

Київський національний університет
імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

Петрушенко А.М.

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ З ДИСЦИПЛІНИ
“СТРУКТУРНА ТЕОРІЯ ЦИФРОВИХ АТОМАТІВ”**

КИЇВ – 2021

Петрушенко А.М. Методичні рекомендації до виконання лабораторних робіт з дисципліни “Структурна теорія цифрових автоматів” / А.М. Петрушенко. – Київ: електронна публікація на сайті факультету, 2021. – 85 с.

Рецензенти:

Провотар О.І., доктор фіз.-мат. наук, професор, завідувач кафедри інтелектуальних програмних систем Київського національного університету імені Тараса Шевченка

Ролік О.І., доктор техн. наук, професор, завідувач кафедри Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

*Рекомендовано до друку вченою радою
факультету комп’ютерних наук та кібернетики
12 квітня 2021 року, протокол № 13*

Автор:

Петрушенко А.М., кандидат фіз.-мат. наук, доцент кафедри інтелектуальних програмних систем факультету комп’ютерних наук та кібернетики

У методичних рекомендаціях до виконання лабораторних робіт з дисципліни “Структурна теорія цифрових автоматів” наведені теоретичні відомості та алгоритми синтезу керуючих автоматів та операційних вузлів операційних автоматів. Алгоритми синтезу як операційних автоматів так і операційних пристроїв загалом (як композиції зі зворотнім зв’язком керуючого та операційного автоматів) захищаються студентами в усній формі.

ЗМІСТ

Вступ.....	4
Лабораторна робота № 1.....	6
Лабораторна робота № 2.....	18
Лабораторна робота № 3.....	32
Лабораторна робота № 4.....	46
Лабораторна робота № 5.....	54
Лабораторна робота № 6.....	70
Лабораторна робота № 7.....	78
Список використаних джерел.....	85

ВСТУП

Навчальна дисципліна “Структурна теорія цифрових автоматів” є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти галузі знань 12 - “Інформаційні технології” зі спеціальності 121 - “Інженерія програмного забезпечення”, освітньо-професійної програми “Програмна інженерія”. Дисципліна є вибірковою навчальною дисципліною за названою освітньо-професійною програмою, викладається у 5 семестрі в обсязі – 90 год. (3 кредити ECTS), у тому числі: лекцій – 26 год., лабораторних робіт – 14 год., самостійної роботи – 50 год. Завершується дисципліна заліком.

Мета дисципліни – вивчення теоретичних основ і оволодіння навичками схемотехнічного проектування цифрових комп’ютерів (насамперед процесорів) заданої архітектури, що реалізують задану систему команд у заданому структурному базисі, з оптимізацією по швидкодії, продуктивності, затратам обладнання. Крім цього, метою дисципліни є також демонстрація подібності моделей і методів проектування комп’ютерів (hardware) і програм (software).

В результаті вивчення дисципліни студенти повинні знати:

- 1) тенденції розвитку науки і техніки в області комп’ютерної інженерії;
- 2) методи проектування типових операційних вузлів комп’ютерів: тригерів, регістрів, лічильників, суматорів і т.д.;
- 3) принципи побудови і методи проектування блоків керування;
- 4) принципи побудови і методи проектування операційних автоматів;
- 5) принципи побудови і методи проектування операційних пристроїв як композиції із зворотнім зв’язком керуючих і операційних автоматів;
- 6) математичні методи оптимізації засобів обчислювальної техніки по найбільш важливим структурно-функціональним і техніко-економічним характеристикам.
- 7) сучасні методи автоматизації проектування в області комп’ютерної інженерії.

В результаті вивчення дисципліни студенти повинні вміти:

- 1) працювати з технічною літературою;
- 2) ставити завдання, давати порівняльну характеристику різноманітним варіантам рішень на етапах проектування;
- 3) використовувати сучасний математичний апарат для вирішення інженерних і наукових задач, що виникають при розробці засобів обчислювальної техніки;
- 4) розробляти алгоритми функціонування різноманітних пристроїв комп'ютерів;
- 5) досліджувати і розробляти типові вузли й пристрої комп'ютерів;
- 6) досліджувати якість вузлів і пристроїв комп'ютерів;
- 7) використовувати сучасні методи автоматизації проектування засобів обчислювальної техніки.

Дисципліна “Структурна теорія цифрових автоматів” є базовою для засвоєння дисциплін “Системне програмування та операційні системи”, “Комп'ютерні мережі”, “Трансформаційні методи синтезу обчислювальних систем”, а також деяких інших дисциплін спеціалізації та вільного вибору студента циклу професійної підготовки фахівців освітньо-кваліфікаційних рівнів “бакалавр” та “магістр”, що передбачають (дисципліни) знання апаратної частини комп'ютера.

Звіт з виконаних лабораторних робіт складається з текстової частини, в яку включаються формальні викладки вирішення задач, і електронної частини, в яку включаються відлагоджені в системі Electronics Workbench схеми. Номер варіанту контрольного завдання визначається порядковим номером студента в журналі групи.

ПРОЕКТУВАННЯ І ДОСЛІДЖЕННЯ ТРИГЕРІВ

Мета роботи: вивчення функціонування і схемних різновидів тригерів, оволодіння методами їхнього проектування, придбання навичок у налагодженні і визначенні характеристик тригерів.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

У схемах цифрової обчислювальної техніки в якості запам'ятовуючих елементів широко використовуються тригери. *Тригер* – це пристрій із двома

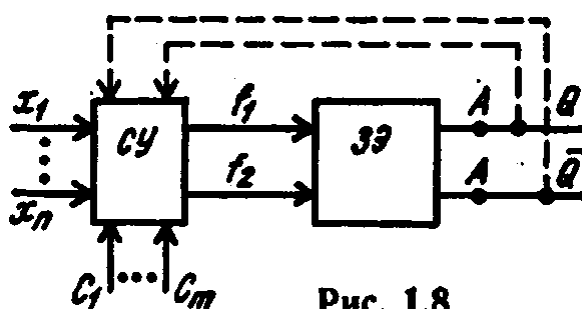


Рис. 1.8

стійкими станами, що містить запам'ятовуючий елемент (ЗЭ) і схему керування (СК), як показано на рис. 1.8, де x_1, \dots, x_n - інформаційні входи тригера; C_1, \dots, C_m - тактуючі входи; Q і \bar{Q} - відповідно прямий і інверсний виходи тригера; f_1 і f_2 - функції збудження ЗЕ.

Тригери класифікуються по ряду ознак. У залежності від наявності тактуючих входів розрізняють *синхронні* (є тактуючі входи) і *асинхронні* (немає тактуючих входів) тригери. Найбільше поширення серед синхронних одержали однотактні тригери, що мають тільки один тактуючий вхід.

Функціональна класифікація характеризує стан входів і виходів тригера в момент часу до його переключення (S) і після його переключення ($S + 1$). При наявності n інформаційних входів можна одержати 5^{2^n} типів тригерів. На практиці ж використовують невелике число типів тригерів. До них можна віднести *RS-*, *JK-*, *T-* і *D-тригери*. *D-тригер* знаходить застосування тільки як синхронний. Інші з зазначених тригерів можуть бути як синхронними, так і асинхронними.

Спосіб функціонування тригерів може бути описаний таблицею переходів. З таблиці переходів асинхронного *RS-тригера* (табл. 1) випливає, що тригер не змінює свого стану в момент часу $S + 1$, тобто $Q^{S+1} = Q^S$, якщо в момент S мало місце $R^S = S^S = 0$. При $S^S = 1$ і $R^S = 0$ тригер встановлюється в одиничний стан ($Q^{S+1} = 1$), а при комбінації сигналів $S^S = 0$ і $R^S = 1$ - у нульовий ($Q^{S+1} = 0$). При $R^S = S^S = 1$ стан тригера в момент часу $S + 1$ не визначений ($Q^{S+1} = *$). Така комбінація сигналів для *RS-тригера* є забороненою.

Існують різновидності *RS-тригерів*, для яких комбінація сигналів $R^S = S^S = 1$ не є забороненою. *R-тригер* відрізняється від *RS-тригера* тим, що при комбінації вхідних сигналів $R^S = S^S = 1$ він переходить у нульовий стан ($Q^{S+1} = 0$) (табл. 2). *S-тригер* (табл. 3) у цьому випадку переходить в одиничний стан ($Q^{S+1} = 1$), а *E-тригер* (табл. 4) не змінює свого стану ($Q^{S+1} = Q^S$).

Синхронний *D-тригер* називають тригером затримки (табл. 5). Він здійснює затримку сигналу на один такт (період проходження синхросигналів). Як асинхронний *D-тригер* не знаходить застосування через відсутність режиму збереження інформації ($Q^{S+1} = Q^S$).

Тригери можуть мати вхід, що дозволяє, V . Наприклад, *D-тригер* із входом, що дозволяє, називають *DV-тригером*. При $V = 1$ він працює як *D-тригер*, а при $V = 0$ зберігає свій стан. *DV-тригер* може бути синхронним і асинхронним.

T-тригер називають також *рахунковим тригером*. Він здійснює підрахунок одиниць, що надходять на вхід T , по модулю два, що видно з табл. 6. Як випливає з таблиці переходів *JK-тригера* (табл. 7), при комбінаціях вхідних сигналів $J = K = 0$, $J = 0$ і $K = 1$, $J = 1$ і $K = 0$ він працює, як *RS-тригер* (вхід J відповідає входіві S , а $K - R$), а при $J = K = 1$ змінює свій стан на протилежний, тобто працює як рахунковий тригер.

Таблиця 1

R^S	S^S	Q^{S+1}
0	0	Q^S
0	1	1
1	0	0
1	1	*

Таблиця 2

R^S	S^S	Q^{S+1}
0	0	Q^S
0	1	1
1	0	0
1	1	0

Таблиця 3

R^S	S^S	Q^{S+1}
0	0	Q^S
0	1	1
1	0	0
1	1	1

Таблиця 4

R^S	S^S	Q^{S+1}
0	0	Q^S
0	1	1
1	0	0
1	1	Q^S

Таблиця 5

C^S	D^S	Q^{S+1}
0	0	Q^S
0	1	Q^S
1	0	0
1	1	1

Таблиця 6

T^S	Q^{S+1}
0	Q^S
1	$\overline{Q^S}$

Таблиця 7

J^S	K^S	Q^{S+1}
0	0	Q^S
0	1	0
1	0	1
1	1	$\overline{Q^S}$

Класифікація тригерів по способу запису інформації характеризує процес переключення тригера. Розрізняють наступні типи тригерів:

1. тригери, керовані рівнем сигналів (без внутрішньої затримки);
2. тригери, керовані перепадом сигналів (із внутрішньою затримкою).

