

В. В. Булатецький, Л. В. Булатецька

Логічні елементи та основні операційні вузли

Лабораторний практикум
з дисципліни “Архітектура ЕОМ”
для студентів спеціальності “Інформатика”

УДК 004.2(076.5)
ББК 32.973.26-02я7
Б 90

Даний практикум призначений для студентів спеціальності “Інформатика” для оволодіння теоретичними та практичними навиками побудови основних компонентів персонального комп’ютера шляхом вивчення та застосування основних методів проектування операційних вузлів обчислювальної системи та їх складових.

Дані методичні рекомендації є логічним доповненням до лекційного курсу і можуть слугувати в якості послідовних вказівок при виконання лабораторних робіт в процесі вивчення дисципліни “Архітектура ЕОМ”. Для виконання завдань рекомендовано використовувати програмний пакет для проектування принципів електричних схем “Electronic Workbench”, проте для різних платформ можливо також застосовувати інші програмні продукти з аналогічними можливостями, наприклад “GPL Electronic Design Automation”, “Spice3”, “XCircuit” тощо.

На початку подано короткий огляд програмного продукту “Electronic Workbench”, на який орієнтовано даний практикум. Кожна лабораторна робота містить теоретичні відомості, практичні завдання до виконання роботи та запитання для самоконтролю. В кінці подано список рекомендованої літератури.

Ці лабораторні роботи можуть бути також використані студентами інших спеціальностей, які вивчають мікроелектроніку, автоматизацію та обчислювальну техніку.

Рецензенти:

Чекурін В.Ф. - доктор фіз.-мат. наук, професор, зав. відділом інституту прикладних проблем математики і механіки ім. М.Я. Підстригача НАН України;

Пех П. А. - кандидат технічних наук, доцент кафедри прикладної математики ЛДТУ;

Мірошніченко І.Г. - канд. пед. наук, проф. кафедри загальної фізики та методики викладання фізики ВДУ імені Лесі Українки;

© Булатецький В.В., Булатецька Л.В.

ЗМІСТ

Вступ.....	4
Лабораторна робота №1. Проектування комбінаційних схем	11
Лабораторна робота №2. Дешифратор. Шифратор	12
Лабораторна робота №3. Мультиплексор. Демультиплексор	26
Лабораторна робота №4. Перетворювачі кодів	32
Лабораторна робота №5. Цифровий компаратор	36
Лабораторна робота №6. Проектування суматорів	41
Лабораторна робота №7. Проектування та дослідження тригерів	46
Лабораторна робота №8. Проектування регістрів	55
Лабораторна робота №9. Проектування лічильників	58
Література	64

ВСТУП

На сьогодні серед існуючих радіотехнічних і радіоелектронних імітаційних, моделюючих і схемотехнічних програмних засобів виділяється програмний пакет Electronics Workbench фірми Interactive Image Technologies, призначений для моделювання аналогових і цифрових радіоелектронних пристроїв різного призначення, які представляють собою віртуальну радіоелектронну лабораторію (Electronics lab in computers). Така лабораторія включає бібліотеки великого числа стандартних радіоелектронних елементів: перемикачів і реле, резисторів і конденсаторів, котушок індуктивності і трансформаторів, діодів і транзисторів, тиристорів, світлодіодів та індикаторів, ОП і компараторів, АЦП і ЦАП, різних логічних елементів, інтегральних мікросхем і т.п. типових елементів, з'єднаних в групи. Необхідно відзначити, що є можливість вибрати модель такого компонента (ідеального або реального з бібліотеки) або створити її самому. Окремо виділені популярні серії аналогових і цифрових мікросхем, а також різні функціональні вузли обробки сигналів (блоки диференціювання, інтеграції, перемножування, частотної фільтрації і т.п.).

Основні групи (рис. а).

До таких груп відносяться: Sources (джерела струму і напруги), Basic (основні пасивні елементи), Diodes (діоди), Transistors (транзистори), Analog Ics (аналогові мікросхеми), Mixed Ics (мікросхеми змішаного типу), Digital Ics (цифрові мікросхе-

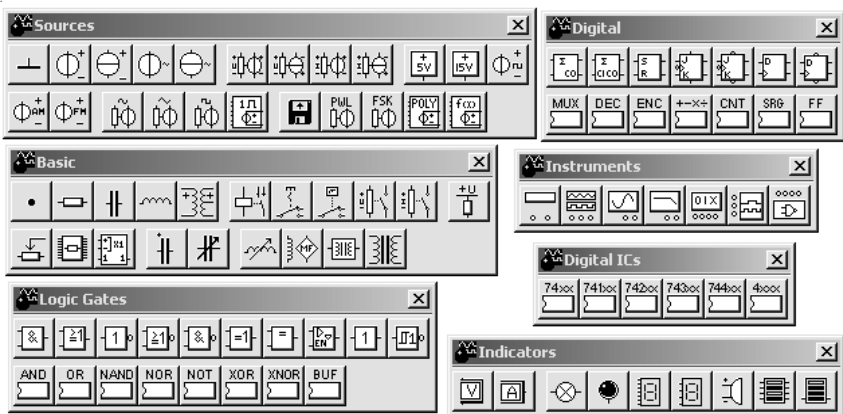


Рис. а.

ми), Logic Gates (логічні елементи), Digital (цифрові елементи), Indicator (індикаторні пристрої), Control (аналогові функціональні пристрої) Miscellaneous (компоненти змішаного типу), Instruments (контрольно-вимірювальні пристрої) і спеціальна група Favorites (бібліотека власноруч створених елементів).

Робоче вікно (рис. b).

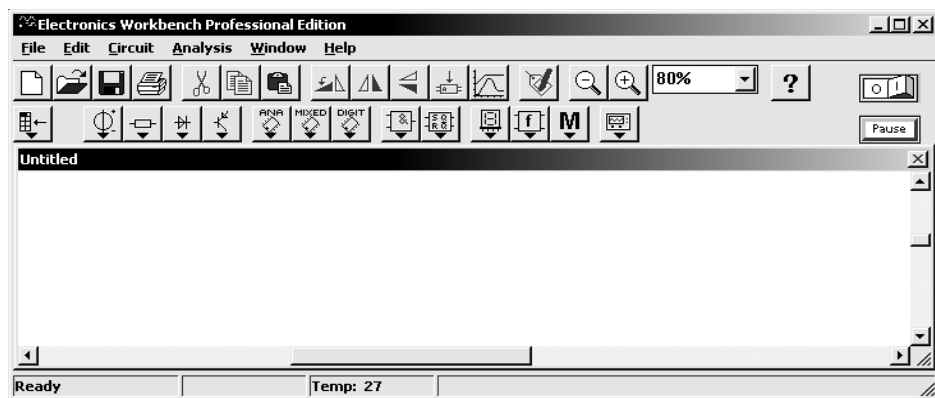


Рис. b.

Набір устаткування включає стандартні прилади з віртуальними панелями управління та індикації: функціональний генератор, логічний аналізатор, перетворювач логічних сигналів (Logic Converter), цифровий генератор слів (Word Generator). і контрольно-тестове обладнання із спрощеною формою управління та індикації: генератори синусоїдальних і прямокутних сигналів, різні джерела струму і напруги, амперметри, вольтметри, і ін. інструменти.

Цифровий(логічний) аналізатор (рис. c).

Логічний аналізатор відображає сигнали, які надходять як прямокутні імпульси, що розгортаються по осі часу.

Логічний перетворювач (конвертор) (рис. d).

Логічний конвертор є могутнім комп'ютерним приладом, здатним проводити кілька трансформацій схемних представлень.

Ви можете використовувати його для перетворення:

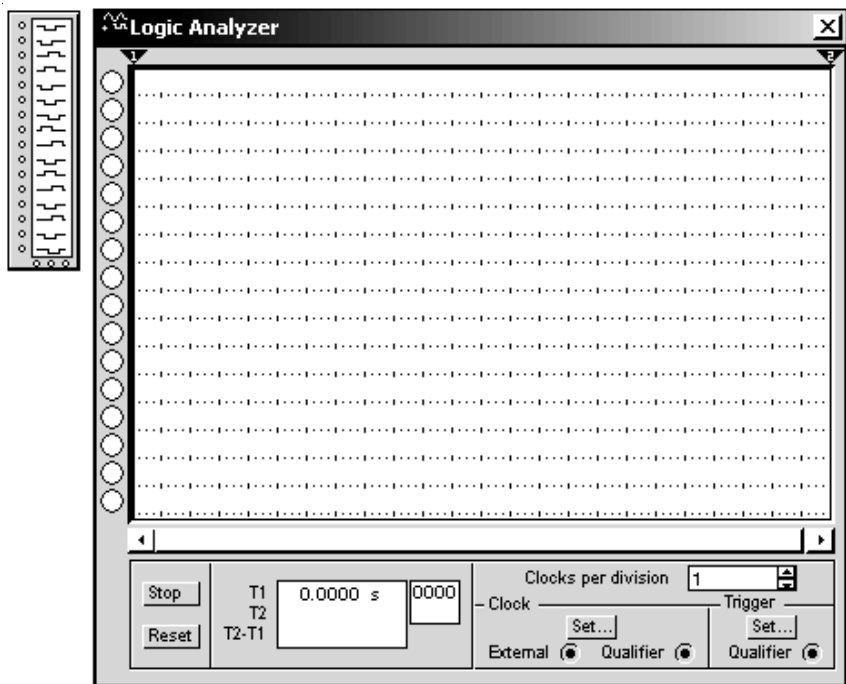


Рис. с.

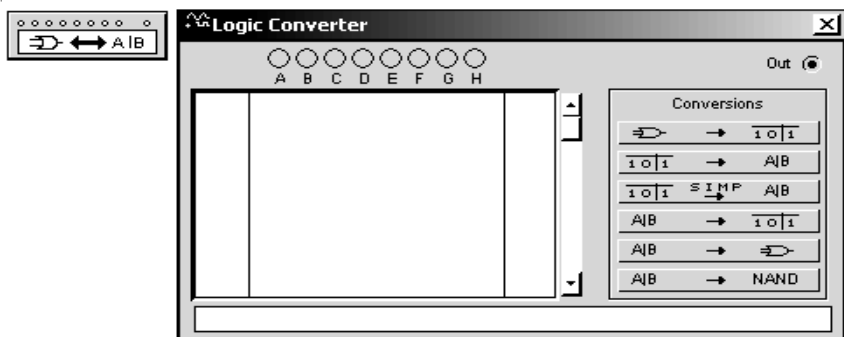


Рис. d.

- схеми в таблицю істинності ;
- таблицю істинності у вираз булевої логіки;
- вираз булевої логіки чи схему в таблицю істинності з проміжною мінімізацією.

Керування логічним конвертором.

У правій частині панелі керування логічного конвертора розташовані кнопки, які можна використовувати для виконання наступних операцій:

1) Конвертування схеми в таблицю істинності.

Логічний конвертор може створювати таблицю істинності для схеми, яка має не більш ніж 8 входів і один вихід. Для цього варто виконати наступні дії:

- приєднати схемні входи до виходів у верхній частині панелі керування логічного конвертора (варто використовувати ліві 8 виходів). Потім приєднати схемний вихід до правого виходу у верхній частині панелі керування;

- Щоб одержати таблицю істинності на дисплеї панелі керування, розташованому ліворуч від кнопок, потрібно натиснути віртуальну кнопку:



2) Задання таблиці істинності.

Для створення таблиці істинності вибрати потрібні входи (канали) логічного конвертора від А до Н, розташовані над дисплеєм. Потім у стовпці ОУТ привласнити потрібним розрядам 1, 0 чи X.

3) Конвертування таблиці істинності у вираз булевої логіки.

Щоб конвертувати таблицю істинності, що міститься на дисплеї логічного конвертора, у вираз булевої логіки слід “натиснути” на кнопку:



Булевий вираз з’явиться внизу у віконці під дисплеєм і кнопками. Його можна потім спростити.

4) Спрощення (мінімізація) булевого виразу. Конвертування таблиці істинності в мінімізований булевий вираз.

Щоб мінімізувати наявний у віконці під дисплеєм і кнопками булевий вираз чи перевести, розташовану на дисплеї панелі керування, таблицю істинності в мінімізований булевий вираз слід “натиснути” кнопку:



Electronics Workbench використовує метод Квайна-Мак Класки для мінімізації булевих виразів. Цей спосіб забезпечує спрощення для систем з великим числом входів, що може бути розраховано вручну за допомогою карт Карно.

5) Конвертація булевого виразу в таблицю істинності.

Булевий вираз можна прямо ввести у віконце, призначене для нього.

Для конвертації булевого виразу в таблицю істинності потрібно “натиснути” на

кнопку:



Якщо ви хочете мінімізувати булевий вираз, його варто спочатку перевести в таблицю істинності.

6) Конвертація булевого виразу в схему.

Щоб виконати цю операцію (при наявному у віконці панелі керування булевого виразу) потрібно “натиснути” на кнопку:



Схема реалізована на логічних елементах з’явиться на робочому столі Electronics Workbench. Компоненти будуть у режимі виділення, тому їх при необхідності можна буде легко перенести в інше місце копіюванням (Copy), вставкою (Paste) чи вставкою в підсхему (Subcircuit).

Для побудови схеми в цьому випадку буде використаний базис: I, АБО, НЕ. Якщо ж потрібно побудувати схему в базисі тільки I-НЕ див. нижче.

7) Конвертація булевого виразу в схему в базисі I-НЕ.

Для виконання цієї операції (при наявному у віконці булевого виразу) потрібно “натиснути” кнопку:



Генератор цифрових слів (рис е.).

Генератор слів використовується для посилання послідовностей бітів у схеми. Його ліва частина містить 16 рядів (слів) по 8 біт кожний. Коли генератор активований слова одне за одним посилаються в схему. Крім цього, пристрій має вихід внутрішнього генератора тимчасових імпульсів, використовуваного для синхронізації.

Керування генератором слів:

1) Введення слів.

Для введення послідовності бітів у генератор слів, варто навести курсор мишки на потрібний біт і натиснути ліву кнопку “миші”, обраний біт виявиться в режимі виділення - можна вводити 1 чи 0. Вибравши один раз точку для введення 0 чи 1, ви зможете переміщуватися по полю слів за допомогою клавіш керування курсором.

2) Активація.

У верхній правій частині панелі керування генератора слів у другому рядку зна-

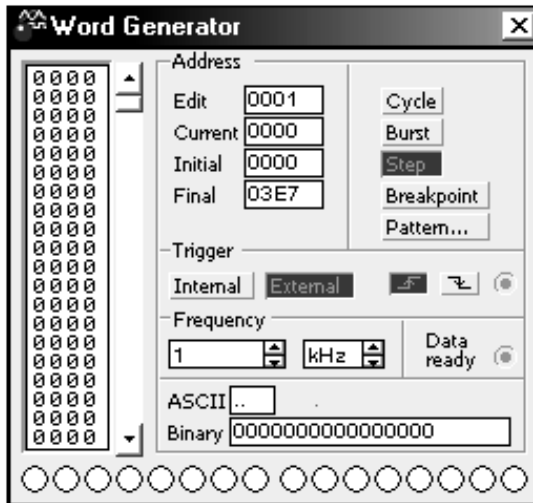


Рис. е.

ходяться три кнопки керування.

- Step - у схему буде подане виділене в даний момент слово;
- Burst - активація цієї панелі викликає подання всіх 16 слів один за одним, причому старт відбудеться з виділеного слова;
- Cycle - подання беззупинної послідовності слів

Значення бітів кожного поточного слова індукуються у вікно внизу панелі керування генератора слів.

3) Збереження, завантаження й очищення послідовностей слів.

Для очищення збереження і завантаження послідовностей слів використовуються кнопки вгорі праворуч у першому ряді панелі керування генератора слів Save, Load і Clear відповідно. Після вибору Save чи Load на екрані з'являється стандартне діалогове вікно Windows для збереження чи завантаження. Файли послідовностей слів мають розширення DP.

4) Тимчасові імпульси.

