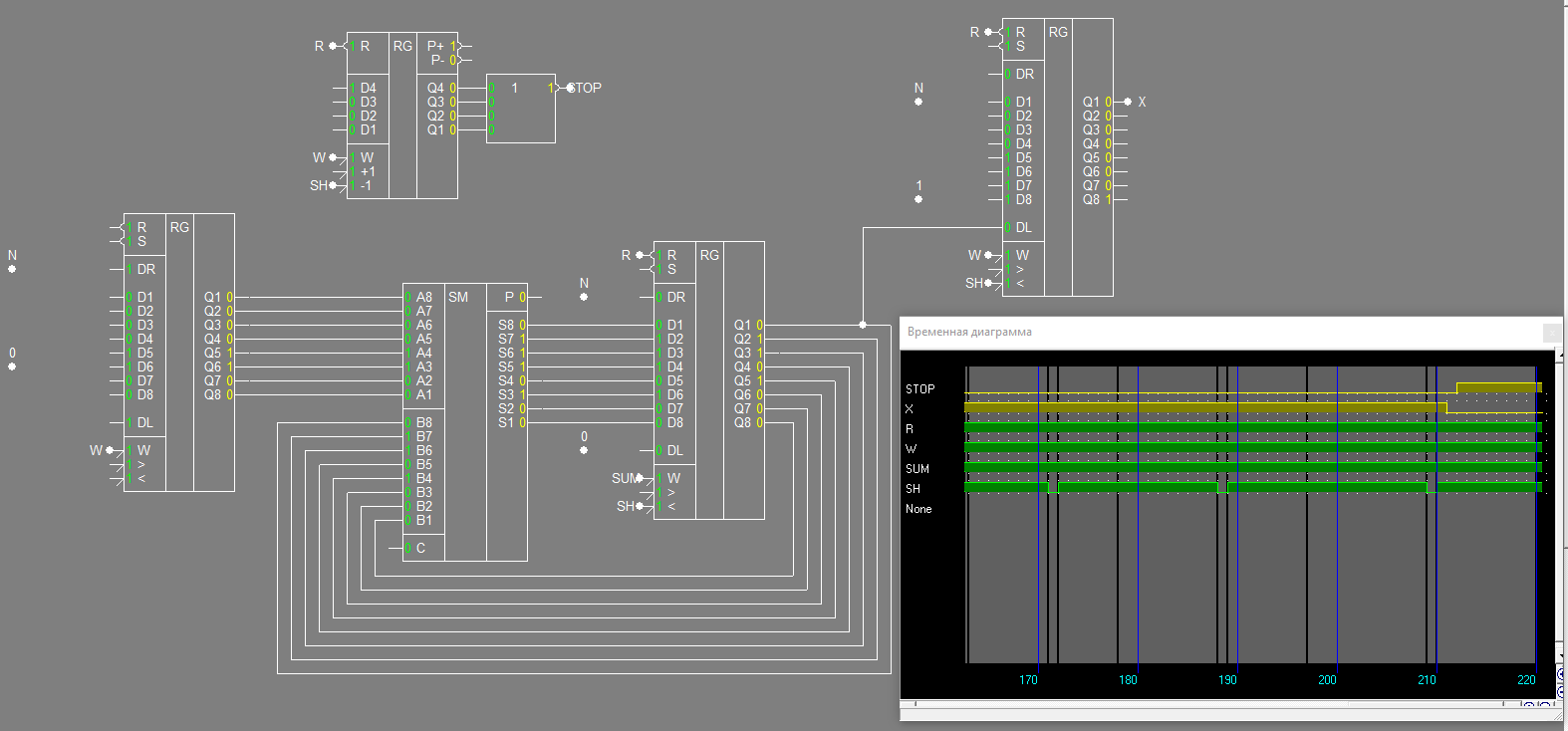


Рис 2.3: Стан схеми після множення

Поглянемо на результат: у результуючому регістрі (на рис 2.1 червоний) має бути записана відповідь.

 RG1=00010110, це трохи не сходиться із тим, що ми отримали на листочку. В наслідок останнього зсуву частина відповіді просто перемістилась на виходи регістра (на рис 2.1 помаранчевий).

Отже, ми отримали наступну відповідь: 0, 10110100 яка є правильною.