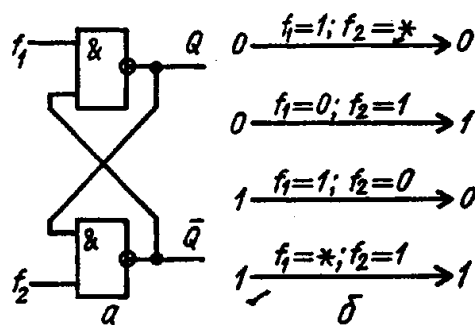


Рис. 1.9

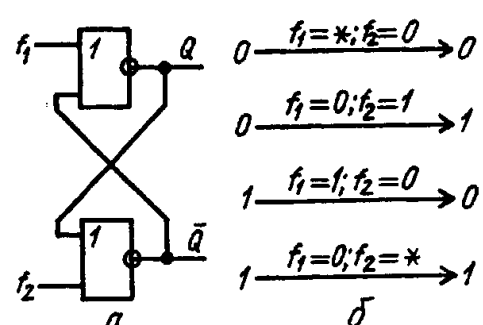


Рис. 1.10

Активним назвемо такий вектор (набір) вхідних сигналів, що може викликати зміна стану тригера. У цьому випадку мають місце співвідношення $Q^{S+1} = 0$, $Q^{S+1} = 1$ і $Q^{S+1} = \overline{Q^S}$. Вектор вхідних сигналів, що відповідає режимові збереження інформації ($Q^{S+1} = Q^S$), будемо вважати пасивним. Наприклад, для

тригера, що працює відповідно до табл. 5, вектор CD є активним при значеннях 10 і 11, а при 00 і 01 - пасивним. Для T -тригера (табл. 6) значення 1 відповідає активному сигналові T , а 0 - пасивному. Для правильної роботи тригерів необхідно, як правило, чергувати активні набори сигналів з пасивними.

Тригери, керовані рівнем сигналів, змінюють свій стан безпосередньо після надходження на їхні входи активного вектора вхідних сигналів. У тригерах, керованих перепадом, сигнали на виході змінюються тільки після зняття активного вектора вхідних сигналів.

У синхронних тригерах при зміні вектора вхідних сигналів необхідно забезпечити зміну тільки тактуючого сигналу. Якщо тригер змінює свій стан при переході тактуючого сигналу з 0 у 1, то говорять, що тригер спрацьовує по позитивному перепаді, а при переході з 1 у 0 - по негативному перепаді тактуючого сигналу.

Характерна риса синхронних тригерів, керованих рівнем сигналу C , полягає в тому що при сталому активному рівні сигналу C вони можуть переключатися стільки разів, скільки разів змінюються інформаційні сигнали. У тригерах, керованих перепадом, у цьому випадку вихідні сигнали Q і \bar{Q} змінюються тільки після зняття активного рівня тактуючого сигналу. У деяких реальних схемах тригерів із внутрішньою затримкою зміна інформаційних сигналів при сталому рівні тактуючого сигналу може спричинити переключення ряду логічних елементів, однак на виходах тригера сигнали при цьому не змінюють свого значення.

Проектування (синтез) тригерних пристроїв полягає у виборі ZE і синтезі SK в заданому елементному базисі.

У тригерах, керованих рівнем сигналу, використовуються елементарні запам'ятовуючі елементи на вентилях І-НІ (рис. 1.9, а),

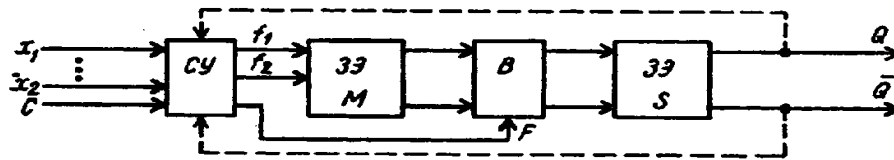


Рис. 1.11

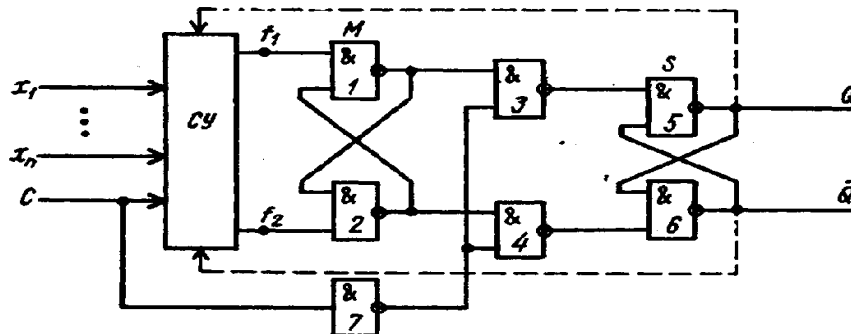


Рис. 1.12

а також АБО-НІ (рис. 1.10,а). Спосіб їхнього переключення описується системами підграфів переходів на рис. 1.9, б і рис. 1.10, б.

Якщо в стовпці Q^{S+1} таблиці переходів проєктованого тригера мається значення \overline{Q}^s (табл. 6 і 7), то Q є аргументом функцій f_1 і f_2 . Для забезпечення правильного переключення тригера в точках A (рис. 5) у цьому випадку необхідно включити елементи затримки. Аналогічна ситуація виникає в тому випадку, коли аргументами функцій f_1 і f_2 є сигнали Q інших тригерів, що переключаються в процесі роботи одночасно з даним тригером. Наприклад, такими «залежними» є тригери в регістрах, що зсувають, лічильниках і т.д.

Однак при побудові тригерів на потенційних елементах не можуть бути використані елементи затримки, що містять реактивні компоненти, тому що це накладало б обмеження на тривалості входних сигналів тригера. У потенційній системі елементів повинна бути забезпечена правильна робота тригера при будь-якій тривалості входних сигналів, якщо вона перевищує деяке мінімально припустиме значення. У цьому випадку на елементах І-НІ і АБО-НІ тригери будують по *MS-схемі* або за *схемою трьох елементарних тригерів*.

Перший спосіб полягає у використанні двох елементарних ЗЭ: основного М-тригера і допоміжного S-тригера. Структура одноктактного MS-тригера пояснюється рис. 1.11. Запис у *М-тригер* тактується сигналом C , а в S-тригер - сигналом F . Передача інформації з М-тригера в S-тригер здійснюється через

вентилі *B*. Найбільше поширення одержали MS-тригери з інвертором у ланцюзі *C* і MS-тригери зі зв'язками, що забороняють. Схеми зазначених тригерів на елементах І-НІ приведені відповідно на рис.1.12 і 1.13, де елементи 1, 2 утворюють *M-тригер*, а елементи 5, 6 - *S-тригер*. Вентилі 3, 4 у схемі на рис. 1.12 керують сигнали з виходу інвертора 7, а в схемі на рис. 1.13 - сигнали f_1 і f_2 . За схемою на рис. 1.13 можуть бути побудовані як синхронні, так і асинхронні тригери. У схемах на рис. 1.12 і 1.13 елементи І-НІ можуть бути замінені на елементи АБО-НІ.

Тригери, виконані по MS-схемі, називають тригерами з потенційним

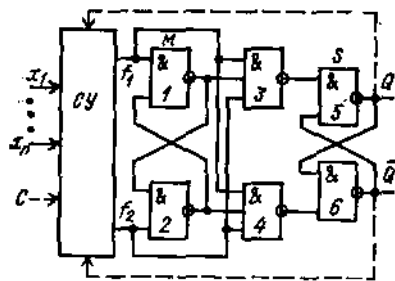


Рис. 1.13

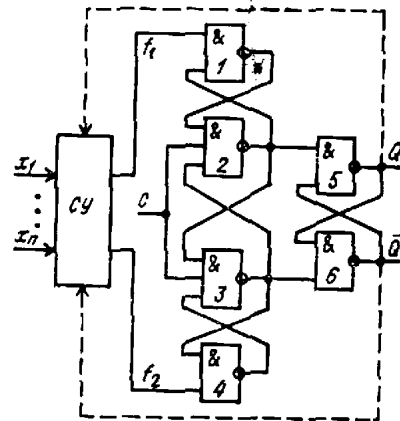


Рис. 1.14

записом інформації. Запис інформації в *M-тригері* здійснюється так само, як і в елементарні ЗЕ (рис. 6 і 7). Перепис інформації з *M-тригерів* у *S-тригери* здійснюється при знятті активного вектора вхідних сигналів тригера.

Тригер із внутрішньою затримкою, на елементах І-НІ, виконаний за схемою трьох елементарних тригерів, зображений на рис. 1.14. У даному тригері сигнали, що відповідають новому стану, установлюються при переході тактуючого сигналу з 0 у 1. Спосіб запису інформації в тригерах такого типу називають динамічним. При $C = 0$ на виходах вентилів 2 і 3 є присутнім одиничний сигнал, тобто тригер на вентиліях 5 і 6 не змінює свого стану, а вентилі 1 і 4 виконують функцію інверторів. Переключення тригера здійснюється відповідно до рис. 1.9, б.

Наприклад, якщо $f_1 = 0$ і $f_2 = 1$, то при переході сигналу C з 0 у 1 на виході вентиліа 2 установлюється нульовий сигнал, що переключає в одиничний стан

тригер на вентилях 5, 6 і підтверджує одиничний сигнал на виході вентиля 1. Після цього сигнали f_1 і f_2 можуть змінюватися що не вплине на стан основного тригера (вентилі 5 і 6) доти, поки не здійсниться черговий перехід сигналу C з 0 у 1. Розглянутий тригер можна побудувати на вентилях АБО-НІ. У цьому випадку він змінює свої стани відповідно до рис. 1.10, б.