Внутрішній генератор тимчасових імпульсів має вихід позначений на панелі ке-

рування Clk. Кожен часовий цикл містить два рівні спочатку йде 1 потім 0.

5) Синхронізація (Trigger).

Генератор слів може бути синхронізований як зсередини своїми тимчасовими імпульсами, так і зовнішніми сигналами, що надходять на вхід зовнішньої синхронізації (вихід праворуч на панелі керування). Якщо використовується зовнішня синхронізація, генератор слів пересилає кожне слово з приходом зовнішнього високого чи низького, у залежності від обраного режиму, сигналу:



- спрацьовує по високому рівні;



- спрацьовує по низькому рівні.

6) Частота.

Частоту внутрішнього тимчасового генератора можна змінювати у вікні під написом Frequency. Змінювати можна як порядок Mhz, Khz, Hz, так і цифри.

ЛАБОРАТОРНА РОБОТА №1

Тема: Основи проектування комбінаційних схем.

Мета: Вивчення методів проектування комбінаційних схем в заданому наборі логічних елементів

Теоретичні відомості

Аналіз комбінаційних пристроїв і цифрових логічних схем найпростіше проводити з допомогою булевої математики, яка оперує тільки двома поняттями: істина (логічна 1) і неістина (логічний 0). В результаті функції, які відображають інформацію приймають в кожний момент часу тільки значення 0 і 1. Таблиця, в якій кожному набору вхідних сигналів ставиться у відповідність вихідний сигнал, називається таблицею істинності.

Аксіоми алгебри логіки

Змінні, розглянуті в алгебрі логіки, можуть приймати тільки два значення - 0 або 1. В алгебрі логіки визначені: відношення еквівалентності (позначається знаком \Rightarrow) і операції: додавання (диз'юнкція), яка позначається знаком \vee (+), множення (кон'юнкція), яка позначається знаком $\&$ або крапкою, і заперечення (або інверсії), яка позначається рискою зверху або апострофом.

Алгебра логіки визначається наступною системою аксіом:

$$\begin{cases} x = 0, & \text{якщо } x \neq 1, \\ x = 1, & \text{якщо } x \neq 0; \end{cases} \quad \begin{cases} 1 \vee 1 = 1, \\ 0 \vee 0 = 0, \\ 0 \vee 1 = 1 \vee 0 = 1; \end{cases}$$

$$\begin{cases} \bar{\bar{0}} = 1, \\ \bar{\bar{1}} = 0; \end{cases} \quad \begin{cases} 0 \cdot 0 = 0, \\ 1 \cdot 1 = 1, \\ 1 \cdot 0 = 0 \cdot 1 = 0. \end{cases}$$

Логічні вирази

Запис логічних виразів, за звичай, здійснюють у кон'юнктивній або диз'юнктивній нормальних формах. У диз'юнктивній формі логічні вирази записуються як логічна сума логічних добуток, у кон'юнктивній формі - як логічний добуток логічних сум. Порядок дій такий же, як і у звичайних алгебраїчних виразів.

Логічні вирази зв'язують значення логічної функції зі значеннями логічних змінних.

Логічні тотожності

При перетвореннях логічних виразів використовуються логічні тотожності:

$$\begin{aligned}x &= x; & x \vee 1 &= x; & x \vee 0 &= x; & x \cdot 1 &= x; \\x \vee x &= x; & x \cdot x &= x; & x \vee x \cdot y &= x; \\x y \vee \overline{x y} &= x; & (x \vee y)(\overline{x \vee y}) &= x; & x \vee \overline{x y} &= x \vee y; \\x y &= \overline{\overline{x} \vee \overline{y}}, & \overline{x \vee y} &= \overline{x} \overline{y}.\end{aligned}$$

Логічні функції

Будь-який логічний вираз, складений з n змінних x_n, x_{n-1}, \dots, x_1 за допомогою кінцевого числа операцій алгебри логіки, можна розглядати як деяку функцію n змінних. Таку функцію називають логічною. Відповідно до аксіом алгебри логіки, функція може приймати, залежно від значення змінних, значення 0 або 1. Функція n логічних змінних може бути визначена для 2^n значень змінним, відповідним всім можливим значенням n – розрядних двійкових чисел. Основний інтерес представляють наступні функції двох змінних x та y .

$$\begin{aligned}f_1(x, y) &= x \cdot y - \text{логічне множення (кон'юнкція)}, \\f_2(x, y) &= x \vee y - \text{логічне додавання (диз'юнкція)}, \\f_3(x, y) &= \overline{x \cdot y} - \text{логічне множення з інверсією}, \\f_4(x, y) &= \overline{x \vee y} - \text{логічне додавання з інверсією}, \\f_5(x, y) &= x \oplus y = x \overline{y} \vee \overline{x} y - \text{підсумовування по модулю 2}, \\f_6(x, y) &= x \oplus y = x y \vee \overline{x} \overline{y} - \text{рівнозначність}.\end{aligned}$$

Якщо булева функція представлена таблицею істинності, то вона може бути представлена в аналітичній формі за допомогою операцій кон'юнкції, диз'юнкції та інверсії за допомогою наступних правил:

Кожній одиниці в таблиці істинності ставиться у відповідність кон'юнкція рангу n , де n кількість аргументів функції. Рангом кон'юнкції називається число аргументів, які входять в кон'юнкцію, причому аргумент входить без інверсії, якщо він у відповідному наборі приймає значення 1 та з інверсією, якщо приймає значення 0. Всі отримані кон'юнкції об'єднуються знаками диз'юнкції.

Такий аналітичний вираз називають досконалою диз'юнктивною нормальною формою (ДДНФ). Досконалою, тому що всі кон'юнкції мають ранг n , нормальною

тому що інверсії застосовуються тільки до окремих аргументів.

Якщо в таблиці істинності нулів набагато менше ніж одиниць, то використовують аналітичний запис у вигляді досконалої кон'юнктивної нормальної форми (ДКНФ). Вона будується слідуєчим чином: кожному нулю в таблиці істинності ставиться у відповідність диз'юнкція рангу n , де n кількість аргументів функції. Рангом диз'юнкції називається число аргументів, які входять в диз'юнкцію, при чому аргумент входить без інверсії, якщо він у відповідному наборі приймає значення 0 та з інверсією, якщо приймає значення 1. Всі отримані диз'юнкції об'єднуються знаками кон'юнкції.

Логічні схеми

Фізичний пристрій, що реалізує одну з операцій алгебри логіки або найпростіша логічна функція, називається логічним елементом (рис. 1.1). Схема, складена з кінцевого числа логічних елементів за певними правилами, називається логічною схе-

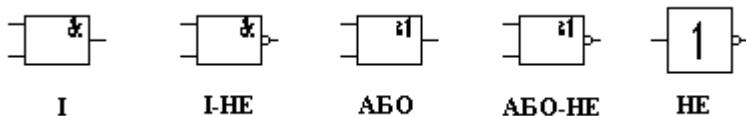


Рис.1.1.

мою.

Основним логічним функціям відповідають виконуючі їх схемні елементи.

Логічне проектування

Зазвичай, логічне проектування виконується в наступній послідовності:

- 1) складання таблиці істинності синтезованого вузла відповідно до його означення, призначення і (словесного) опису принципу роботи;
- 2) складання математичної формули для логічної функції, що описує роботу синтезуючого вузла, відповідно до наявної таблиці істинності;
- 3) аналіз отриманої функції з метою побудови різних варіантів її математичного виразу (на підставі законів булевої алгебри) і знаходження найкращого з них відповідно до того чи іншого критерію ;
- 4) складання функціональної (логічної) схеми вузла із заздалегідь заданим набомою.

ром логічних елементів.

Синтез комбінаційних пристроїв, зазвичай, починається з табулювання значень істинності всіх вхідних і вихідних величин. Табличне задання закону функціонування деякого пристрою є найбільш наочним і універсальним засобом опису його роботи. Результатом розглянутого етапу є таблиця істинності, що зв'яже всі можливі комбінації значень аргументів і функцій.

Та сама функція може бути представлена різними формулами. Кожній формулі відповідає своя суперпозиція і, отже, своя схема з'єднань елементів. Очевидно, серед схем, що реалізують дану функцію, є більш проста. Пошук логічної формули, що відповідає цій схемі, представляє великий практичний інтерес, а перетворення формул булевих функцій засновано на використанні законів булевої алгебри. Виходячи з заданої таблиці істинності, знаходимо за допомогою діаграми Вейча МДФ функції. Даний метод дозволяє швидко одержувати мінімальні ДНФ булевої функції f невеликого числа змінних. В основі методу лежить задання булевих функцій діаграмами деякого спеціального виду, що одержали назву діаграм Вейча. Для булевої функції двох змінних

	x_1	\bar{x}_1
x_2	11	01
\bar{x}_2	10	00

Рис.1.2

діаграма Вейча має вид рис.1.2. Кожна клітка діаграми відповідає набору змінних булевої функції в її таблиці істинності. У клітині діаграми Вейча ставиться одиниця, якщо булева функція приймає одиничне значення для відповідного набору. Нульові значення булевої функції в діаграмі Вейча не ставляться. Для булевої функції трьох змінних діаграма Вейча має наступний вид рис.1.3.

	x_1			
	110	111	011	010
x_2	100	101	001	000
	x_3			

Рис.1.3.

Додавання до неї ще такої ж таблиці дає діаграму для функції 4-х змінних (рис.1.4).

	x_2				
	1100	1101	1001	1000	
	1110	1111	1011	1010	
x_1	0110	0111	0011	0010	x_3
	0100	0101	0001	0000	
	x_4				

Рис.1.4.

У такий же спосіб, тобто приписуванням ще однієї діаграми 3-х змінних, можна одержати діаг-

раму для функції 5-ти змінних і т.д., однак діаграми для функцій з числом змінних більше 4-х використовуються рідко. Після того як діаграма заповнена, робимо відповідні склеювання. Склеювати (об'єднувати) можна дві, чотири, вісім, і т.д. клітини, які є між собою сусідами.

Найбільш очевидним способом мінімізації ДДНФ є виконання перетворень, аналогічних перетворенням звичайної алгебри. Це можливо, оскільки для операцій І і АБО справедливі асоціативний та дистрибутивний закони. Мова йде про те, щоб перейти від ДДНФ до ДНФ із мінімумом доданків, при цьому кількість множників у кожному доданку повинна бути також мінімальною (позбутися від “досконалості”), тобто максимально зменшити кількість змінних і операцій у ДДНФ.

Аналіз і оптимізація (мінімізація) логічних функцій є дуже важливими компонентами синтезу цифрових автоматів без пам'яті.

До побудови функціональної схеми синтезованого вузла, в принципі, можна переходити відразу ж, як тільки стає відомим аналітичний опис його роботи. Побудова схеми заснована на прямому заміщенні елементарних добутоків, сум і заперечень відповідно кон'юнкторами, диз'юнкторами і інверторами.

Завдання до лабораторної роботи

Таблиця 1.1

x_1	x_2	x_3	x_4	y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Варіант І.

1. Побудувати на елементах АБО та АБО-НЕ однотактний пристрій, який реалізує слідуючий алгоритм роботи. На вхід подається 5-розрядний двійковий код (x_1, x_2, x_3, x_4, x_5). На виході маємо 1, якщо число одиниць в коді рівне 4 чи 5. В інших випадках на виході маємо 0.

2. На логічних елементах побудувати схему, яка реалізує функцію

$$y = \overline{x_3 x_2 x_1} + \overline{x_3 x_2 x_1} + x_2 x_1$$

3. Отримати операторне пред-

ставлення функції, заданої в таблиці 1.1, яка може бути реалізована на елементах

- а) І та інвертор
- б) АБО та інвертор

Побудувати схему для реалізації даної функції. Врахувавши, що на її вхід можуть подаватися прямі та інверсні значення змінних.

Варіант II.

1. Побудувати на елементах АБО та АБО-НЕ однотактний пристрій, який реалізує слідуєчий алгоритм роботи. На вхід подається 5-розрядний двійковий код (x1, x2, x3, x4, x5). На виході маємо 1, якщо число одиниць в коді рівне 3. В інших випадках на виході маємо 0.

2. На логічних елементах побудувати схему, яка реалізує функцію

$$y = \overline{x_3 x_2 x_1} + \overline{x_3 x_1} + \overline{x_3 x_2 x_1} + x_2$$

3. Отримати операторне представлення функції, заданої в таблиці 1.2, яка може бути реалізована на елементах

- а) І та інвертор
- б) АБО та інвертор

Таблиця 1.2.

X1	X2	X3	X4	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Побудувати схему для реалізації даної функції. Врахувавши, що на її вхід можуть подаватися прямі та інверсні значення змінних.

Варіант III.

1. Побудувати на елементах АБО та АБО-НЕ однотактний пристрій, який реалізує слідуєчий алгоритм роботи. На вхід подається 5-розрядний двійковий код (x1, x2, x3, x4, x5). На виході маємо 1, якщо число одиниць в коді рівне 2. В інших випадках на виході маємо 0.

2. На логічних елементах побуду-

Таблиця 1.3.

X1	X2	X3	X4	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

вати схему, яка реалізує функцію

$$y = \overline{x_3 x_2 x_1} + \overline{x_2 x_1} + \overline{x_2 x_1}$$

3. Отримати операторне представлення функції, заданої в таблиці 1.3, яка може бути реалізована на елементах

а) І та інвертор

б) АБО та інвертор

Побудувати схему для реалізації даної функції. Врахувавши, що на її вхід можуть подаватися прямі та інверсні значення змінних.

Питання для самоконтролю

1. Що таке логічна змінна і логічний сигнал? Які значення вони можуть приймати?
2. Що таке таблиця істинності логічного елемента чи пристрою, який виконує деяке логічне перетворення?
3. Що таке логічна функція?
4. Чи може бути логічним сигналом рівень напруги? Стан контакту?
5. Як буде поводитися схема І, якщо на одному із входів, внаслідок внутрішньої несправності, буде постійно присутня логічна одиниця? Логічний нуль? Складіть таблицю істинності для несправної схеми І. Визначите поведження схеми І-НЕ за тих самих умов.
6. Як буде поводитися схема АБО, якщо на одному із входів, внаслідок внутрішньої несправності буде, постійно присутня логічна одиниця? Логічний нуль?
7. Поясніть, чому невикористані входи логічних елементів АБО, АБО-НЕ з'єднуються з корпусом (рівнем логічного "0"), а на невикористані входи логічних елементів І, І-НЕ подається напруга рівня логічної "1"?

ЛАБОРАТОРНА РОБОТА №2

Тема: Дешифратор. Шифратор.

Мета: Вивчення принципів проектування дешифраторів та шифраторів в заданому базисі логічних елементів, а також дослідження функціонування спроектованих дешифраторів та шифраторів та їх інтегральних схем.

Теоретичні відомості

Дешифратори і шифратори (як і елементи І, АБО, НЕ, І-НЕ, АБО-НЕ) є комбінаційними елементами: сигнали на їх виходах залежать від стану входів, такі елементи не зберігають попередній стан після зміни сигналу на входах, тобто не мають пам'яť.

Дешифратори можуть бути повними й неповними. Повні дешифратори реагують на всі вхідні коди, неповні - на коди, величина яких не переверщує деякого заздалегідь встановленого значення. Виходи дешифраторів можуть бути прямими і інверсними.

Шифратори випускаються пріоритетними і не пріоритетними. У пріоритетного шифратора входи мають різний пріоритет. Збуджений вхід з більшим пріоритетом придушує дію раніше збудженого і встановлює на виходах код, який відповідає його значенню.

Дешифратори

Дешифратор призначений для перетворення двійкового коду на вході в керуючий сигнал на одному з виходів. Дешифратор широко застосовують в пристроях управління, в системах цифрової індикації з газорозрядними індикаторами, для по-

Таблиця 2.1.

<i>X1</i>	<i>X2</i>	<i>X3</i>	<i>Y0</i>	<i>Y1</i>	<i>Y2</i>	<i>Y3</i>	<i>Y4</i>	<i>Y5</i>	<i>Y6</i>	<i>Y7</i>
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

будови розподільників імпульсу.