На діаграмі можна побачити такі режими:

* X – показує, коли потрібно виконати режим сумування (SUM).
* R – режим скидання.
* W – режим запису.
* SH – режим зсуву.
* SUM – режим сумування.
* STOP – показує, коли множення було закінчено.

Прокоментую часову діаграму: спочатку, натискаючи на R, я скидаю стан системи (STOP = 0 – низька напруга; висока напруга STOP буде означати те, що ми вже закінчили множення). Починаю вмикати режим зсування SH до тих пір, поки на Х не з’явиться висока напруга. Саме тепер можна почати сумування, натискаючи на SUM. Після цього виконую скидання, натискаючи на SH. Повторюю процес, поки на STOP не з’явиться висока напруга.

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

**Лабораторна робота № 6**

Функціональна схема для реалізації обчислювальної функції

**Завдання:**

Обчислити вираз: (a \* b) + (a + b) = y

а та b обрати за варіантом (таблиця 1).

|  |  |  |
| --- | --- | --- |
| **Варіант** | **a** | **b** |
| 1 | 1 | 15 |
| 2 | 2 | 13 |
| 3 | 3 | 11 |
| 4 | 4 | 9 |
| 5 | 5 | 7 |
| 6 | 6 | 5 |
| 7 | 7 | 3 |
| 8 | 8 | 1 |
| 9 | 9 | 14 |
| 10 | 10 | 12 |
| 11 | 11 | 10 |
| 12 | 12 | 8 |
| 13 | 13 | 6 |
| 14 | 14 | 4 |
| 15 | 15 | 2 |
| 16 | 1 | 14 |
| 17 | 2 | 12 |
| 18 | 3 | 10 |
| 19 | 4 | 8 |
| 20 | 5 | 6 |
| 21 | 6 | 4 |
| 22 | 7 | 2 |
| 23 | 8 | 15 |
| 24 | 9 | 13 |
| 25 | 10 | 11 |
| 26 | 11 | 9 |
| 27 | 12 | 5 |
| 28 | 13 | 3 |
| 29 | 14 | 1 |
| 30 | 15 | 7 |

**Таблиця** 1- розподіл значень а та b обрати за варіантами

Перемножу числа a = 7, b = 15.

Продемонструю схему для обчислення виразу (рис 1.)