Для побудови ЗЕ крім вентилів І-НІ і АБО-НІ використовують і інші вентиля. Наприклад, синхронний тригер із внутрішньою затримкою і динамічним записом інформації можна побудувати за схемою на рис. 1.15, де запам'ятовуючий елемент виконаний на вентилях І-АБО-НІ. Такий тригер переключується по негативному перепаді синхросигнала C в відповідності з рис. 1.9, б. Для правильної роботи тригера необхідно, щоб функції f_1 і f_2 зберігали своє значення після негативного перепаду сигналу C не менш $2t$ (t - час затримки сигналів на вентилі І-АБО-НІ). Якщо ця умова при синтезі СК автоматично не виконується, то в схему вводяться додаткові елементи (інвертори, повторювачі і т.д.) для забезпечення необхідної затримки.

При синтезі СК для будь-якого типу тригера спочатку на підставі скороченої таблиці переходів будується повна таблиця переходів тригера, у якій відбивають значення Q^s для кожного вектора вхідних сигналів. Потім одержують вирази для f_1 і f_2 і реалізують ці функції в заданому елементному базисі. В якості прикладу розглянемо процес проектування *JK-тригера* на елементах І-НІ.

Оскільки в табл. 7 присутнє значення \overline{Q}^s , то тригер повинен мати внутрішню затримку. Вибираємо структуру тригера, представлену на рис. 1.12, як одну з можливих. Порядок переходів тригера M на вентилях І-НІ визначається системою підграфів на рис. 1.9, б.

Для розглянутого тригера одержуємо повну таблицю переходів (табл. 8), побудовану відповідно до табл. 7.

Таблиця 8.

C^S	J^S	K^S	Q^S	Q^{S+1}	f_1	f_2
0	0	0	0	0	1	*
0	0	0	1	1	*	1
0	0	1	0	0	1	*
0	0	1	1	1	*	1
0	1	0	0	0	1	*
0	1	0	1	1	*	1
0	1	1	0	0	1	*
0	1	1	1	1	*	1
1	0	0	0	0	1	*
1	0	0	1	1	*	1
1	0	1	0	0	1	*
1	0	1	1	0	1	0
1	1	0	0	1	0	1
1	1	0	1	1	*	1
1	1	1	0	1	0	1
1	1	1	1	0	1	0

На підставі визначеного вище порядку переходів М-тригера заповнюємо в табл. 8 графи для f_1 і f_2 , аналізуючи переходи $Q^S \rightarrow Q^{S+1}$ у кожному рядку таблиці.

За допомогою діаграм Вейча (рис. 1.16) знаходимо мінімальну диз'юнктивну нормальну форму (МДНФ) функцій f_1 і f_2 , (індекси S при цьому опускаємо):

$$f_1 = \bar{C} \vee Q \vee \bar{J};$$

$$f_2 = \bar{C} \vee \bar{Q} \vee \bar{K}.$$

Приводимо отримані функції до виду, зручному для реалізації на елементах І-НІ:

$$f_1 = \overline{\overline{C \vee Q \vee J}} = \overline{C \overline{Q} J};$$

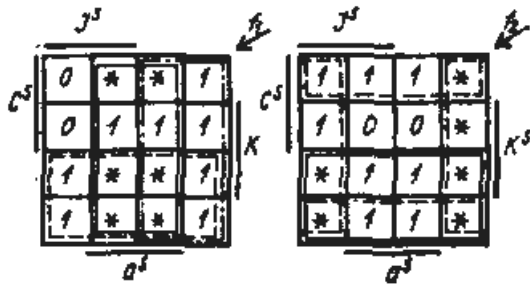


Рис. 1.16

$$f_2 = \overline{\overline{C \vee Q \vee K}} = \overline{C \overline{Q} K}.$$

Отримана схема тригера приведена на рис. 1.17.

Синтез здійснюється аналогічно, якщо в якості вихідної прийняти схему, приведену на рис. 1.15. Тригер у цьому випадку буде мати вигляд, показаний на рис. 1.18. Для правильної роботи схеми необхідно, щоб затримка сигналів на вентилях І-НІ була, принаймні, удвічі більше затримки сигналів на вентилі І-АБО-НІ. Якщо ця умова не виконується, то в крапки *A* необхідно включити додаткові елементи затримки (наприклад, повторювачі або інвертори).

Якщо в якості базової вибирається схема трьох тригерів (рис. 1.14), то для синтезу СК варто скористатися табл. 9, де відсутнє значення *C*, тому що відповідний сигнал на СК в тригерах такого типу не надходить. З табл. 9 отримуємо $f_1 = \overline{J \overline{Q}}$ і $f_2 = \overline{K \overline{Q}}$. Схема тригера показана на рис. 1.19.

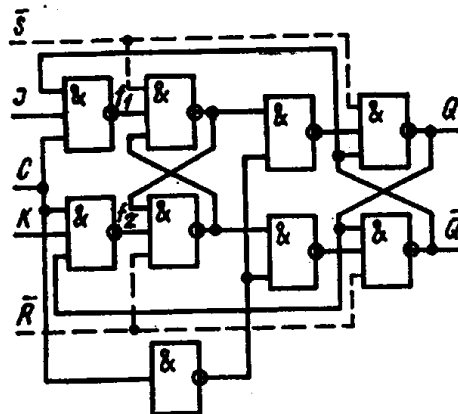


Рис. 1.17

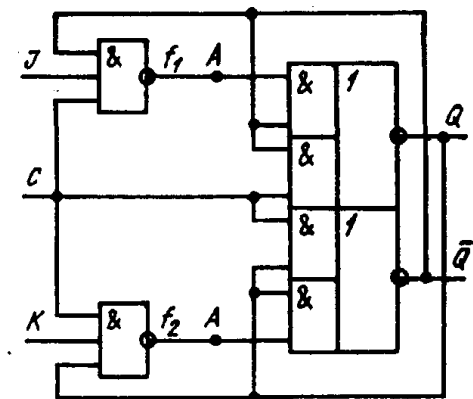


Рис. 1.18

Таблиця 9

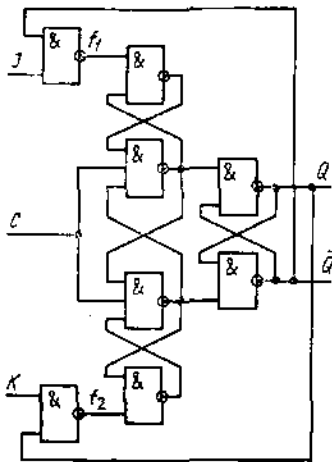


Рис. 1.19

J^S	K^S	Q^S	Q^{S+1}	f_1	f_2
0	0	0	0	1	*
0	0	1	1	*	1
0	1	0	0	1	*
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	1	*	1
1	1	0	1	0	1
1	1	1	0	1	0

Синхронні тригери можуть мати асинхронні входи попередньої установки тригера в 0 (вхід R) і в 1 (вхід S). Сигнали, що надходять на ці входи, незалежно від стану інших входів тригера переключають тригер у новий стан, тобто мають пріоритет стосовно інших сигналів. Як приклад організація асинхронних входів показана пунктиром на рис. 1.17.

ПІДГОТОВКА ДО РОБОТИ

1. Побудувати і замалювати в протоколі синхронні тригери, керовані рівнем тактуючого сигналу, на елементах І-НІ відповідно до табл. 1.4 і 1.5, а також на елементах АБО-НІ по табл. 1.6 і 1.7. Для кожної з чотирьох отриманих схем за допомогою тимчасових діаграм визначити час переключення тригера, мінімальний тривалість і максимальну частоту тактуючих сигналів. Затримку сигналів одним елементом вважати рівної t .

2. Побудувати синхронний D-тригер із внутрішньою затримкою па

елементах АБО-НІ за схемою трьох тригерів, асинхронний T -тригер за схемою на рис. 1.13 і синхронний Γ -тригер за схемою на рис. 1.15. Для побудови СУ T -тригерів використовувати елементи І-НІ. Для кожної схеми визначити час переключення тригера, мінімальну тривалість і максимальну частоту проходження вхідних сигналів. Пунктирними лініями показати, як організуються асинхронні настановні входи тригерів.

3. Відповідно до таблиці варіантів (табл. 1.13) побудувати синхронні тригери з використанням для СУ елементів І-НІ, а також АБО-НІ. Обґрунтувати обрану структуру тригерів і визначити з використанням тимчасових діаграм їхні тимчасові параметри.

Таблиця 1.13

x_1^S	x_2^S	Q^{S+1}															
		варіант															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	0	\bar{Q}^S	Q^S	1	1	\bar{Q}^S	Q^S	1	1	\bar{Q}^S	1	0	\bar{Q}^S	1	0	0
0	1	1	0	\bar{Q}^S	Q^S	0	1	\bar{Q}^S	Q^S	1	Q^S	\bar{Q}^S	0	0	\bar{Q}^S	1	0
1	0	Q^S	1	0	\bar{Q}^S	Q^S	0	1	\bar{Q}^S	\bar{Q}^S	0	Q^S	1	0	0	\bar{Q}^S	\bar{Q}^S
1	1	\bar{Q}^S	Q^S	1	0	\bar{Q}^S	Q^S	0	1	Q^S	1	0	\bar{Q}^S	1	0	0	1

x_1^S	x_2^S	Q^{S+1}															
		варіант															
		17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
0	0	1	\bar{Q}^S	0	1	\bar{Q}^S	0	1	1	\bar{Q}^S	*	0	1	*	0	\bar{Q}^S	Q^S
0	1	0	1	\bar{Q}^S	1	1	\bar{Q}^S	0	0	1	\bar{Q}^S	*	*	1	\bar{Q}^S	0	\bar{Q}^S
1	0	0	0	1	0	1	1	\bar{Q}^S	*	0	1	\bar{Q}^S	0	\bar{Q}^S	1	*	*
1	1	\bar{Q}^S	0	0	\bar{Q}^S	0	1	1	\bar{Q}^S	*	0	1	\bar{Q}^S	0	*	1	1

ПОРЯДОК ВИКОНАННЯ РОБОТИ

Зібрати декілька (за завданням викладача) схеми тригерів, налагодити їх і дослідити.