Якщо дешифратор має n входів, то вихідних шин повинно бути $N = 2^n$, так як для кожного із 2^n значень вхідного коду повинен відповідати одиничний сигнал на одному із виходів дешифратора. Дешифратор працює відповідно до таблиці 2.1.

Основу структури дешифратора можуть становити елементи І; вихід кожного з них є виходом дешифратора. Якщо цей вихід повинен бути збуджений, то на входах елемента І повинні збиратися логічні одиниці. При цьому, розряди вхідного коду, в якому присутні логічні одиниці, повинні надходити на входи елемента І безпосередньо, а нульові розряди повинні інвертуватися.

Викладений принцип покладений в основу побудови схеми, зображеної на рис.2.1. Логічна 1 на виході Y_0 повинна з'явитися, коли на входах X_3, X_2, X_1 присутній двійковий код 000 десяткового числа 0. Тому входи верхнього (за схемою) кон'юнктора повинні бути з'єднані з лініями $\overline{X_1}, \overline{X_2}, \overline{X_3}$, на кожній з яких є логічна 1, коли на

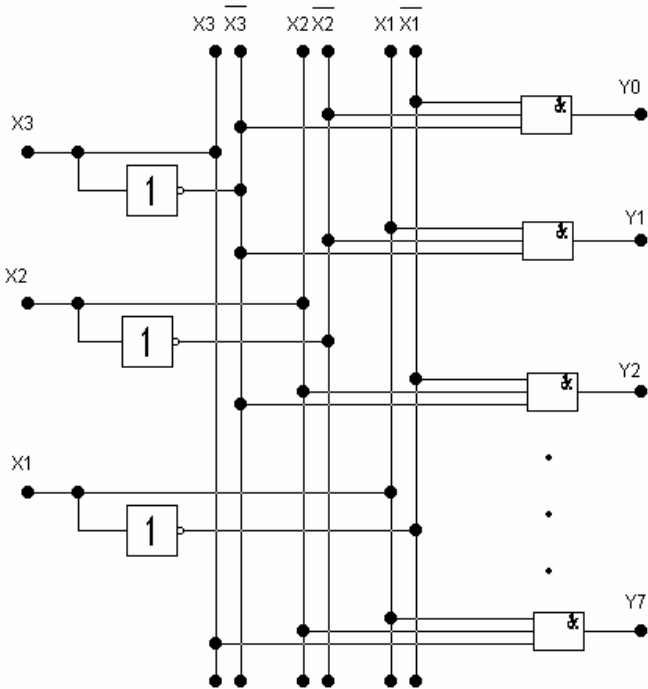


Рис.2.1.

входах $X_3=X_2=X_1=0$. Логічна 1, наприклад, на виході Y_2 повинна з'явитися, коли на входах X_3, X_2, X_1 встановлюється код 010 десятичного числа 2, тому входи відповідного кон'юнктора повинні бути з'єднані з лініями $\overline{X_1}, X_2, \overline{X_3}$, на кожній з яких є логічна 1, коли $X_3=0, X_2=1, X_1=0$. Аналогічно з'єднуються з лініями входи інших кон'юнкторів.

Деякі типи дешифраторів мають інверсні виходи: на збудженому (активізованому) виході присутній логічний 0, в той час як на всіх інших - логічна 1. Такі дешифратори зручно використовувати, коли активним сигналом для вибору (запровадження в дію, ініціалізації) пристрою з виходу дешифратора є логічний 0.

Основне призначення дешифратора полягає в тому, щоб вибрати (адресувати) один об'єкт із багатьох які є в пристрої.

Розширення розрядності дешифратора

Загальний випадок розширення розрядності дешифраторів ілюструє рис.2.2. Лівий (за схемою) дешифратор постійно активізований логічною 1 на вході V. Кодами на його адресних входах може бути активізований (обраний) кожен з дешифраторів DC0...DC15. Вибір одного з виходів 0...15 кожного з них визначається кодом на об'єднаних входах 1, 2, 4, 8. Таким чином, кожен з 256 (28) виходів може бути активізований восьмиразрядним кодом, чотири розряди якого вибирають номер дешифратора, а чотири – номер його виходу.

Застосування дешифраторів

Основне призначення дешифратора полягає в тому, щоб вибрати (адресувати, ініціалізувати) один об'єкт із безлічі, які знаходяться. Кожному об'єкту присвоюють конкретну адресу (номер). Коли на входи дешифратора надходить двійковий код адреси, відповідний елемент активується за рахунок появи логічного 0 на зв'язаному з ним виході дешифратора, а інші елементи залишаються заблокованими.

Можна передбачити, щоб з одного з виходів дешифратора на визначений блок надходив керуючий сигнал, коли на входах дешифратора з'являється визначений код, що відповідає, наприклад, перевищенню якогось параметра (температури, напруги і т.д.), що повинен бути приведений до нормального рівня зазначеним блоком.

На дешифраторі можуть бути реалізовані логічні функції.

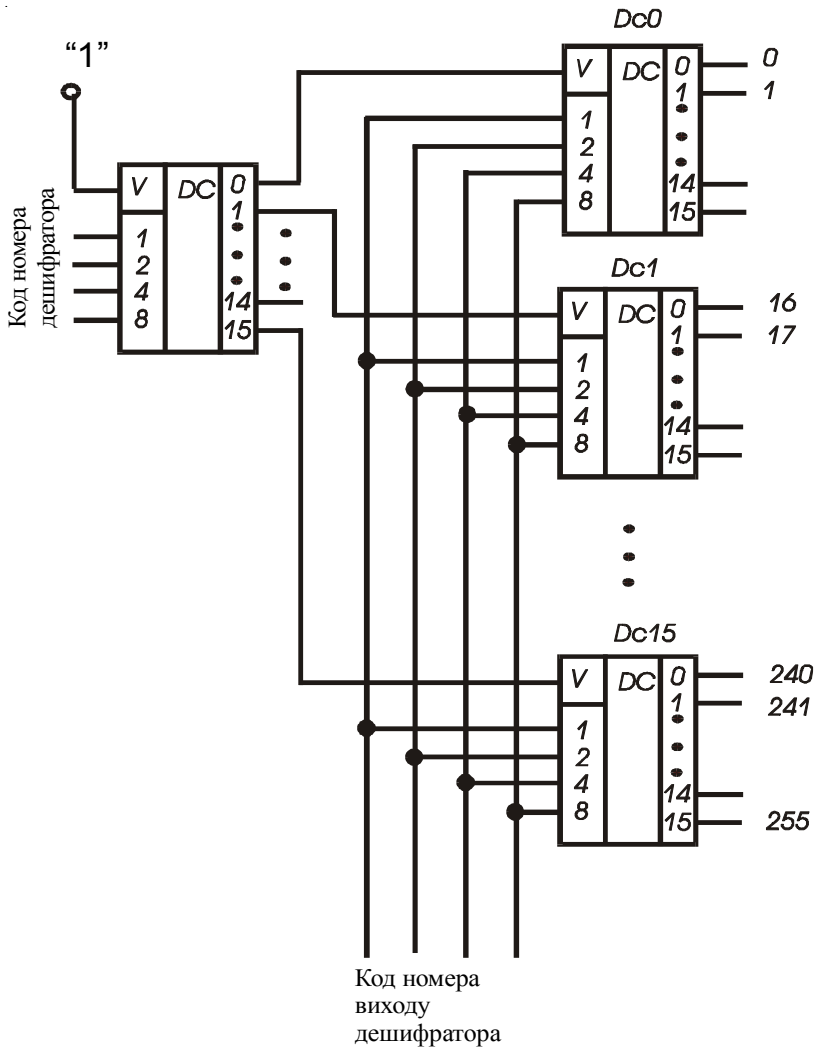


Рис.2.2.

Наприклад: $y = \overline{x_3}x_2x_1 + x_3\overline{x_2}x_1 + x_3x_2\overline{x_1}$

Логічні змінні подаються на адресні входи дешифратора. Перша кон'юнкція збуджує вихід №2, друга – вихід №3, третя – вихід №5. Так як умова $y = 1$ повинна мати місце при наявності кожної з цих кон'юнкцій, то виходи 2, 3 і 5 треба об'єднати диз'юнкцією.

Таблиця 2.2.

Шифратор			
Десяткове число	Двійковий код		
X	Y2	Y1	Y0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Шифратори

Шифратор (кодер) перетворює одиничний сигнал на одному із входів в n -розрядний двійковий код.

Шифратор розв'язує задачу, обернену дешифратору: зокрема, на його виходах встановлюється двійковий код, що відповідає

десятьковому номеру збудженого інформаційного входу (таблиця 2.2.).

При побудові шифратора для одержання на виході натурального двійкового коду враховують, що одиницю в молодшому розряді такого коду мають непарні десяткові цифри 1, 3, 5, 7, ... , тобто на виході молодшого розряду повинна бути 1, якщо вона є на вході № 1 або на вході № 3 і т.д. Тому входи під вказаними номерами через елемент АБО з'єднуються з виходом молодшого розряду. Одиницю в другому розряді двійкового коду мають десяткові цифри 2, 3, 6, 7, ... ; входи з цими номерами через елемент АБО повинні підключатися до виходу шифратора, на якому встановлюється другий розряд коду. Аналогічно, входи 4, 5, 6, 7, ... через елемент АБО повинні бути з'єднані з виходом, на якому встановлюється третій розряд, тому що їхні коди мають в цьому розряді одиницю, і т.д.

Застосування шифраторів

Найбільше застосування він знаходить у пристроях введення інформації (пультах управління) для перетворення десяткових чисел в двійкову систему числення. Припустимо, на пульті десять клавіш з гравіюванням від 0 до 9. При натисненні будь-якої із них на вхід шифратора подається одиничний сигнал (X_0, \dots, X_9). На виході шифратора повинен з'явитися двійковий код (Y_0, \dots, Y_3) цього десяткового числа.

Шифратор може бути організований не тільки для представлення (кодування) десяткового числа двійковим кодом, але і для видачі певного коду (його значення заздалегідь вибирається), наприклад, при натисненні клавіші з відповідним символом. З появою даного коду система сповіщається про те, що натиснено певну клавішу.

ішу клавіатури. Шифратори застосовуються в пристроях, що перетворюють один вид коду в інший. При цьому спочатку дешифрується комбінація вихідного коду, у результаті чого на відповідному виході дешифратора з'являється логічна 1. Це відображення вхідного коду, значення якого визначено номером збудженого виходу дешифратора, подається на шифратор, організований з таким чином, щоб кожний вхідний код викликав появу заданого вихідного коду.

Завдання до лабораторної роботи

Варіант I.

1. Побудувати дешифратор на 4 входи на базі логічних елементів.
2. Побудувати шифратор на 4 входи на базі логічних елементів.
3. Розглянути принцип дії дешифратора реалізованого на мікросхемі 74154
4. На основі дешифратора реалізованого на мікросхемі 74154 продемонструвати розширення розрядності дешифраторів (збільшити кількість виходів до 32).

Варіант II.

1. Побудувати дешифратор на 2 входи на базі логічних елементів.
2. На основі побудованого дешифратора продемонструвати розширення

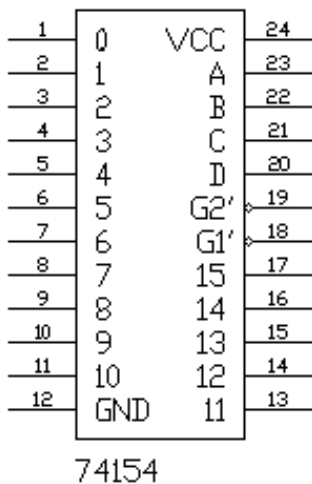


Рис.2.3.

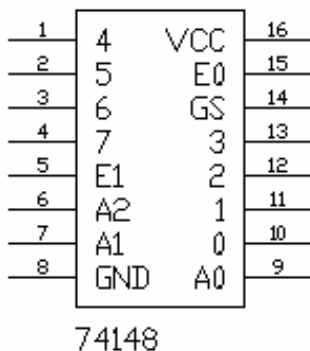


Рис.2.4.

розрядності дешифратора.

3. Побудувати шифратор на 16 входів на базі логічних елементів.
4. Розглянути принцип дії шифратора реалізованого на мікросхемі 74147

Варіант III.

1. Побудувати дешифратор на 3 входи на базі логічних елементів.
2. На основі побудованого дешифратора побудувати логічну функцію
$$y = \overline{x_3}x_2x_1 + x_3\overline{x_2}x_1 + x_3x_2\overline{x_1}$$
3. Побудувати шифратор на 8 входів на базі логічних елементів.
4. Розглянути принцип дії шифратора реалізованого на мікросхемі 74147.

Примітка

Щоб вибрати 74154 мікросхему виберемо меню Digital ICs, в якому винесемо (тримаючи ліву кнопку миші) в робоче вікно програми пункт 741xx. Після чого з'явиться список мікросхем (дешифраторів). В якому виберемо дану мікросхему. (рис.2.3.).

0..15 – виходи мікросхеми; A, B, C, D – входи мікросхеми; G2', G1' - дозволяючі входи; VCC, GND - живлення мікросхеми (подається напруга 5V).

Вхідні сигнали будемо подавати використовуючи Word Generator.

Щоб вибрати мікросхему 74147 виберемо меню Digital ICs, в якому винесемо (тримаючи ліву кнопку миші) в робоче вікно програми пункт 741xx, де виберемо дану мікросхему.

0..7 – входи мікросхеми; A0, A1, A2 – виходи; E1 – дозволяючий вхід; E0, GS – виходи для каскадування шифраторів; VCC, GND – живлення мікросхеми. При моделюванні необхідно звернути увагу на принцип пріоритету старшого розряду, при цьому необхідно врахувати, що всі входи і виходи інверсні.

Питання для самоконтролю

1. Дайте визначення дешифратора та шифратора.
2. Як працює дешифратор?
3. Яку мінімальну розрядність повинен мати дешифратор для адресації 11-ти

пристроїв?

4. Скільки входів повинен мати неповний дешифратор, що має 10 виходів?
5. В яких пристроях використовується дешифратор?
6. Як працює шифратор? При розв'язуванні яких задач він використовується?
7. Який номер збудженого входу шифратора, якщо на виході встановився код 0110?
8. В чому полягає зміст слова “пріоритетний” в назві шифратора типу 74148?
9. Як на базі дешифратора можуть бути реалізовані логічні функції?

ЛАБОРАТОРНА РОБОТА №3

Тема: Мультиплексор. Демультимплексор.

Мета: Вивчення принципів проектування і дослідження роботи мультиплексора та демультимплексора.

Теоретичні відомості

Мультиплексор є комутатором з одного із декількох інформаційних входів на єдиний вихід. Вибір того або іншого входу визначається кодом, який встановлюється на адресних входах мультиплексора. Це дозволяє при зміні кодів передавати на вихід цифрову інформацію то з одного, то з іншого каналу.

Демультимплексор є комутатором єдиного інформаційного входу на один із декількох виходів. Вибір того або іншого виходу визначається кодом на адресних входах демультимплексора. Таким чином, демультимплексор вирішує задачу, обернену мультиплексору: при зміні кодів він може передавати цифрову інформацію то в один, то в інший канал з одного входу.

Мультиплексори

Структура мультиплексора.

Мультиплексор (рис.3.1.) комутує на вихід Y один із входів $D1, D2, \dots$, який вибирається (адресується) двійковим кодом на адресних входах $A0, A1, A2$.

На рис. 3.2. приведена функціональна схема мультиплексора. Він має вісім інформаційних ($D0 \dots D7$) і три адресні ($A1 \dots A3$) входи.

На адресні входи поступає трьохрозрядний цифровий код, повне число комбінацій якого рівне вісьми. Код 111 повинен забезпечити з'єднання виходу Y з входом $D7$, код 110 із входом $D6$ і т. д., код 000 - з входом $D0$.