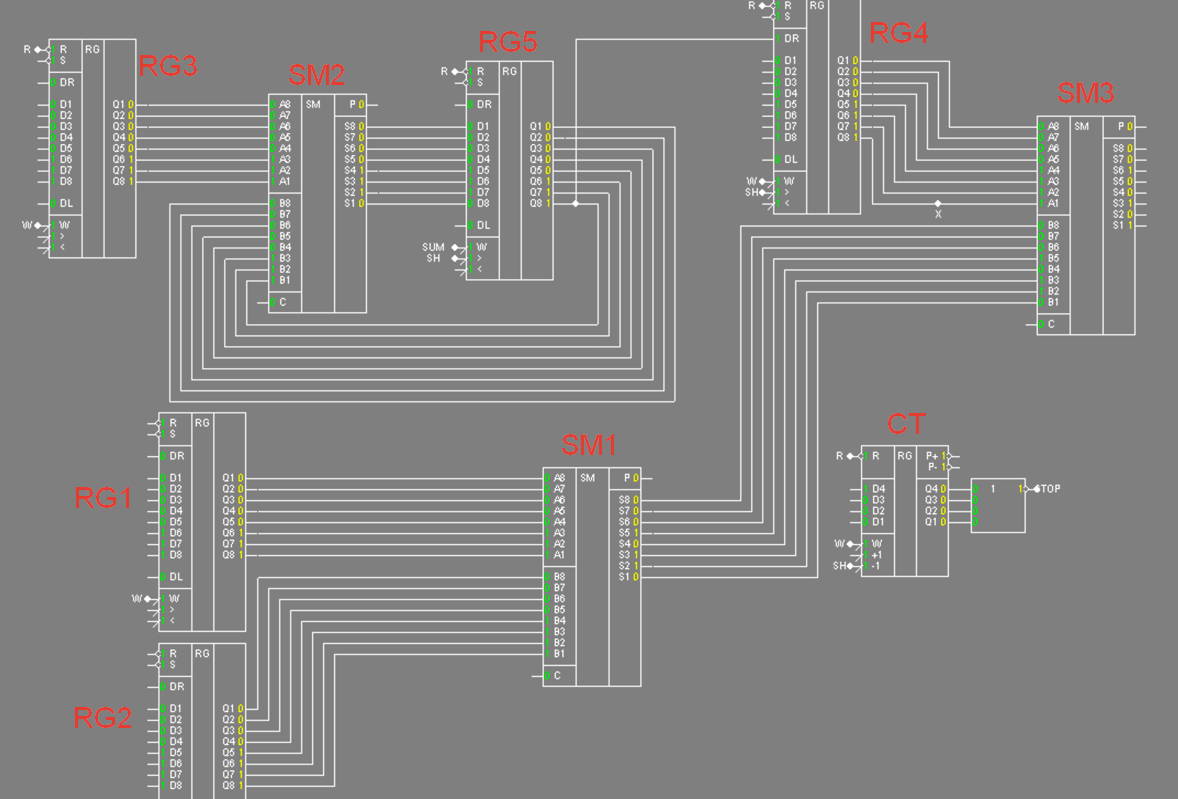


Рис 1: Схема для обчислення виразу

Число А буде записуватись в регістри: **RG1, RG3**.

Число В – в регістри: **RG2, RG4** .