ЗМІСТ ЗВІТУ

Звіт повинен містити короткі теоретичні зведення, необхідні для виконання лабораторної роботи і відповіді на контрольні питання, усі таблиці, схеми і діаграми, отримані при виконанні завдання, а також висновки по роботі.

КОНТРОЛЬНІ ПИТАННЯ

1. Складіть таблиці переходів RS -, R -, S -, E -, D -, T -; DV - і JK -тригерів.
2. У чому розходження між синхронними й асинхронними тригерами?
3. У чому розходження між тригерами, керованими рівнем тактуючого сигналу, і тригерами з внутрішньою затримкою? У яких випадках використовуються зазначені типи тригерів?
4. Пояснити роботу синхронних тригерів, виконаних по MS -схемі, за схемою трьох тригерів із ЗЭ на вентилях І-АБО-НІ.
5. Укажіть розходження між синхронними й асинхронними входами синхронного тригера.
6. Поясните, як будується часова діаграма роботи тригера.
7. Охарактеризуйте етапи проектування тригерних схем. Побудуйте тригери по заданій таблиці переходів.
8. Назвіть основні часові характеристики тригерів.
9. Порівняйте по швидкодії тригери з внутрішньою затримкою, виконані по різних схемах. Які часові умови є основними для правильної роботи цих тригерів?
10. Як перейти до базису І-НІ і АБО-НІ, якщо функція представлена в МДНФ?
11. Як побудувати T -тригер на основі RS -, D - і JK -тригерів?
12. Які зміни необхідно здійснити в схемі тригера з внутрішньою затримкою, щоб змінити фронт тактуючого сигналу, по якому здійснюється переключення тригера?
13. Чи можна в цифровому пристрої замінити синхронний JK -тригер на синхронний JK -тригер (E -, R -, S -тригер), не порушуючи правильності роботи пристрою?

ПРОЕКТУВАННЯ І ДОСЛІДЖЕННЯ РЕГІСТРІВ

Мета роботи: вивчення схемних різновидів регістрів, мікрооперацій, що виконуються на них, і оволодіння методами проектування регістрів.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Регістром називається упорядкована послідовність тригерів, призначена для збереження слів інформації і виконання мікрооперацій над ними.

Мікрооперація - це елементарна машинна дія, у результаті якої змінюється значення слова чи здійснюється його пересилання.

Узагальнена структура регістра представлена на рис. 1, де КС - комбінаційна схема; Q_i - виходи i -х тригерів ($i = 1, \dots, n$); f_i - функції збудження тригерів (наприклад, J_i, K_i, R_i, S_i і т.д.); С і Т - тактуючі входи, відповідно, тригерів і регістра (у конкретних схемах можуть бути відсутніми). Число розрядів (тригерів) називають довжиною регістра. Регістр може знаходитися в 2^n станах, тобто в нього можна записати 2^n різних слів.

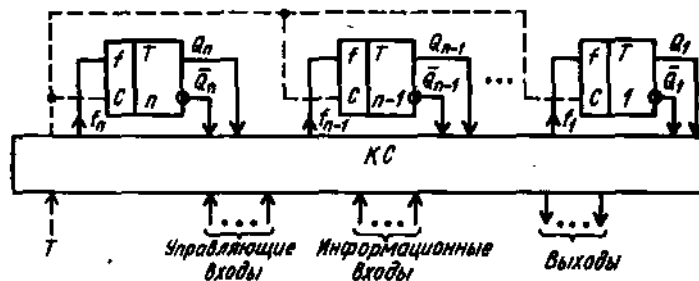


Рис.1 Узагальнена структура регістра.

Розрізняють *синхронні й асинхронні регістри*. У синхронних регістрах мікрооперації виконуються по тактуючому сигналу Т. Настроювання регістра на відповідну мікрооперацію здійснюється попередньою установкою на керуючих входах коду мікрооперації $S_1 \dots S_m$. Число розрядів у коді визначається зі співвідношення $m \geq \lceil \log_2 k \rceil$, де k - кількість мікрооперацій.

В асинхронних регістрах тактуючий вхід Т відсутній. Кожна i -а мікрооперація виконується під дією власного керуючого сигналу u_i . Кількість таких сигналів дорівнює числу мікрооперацій k , тобто, в загальному випадку,

більше, ніж розрядів у кодї мікрооперації $S_1 \dots S_m$ синхронного регістра. При виконанні мікрооперацій у кожному розрядї регістра здійснюється однакове перетворення інформації. Характер мікрооперації можна описати через стани сигналів у момент часу, що передує виконанню мікрооперації (S), і в момент завершення мікрооперації ($S+1$).

Найбільше часто на регістрах виконують мікрооперації занесення (прийому, запису) слова паралельним кодом, зсуву слова, а також установки вхідного (як правило, нульового) стану.

Занесення слова здійснюється через інформаційні входи D_i ($i = 1, \dots, n$). Для i -го розряду регістра можна записати $Q_i^{S+1} = D_i^S$, тобто при виконанні цієї мікрооперації в i -й розряд регістра записується значення сигналу на вході D_i .

Регістри, на яких виконуються мікрооперації зсуву, називаються зсуваючими. Зсув слова може бути здійснений уліво (у бік старших розрядів) чи вправо (у бік молодших розрядів) на i розрядів одночасно, де $i = 1, \dots, n-1$. Регістри, що мають ланцюги як лівого, так і правого зсуву, називаються реверсивними. Зсув слова вліво і вправо, наприклад, на один розряд можна описати відповідно як $Q_i^{S+1} = Q_{i-1}^S$ і $Q_i^{S+1} = Q_{i+1}^S$.

Установка нульового стану регістра ($Q_i^{S+1} = 0$) здійснюється, як правило, через асинхронні настановні входи тригерів.

За допомогою регістрів можна виконувати й інші перетворення інформації, наприклад, порозрядні логічні операції: кон'юнкція ($Q_i^{S+1} = Q_i^S D_i^S$); диз'юнкція ($Q_i^{S+1} = Q_i^S \vee D_i^S$); нерівнозначність ($Q_i^{S+1} = Q_i^S \oplus D_i^S$); інвертування розрядів і т.д.

Виходами регістра бувають безпосередньо виходи тригерів, але в ряді випадків КС (рис. 1) включає елементи, що здійснюють видачу інформації. Видача слова може бути здійснена в прямому, оберненому чи парафазному кодї. Крім того, регістр може мати виходи з трьома станами (стан логічного нуля, логічної одиниці і високоомний стан). Для видачі слова в прямому кодї до виходів регістра підключають прямі виходи тригерів Q_i , у оберненому кодї - інверсні виходи $\overline{Q_i}$, а в парафазному кодї - прямі й інверсні.

Як приклад на рис. 2, а показана організація ланцюгів видачі прямого і зворотного коду на елементах І-АБО-НІ. Для видачі прямого коду подається керуючий сигнал ψ , а для видачі зворотного - $\bar{\psi}$. Якщо $\psi = \bar{\psi} = 0$, то на усіх виходах регістра будуть присутні одиниці. Ланцюга видачі прямого і зворотного коду можна реалізувати і на елементах ВИКЛЮЧАЮЧЕ АБО (рис. 2, б). У цьому випадку при $\psi = 0$ з регістра видаються значення Q_i , при $\psi = 1$ - значення \bar{Q}_i , а заборона видачі інформації не забезпечується. Проектування регістрів зводиться до вибору типу тригерів і синтезу КС.

Для побудови регістрів у залежності від виконуваних мікрооперацій можуть бути використані тригери, різні по функціональній ознаці (наприклад, RS-, JK-, T-, D-типу) і по організації (синхронні й асинхронні, із внутрішньою затримкою і без внутрішньої затримки).

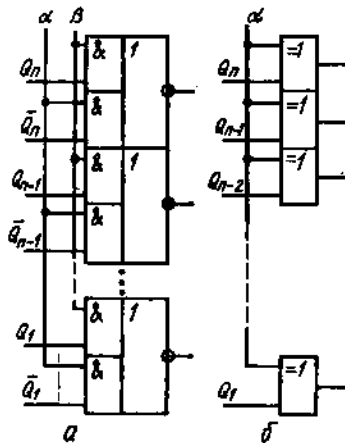
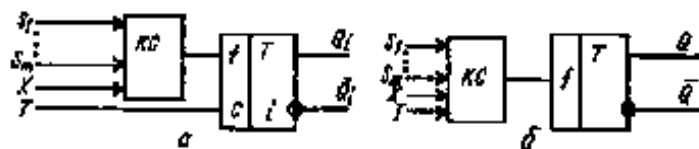


Рис.2

Звичайно в регістрах усі розряди виконуються однаково. Розряд синхронного регістра (без обліку ланцюгів видачі інформації) може бути побудований як на синхронному (рис. 3, а), так і на асинхронному тригері

Рис. 3



(рис.3, б). Аргументами функцій збудження f є розряди S_i , код мікрооперації і множина сигналів X , що містить значення D_i, Q_i, Q_j ($j \neq i$) і т.д. і

визначається системою мікрооперацій. Якщо використовується асинхронний тригер, то тактуючий сигнал Т подається на вхід КС.

Асинхронні регістри також можуть бути реалізовані з використанням і синхронних і асинхронних тригерів. Синхронні тригери найбільш зручні, коли на регістрі виконується одна мікрооперація (наприклад, мікрооперація зсуву). При цьому (рис. 4, а) сигнал мікрооперації Y підключається безпосередньо до входу C тригера. При використанні асинхронних тригерів розряд регістра будується відповідно до рис. 4, б. У цьому випадку сигнали Y_i надходять на входи КС.

Синтез функцій зсуву тригерів виконують у наступній послідовності:

складають таблицю переходів i -го розряду регістра, у якій відображають стан тригера Q_i і значення сигналів з множини X^S (рис. 3 і 4) у момент часу S , а для моменту часу $S+1$ – новий стан тригера Q_i^{S+1} ;

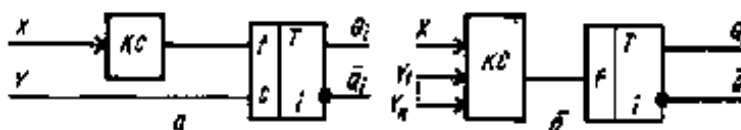


Рис. 4

відповідно до системи підграфів переходів використовуваного тригера для кожного рядка отриманої таблиці (для кожного переходу $Q_i^S \rightarrow Q_i^{S+1}$) записують необхідні значення функцій збудження тригера;

виконують синтез КС у заданому елементному базисі.