Як і в дешифраторі, для розблокування кон'юнктора його входи треба з'єднати безпосередньо з тими адресними входами, на яких при даному коді присутні одиниці, і через інвертори - з тими входами, на яких знаходяться нулі. Так, наприклад, на другий зверху кон'юнктор (рис. 3.2) безпосередньо поступає сигнал з входу $A1$ і через інвертори - з входів $A3$ і $A2$, що при коді адреси 001 забезпечить на цьому кон'юнкторі три логічні 1, тобто підключо-

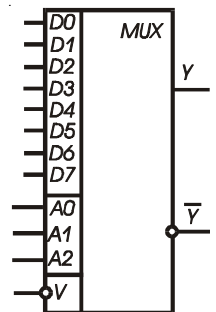


Рис.3.1.

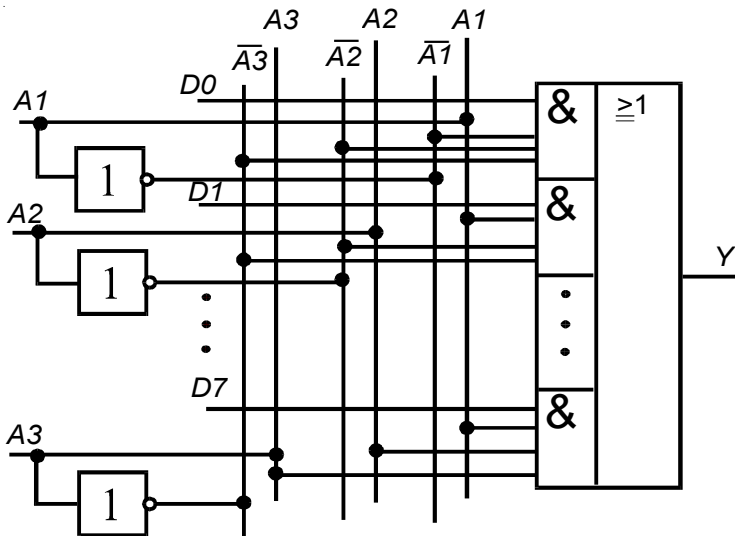


Рис.3.2.

чення до виходу мультиплексора входу D1. Входи нижнього кон'юнктора сполучені з адресними входами безпосередньо, що при коді адреси 111 забезпечить підключення до виходу мультиплексора інформаційного входу D7. За наявності на вході V логічної 1 мультиплексор блокується: на прямому виході встановлюється логічний 0 незалежно від потенціалів на інформаційних входах.

Розширення розрядності мультиплексора.

Розширення розрядності мультиплексорів в загальному випадку ілюструє рис. 3.3. Тут “мультиплексорне дерево” містить чотири чотиривхідних мультиплексори MUX1...MUX4 із під'єднаними паралельно адресними входами A0, A1 якими одночасно вибирається один з входів D0...D3 всіх чотирьох елементів, а вихідний мультиплексор кодом на адресних входах A2, A3 вибирає один з виходів Y0...Y3. Таким чином, чотирьохрозрядний код на входах A0...A3 сполучає з виходом Y тільки один з 16 входів D0...D15.

Застосування мультиплексорів

Сукупність мультиплексорів із з'єднаними адресними входами може бути використана для послідовної передачі на їх виходи декількох багаторозрядних кодів. Таку

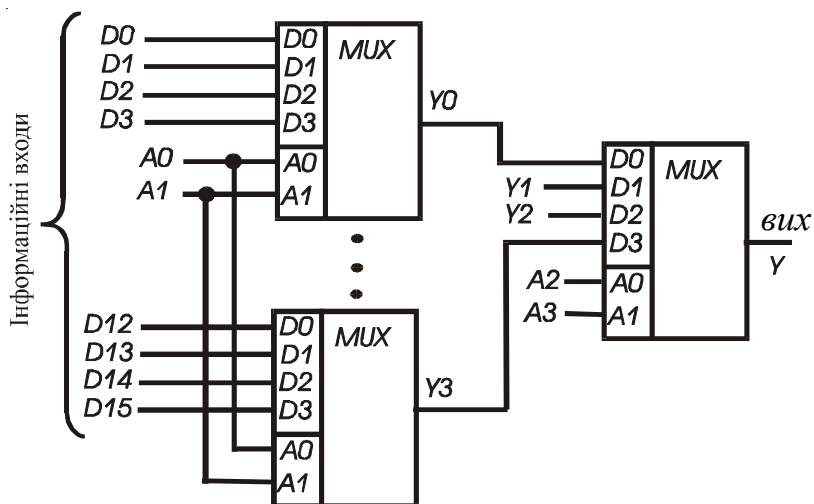


Рис. 3.3.

можливість забезпечить схема на рис. 3.3 якщо з неї виключити крайовий мультиплексор. При цьому розряди першого коду подають на входи D0 всіх мультиплексорів, розряди другого коду - на входи D1 і т.д. При зміні адреси на входах A0 A1 на виходи мультиплексорів передаватимуться розряди то першого, то другого, то третього коду. Таке часове мультиплексування використовують при необхідності передавати на одні і ті ж входи наступного пристрою то один, то інший код. Окрім комутації з одного з n входів на один вихід, мультиплексор може використовуватися для перетворення паралельного коду, розряди якого подаються на входи D0, D1 ..., в послідовний код на виході Y. Для цього код на адресних входах повинен циклічно змінюватися, приймаючи всі послідовні значення. Таку зміну можна забезпечити, приєднавши до адресних входів виходи лічильника, які послідовно змінюють свій стан під дією імпульсів генератора.

На мультиплексорі можуть бути реалізовані логічні функції. Нехай, наприклад, задана функція

$$y = \overline{x_3}x_2x_1 + x_3\overline{x_2}x_1 + x_3x_2\overline{x_1}$$

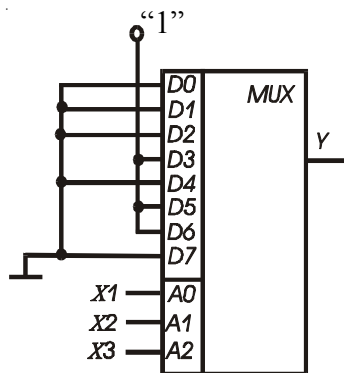


Рис.3.4

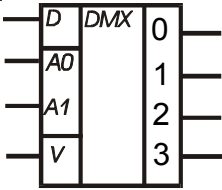


Рис. 3.5.

Логічні змінні x_3, x_2, x_1 подають на адресні входи A_0, A_1, A_2 . Коли набір x_3, x_2, x_1 складатиме кожну з приведених кон'юнкцій, на вихід Y по черзі комутуватимуться відповідно входи D_3, D_5, D_6 . Оскільки кожна з них повинна забезпечити $Y = 1$, то на вказані входи слід подати логічні одиниці, а на ті, що залишилися - логічні нулі (рис. 3.4).

Демультиплексори

Структура демультиплексора

Демультиплексор (Рис. 3.5) виконує задачу, обернену мультиплексору: він комутує єдиний інформаційний вхід D на один з виходів, який адресується двійковим кодом на адресних входах A_0, A_1 .

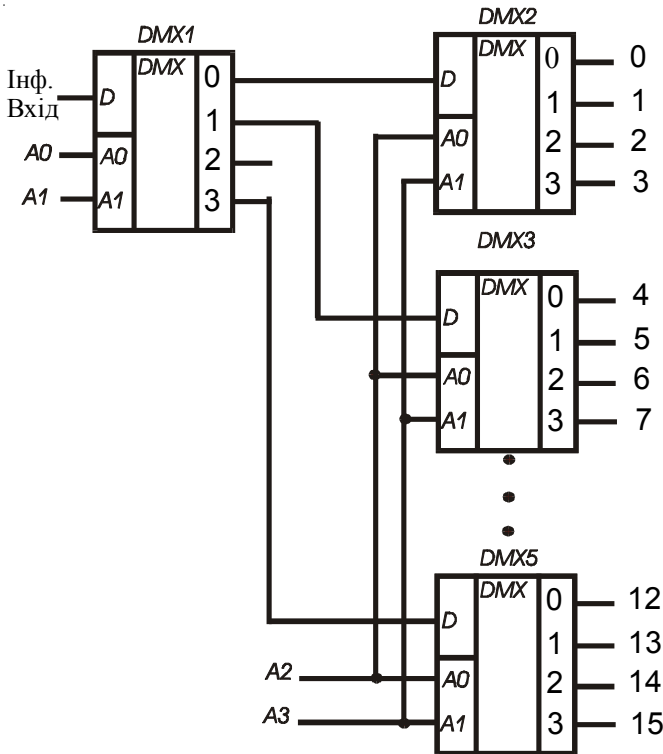


Рис. 3.6.

Розширення розрядності демультиплексора

На рис. 3.6. показаний загальний випадок нарощування розрядності демультиплексорів. На інформаційний вхід D поступають логічні 1 і 0. Кодом на адресних входах A0, A1 вибирається один з виходів DMX1, а кодом на A2 A3 одночасно вибираються чотири однойменні виходи всіх чотирьох демультиплексорів DMX2 ... DMX5. В результаті кодом A0.A3 вибирається один з 16 виходів.

Застосування демультиплексорів

Окрім прямого призначення, демультиплексор в сукупності з мультиплексором дозволяє скомутувати будь-який вхід мультиплексора з будь-яким виходом демультиплексора. Для цього вихід мультиплексора треба з'єднати з інформаційним входом демультиплексора.

Завдання до лабораторної роботи

Варіант I.

1. Побудувати чотиріканальний мультиплексор на логічних елементах.
2. На базі отриманого мультиплексора складіть схему, яка реалізує функцію
$$y = x_2x_1 + x_2x_1 + x_2x_1$$
3. Побудувати демультиплексор на 8 виходів .
4. На основі мікросхеми мультиплексора продемонструйте розширення розрядності мультиплексора. Щоб вибрати мікросхему мультиплексора в меню Digital, винесемо (тримаючи ліву кнопку миші) в робоче вікно програми пункт MUX якому виберемо дану мікросхему (рис 3.7).

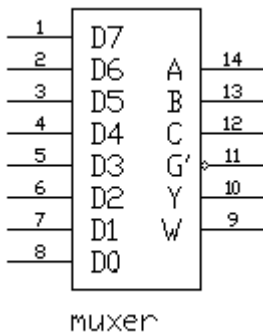


Рис.3.7.

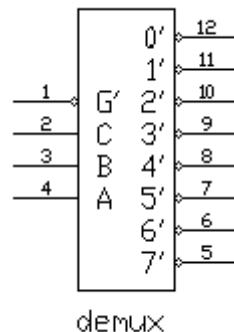


Рис.3.8.

Варіант II.

1. Побудувати восьми канальний мультиплексор.
2. На базі побудованого мультиплексора складіть схему, яка реалізує функцію $y = x_3x_2x_1 + x_3x_2x_1 + x_3x_2x_1 + x_3x_2x_1$
3. Побудувати демультимплексор на 4 виходи.
4. На основі мікросхеми демультимплексора продемонструйте розширення розрядності демультимплексора. Щоб вибрати мікросхему демультимплексора в меню Digital, винесемо (тримаючи ліву кнопку миші) в робоче вікно програми пункт DEC якому виберемо дану мікросхему (рис.3.8).

Варіант III.

1. Побудувати 8-канальний мультиплексор.
2. На базі побудованого мультиплексора складіть схему, яка реалізує функцію $y = x_3x_2x_1 + x_3x_2x_1 + x_3x_2x_1 + x_3x_2x_1 + x_3x_2x_1$
3. Побудувати демультимплексор на 4 виходи.
4. На основі побудованого демультимплексора продемонструйте розширення розрядності демультимплексора.

Питання для самоконтролю

1. Що таке мультиплексор і яке його призначення?
2. Скільки адресних входів повинен мати мультиплексор, якщо кількість інформаційних входів рівна вісім?
3. Задайте послідовність кодів на адресних входах мультиплексора, щоб 4-х розрядний код на інформаційних входах перетворити в послідовний код на виході.
4. Що таке демультимплексор і для розв'язку яких задач він використовується ?
5. Яка особливість демультимплексора відрізняє його від дешифратора?
6. Як на базі мультиплексора можуть бути реалізовані логічні функції?
7. В чому полягає принцип каскадування мультиплексорів?

ЛАБОРАТОРНА РОБОТА №4

Тема: Перетворювачі кодів

Мета: Розглянути принцип роботи перетворювачів кодів на прикладі перетворювачів побудованих на дешифраторі та шифраторі та перетворювачі двійково-десятькового коду в код семисегментного індикатора

Теоретичні відомості

Комбінаційні перетворювачі кодів призначені для перетворення m - елементного паралельного коду на вході в n - елементний паралельний код на виході. Зв'язок між вхідними і вихідними сигналами можна задати таблицями істинності або логічними функціями. Розглянемо найпоширеніші види перетворювачів кодів.

Перетворювачі на дешифраторі та шифраторі.

Перетворювач кодів описується системою логічних рівнянь, де аргументами є елементи перетворюваного коду, а функцією – кожний елемент перетвореного.

Принцип перетворення на дешифраторі і шифраторі полягає в тому, що кожному вхідному коду дешифратор ставить у відповідність збуджений вихід, якому шифратор ставить у відповідність вихідний код.

Перетворювач двійково-десятькового коду в код семисегментного індикатора.

Індикація чисел на табло і пультах проводиться, як правило, в десятковому вигляді. Ми знаємо, що для цього можна використовувати семисегментні світлодіодні або рідкокристалічні індикатори. Подаючи управляючу напругу на окремі елементи індикатора і викликаючи його свічення (світлодіодні індикатори) або змінюючи його забарвлення (рідкокристалічні індикатори), ми можемо одержувати зображення десяткових цифр 0, 1, ..., 9. (рис. 4.1.);

Для зручності перекладу двійкової інформації в десятковий вигляд часто використовують двійково-десятьковий код (або код 8421), тобто представлення десяткових



Рис. 4.1.

Таблиця 4.1.

цифра	Двійково-десятковий код "8421"				Семисегментний код							
	X4	X3	X2	X1	a	b	c	d	e	f	g	
0	0	0	0	0	1	1	1	1	1	1	0	
1	0	0	0	1	0	1	1	0	0	0	0	
2	0	0	1	0	1	1	0	1	1	0	1	
3	0	0	1	1	1	1	1	1	0	0	1	
4	0	1	0	0	0	1	1	0	0	1	1	
5	0	1	0	1	1	0	1	1	0	1	1	
6	0	1	1	0	1	0	1	1	1	1	1	
7	0	1	1	1	1	1	1	0	0	0	0	
8	1	0	0	0	1	1	1	1	1	1	1	
9	1	0	0	1	1	1	1	1	0	1	1	

Таблиця 4.2.

8421				2421			
X4	X3	X2	X1	Y4	Y3	Y2	Y1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	1	0	0
0	1	1	1	1	1	0	1
1	0	0	0	1	1	1	0
1	0	0	1	1	1	1	1

чисел у вигляді чотирьохрозрядних двійкових чисел.

$$\text{Наприклад, } 9_{(10)} = 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 1001_{(2)}$$

$$75_{(10)} = 0111\ 0101;$$

$$910_{(10)} = 1001\ 0001\ 0000.$$

Закон функціонування перетворювача двійково-десятькового коду в код семисег-

ментного індикатора працює відповідно до таблиці 4.1.

Перетворювач коду з 8421 в 2421

Код 2421 утворюється з коду 8421. До 4-х він повторює код 8421, а далі утворюється як інверсія доповнення до 9-и. Код 8421 є двійково-десятковим кодом, де коефіцієнти 8,4,2 і 1 є ваговими коефіцієнтами, тобто відповідають “вартості” кожного розряду. Сполучення: 1010, 1011, 1100, 1101, 1110, 1111 є забороненими. Перетворювачі кодів застосовуються в системах інтерфейсів.

Перетворювач працює відповідно до таблиці. 4.2.

Завдання до лабораторної роботи

Варіант I.

1. Побудувати перетворювач кодів за таблицею 4.3. з використанням елементів І, І-НЕ і чотирирівідного дешифратора з інверсними виходами. Дешифратор розміщений в корпусі з 24 виходами, він реалізований на мікросхемі 74154

Таблиця 4.3.