**SM1** – суматор з результатом суми числа А з числом В (А+В)

**RG5, RG4**  - регістри в яких буде зберігатись результат множення чисел (А\*В)

**SM3** – суматор з результатом виразу

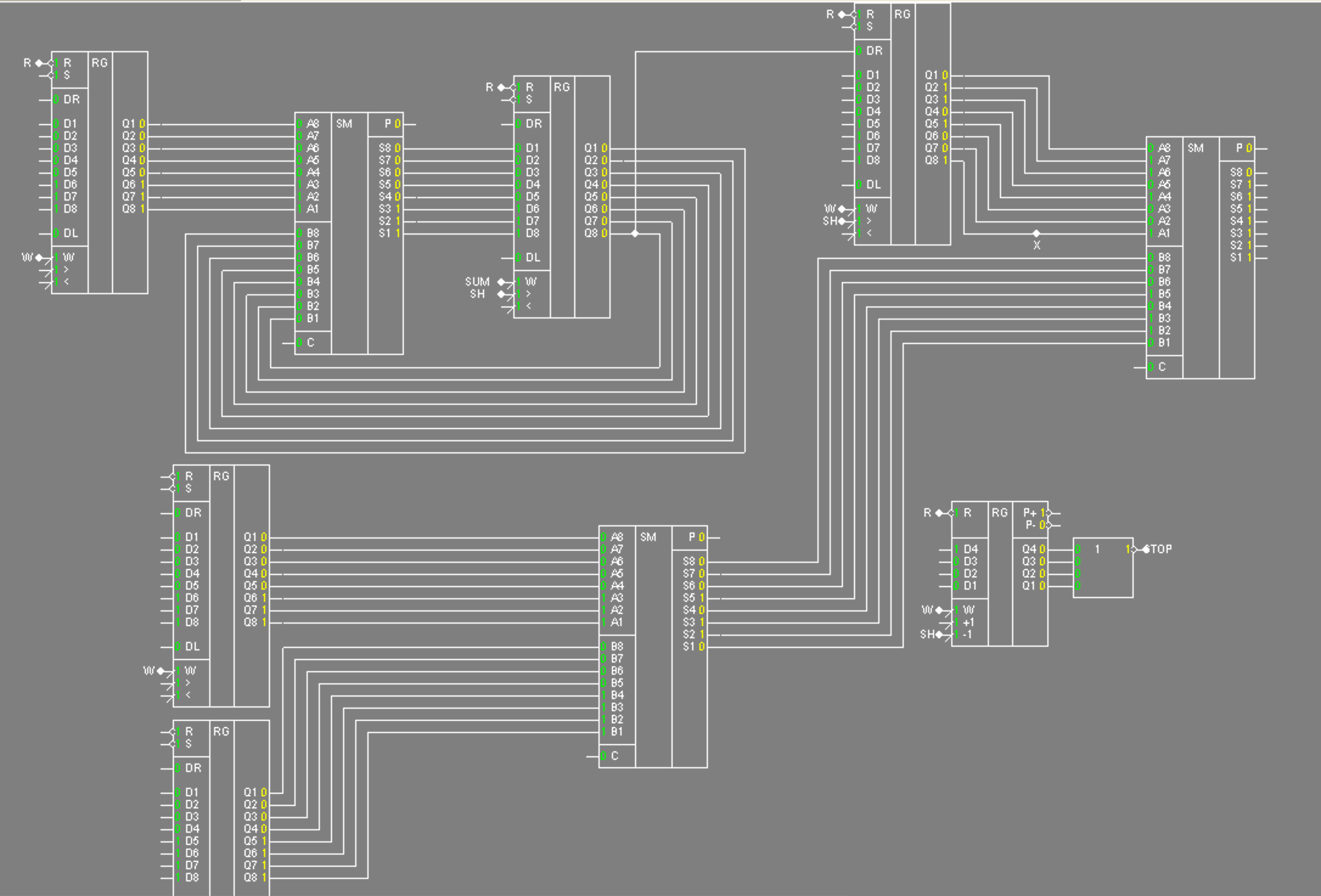


Рис 2: Стан схеми після обчислення виразу.

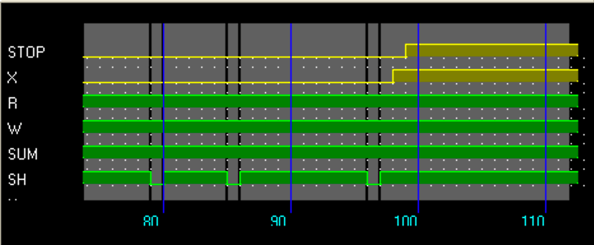
****

Рис 3: Часова діаграма

На рисунку 2 в **SM3** можна побачити результат.

Перевіримо результат на правильність:

(7 \* 15) + (7 + 15) = 127

Переведемо 127 в двійкому систему числення

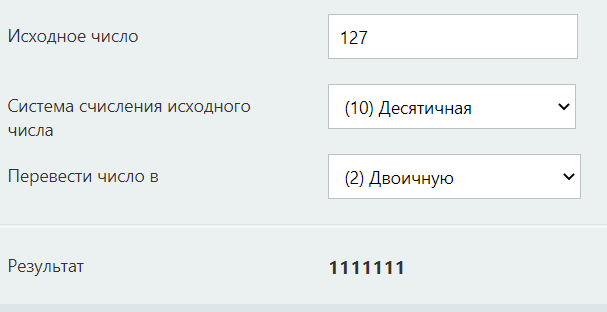


Рис 4: 127 в двійковій системі числення

Саме таке значення записано в **SM3**, отже обрахунки були проведені коректно.

На діаграмі (рис. 3) можна побачити такі режими:

* X – показує, коли потрібно виконати режим сумування (SUM).
* R – режим скидання.
* W – режим запису.
* SH – режим зсуву.
* SUM – режим сумування.
* STOP – показує, коли множення було закінчено.

Прокоментую часову діаграму: спочатку, натискаючи на R, ми скидаємо стан системи (STOP = 0 – низька напруга; висока напруга STOP буде означати те, що ми вже закінчили операцію). Натиснемо на W, щоб записати числа. Починаємо вмикати режим зсування SH до тих пір, поки на Х не з’явиться висока напруга.

Після того, як на Х з’явилась висока напруга, натиснемо на SUM, після цього подивимося на **SM1** (суматор з результатом суми числа А з числом В (А+В)) та на **SM3** (суматор з результатом виразу), і побачимо, що значення в цих суматорах змінилось, адже суматори почали записувати суму початкових чисел. Після сумування, потрібно зробити зсув чисел, натискаючи SH, і після цього знову натиснемо на SUM.

Також, після кожного зсуву змінюється значення в регістрах **RG4** та **RG5**, які зберігають результат множення чисел, адже ми змінюємо числа, тому і змінюється результат множення.

Повторюємо операції зсуву та сумування до тих пір, поки на STOP не з’явиться висока напруга, це означає, що операція закінчилась, і в суматорі **SM3** з’явиться остаточне значення, яке дорівнює результату виразу.