Якщо аргументами функцій зсувів тригерів є значення Q_i , то тригери повинні мати внутрішню затримку.

Складність і швидкодія КС при заданому наборі мікрооперацій визначається функціональним типом тригерів. Якщо для побудови регістра можна використовувати тригери різного типу, доцільно одержати кілька варіантів схем, а потім вибрати з них ту, котра має необхідні параметри.

Системи підграфів переходів для JK-, RS-, T- і D-тригерів представлені на

рис. 5, де знаком * відмічені довільні значення функцій збудження.

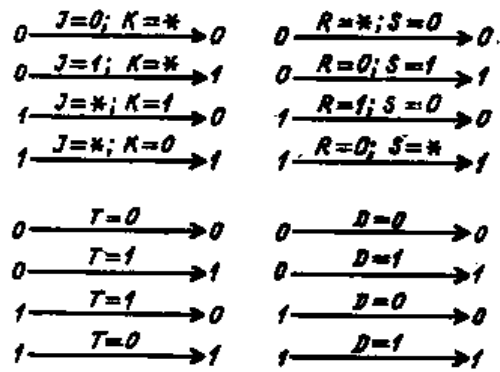


Рис. 5

Нехай необхідно побудувати синхронний регістр для виконання мікрооперацій паралельного занесення і зсуву слів вправо на один розряд з використанням елементів І-НІ, а також синхронних RS-, JK-, T-, чи D-тригерів.

Будемо вважати, що значення 00, 01 і 10 коду мікрооперації S_2S_1 відповідають режиму збереження інформації, паралельного занесення і зсуву. Значення слова $S_2S_1 = 11$ є забороненим.

У таблиці переходів і-го розряду регістра (табл. 1) значення Q_i^{S+1} при $S_2S_1 = 00$ повторюють значення Q_i^S , що відповідає режиму збереження інформації. Для $S_2S_1 = 01$ справедлива рівність $Q_i^{S+1} = D_i$, що відповідає мікрооперації занесення, а при $S_2S_1 = 10$ використовується співвідношення $Q_i^{S+1} = Q_{i+1}^S$, що характеризує зсув інформації в регістрі на один розряд. Значення функцій збудження тригерів $T_i, J_i, K_i, R_i, S_i, i D_i'$ (на відміну від інформаційного входу D_i регістра, функція збудження D-тригера відмічена штрихом) для кожного переходу $Q_i^S \rightarrow Q_i^{S+1}$ занесені в таблицю відповідно до рис. 5. На підставі табл. 1 одержуємо діаграми Вейча для функцій збудження тригерів (рис. 6) і знаходимо операторні представлення функцій у формі І-НІ/І-НІ (опускаючи індекси S):

$$\begin{aligned}
 D_i' &= S_1 D_i \vee S_2 Q_{i+1} \vee \overline{S_2} \overline{S_1} Q_i = \overline{\overline{S_1} D_i \vee \overline{S_2} Q_{i+1} \vee \overline{S_2} \overline{S_1} Q_i}; \\
 R_i &= K_i = S_1 \overline{D_i} \vee S_2 \overline{Q_{i+1}} = \overline{\overline{S_1} \overline{D_i} \vee \overline{S_2} \overline{Q_{i+1}}}; \\
 S_i &= J_i = S_1 D_i \vee S_2 Q_{i+1} = \overline{\overline{S_1} \overline{D_i} \vee \overline{S_2} \overline{Q_{i+1}}}; \\
 T_i &= S_2 \overline{Q_{i+1}} Q_i \vee S_2 Q_{i+1} \overline{Q_i} \vee S_1 D_i \overline{Q_i} \vee S_1 \overline{D_i} Q_i = \\
 &= \overline{\overline{S_2 \overline{Q_{i+1}} Q_i \vee S_2 Q_{i+1} \overline{Q_i} \vee S_1 D_i \overline{Q_i} \vee S_1 \overline{D_i} Q_i}}.
 \end{aligned}$$

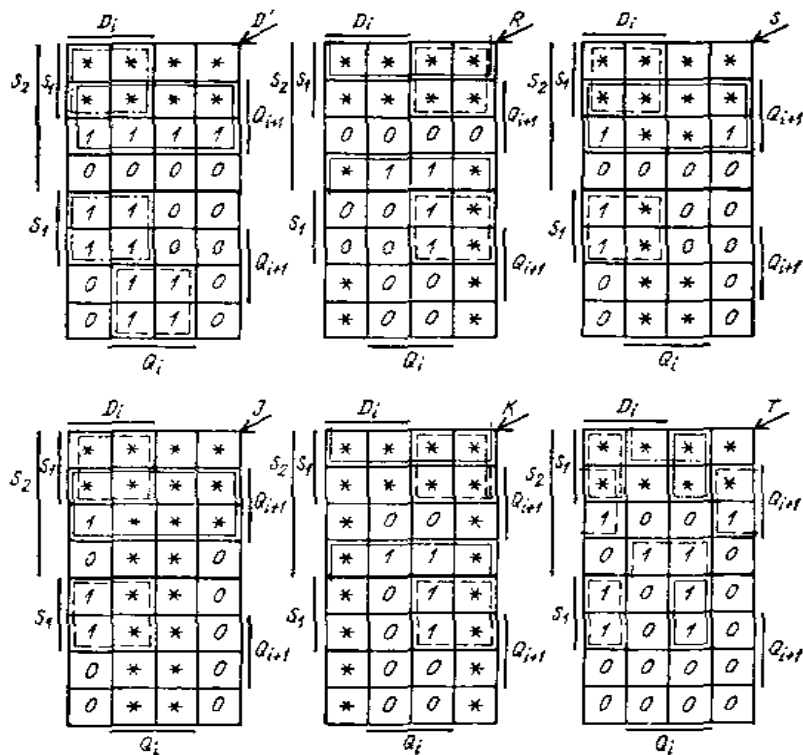


Рис. 6.

Очевидно, що найбільш складною є КС Т-тригерів. Для JK- і RS-тригерів складність КС однакова. Найбільш простий є КС при використанні D-тригерів. В усіх випадках аргументами функцій збудження є Q_i . Отже, для побудови регістра необхідно використовувати тригери, керовані перепадом тактуючого сигналу (із внутрішньою затримкою).

Схема 4-розрядного регістра на D-тригерах представлена на рис. 7, де DR - вхід занесення інформації послідовним кодом при виконанні мікрооперації зсуву.

Аналогічно виконується синтез асинхронних регістрів. Нехай необхідно побудувати асинхронний регістр, використовуючи асинхронні RS- чи Т-тригери, двохвходові елементи І і АБО, а також інвертори. Регістр повинен забезпечувати виконання мікрооперацій y_1, y_2, y_3 де y_1 - занесення інформації ($Q_i^{S+1}=D_i$); y_2 - інвертування розрядів регістра ($Q_i^{S+1}=\overline{Q_i^S}$) і y_3 - нерівнозначність ($Q_i^{S+1} = Q_i \text{ І } D_i^S$).

Відповідно до табл. 2 і рис. 8 одержуємо операторні представлення функцій:

$$\begin{cases}
 T_i = y_1 (\bar{D}_i Q_i \vee D_i \bar{Q}_i) \vee (y_2 \vee y_3 D_i); \\
 R_i = (y_1 \bar{D}_i \vee y_2 Q_i) \vee (y_3 D_i) Q_i; \\
 S_i = (y_1 D_i \vee y_2 \bar{Q}_i) \vee (y_3 D_i) \bar{Q}_i.
 \end{cases}$$

При цьому складність КС менше при використанні Т-тригерів. Помітимо, що при виборі варіанта побудови регістра в разі потреби оцінюється складність не тільки КС, а всієї апаратури, включаючи тригери. Схема одного розряду на Т-тригері представлена на рис. 9. Усі розряди регістра будуються аналогічно. Як приклад показана організація виходів регістра з трьома станами. Керування станами здійснюється сигналом а.

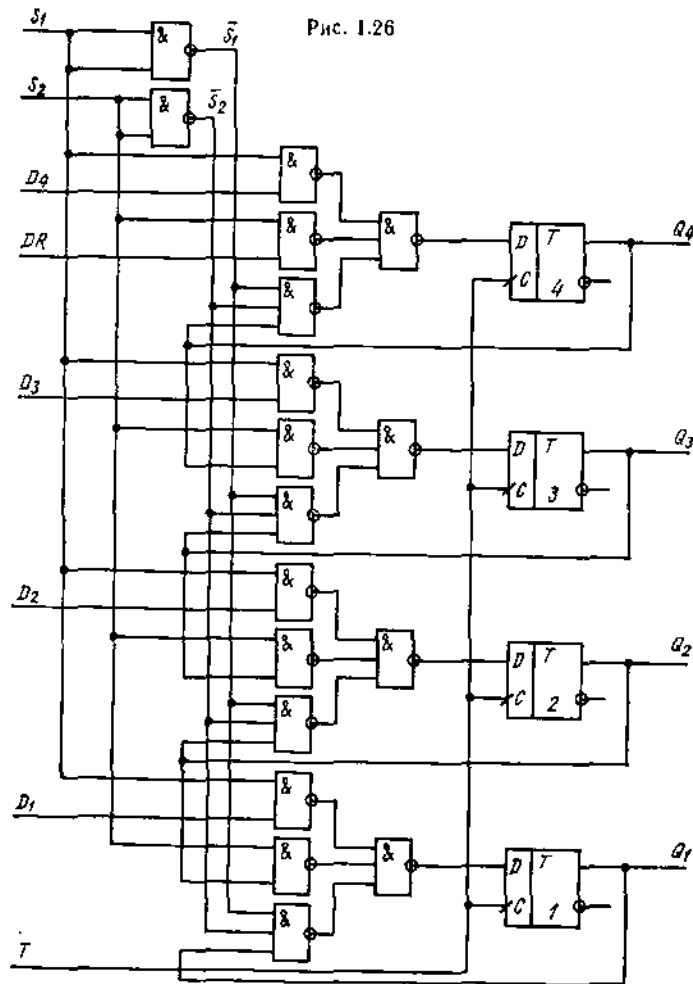


Рис. 7.