інформація							
на вході				на виході			
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	1
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	1
1	1	0	0	1	1	1	1
1	1	0	1	0	0	0	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	1	0

2. Побудувати перетворювач двійково-десятькового коду в код семисегментного індикатора на базі логічних елементах.

Варіант II.

1. Побудувати перетворювач кодів за таблицею 4.3. з використанням елементів АБО, АБО-НЕ і чотиривхідного дешифратора з інверсними виходами. Дешифратор розміщений в корпусі з 24 виходами, він реалізований на мікросхемі 74154

2. Побудувати перетворювач двійково-десятькового коду в код семисегментного індикатора на базі логічних елементах.

Варіант III.

1. Побудувати перетворювач коду з 8421 в 2421 на логічних елементах АБО та АБО-НЕ

2. Побудувати перетворювач двійково-десятькового коду в код семисегментного індикатора на базі чотиривхідного дешифратора з інверсними виходами та з використанням логічних елементів АБО, АБО-НЕ. Дешифратор розміщений в корпусі з 24 виходами, він реалізований на мікросхемі 74154.

Питання для самоконтролю

1. Які є види перетворювачів кодів?
2. В чому полягає принцип роботи перетворювача коду на дешифраторі та шифраторі?
3. В якому вигляді проводиться індикація чисел на табло?
4. Які є види індикаторів?

ЛАБОРАТОРНА РОБОТА № 5

Тема: Цифровий компаратор

Мета: Вивчення правил порівняння двійкових чисел та методів проектування компараторів. Побудова та дослідження роботи цифрових компараторів

Теоретичні відомості

Цифровий компаратор призначений для порівняння двох двійкових чисел.

Він має дві групи входів. На одну з них надходять розряди першого числа (А), на другу групу – розряди другого числа (В). Три виходи компаратора появою логічної 1 фіксують результат порівняння. На одному виході вона встановлюється при рівності чисел (А=В), на іншому – при А>В, на третьому – при А<В.

Цифровий компаратор може використовуватися, наприклад, у системах автоматичного контролю і регулювання. При цьому число А є параметром деякого процесу, а число В – порогом, якого (відповідно до умов задачі) цей параметр не повинен перевищувати чи опускатися нижче його.

Таблиця 5.1.

Входи		Виходи		
a	b	F _{a>b}	F _{a=b}	F _{a<b}
1	1	0	1	0
1	0	1	0	0
0	1	0	0	1
0	0	0	1	0

У таблиці 5.1. подається зв'язок між сигналами на виходах і входах компаратора при порівнянні однорозрядних чисел а і b, що можуть бути рівні одиниці чи нулю. На відповідному виході з'являється логічна одиниця, коли в належному співвідношенні знаходяться коди на

входах. Так, якщо $a=1, b=1$ (числа однакові), то функція, яка характеризує рівність чисел, $F_{a=b}=1$, а функції, що характеризують їхню нерівність, $F_{a>b}=0$ і $F_{a<b}=0$. Аналогічно заповнюються інші рядки таблиці. За відомими правилами з таблиці 5.1 можна записати наступні логічні функції, що характеризують співвідношення однорозрядних чисел:

$$F_{a>b} = a\bar{b}, \quad F_{a=b} = ab + \bar{a}\bar{b}, \quad F_{a<b} = \bar{a}b$$

Якщо значення а і b такі, що права частина функції дорівнює 1, то співвідношення, зазначене в лівій частині, виконується. Якщо права частина функції дорівнює 0, то співвідношення між а і b протилежні зазначеному. Схема однорозрядного компаратора, що реалізує приведені функції, зображена на рис. 5.1.

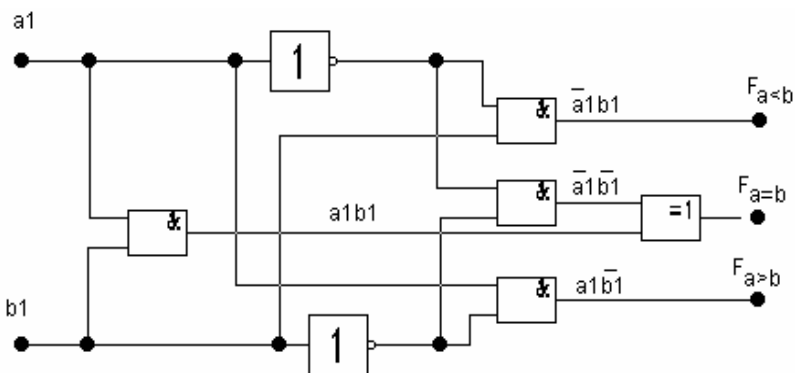


Рис. 5.1.

Зупинимося докладніше на рівності чисел. Відмітимо, що функція $F_{a=b}$ – функція “Рівнозначність”. За змістом вона протилежна функції “Нерівнозначність” (“Виключаюче АБО”):

$$F_{a \neq b} = a_1 \bar{b}_1 + \bar{a}_1 b_1 = a_1 \otimes b_1 \text{ тобто}$$

$$F_{a=b} = \bar{F}_{a \neq b} = a_1 \bar{b}_1 + \bar{a}_1 b_1 = a_1 \otimes b_1$$

Тому перевірку рівності пари однойменних розрядів двох чисел можна здійснити, використовуючи елемент “Рівнозначність” (рис. 5.2.а) чи елемент “Нерівнозначність”, доповнений інвертором (рис. 5.2.б.).

Два числа A і B рівні, якщо їхні однойменні розряди містять однакові цифри ($a_0=b_0$ і $a_1=b_1$ і ... $a_{n-1}=b_{n-1}$), тобто функція, яка характеризує співвідношення чисел, повинна бути кон’юнкцією функцій, які характеризують співвідношення цифр в їхніх однойменних розрядах:

$$F_{A=B} = F_{a_0=b_0} F_{a_1=b_1} \dots F_{a_{n-1}=b_{n-1}}$$

Коли цифри в однойменних розрядах чисел A і B однакові, на виходах всіх елементів “Рівнозначність” (рис.5.2) будуть логічні одиниці і $F_{A=B}=1$. Якщо хоча б в одній парі розрядів знаходяться різні цифри, то на виході відповідного елемента “Рівнозначність” буде логічний 0 і функція $F_{A=B}=0$, що вказує на нерівність чисел A і B .

Розглянемо випадки нерівності чисел. Виявлення більшого з двох багаторозрядних чисел A і B починається із старших розрядів; якщо вони рівні, то порівнюється наступна пара однойменних розрядів і т.д.

Так, у випадку трьохрозрядних чисел $A > B$, якщо мають місце :

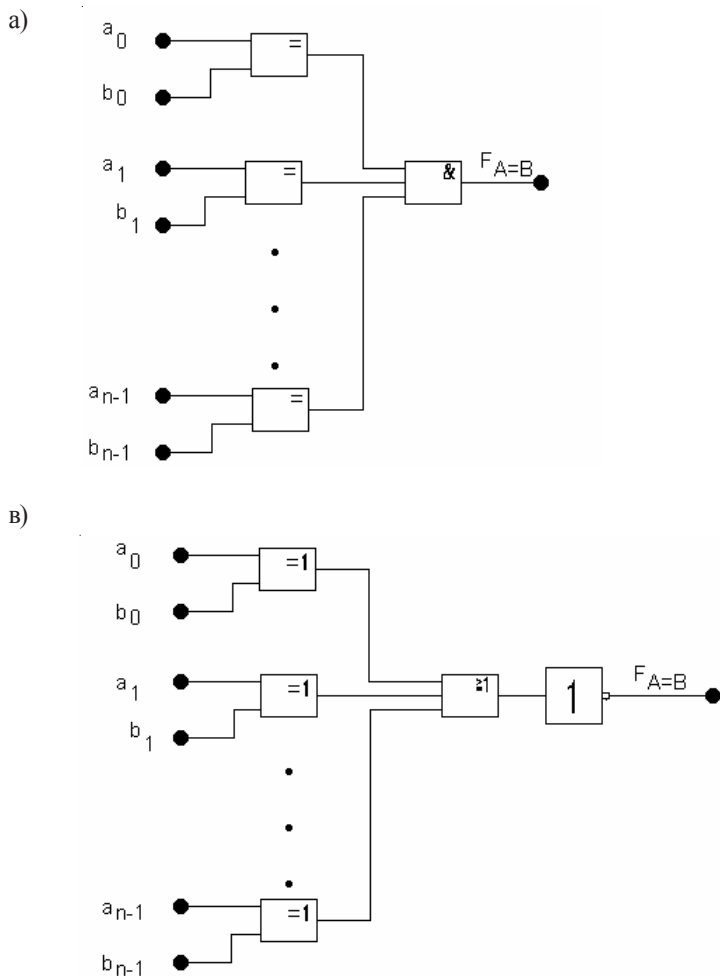


Рис. 5.2.

а) нерівність старших розрядів ($a_2 > b_2$, $F_{a_2 > b_2} = a_2 \bar{b}_2 = 1$);

б) чи при рівності старших розрядів ($a_2 = b_2$, $F_{a_2 = b_2} = a_2 b_2 + \bar{a}_2 \bar{b}_2 = 1$); існує нерівність розрядів a_1, b_1 ($a_1 > b_1$, $F_{a_1 > b_1} = a_1 \bar{b}_1 = 1$);

в) чи при рівності ($a_2 = b_2$, $F_{a_2 = b_2} = a_2 b_2 + \bar{a}_2 \bar{b}_2 = 1$); і ($a_1 = b_1$,

$F_{a_1 = b_1} = a_1 b_1 + \bar{a}_1 \bar{b}_1 = 1$); існує нерівність розрядів a_0, b_0 ($a_0 > b_0$, $F_{a_0 > b_0} = a_0 \bar{b}_0 = 1$);

Позначивши для стислості, $F_{a_2 > b_2} = a_2 b_2 + \bar{a}_2 \bar{b}_2 = F_2$, $F_{a_1 = b_1} = a_1 b_1 + \bar{a}_1 \bar{b}_1 = F_1$

запишемо приведені умови у вигляді

$$F_{A>B} = a_2 \bar{b}_2 + a_1 \bar{b}_1 F_2 + a_0 \bar{b}_0 F_2 F_1$$

При $a_2 > b_2$ ($a_2=1, b_2=0$) кон'юнкція $a_2 \bar{b}_2 = 1$ - функція $F_{a>b} = 1$, що вказує на справедливність нерівності $A > B$.

Коли $a_2 = b_2$ ($F_2 = 1, a_2 \bar{b}_2 = 0$), але $a_1 > b_1$ ($a_1 = 1, b_1 = 0, a_1 \bar{b}_1 = 1$) на справедливність нерівності $A > B$ вказує другий член записаної функції: $a_1 \bar{b}_1 F_2 = 1$.

Якщо $a_2 = b_2$ ($F_2 = 1, a_2 \bar{b}_2 = 0$) і $a_1 = b_1$ ($F_1 = 1, a_1 \bar{b}_1 = 0$), але

$a_0 = b_0$ ($a_0 = 1, b_1 = 0, a_0 \bar{b}_0 = 1$), то $a_0 \bar{b}_0 F_2 F_1 = 1$ і $F_{A>B} = 1$

Функція (1) реалізується схемою рис. 5.3. На рис. 5.4. вона доповнена елементом "Рівнозначність", на входи якого подаються розряди a_0, b_0 ; кон'юнктом, на виході

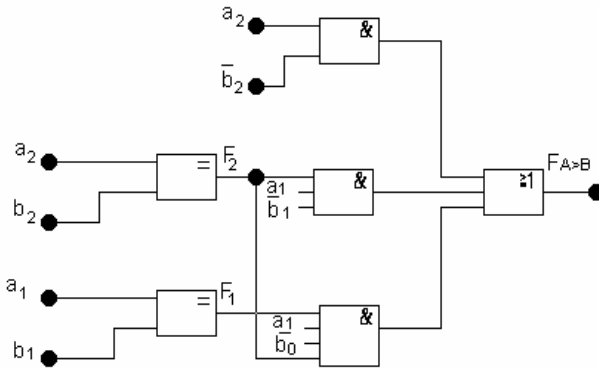


Рис.5.3.

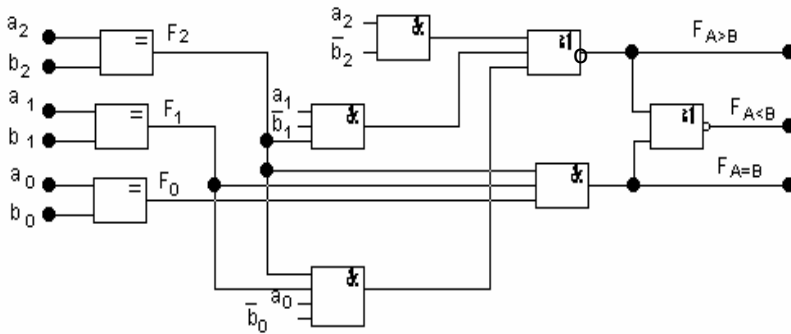


Рис.5.4.

якого формується функція $F_{A=B}$, і елементом АБО-НЕ, на виході якого формується функція $F_{A<B}$. Коли $a_2=b_2$ ($F_2=1$), $a_1=b_1$ ($F_1=1$) і $a_0=b_0$ ($F_0=1$), то $F_{A=B}=F_2 F_1 F_0=1$, тобто $A=B$. Якщо в результаті порівняння чисел $F_{A>B}=0$ і $F_{A=B}=0$, то на виході АБО-НЕ буде логічна 1 ($F_{A<B}=1$), тобто $A<B$.

По аналогічних схемах (рис. 5.4) виконуються компаратори для порівняння чисел з більшою розрядністю.

Завдання до лабораторної роботи (на основі елементів XOR і XNOR)

Варіант I.

1. Скласти схему цифрового компаратора для порівняння на рівність двох трирозрядних кодів.
2. Скласти схему цифрового компаратора для порівняння на більше, менше або дорівнює двох чотирирозрядних кодів.
3. Використовуючи побудований цифровий компаратор визначити найбільше з трьох чотирирозрядних двійкових чисел.

Варіант II.

1. Скласти схему цифрового компаратора для порівняння на рівність двох чотирирозрядних кодів.
2. Скласти схему цифрового компаратора для порівняння на більше, менше або дорівнює двох дворозрядних кодів.
3. Використовуючи побудований цифровий компаратор визначити найменше з 4 двохрозрядних двійкових чисел.

Варіант III.

1. Скласти схему цифрового компаратора для порівняння на рівність двох чотирирозрядних кодів.
2. Скласти схему цифрового компаратора для порівняння на більше, менше або дорівнює двох трирозрядних кодів.
3. Використовуючи побудований цифровий компаратор визначити найменше з трьох трирозрядних двійкових чисел.

Питання для самоконтролю

1. Для чого призначений компаратор?
2. Які функції виконує цифровий компаратор, в яких пристроях він може бути використаний?
3. Запишіть умову рівності (нерівності) двох однорозрядних чисел.

ЛАБОРАТОРНА РОБОТА №6

Тема: Проектування суматорів.

Мета: Вивчення правил виконання арифметичних дій над двійковими числами. Дослідження методів проектування суматорів, отримання навиків у побудові однорозрядних суматорів.

Теоретичні відомості

Основною дією над двійковими числами є додавання. Воно використовується як саме по собі, так і в операціях віднімання, а також лежить в основі множення і ділення чисел.

Суматор призначений для арифметичного додавання двох чисел. При додаванні двох багаторозрядних двійкових чисел по модулю два (доданки і перенос, який поступив з молодшого розряду) і формується сигнал переносу в старший розряд.

Таблиця 6.1.

Вхід		Вихід		
Доданки	Перенос	Сума	Перенос	
x_i	y_i	z_i	s_i	p_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Таблиця 6.1. таблиця істинності однорозрядного суматора.

Так як цей суматор буде виконувати додавання чисел в двійковій системі числення, то коефіцієнт $k=2$. Виходячи з таблиці, складемо діаграму Вейча для функцій s_i і p_i (рис.6.1).