Часові характеристики регістрів визначають за допомогою часових діаграм. До основних з них відносять мінімальний час виконання мікрооперацій t і максимальну частоту переключення регістра f .

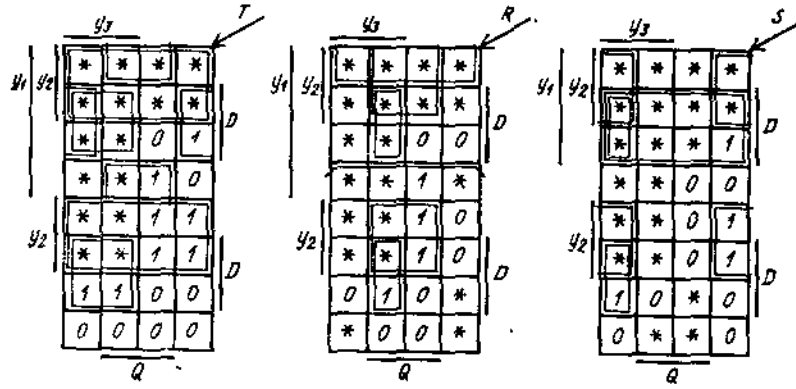


Рис. 8.

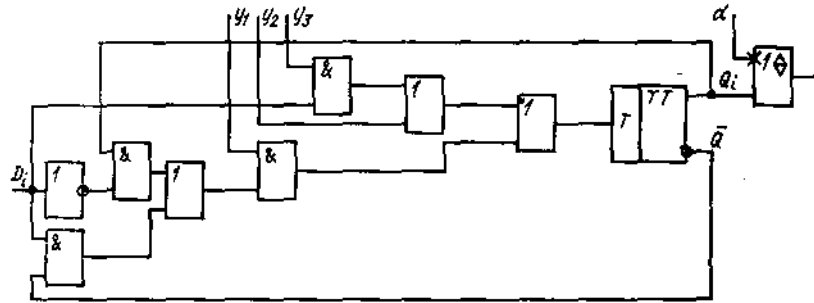


Рис. 9

Таблиця 1.

S_1^S	S_2^S	D_i^S	Q_{i+1}^S	Q_i^S	Q_i^{S+1}	D_i'	R_i	S_i	J_i	K_i
0	0	0	0	0	0	0	*	0	0	*
0	0	0	0	1	1	1	0	1	*	0
0	0	0	1	0	0	0	*	0	0	*
0	0	0	1	1	1	1	0	*	*	0
0	0	1	0	0	0	0	*	0	0	*
0	0	1	0	1	1	1	0	0	*	0
0	0	1	1	0	0	0	*	0	0	*
0	0	1	1	1	1	1	0	*	*	0
0	1	0	0	0	0	0	*	0	0	*
0	1	0	0	1	0	0	1	0	*	1
0	1	0	1	0	0	0	*	0	0	*
0	1	0	1	1	0	0	1	0	*	1
0	1	1	0	0	1	1	0	1	1	*
0	1	1	0	1	1	1	0	*	*	0
0	1	1	1	0	1	1	0	1	1	*
0	1	1	1	1	1	1	0	*	*	0
1	0	0	0	0	0	0	*	0	0	*
1	0	0	0	1	0	0	1	0	*	1
1	0	0	1	0	1	1	0	1	1	*
1	0	0	1	1	1	1	0	*	*	0
1	0	1	0	0	0	0	*	0	0	*
1	0	1	0	1	0	0	1	0	*	1
1	0	1	1	0	1	1	0	1	1	*
1	0	1	1	1	1	1	0	*	*	0
1	1	0	0	0	*	*	*	*	*	*
1	1	0	0	1	*	*	*	*	*	*
1	1	0	1	0	*	*	*	*	*	*
1	1	0	1	1	*	*	*	*	*	*
1	1	1	0	0	*	*	*	*	*	*
1	1	1	0	1	*	*	*	*	*	*
1	1	1	1	0	*	*	*	*	*	*
1	1	1	1	1	*	*	*	*	*	*

Таблица 2

y_1^S	y_2^S	y_3^S	D_i^S	Q_i^S	Q_i^{S+1}	T_i	R_i	S_i
0	0	0	0	0	0	0	*	0
0	0	0	0	1	1	0	0	*
0	0	0	1	0	0	0	*	0
0	0	0	1	1	1	0	0	*
0	0	1	0	0	0	0	*	0
0	0	1	0	1	1	0	0	*
0	0	1	1	0	1	1	0	1
0	0	1	1	1	0	1	1	0
0	1	0	0	0	1	1	0	1
0	1	0	0	1	0	1	1	0
0	1	0	1	0	1	1	0	1
0	1	0	1	1	0	1	1	0
0	1	1	0	0	*	*	*	*
0	1	1	0	1	*	*	*	*
0	1	1	1	0	*	*	*	*
0	1	1	1	1	*	*	*	*
1	0	0	0	0	0	0	*	0
1	0	0	0	1	0	1	1	*
1	0	0	1	0	1	1	0	0
1	0	0	1	1	1	0	0	1
1	0	1	0	0	*	*	*	*
1	0	1	0	1	*	*	*	*
1	0	1	1	0	*	*	*	*
1	0	1	1	1	*	*	*	*
1	1	0	0	0	*	*	*	*
1	1	0	0	1	*	*	*	*
1	1	0	1	1	*	*	*	*
1	1	1	0	0	*	*	*	*
1	1	1	0	1	*	*	*	*
1	1	1	1	1	*	*	*	*
1	1	1	1	1	*	*	*	*

ПІДГОТОВКА ДО РОБОТИ

1. Побудувати чотирирозрядний синхронний регістр для виконання мікрооперацій, заданих у табл. 3, де y_1 - зсув вліво на один розряд; y_2 - зсуву вправо на один розряд; y_3 - прийом слова паралельним кодом; y_4 - диз'юнкція; y_5 - кон'юнкція; y_6 - нерівнозначність (\oplus - додавання по модулю два); y_7 - рівнозначність (\ominus - еквівалентність); y_8 - інвертування розрядів.

Для побудови регістра можна використовувати логічні елементи, зазначені в табл. 4, а також RS-, JK-, T- чи D-тригери.

Значення \checkmark_i в табл 3 і 4 визначають варіант завдання. Для вибору варіанта необхідно його номер представити в двійковій системі числення і позначити шістьох молодших розрядів у виді слова $\checkmark_6 \checkmark_5 \checkmark_4 \checkmark_3 \checkmark_2 \checkmark_1$.

У процесі синтезу попередньо одержати операторні представлення функцій збудження для усіх функціональних типів тригерів, а потім вибрати тригер, що забезпечує мінімальну складність КС.

Визначити необхідну організацію тригерів (із внутрішньою чи затримкою без внутрішньої затримки) і побудувати схему регістра з ланцюгами видачі прямого коду. Передбачити установку регістра в кульовий стан, використовуючи асинхронні входи \bar{R} . За допомогою тимчасової діаграми визначити параметри регістра t і f , позначивши затримку сигналів логічними елементами через AND , OR , NOT , NAND , NOR . а час переключення тригера як t_{tr} .

2. Побудувати 4-розрядний синхронний регістр відповідно до табл. 3 і 4, виконавши етапи, аналогічні п. 1 завдання. Регістр повинен мати виходи з трьома станами. Для побудови регістра використовувати асинхронні RS-, JK- чи T-тригери.

3. Аналогічно вимогам п. 1 і п. 2 завдання виконати синтез 4-розрядного асинхронного регістра на асинхронних RS-, JK-, чи T-тригерах відповідно до табл. 3 і 4. Передбачити в регістрі ланцюга видачі інформації в прямому і зворотному коді.

ЗМІСТ ЗВІТУ

Звіт повинен містити короткі теоретичні відомості, необхідні для виконання лабораторної роботи і відповіді на контрольні питання, усі схеми, формули, таблиці і графіки, отримані при виконанні завдання, а також висновки.

Таблиця 3

✓ ₃	✓ ₂	✓ ₁	Мікрооперації		
			Номер пункту завдання		
			1	2	3
0	0	0	у ₁ , у ₃ , у ₄	у ₃ , у ₈	у ₂ , у ₃
0	0	1	у ₁ , у ₃ , у ₅	у ₃ , у ₇	у ₂ , у ₄
0	1	0	у ₁ , у ₃ , у ₆	у ₃ , у ₆	у ₂ , у ₅
0	1	1	у ₁ , у ₃ , у ₇	у ₃ , у ₅	у ₂ , у ₆
1	0	0	у ₂ , у ₃ , у ₈	у ₃ , у ₄	у ₁ , у ₇
1	0	1	у ₂ , у ₃ , у ₄	у ₁ , у ₈	у ₁ , у ₈
1	1	0	у ₂ , у ₃ , у ₅	у ₂ , у ₈	у ₁ , у ₆
1	1	1	у ₂ , у ₃ , у ₆	у ₁ , у ₇	у ₁ , у ₅

Таблиця 4

✓ ₆	✓ ₅	✓ ₄	Логічні елементи		
			Номер пункту завдання		
			1	2	3
0	0	0	2І-НІ, 3І	3АБО-НІ	3І, 2АБО, НІ
0	0	1	3І, 2АБО, НІ	2І-НІ	3АБО-НІ
0	1	0	3АБО-НІ	3І, 2АБО, НІ	2І-НІ
0	1	1	3І-НІ	2АБО-НІ	2І, 3АБО, НІ
1	0	0	2І, 3АБО, НІ	3І-НІ	2АБО-НІ
1	0	1	2АБО-НІ	2І, 3АБО-НІ	3І-НІ
1	1	0	2І-НІ	2АБО-НІ	3І, 3АБО, НІ
1	1	1	2І, 2АБО, НІ	3І-НІ	3АБО-НІ

КОНТРОЛЬНІ ПИТАННЯ

1. Скільки різних слів можна записати в регістр довжиною 1?
2. Які мікрооперації можна виконувати на регістрах?
3. Намалюйте узагальнену логічну структуру регістра на синхронних і асинхронних тригерах.
4. Від чого залежить складність комбінаційної схеми тригера?
6. Які регістри називаються зсуваючими?
6. Охарактеризуйте, які перетворення інформації здійснюються при виконанні різних мікрооперацій?
7. У яких випадках можна здійснити прийом слова на регістр в однофазному коді й у яких випадках - у паралельному коді?
8. Охарактеризуйте етапи синтезу комбінаційної схеми регістра.
9. Чим відрізняється процес синтезу комбінаційної схеми при використанні для побудови регістра синхронних і асинхронних тригерів?
10. Побудувати регістр на тригерах заданого типу для виконання визначеної мікрооперації.
11. Складіть таблицю переходів для *RS*-, *JK*-, *T*- і *D*-тригерів;.
12. У яких випадках для побудови регістра необхідно використовувати тригери з внутрішньою затримкою?
13. У чому відмінність тригерів із внутрішньою затримкою від тригерів без внутрішньої затримки?
14. Назвіть основні часові характеристики регістрів. Як їх визначити?
15. Як побудувати часову діаграму роботи регістра з урахуванням часу затримки сигналів логічними елементами і часу переключення тригерів?

ПРОЕКТУВАННЯ І ДОСЛІДЖЕННЯ ЛІЧИЛЬНИКІВ

Мета роботи: вивчення різних типів лічильників у потенційній елементній базі, оволодіння методами їхнього проектування і налагодження.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Лічильником називають схему, що призначена для виконання мікрооперації рахунку і збереження слів.

Число дозволених станів лічильників називають його *періодом*, *модулем* чи *коефіцієнтом перерахунку* K .

Лічильники *можуть бути побудовані* на основі рахункових тригерів зі спеціальними міжрозрядними зв'язками, на основі регістрів, що зсувають (кільцеві лічильники) і на основі багатостійких елементів. У даній роботі розглядаються лічильники двох перших типів.

Основними часовими характеристиками лічильників є:

f - максимальна частота надходження рахункових сигналів;

t - час переходу лічильника з одного стану в інший.

Лічильники зі спеціальними міжрозрядними зв'язками класифікуються по різних ознаках.

По *характеру мікрооперації рахунку* лічильники підрозділяються на *збільшуючі*, *зменшуючі* і *реверсивні*.

При надходженні чергового рахункового сигналу X вміст збільшуючого лічильника збільшується на 1, а зменшуючого лічильника - зменшується на 1. Реверсивний лічильник може виконувати як мікрооперацію додавання, так і мікрооперацію віднімання, у залежності від значення сигналу на керуючому вході Y (наприклад, при $Y = 1$ виконується додавання, а при $Y=0$ - віднімання).

У *залежності від основи системи числення*, у якій здійснюється мікрооперація рахунку, розрізняють *двійкові* лічильники, *двійково- n 'ятиркові*, *двійково-десяткові* і так далі.

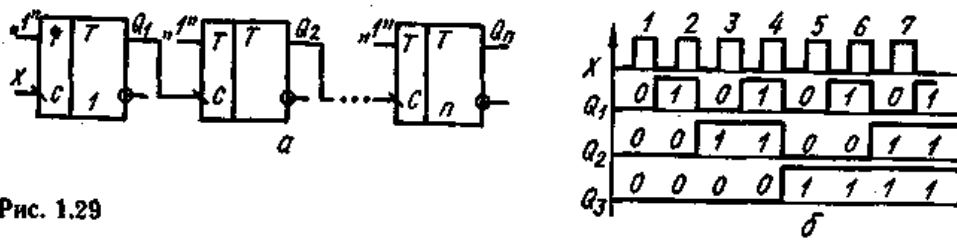


Рис. 1.29

Лічильники класифікуються і по *схемних ознаках*. Для побудови лічильників у потенційній елементній базі застосовуються переважно синхронні тригери з внутрішньою затримкою, що дозволяє використовувати на один розряд двійкового лічильника один тригер.

По способу організації ланцюгів переносу (позики) між розрядами лічильники підрозділяються на наступні типи: з *послідовним переносом*; з *наскрізним переносом*; з *паралельним переносом*; із *груповим переносом*.

У лічильниках з *послідовним переносом* перенос (позика) у сусідній старший розряд формується тільки після переключення тригера в попередньому розряді, тобто тригери переключаються не одночасно. При проектуванні таких лічильників виникають труднощі, пов'язані з необхідністю аналізу не тільки логічного рівня сигналів, що формуються в схемі, але і моментів зміни рівнів сигналів. На рис. 1.29, а представлена функціональна схема n-розрядного збільшуючого лічильника з послідовним переносом, побудованого на синхронних Т-тригерах, що переключаються по від'ємному перепаду тактуючого сигналу. Часова діаграма зміни сигналів на виходах Q_i ($i = \overline{1, 3}$, без обліку часу переключення тригерів) показана на рис. 1.29, б.

У лічильниках з *паралельним переносом* аргументами функцій переносів для кожного розряду є тільки сигнали на виходах тригерів відповідних розрядів. Переноси для всіх розрядів лічильника формуються одночасно (за умови, що всі логічні елементи в схемі мають однаковий час переключення).

Ланцюги *наскрізного переносу* організуються таким чином, щоб функція переносу i-го розряду лічильника була аргументом функції переносу (i + 1)-го розряду. У цьому випадку сигнали переносів для кожного розряду формуються по черзі, починаючи з молодших розрядів лічильника. Лічильники з наскрізним переносом вимагають меншого числа входів логічних елементів для організації

ланцюгів переносу, але уступають лічильникам з паралельним переносом у швидкодії.

У лічильниках із *груповим переносом* розряди розбиваються на групи (наприклад, n розрядів розбиваються на m груп). У межах однієї групи, як правило, організується паралельний перенос, а між групами - послідовний чи наскрізний. По такому принципу будуються і лічильники для систем числення з основою $K > 2$. У цьому випадку роль груп виконують K -ічні розряди.

Якщо мікрооперація рахунку виконується в двійково-кодованій системі числення (двійково-п'ятірковій, двійково-десятковій і т. д.), то для побудови одного розряду лічильника необхідно не менш $n = \lceil \log_2 K \rceil$ двійкових тригерів. Наприклад, для побудови одного розряду десяткового лічильника потрібно не менш чотирьох двійкових тригерів. Таким чином, один розряд K -ічного лічильника являє собою двійковий лічильник з коефіцієнтом перерахунку K , що виконує мікрооперацію рахунку у відповідному кодї.

Якщо мікрооперація рахунку виконується в канонічній двійковій системі числення (в однорідній позиційній двійковій системі числення з природним порядком ваг), то такий лічильник називають лічильником із *природним порядком рахунку*.

Стани чотирьохрозрядного лічильника з природним порядком рахунку ілюструються табл. 1.18.

Якщо мікрооперація рахунку виконується в неканонічних системах (наприклад, символічних, зі штучним порядком ваг), то *порядок рахунку вважається штучним*. Стани чотирьохрозрядного лічильника зі штучним порядком рахунку по модулю 2^n , що виконує мікрооперацію рахунку в кодї Грея, наведені в табл. 1.19.

З табл. 1.18 і 1.19 випливає, що стани лічильників повторюються з періодом 2^n . Для скорочення таблиць станів у них заносяться тільки стани для одного періоду.

Лічильники з природним і штучним порядком рахунку можуть мати коефіцієнт перерахунку $K \neq 2^n$

Таблиця 1.18

Количество счетных сигналов	Состояние счетчика	
	суммирующего	вычитающего
0	0 0 0 0	0 0 0 0
1	0 0 0 1	1 1 1 1
2	0 0 1 0	1 1 1 0
3	0 0 1 1	1 1 0 1
4	0 1 0 0	1 1 0 0
5	0 1 0 1	1 0 1 1
6	0 1 1 0	1 0 1 0
7	0 1 1 1	1 0 0 1
8	1 0 0 0	1 0 0 0
9	1 0 0 1	0 1 1 1
10	1 0 1 0	0 1 1 0
11	1 0 1 1	0 1 0 1
12	1 1 0 0	0 1 0 0
13	1 1 0 1	0 0 1 1
14	1 1 1 0	0 0 1 0
15	1 1 1 1	0 0 0 1
16	0 0 0 0	0 0 0 0
17	0 0 0 1	0 1 1 1
18	0 0 1 0	0 1 1 0
19	0 0 1 1	0 1 0 1

Таблиця 1.19

Количество счетных сигналов	Состояния счетчика
0	0 0 0 0
1	0 0 0 1
2	0 0 1 1
3	0 0 1 0
4	0 1 1 0
5	0 1 1 1
6	0 1 0 1
7	0 1 0 0
8	1 1 0 0
9	1 1 0 1
10	1 1 1 1
11	1 1 1 0
12	1 0 1 0
13	1 0 1 1
14	1 0 0 1
15	1 0 0 0
16	0 0 0 0
17	0 0 0 1
18	0 0 1 1
19	0 0 1 0
...	...

Найбільш простими є схеми лічильників із природним порядком рахунку, побудовані на основі тригерів з рахунковим входом (T - і JK -тригерів).

На рис. 1.30 представлена узагальнена структура лічильника на T -тригерах (KC - комбінаційна схема, що формує функції збудження D , що надходять на рахункові входи i -х тригерів). У JK -тригерах рахунковий вхід організується шляхом з'єднання входів J і K .

З табл. 1.18 випливає, що переключення тригера молодшого розряду здійснюється з приходом кожного рахункового сигналу, а інших тригерів - тільки в тому випадку, коли всі тригери молодших розрядів встановлені в 1 (збільшуючий) чи в 0 (що віднімає) лічильник.