Виконавши відповідні склеювання одержимо мінімальні диз'юнктивні нормальні форми (МДНФ) функцій s_i і p_i .

$$s_i = x_i \bar{y}_i \bar{z}_i \vee \bar{x}_i \bar{y}_i z_i \vee \bar{x}_i y_i \bar{z}_i \vee x_i y_i z_i$$

$$p_i = x_i y_i \vee x_i z_i \vee y_i z_i$$

При використанні елементів І-АБО/НЕ, однорозрядний комбінаційний суматор (рис.6.2.) можна побудувати виходячи з таких виразів:

$$s_i = x_i y_i z_i \vee x_i \bar{p}_i \vee y_i \bar{p}_i \vee z_i \bar{p}_i$$

Для того, щоб не ускладнювати конструкцію АЛП (арифметично логічного при-

строю), операцію віднімання замінюють додаванням (що виконується суматором) зменшеного з від'ємником, представленим в спеціальному коді. Розглянемо це більш детально.

Віднімання двійкових чисел, записаних в прямому коді, подібне до віднімання в десятковій системі Рис.6.3.

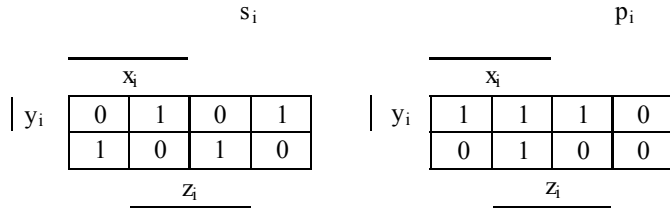


Рис.6.1.

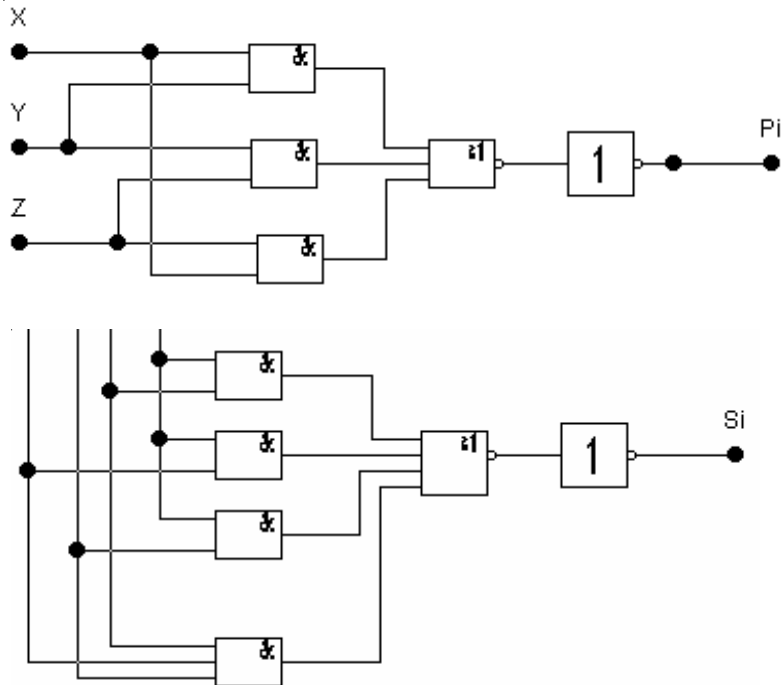


Рис.6.2.

	⇩	⇩
_ A	_ 10111	_ 23
B	1110	14
— F	— 1001	9

Рис.6.3.

Стрілками показана операція “позичання”, яка виконується для тих розрядів, в яких від’ємник більший за зменшуване. В десятковій системі позичається одиниця старшого розряду, яка дорівнює десяти одиницям сусіднього молодшого розряду, а в двійковій – двом одиницям молодшого розряду.

Для заміни операції віднімання операцією додавання потрібно подати від’ємник В в додатковому коді. Додатковий код утворюється з оберненого (інверсного) кода, додаванням до нього одиниці. Так, чотирьохрозрядний від’ємник В, представлений в прямому коді $B_{пр} = B_4 B_3 B_2 B_1$, може бути поданий і в оберненому коді $B_{обер} = \bar{B}_4 \bar{B}_3 \bar{B}_2 \bar{B}_1$, і в додатковому коді $B_{дод} = \bar{B}_{обер} + 1$. Очевидно, для чотирьохрозрядних чисел, записаних в цих кодах, справедливі рівності:

$$B_{пр} + B_{обер} = 1111;$$

$$B_{пр} + B_{дод} = B_{пр} + B_{обер} + 1 = 1111 + 1 = 10000;$$

$$B_{пр} = 10000 - B_{дод} = 10000 - B_{обер} - 1.$$

А тому, операцію віднімання можна подати у вигляді:

$$A_{пр} - B_{пр} = A_{пр} + B_{дод} - 10000.$$

Таким чином, в АЛП при виконанні операції віднімання, вхідний операнд В перетворюється в додатковий код, а віднімання числа 10000 виконується без допомоги спеціальних схем, тільки з використанням сигналу переноса в старший (п’ятий) розряд. Але, при цьому результат арифметичних дій на виході АЛП буде також поданий в оберненому коді.

Завдання до лабораторної роботи

Варіант І.

1. Використовуючи логічні елементи І, НЕ, І-НЕ, побудувати однорозрядний повний суматор.

2. На основі побудованого однорозрядного повного суматорсуматора побудувати схему для демонстрації додавання двох трирозрядних двійкових чисел.

3. За допомогою трирозрядного суматора продемонструвати віднімання двох трирозрядних чисел.

Варіант II.

1. Використовуючи логічні елементи АБО, НЕ, АБО-НЕ, побудувати однорозрядний повний суматор.

2. На основі побудованого однорозрядного повного суматорсуматора побудувати схему для демонстрації додавання двох чотирирозрядних двійкових чисел.

3. За допомогою чотирирозрядного суматора продемонструвати віднімання двох чотирирозрядних чисел.

Варіант III.

1. Використовуючи логічні елементи І, НЕ, АБО-НЕ, побудувати однорозрядний повний суматор.

2. На основі побудованого однорозрядного повного суматорсуматора побуду-

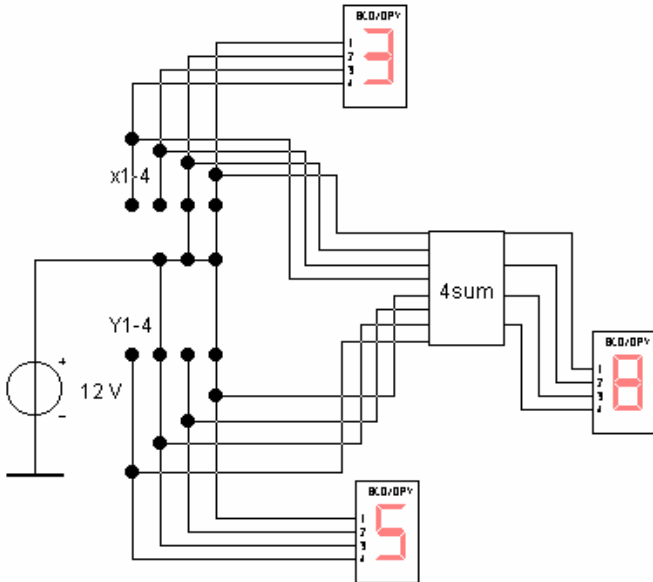



Рис.6.4.

вати схему для демонстрації додавання двох трирозрядних двійкових чисел.

3. За допомогою трирозрядного суматора виконати дії над трьома трирозрядними числами $A+B-C$.

Примітка

Зібравши повністю однорозрядний суматор перевірте його таблицю істинності використавши генератор слів або джерело напруги. Після проведення перевірки помістіть ваш суматор у бібліотеку Workbench через меню Circuit/Create Subcircuit (Ctrl+B). Далі виведіть назовні на уніфікованому вами суматорі входи і виходи.

В результаті схема вашого однорозрядного суматора буде використовуватись надалі в вигляді: 

Для побудови демонстраційної схеми додавання двох чотирирозрядних (трирозрядних) чисел найкраще використати чотири (три) ваших уніфікованих суматори (відповідним чином їх з'єднавши і теж помістивши у бібліотеку Workbench), три семисегментні індикатори (два для контролю аргументів і один для контролю результату) та джерело напруги у ролі джерел чотирирозрядних двійкових чисел, що будуть виступати аргументами.

Для перевірки подайте певним чином на входи x_1-x_4 та на y_1-y_4 напругу, що буде еквівалентно деяким двійковим числам на цих входах. Індикатори на входах відобразять шістнадцятковий еквівалент цих чисел. На виході ви отримаєте шістнадцятковий результат. Таку операцію проробіть для різних комбінацій аргументів.

На рис.6.4. зображена схема для перевірки чотирирозрядного суматора.

4sum – уніфікований вами чотирирозрядний суматор складений з однорозрядних, з'єднаних відповідним чином.

3. За допомогою чотирьохрозрядного суматора продемонструвати віднімання двох чотирьохрозрядних чисел.

Питання для самоконтролю

1. Для чого призначений суматор?
2. Скільки входів та виходів має однорозрядний суматор?
3. Для чого використовуються семисегментні індикатори при побудові схеми для демонстрації додавання двох чотирирозрядних двійкових чисел?
4. Як утворюється додатковий код?

ЛАБОРАТОРНА РОБОТА №7

Тема: Проектування та дослідження тригерів.

Мета: Вивчення функціонування і схемних різновидностей тригерів, оволодіння методами їх проектування.

Теоретичні відомості

Тригер - найпростіша цифрова схема послідовного типу. У тригерів стан виходу Y у в будь-який момент часу визначається поточним станом входу X та внутрішнім станом схеми Q : $Y=F(X,Q)$

Іншими словами, цифровий автомат є не тільки перетворювачем, але й зберігає попередню й поточну інформацію (стани). Ця властивість забезпечується наявністю в схемах зворотних зв'язків.

Основою послідовних схем є тригери. Тригер має два стійких стани $Q=1$ і $Q=0$, тому його іноді називають бістабільною схемою. В якому з цих станів опиниться тригер, залежить від сигналів на входах тригера і від його попереднього стану, тобто він має пам'ять. Можна сказати, що тригер є елементарною коміркою пам'яті.

Тип тригера визначається алгоритмом його роботи. В залежності від алгоритму роботи, тригер може мати настановні, інформаційні та керуючі входи. Наставновні входи встановлюють стан тригера незалежно від стану інших входів. Входи керування дозволяють запис даних, які надходять на інформаційні входи.

Асинхронний RS – тригер.

RS - тригер має два входи S і R , основний і інверсний виходи. Стан тригера визначається по сигналу на основному вході. Тригер має два настановні входи: установки S (set - установка) і скиду R (reset - скид), на які подаються вхідні сигнали від зовнішніх джерел. При подачі керуючого сигналу на вхід S на основному виході встановлюється логічна одиниця або ця одиниця підтверджується, якщо вона там була (Таблиця.7.1.). При подачі керуючого сигналу на вхід R , на основному виході з'являється логічний нуль, як говорять, тригер скидається. Якщо тригер був уже скинутий, то скидання підтверджується. Подача керуючих сигналів одночасно на входи S і R заборонена. У відсутності керуючих сигналів стан тригера змінитися не може, тригер знаходиться в режимі збереження інформації. В залежності від типу логічних елементів, на яких зібраний тригер, керуючими сигналами можуть бути як нулі, так і

Таблиця 7.1.

Q^t	S^t	R^t	Q^{t+1}	
0	0	0	0	Збереження інформації
0	0	1	0	Підтвердження 0
0	1	0	1	Установка в 1
0	1	1	X	Заборона
1	0	0	1	Збереження інформації
1	0	1	0	Скидання в 0
1	1	0	1	Підтвердження 1
1	1	1	X	Заборона

одиниці. У таблиці 7.1. Q^t - це значення вихідного сигналу до моменту подачі керуючих сигналів S^t і R^t , або його вихідний стан. Q^{t+1} – новий стан тригера після подачі керуючих сигналів, якими є логічні одиниці.

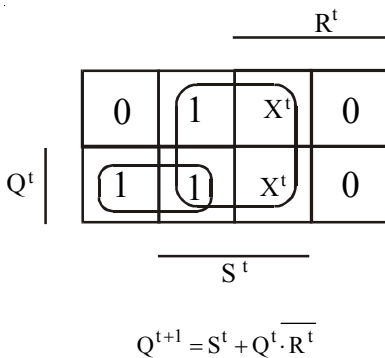


Рис. 7.1.

На рис.7.1. показана процедура мінімізації функції Q^{t+1} з використанням карти Карно. Отримана формула описує роботу RS- тригера, але схеми тригерів будують після перетворення цієї формули, замінюючи операцію множення на додавання або додавання замінюють множенням. Після заміни множення на додавання по 16-й теоремі Булевої алгебри можна одержати наступну формулу $Q^{t+1} = S^t + \overline{Q^t} + R^t$. Якщо замінити додавання на множення, то одержимо $Q^{t+1} = \overline{\overline{S^t} \cdot \overline{Q^t} \cdot \overline{R^t}}$. Схеми тригерів, побудо-

вані по цих формулах показані на рис. 7.2. Перша зі схем побудована на елементах АБО-НЕ, цей тригер керується логічними одиницями. Таблиця його функціонування приведена в таблиці 7.1. Схема тригера, побудована по другій формулі на елементах І-НЕ найпоширеніша, цей тригер керується логічними нулями, тобто має інверсні входи. Таблиця 7. 2. – таблиця функціонування даного RS – тригера. Широкому використанню асинхронного RS – тригера в якості самостійного при-

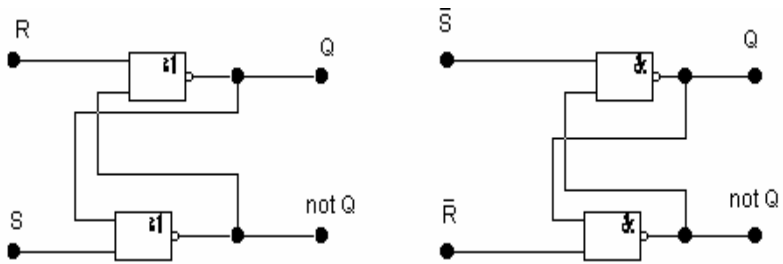


Рис.7.2

Таблиця 7.2.

Q^t	S^t	R^t	Q^{t+1}	
0	0	0	X	Заборона
0	0	1	1	Установка в 1
0	1	0	0	Підтвердження 0
0	1	1	0	Збереження інформації
1	0	0	X	Заборона
1	0	1	1	Підтвердження 1
1	1	0	0	Скидання
1	1	1	1	Збереження інформації

строю заважають серйозні його недоліки: наявність заборонених комбінацій вхідних сигналів, подача інформації по двох окремих входах.

Синхронний RS – тригер.

RS – тригер може бути синхронним. У цьому випадку крім двох інформаційних входів S і R тригер має ще вхід синхронізації. Сигнали на входах S і R лише готують тригер до потрібного переключення, а саме переключення відбувається тільки в момент подачі синхронізуючого імпульсу. Синхронізація організовується за допомогою двох додаткових елементів І-НЕ. При відсутності сигналу синхронізації ($C = 0$) на входах асинхронного RS – тригера встановлюються дві одиниці, що забезпечує в ньому збереження інформації. При подачі синхронізуючого сигналу ($C = 1$) тригер переключається відповідно поданій інформації на входи S і R.

Синхронний D – тригер.

D – тригер, називається ще тригером затримки, може бути асинхронним і синхронним. Однак асинхронний D – тригер змісту не має, тому що він містить один інформаційний вхід D та основний і інверсний виходи. Опис роботи тригера при різних комбінаціях вхідних сигналів представлений в таблиці 7.3. Сигнал (інформація) на виході завжди збігається з інформацією на вході, тобто $Q^{t+1} = D^t$. Зміст має тільки синхронний D – тригер, у якого, крім інформаційного входу D, є вхід синхронізації C. Інформація з входу D передається на основний вихід (записується в тригер) у момент приходу синхронізуючого імпульсу. Структурна формула, що описує роботу синхронного D – тригера наступна: $Q^{t+1} = Q^t \cdot \bar{C}^t + C^t \cdot D^t$. З формули видно, що при $C = 0$ стан тригера не змінюється $Q^{t+1} = Q^t$, а при $C = 1$ стан тригера збігається із значенням інформації на вході D $Q^{t+1} = D^t$. У такий спосіб, при відсутності синхронізуючого імпульсу, стан тригера не змінюється, інформація записана в тригер зберігається (затримується) на період проходження синхронізуючих імпульсів. На рис. 7.3. показаний один з варіантів схеми D – тригера. При $C = 0$, на входах асинхронного RS – тригера, що входить до складу D – тригера, встановлюються дві одиниці, що означає збереження інформації. Можна простежити за схемою, що при $C = 1$ тригер встановиться в 1, якщо на його вході D була 1, і скинеться в 0, якщо на вході D був логічний 0. Якщо сигнал зміниться під час дії синхроімпульсу

Таблиця 7.3.

C^t	D^t	Q^{t+1}	
0	0	Q^t	Збереження інформації
0	1	Q^t	Збереження інформації
1	0	0	Запис 0
1	1	1	Запис 1

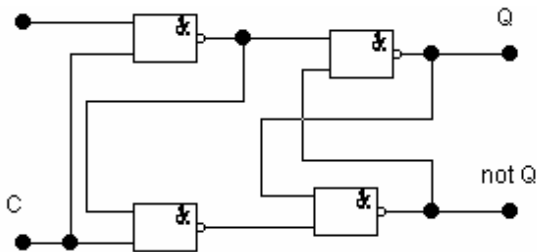


Рис. 7.3.

су, то в тригері виявиться записаною та інформація, яка була перед закінченням синхроімпульсу. Завдяки цій властивості (зміни інформації на протязі всього часу поки $C=1$) даний тригер називається статичним синхронним D – тригером. Для нормальної роботи статичного D – тригера необхідно, щоб зміна інформації на D – вході відбувалась тільки при $C=0$.

Динамічний синхронний D – тригер

Динамічний синхронний D – тригер виключає наскрізну передачу сигналу з D входу на вихід тригера в момент дії синхроімпульсу. В тригері з динамічним керуванням інформація записується тільки в момент перепаду напруги на вході сигналу синхронізації. Схема динамічного синхронного D – тригера показана на рис.7.4. Умовне графічне позначення рис.7.5.а.

Динамічний D – тригер складається з трьох статичних RS – тригерів. Перших два (складених з елементів D1, D3 та D2, D4) готують інформацію. Третій тригер (D5, D6) записує попередньо оброблену інформацію. Саме така двохступінчаста побудова

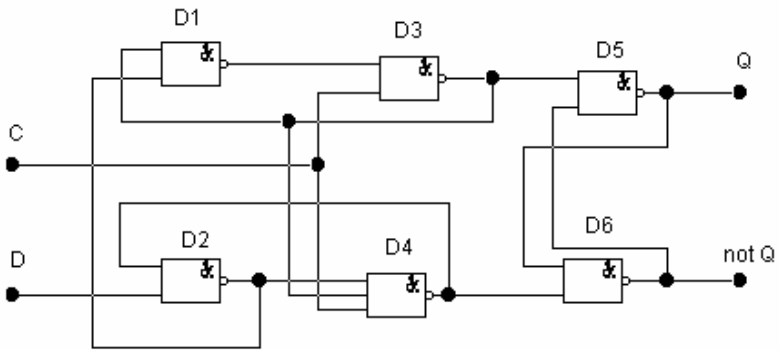


Рис. 7.4.

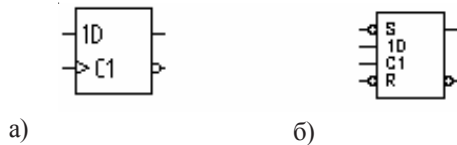


Рис. 7.5.

динамічного тригера і дозволяє позбутися від прямого проходження сигналу з D – входу на вихід тригера в період дії синхроімпульсу.

Ускладнення схеми рис.7.4. дозволяє отримати універсальний D – тригер рис.7.5.б., який виконує функції як RS – тригера так і динамічного D – тригера. Модернізація заключається в заміні всіх двовхідних елементів І-НЕ на трьохвхідні елементи І-НЕ. Додаткові входи елементів D1, D5 будуть входами сигналу not S, а входи D2, D3, D6 – входами сигналу not R. Поки сигнал на not S і not R рівний 1, універсальний тригер працює як динамічний D – тригер. Як тільки на один з входів not S або not R поступає сигнал, рівний 0, то тригер зразу перестає реагувати на сигнали С і D і приймає стан який визначається сигналами not S або not R. Стан not S=not R =0 вважається забороненим.

T-тригер(лічильний тригер)

Несинхронний T – тригер має один вхід T, основний і інверсний виходи. Інформація на виходах змінюється на протилежну при кожному позитивному перепаді напруги на вході T, тому цей тригер використовують в якості подільника частоти вхідного сигналу. Тригер такого типу може бути створений з D – тригера з динамічним управлінням, якщо його інверсний вихід з'єднати з інформаційним входом. При цьому, якщо в початковий момент часу на виході Q був нульовий рівень, то на вході D був рівень not Q=1. По фронту першого синхроімпульсу одиниця з D-входу переписується (з запізненням, рівним затримці одного логічного елемента) на вихід Q. Відповідно, на виході і вході D появиться нульовий рівень (з запізненням, рівним затримці двох логічних елементів). В наступному такті на вихід Q буде переписано нульове значення з входу D і т. д.

Створити лічильний тригер на базі статичного D-тригера, зворотнім зв'язком з виходу not Q на вхід D, не можливо. Так як статичний тригер має потенційне управління, то при C=1 напруга на виході, за рахунок впливу зворотного зв'язку, буде постійно змінюватися на протилежну, тобто виникне високочастотне коливання.

JK-тригер

Несинхронний JK – тригер має два входи J і K, основний і інверсний виходи. Вхід J аналогічний входіві S RS – тригера. По цьому вході тригер встановлюється в стан «1». По входу K тригер скидається в «0», як і RS – тригер по входу R.

Відмінність від RS – тригера полягає в тому, що цей тригер не має заборонених комбінацій сигналів на входах, а при подачі керуючих сигналів одночасно на обидва входи, тригер переключиться в протилежний стан (Таблиця.7.4). Синхронний JK – тригер має ще один вхід С – вхід синхронізації і переключиться тільки при подачі імпульсу на цей вхід.

Структурна формула, що описує роботу несинхронного JK – тригера має такий вигляд: $Q^{t+1} = J^t \bar{Q}^t + \bar{K}^t Q^t$. JK – тригер називають універсальним, тому що з нього можна зробити будь-який тип тригера. RS – тригер виходить з JK – тригера, коли входи JK використовуються, як входи S і R відповідно, а заборонена комбінація не подається.

Таблиця 7.4.

Q^t	J^t	K^t	Q^{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Якщо у формулі несинхронного JK – тригера J назвати входом D, а на вхід K подати \bar{D} , то одержимо: $Q^{t+1} = D(\bar{Q}^t + Q^t) = D$, що відповідає несинхронному D – тригеру, але тому що несинхронний D – тригер змісту не має, то для одержання

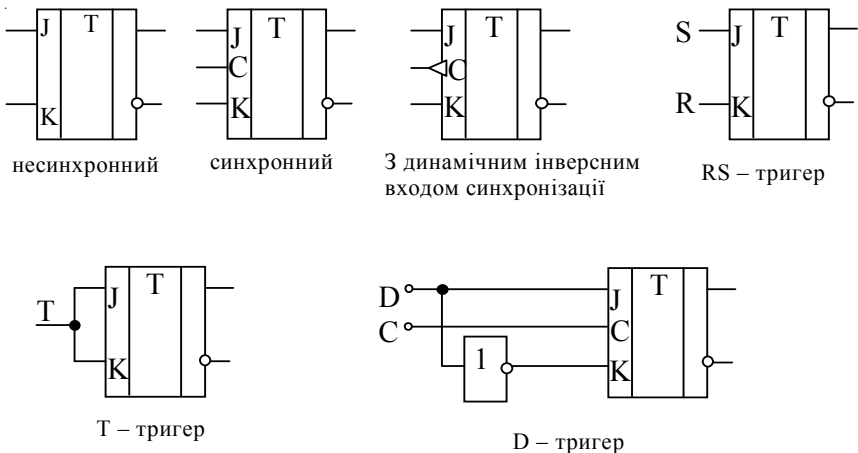


Рис.7.6.

синхронного D – тригера потрібно використовувати синхронний JK – тригер. Для одержання T – тригера досить об'єднати входи J і K і назвати цей вхід входом T по якому тригер буде переключатися в протилежний стан, як це повинен робити T – тригер.

На рис.7.6. показані умовні позначки JK – тригерів і виконання на основі JK – тригерів інші типи тригерів.

Завдання до лабораторної роботи

Варіант I.

1. Побудувати та розглянути принцип роботи асинхронного RS – тригера на елементах I-НЕ, та RS – тригера на елементах АБО-НЕ.
2. Побудувати та розглянути принцип роботи двохступінчатого синхронного RS – тригера.
3. Побудувати D – тригер статичний та динамічний.
4. Побудувати універсальний D – тригер, який виконує функції як RS – тригера, так і динамічного D – тригера.
5. На основі D – тригера побудувати T-тригер, та розглянути принцип його роботи.

Варіант II.

1. Побудувати та розглянути принцип роботи асинхронного RS – тригера на елементах I-НЕ, та RS – тригера на елементах АБО-НЕ.
2. Побудувати та розглянути принцип роботи синхронного RS – тригера на елементах АБО -НЕ.
3. Побудувати D – тригер статичний та динамічний.
4. Побудувати Універсальний JK – тригер.
5. На основі JK – тригера побудувати T-тригер, та розглянути принцип його роботи.

Варіант III.

1. Побудувати та розглянути принцип роботи асинхронного RS – тригера на елементах I-НЕ, та RS – тригера на елементах АБО-НЕ.
2. Побудувати та розглянути принцип роботи синхронного RS – тригера на елементах I-НЕ.

3. Побудувати D – тригер статичний та динамічний.
4. Побудувати синхронний JK –тригер та розглянути принцип його роботи.
5. На основі JK – тригера побудувати T-тригер, та розглянути принцип його роботи.

Питання для самоконтролю

1. Чим відрізняється робота RS-тригера з прямими входами від роботи з інверсними входами?
2. Чому комбінація сигналів 11 на входах RS-тригера називається “забороненою”?
3. Як властивість запам’ятовування відображається в характеристичних рівняннях тригерів?
4. В чому принципова різниця роботи синхронних тригерів від асинхронних ?
5. Який пріоритет інформаційних і установочних входів в синхронних тригерах?
6. Чому JK-тригер при $J=K=1$ не перетворюється в автогенератор?
7. Чому T- тригер отримав назву лічильного ? Яке число імпульсів він може полічити?
8. Як працює D- тригер, якщо $D=Q$?

ЛАБОРАТОРНА РОБОТА №8

Тема: Проектування та дослідження регістрів.

Мета: Вивчення схемних різновидностей регістрів, оволодіння методами їх проектування.

Теоретичні відомості

Регістром називається пристрій, що здійснює прийом, збереження, перетворення і видачу чисел у двійковому коді. Інформація в регістрі зберігається у вигляді числа. Він містить у собі окремі тригери, кількість яких відповідає числу розрядів двійкового коду, і логічні елементи.

Регістри виконують ряд мікрооперацій над словами:

1. Прийом слова в регістр у прямому і оберненому коді, дані зберігаються в регістрі поки не з'явиться команда на їх зміну.
2. Видача слова з регістра в прямому і оберненому коді.
3. Виконання порозрядних логічних операцій над декількома словами.
4. Зсув коду вправо чи вліво на необхідне число розрядів, перетворення паралельного коду в послідовний і навпаки.

За способом запису і зчитування коду числа в регістрі, варто розрізнити паралельні і послідовні регістри:

У паралельному регістрі на тактуючих D-тригерах (рис. 8.1) код числа, що запам'ятовується, подається на інформаційні входи всіх тригерів і записується в регістр із приходом тактового імпульсу. Вихідна інформація змінюється з подачею нового

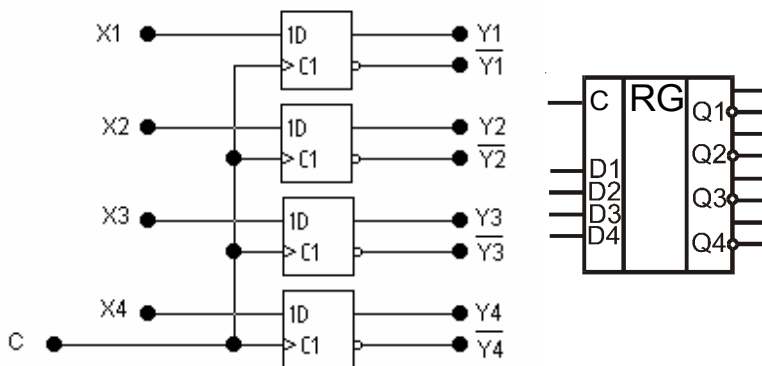


Рис.8.1

вхідного слова і приходом наступного імпульсу запису. Такі регістри використовують у системах оперативної пам'яті. Число тригерів в них дорівнює максимальній розрядності збережених слів.

Схема послідовного регістра на D-тригерах з динамічним керуванням, наведена на рис. 8.2. По приході тактового імпульсу С перший тригер записує код X (0 або 1), що перебуває в цей момент на його D-вході, а кожний наступний тригер перемикається в стан, у якому до цього перебував попередній. Так відбувається тому, що записуваний сигнал проходить із входу D-тригера до виходу Q із затримкою, більшої тривалості переднього фронту тактового імпульсу (протягом якого й відбувається запис). Кожний тактовий імпульс послідовно зсуває код числа в регістрі на один розряд. Тому для запису n-розрядного коду потрібно n тактових імпульсів. Чотиризначне число 1011 було записане у відповідні розряди регістра (1 - Q4, 0 - Q3, 1 - Q2, 1 - Q1) після приходу четвертого тактового імпульсу. До наступного тактового імпульсу це число зберігається в регістрі у вигляді паралельного коду на виходах Q4, ..., Q1. Якщо необхідно одержати збережену інформацію в послідовному коді, то її знімають із виходу Q4 у моменти приходу наступних чотирьох імпульсів (5 - 8). Такий режим називається режимом послідовного зчитування.

Дуже зручні універсальні регістри, що дозволяють здійснювати як послідовний, так і паралельний запис і зчитування. Такі регістри можна використовувати в якості перетворювачів паралельного коду в послідовний і навпаки.

На базі універсального регістра можна побудувати реверсивний регістр, що здійснює зсув вправо і зсув вліво.

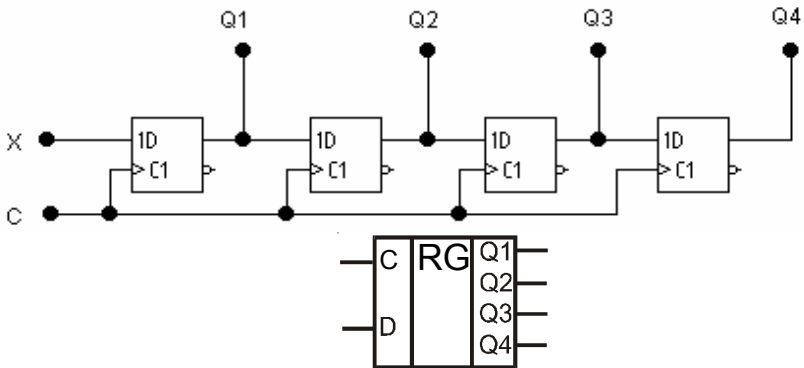


Рис.8.2.

Завдання до лабораторної роботи

Варіант I.

1. Складіть принципову схему 4-розрядного послідовно реєстра на D-тригерах.
2. Складіть принципову схему 6-розрядного послідовно-паралельного реєстра на D-тригерах, в якому запис здійснюється в послідовному коді, а прийом в паралельному.
3. Складіть принципову схему 4-розрядного послідовно-паралельного реєстра на JK-тригерах із двома керуючими входами, один із яких встановлює паралельне введення числа, а інший послідовне.

Варіант II.

1. Складіть принципову схему 8-розрядного паралельного реєстра на D-тригерах.
2. Складіть принципову схему 4-розрядного паралельно-послідовного реєстра на JK-тригерах, в якому запис здійснюється в паралельному коді, а прийом в послідовному.
3. Складіть принципову схему 3-розрядного послідовно-паралельного реєстра на JK-тригерах із двома керуючими входами, один із яких встановлює паралельне введення числа, а інший послідовне.

Варіант III.

1. Складіть принципову схему 8-розрядного паралельного реєстра на JK-тригерах.
2. Складіть принципову схему 4-розрядного послідовно-паралельного реєстра на JK-тригерах, в якому запис здійснюється в послідовному коді, а прийом в паралельному.
3. Складіть принципову схему реверсивного 3-розрядного реєстра зсуву на JK-тригерах.

Питання для самоконтролю

1. Для чого призначені реєстри і яких типів вони бувають?
2. На яких тригерах можуть бути реалізовані реєстри?
3. Якою повинна бути мінімальна розрядність реєстра для запису чисел, десятиковий еквівалент найбільшого з яких рівний 45?
4. Скільки розрядів повинен мати зсувний реєстр, щоб тризначне двійкове число можна було збільшити у вісім разів?

ЛАБОРАТОРНА РОБОТА № 9

Тема: Проектування лічильників.

Мета: Вивчення різних типів лічильників, оволодіння методами їх проектування.

Теоретичні відомості

Лічильником називають операційний елемент послідовної дії, що здійснює рахунок імпульсів, які надходять на його вхід. Результат рахунку зберігається лічильником до приходу наступного імпульсу. Зчитування результату рахунку може відбуватися в проміжках між рахунковими імпульсами.

Лічильники, як і зсувні регістри, складаються з ланцюжка послідовно включених тригерів. Розрядність лічильника, а отже, і число тригерів N визначаються максимальним числом, до якого він повинен рахувати. Це число називається коефіцієнтом (модулем) рахунку – K_p . Якщо число вхідних імпульсів $n > K_p$, то через кожні K_p імпульсів лічильник повертається у вихідний стан і починає рахувати імпульси спочатку.

Велика різномантність типів лічильників викликана їхнім широким використанням як в обчислювальній техніці, так і в різних пристроях автоматики. Вони застосовуються для утворення послідовностей адрес команд, для рахунку числа циклів виконання операцій, для запам'ятовування коду в аналого-цифрових перетворювачах і т.д.

Розглянемо найпоширеніші типи лічильників

Асинхронний (послідовний) двійковий лічильник.

Такий лічильник утворений ланцюжком послідовно ввімкнених лічильних тригерів (рис.9.1). Результат рахунку відображається на виходах лічильника $Q(N-1), \dots, Q_0$ у вигляді паралельного двійкового коду числа порахованих імпульсів. Оскільки число вихідних змінних дорівнює числу тригерів N і кожна змінна може приймати лише два значення, то число можливих станів (коефіцієнт рахунку) дорівнює: $K_p = 2^N$. Тому що з 2^N станів одне доводиться на нульовий стан, то максимальне число, при якому лічильник повністю заповнюється одиницями, дорівнює $(2^N - 1)$.

Найпростішим однорозрядним лічильником з $K_p = 2$ є Т-Тригер, що міняє свій стан на протилежний під дією кожного вхідного сигналу. У результаті перепаду на-

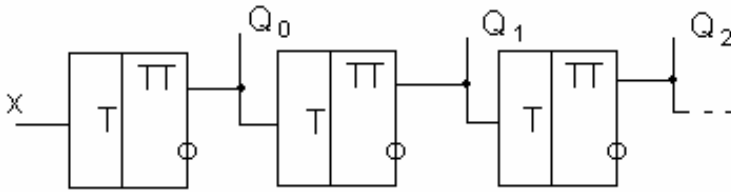


Рис.9.1.

пруги на виході тригери мають удвічі меншу частоту, ніж на вході. По цих перепадах запускається наступний тригер, і на його виході зміни стану відбуваються вже в чотири рази рідше, ніж на вході першого тригера.

З приходом кожного наступного імпульсу паралельний двійковий код на виході лічильника буде збільшуватися на одиницю, поки не наступить переповнення лічильника, при якому всі тригери скинуться в нульовий стан.

Істинна інформація на виходах лічильника встановлюється тільки через час $N \cdot t$, що пройшов після надходження тактового імпульсу. Де t – затримка поширення імпульсу в кожному тригері. При подальшому збільшенні розрядності, сумарна затримка може привести до спотворювання інформації в лічильнику. Тому багато-розрядні лічильники з послідовним переносом рахункових імпульсів від тригера до тригера можуть працювати тільки на знижених частотах, при досить великих періодах проходження імпульсів.

Крім розглянутого підсумовуючого лічильника, є й лічильники, що віднімають, у яких вихідний код зменшується на 1 із приходом кожного рахункового імпульсу. Такий лічильник отримується при подачі інверсних сигналів на тактові входи. Для цього необхідно тактові входи тригерів підключити до інверсних виходів Q попередніх тригерів.

Синхронний (паралельний) двійковий лічильник.

Паралельні лічильники (рис.9.2) мають максимальну швидкодію, оскільки в них усі розряди переключаються одночасно. Паралельний лічильник містить розрядні тригери з кон'юнкторами, що аналізують стан попередніх розрядів. При надходженні вхідного сигналу переключаються тільки ті тригери, для яких усі попередні були в одиничному стані.

Час установки лічильника не залежить від розрядності і дорівнює

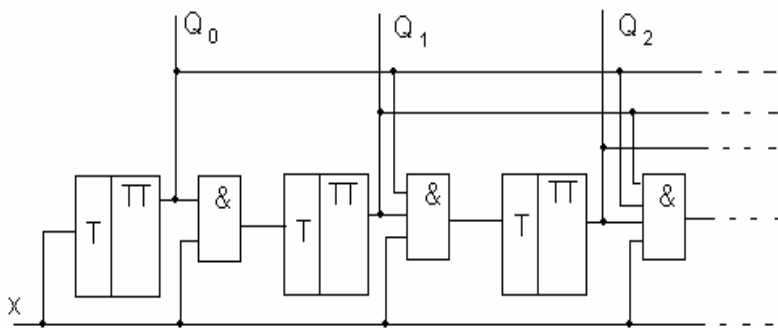


Рис.9.2.

$$t_{3,к} + t_{\text{т}}$$

де $t_{3,до}$ – затримка сигналу кон'юнктором.

t – затримка поширення імпульсу в тригері.

Труднощі реалізації багаторозрядних паралельних лічильників пов'язані з ростом числа входів у кон'юнкторів по мірі збільшення розрядності лічильника. Другий обмежувачий фактор – ріст навантаження на виходи тригерів по мірі збільшення числа розрядів лічильника. Застосування різних схем буферного типу для подолання зазначених обмежень небажано, тому що знижується швидкодія лічильника.

У паралельних лічильниках всі розряди переключаються одночасно, тому їхній структурі властиві тимчасові змагання сигналів. При використанні двоступінчастих тригерів ці недоліки виключаються.

Реверсивні лічильники.

Реверсивні лічильники змінюють напрямок рахунку під впливом керуючого сигналу чи при зміні точки подачі лічильних сигналів. У першому випадку схема має рахунковий і лічильний входи, у другому – два лічильних входи.

Найбільш розповсюджений спосіб побудови реверсивних лічильників – переключення міжрозрядних зв'язків. Лічильники прямого і зворотного рахунку розрізняються лише точкою знімання сигналу, який подається з попереднього розряду на наступний. Якщо керуючий сигнал перебудовує міжрозрядні зв'язки, переносючи точку знімання сигналу з одного виходу тригера на іншій, то реалізується схема реверсивного лічильника.

Послідовний лічильник перетвориться в реверсивний шляхом введення в його

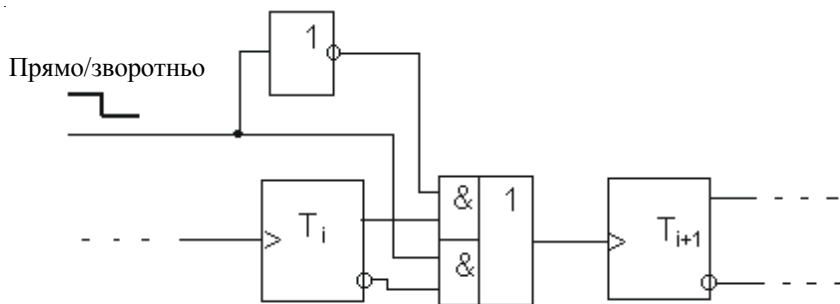


Рис.9.3.

структуру елементів реверса. У зв'язку з появою додаткових затримок введення реверса знижується швидкодія лічильника.

Лічильник з довільним коефіцієнтом рахунку.

Часто потрібні лічильники із числом стійких станів, відмінних від 2^N . Наприклад, в електронних годинниках є мікросхеми з коефіцієнтами рахунку 6 (десятки хвилин), 10 (одиниці хвилин), 7 (дні тижня), 24 (години). Для побудови лічильника з K_p не рівним 2^N можна використовувати пристрій з N-тригерів, для якого виконується умова $2^N > K_p$. Очевидно, такий лічильник має зайві стійкі стани ($2^N - K_p$). Виключити ці непотрібні стани можна використанням зворотних зв'язків, по ланцюгах яких лічильник перемикається в нульовий стан у тому такті роботи, у якому він дораховує до числа K_p .

Для лічильника з $K_p = 10$ потрібні чотири тригери (тому що $2^3 < 10 < 2^4$). Лічильник повинен мати десять стійких станів (0, ..., 9). У тому такті, коли він повинен був би перейти в одинадцятий стійкий стан (число 10), його необхідно встановити у вихідний нульовий стан. Для такого лічильника можна використовувати будь-який чотирирозрядний лічильник з зворотнім зв'язком з виходів, що відповідають числу 10 (тобто 2 і 8), на входи установки лічильника в 0. На самому початку одинадцятого стану (число 10) на обох входах елемента І мікросхеми з'являються логічні 1, що виробляють сигнал скидання всіх тригерів лічильника в нульовий стан.

Розглянутий лічильник є двійковим еквівалентом рахункової декади, що представляє будь-яку десяткову цифру її двійковим кодом. Тому такий лічильник називають двійково-десятковим, а його вихідний код - двійково-десятковим кодом (або ко-

дом 8421).

Якщо двійково-десятковий лічильник призначений для роботи в системах, де потрібна візуальна інформація про число підрахованих імпульсів (наприклад, різні цифрові вимірювальні прилади), то після лічильника ставиться перетворювач двійково-десятькового коду в код семисегментного індикатора.

Лічильник з попередньою установкою.

Лічильник може встановлюватися в початковий стан, рівний будь-якому числу від 0 до $K_p - 1$. Ця операція здійснюється паралельним записом у лічильник коду необхідного числа. Рахунок (додавання або вирахування) буде починатися вже не з нуля, а із встановленого числа. Такий режим роботи лічильника необхідний, команд із заданої початкової адреси.

Лічильники з попередньою установкою, звичайно, є універсальними й можуть працювати в режимах додавання, віднімання, установки заданого коду, установки (скидання) нуля.

Дільники частоти.

Лічильники можуть виконувати функції дільників частоти, тобто пристроїв, що формують із імпульсної послідовності із частотою $f_{вх}$ імпульсну послідовність на виході останнього тригера із частотою $f_{вих}$ в K_p раз меншу вхідний. При такому використанні лічильників не потрібно знати, яке число в ньому записано в даний момент; тому дільники можуть не мати всіх проміжних виходів. Це значно спрощує їхню схему й конструкцію.

Завдання до лабораторної роботи

Варіант I.

1. На D-тригерах побудувати схему сумуючого лічильника з послідовним переносом з дійсним порядком рахунку 16.

2. На JK-тригерах побудувати схему віднімаючого лічильника з паралельним переносом з дійсним порядком рахунку 8.

3. На JK-тригерах побудувати схему реверсивного лічильника який змінює порядок рахунку під впливом керуючого сигналу з дійсним порядком рахунку 8.

4. На JK-тригерах побудувати схему лічильника з коефіцієнтом рахунку 10.

Варіант II.

1. На JK-тригерах побудувати схему віднімаючого лічильника з послідовним переносом з дійсним порядком рахунку 8.
2. На D-тригерах побудувати схему сумуючого лічильника з паралельним переносом з дійсним порядком рахунку 16.
3. На JK-тригерах побудувати схему реверсивного лічильника який змінює порядок рахунку під впливом керуючого сигналу з дійсним порядком рахунку 16.
4. На D-тригерах побудувати схему лічильника з коефіцієнтом рахунку 9.

Варіант III.

1. На D-тригерах побудувати схему сумуючого лічильника з послідовним переносом з дійсним порядком рахунку 8.
2. На D-тригерах побудувати схему віднімаючого лічильника з паралельним переносом з дійсним порядком рахунку 16.
3. На D-тригерах побудувати схему реверсивного лічильника який змінює порядок рахунку під впливом керуючого сигналу з дійсним порядком рахунку 16.
4. На JK -тригерах побудувати схему лічильника з коефіцієнтом рахунку 11.

Питання для самоконтролю

1. Які види лічильників ви знаєте?
2. Де використовуються лічильники?
3. Скільки JK- тригерів потрібно використати для побудови лічильника з дійсним порядком рахунку 16? Чому?
4. Скільки потрібно тригерів для побудови лічильника з коефіцієнтом рахунку рівним 10? Як працює такий лічильник?
5. Як працює подільник частоти? Для чого він призначений?

Список рекомендованої літератури

1. Дебновецкий С.В. “Основы автоматизированного проектирования электронных приборов.” Киев, Вища школа, 1987г.
2. Дмитрова М. Пунджев В. 33 схемы с логическими элементами И-НЕ. – М.: Радио и связь, 1987, - 110 с.
3. Карлашук В.И. Электронная лаборатория на IBM PC. –М.: Солон-Р, 1999.- 510 с.
4. Мальцева Л.А. Фромберг Э.М. Ямпольский В.С. Основы цифровой техники. – Радио и связь, 1986. – 125 с.
5. Микросхемы и их применение. Справочное пособие. // Батушев В.А., Вениаминов В. Н., Ковалев В.Г. и др. – М.: Радио и связь, 1984, - 271 с.
6. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П. Цифровые ЭВМ (Практикум). - К: Вища школа., 1989. - 420 с.
7. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П. Цифровые ЭВМ. - К: Вища школа., 1989. - 420 с.
8. Ямпольський В.С. Основы автоматики и электронно-вычислительной техники. – М., Просвещение, 1991. – 223с.
9. Алексеенко А.Г., Шагурин И.М. Микросхемотехника. М.: Радио и связь, 1982.
10. Вениаминов В.Н., Лебедев О.Н., Мирошниченко А.И. Микросхемы и их применение. М.: Радио и связь, 1989.
11. Савельев П.В., Коняхин В.В. Функционально-логическое проектирование БИС. М.: Высшая школа, 1990.
12. Угрюмов Е.П. Цифровая схематехника. СПб.: БХП Петербург, 2001.
13. Цифровая и вычислительная техника / Под ред. Э.В. Евреинова. М.: Радио и связь, 1991.
14. Электротехника и электроника в экспериментах и упражнениях: В 2 томах / Под ред. Д.Н. Панфилова. Т. 2. Электроника. М.: Додэна, 2000.
15. Янсен И. Курс цифровой электроники: В 4 томах. М.: Мир, 1987.