Отже, для збільшуючих лічильників із природним порядком рахунку по модулю 2^n , що має ланцюги паралельного переносу, перемикальні функції f_i мають вигляд

$$f_i = Q_1 Q_2 \dots Q_{i-1} \quad (i = \overline{2, n}); \quad (1)$$

для лічильників, що віднімають

$$f_i = \overline{Q_1} \overline{Q_2} \dots \overline{Q_{i-1}} \quad (i = \overline{2, n}). \quad (2)$$

а для реверсивних

$$f_i = Q_1 Q_2 \dots Q_{i-1} Y \vee \overline{Q_1} \overline{Q_2} \dots \overline{Q_{i-1}} \overline{Y} \quad (i = \overline{2, n}) \quad (3)$$

Для всіх типів лічильників $f_1 = 1$.

Для лічильників з наскрізним переносом функції (1), (2) і (3) можуть бути представлені відповідно:

$$\begin{aligned} f_i &= f_{i-1} Q_{i-1} \quad (i = \overline{2, n}); \\ f_i &= f_{i-1} \bar{Q}_{i-1} \quad (i = \overline{2, n}); \\ f_i &= f_i \vee f_i' \quad (i = \overline{2, n}), \end{aligned}$$

де $f_1 = 1$; $f_2' = Q_1 Y$; $f_2 = \bar{Q}_1 \bar{Y}$; $f_i = f_{i-1} Q_{i-1}$ ($i = \overline{3, n}$); $f_i = f_{i-1} \bar{Q}_{i-1}$ ($i = \overline{3, n}$)

На рис. 1.31 показана функціональна схема збільшуючого лічильника з паралельним переносом на Т-тригерах, а на рис. 1.32 - реверсивного лічильника, з наскрізним переносом на JK-тригерах (для $n = 4$).

Лічильники з будь-яким коефіцієнтом перерахунку K , і будь-яким порядком рахунку можна побудувати в такий спосіб:

скласти таблицю переходів лічильника (за формою табл. 1.20), записавши у відповідні стовпці таблиці в кожному рядку коди станів лічильника до надходження чергового рахункового сигналу (S -й момент часу) і після його надходження ($(S + 1)$ -й момент часу). Наприклад, у i -му рядку для S -го моменту часу записати значення Q_i^S сигналів на виходах тригерів у вихідному стані, а для $(S + 1)$ -го моменту часу - значення Q_i^{S+1} після надходження першого рахункового сигналу;

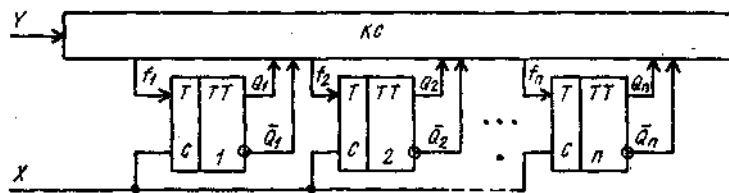


Рис. 1.30

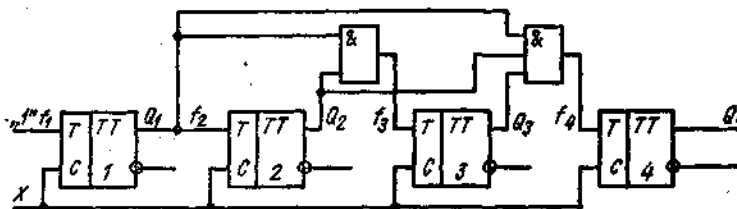


Рис. 1.31

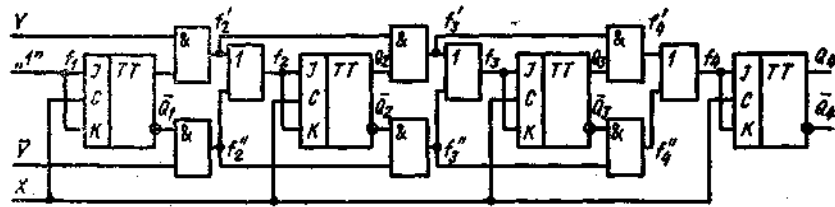


Рис. 1.32

уміст стовпців Q_i^{S+1} 0-го рядка перенести в стовпці Q_i^S 2-го рядка, а в стовпці Q_i^{S+1} цього рядка записати значення виходів тригерів після надходження чергового (у даному випадку другого) рахункового сигналу і т.д.;

для кожного i -го розряду лічильника записати в кожному j -му рядку таблиці значення сигналів на інформаційних входах f_i тригерів (функції збудження тригерів), що забезпечують переключення тригерів зі стану Q_i^S в стан Q_i^{S+1} ;

одержати операторні представлення функцій f_i у заданому елементному базисі для всіх типів тригерів, розглядаючи в якості аргументів значення $Q_1^S, Q_2^S, \dots, Q_n^S$;

вибрати тип тригера і побудувати схему лічильника.

Нехай необхідно побудувати лічильник, що змінює свій стан у наступній послідовності 000, 001, 010, 011, 110, 111, використовуючи для цього двовходові елементи І та АБО, а також T - і JK -тригери.

Складаємо таблицю переходів лічильника (табл. 1.20) і відповідно до рис. 1.24 записуємо для кожного рядка таблиці необхідні значення функцій збудження тригерів. Наприклад, для першого рядка переходи $Q_1^S \rightarrow Q_1^{S+1}$, $Q_2^S \rightarrow Q_2^{S+1}$ і $Q_3^S \rightarrow Q_3^{S+1}$ мають відповідно вид $0 \rightarrow 1$, $0 \rightarrow 0$ і $0 \rightarrow 0$. Для T -тригерів згідно рис. 1.24 одержуємо $T_1=1$, $T_2=0$, $T_3=0$, а для JK -тригерів записуємо в даному рядку $J_1=1$, $K_1=*$, $J_2=0$, $K_2=*$, $J_3=0$ і $K_3=*$.

Розглядаючи як аргументи значення Q_1^S , Q_2^S і Q_3^S , одержуємо операторні представлення функцій (опускаючи індекси 5):

$$\begin{aligned}
 T_1 &= 1; & T_2 &= Q_3 Q_1 \vee \bar{Q}_2 Q_1; & T_3 &= Q_2 Q_1; \\
 \begin{cases} J_1 = 1, \\ K_1 = 1; \end{cases} & \begin{cases} J_2 = Q_1, \\ K_2 = Q_3 Q_1; \end{cases} & \begin{cases} J_3 = Q_3 Q_1, \\ K_3 = Q_1. \end{cases}
 \end{aligned}$$

Схема лічильника на T -тригерах ілюструється рис. 1.33, а на JK -тригерах - рис. 1.34. Очевидно, що при використанні JK -тригерів для побудови

лічильника потрібно менше логічних елементів.

При великій розрядності лічильників розглянутий метод стає дуже трудомістким, зокрема при мінімізації функцій збудження тригерів.

Для побудови лічильників з коефіцієнтом перерахунку $K \neq 2^n$ можна скористатися методом виключення надлишкових станів, що рисо критичний до величини n .

Найбільше просто виключаються стани, що впливають підряд.

Таблиця 1.20

Состояния счетчика						Функции возбуждения триггеров								
Q_3^S	Q_2^S	Q_1^S	Q_3^{S+1}	Q_2^{S+1}	Q_1^{S+1}	T_1	T_2	T_1	J_1	K_1	J_2	K_2	J_1	K_1
0	0	0	0	0	1	0	0	1	0	*	0	*	1	*
0	0	1	0	1	0	0	1	1	0	*	1	*	*	1
0	1	0	0	1	1	0	0	1	0	*	*	0	*	*
0	1	1	1	1	0	1	0	1	1	*	*	0	*	1
1	1	0	1	1	1	0	0	1	*	0	*	0	1	*
1	1	1	0	0	0	1	1	1	*	1	*	1	*	1

При побудові лічильника необхідно, виконати наступні етапи:

визначити число розрядів n лічильника по формулі $n = \lceil \log_2 K \rceil$;

де $\lceil \cdot \rceil$ - функція округлення числа до найближчого більшого цілого, якщо число не ціле;

скласти таблицю станів і одержати функції збудження f_i тригерів заданого типу для всіх розрядів лічильника з природним порядком рахунку по модулі 2^n ;

виключити з таблиці надлишкові стани і позначити через A стан, що передує першому з групи виключених,

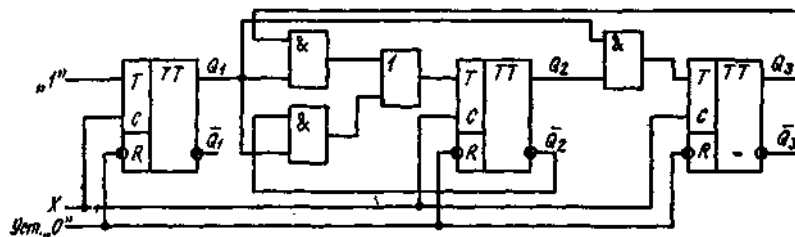


Рис. 1.33

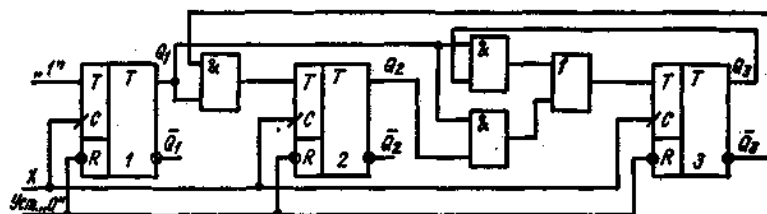


Рис. 1.34

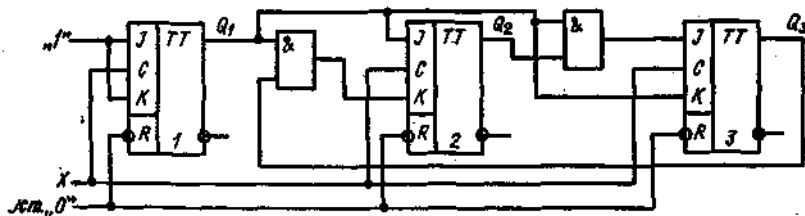


Рис. 1.35

через B - перший в групі виключених, через C - наступний за останнім із групи виключених станів (стан B відповідає природному, а стан C - штучному переходу зі стану A);

визначити скоректовані функції збудження f_i^* для проектованого лічильника за наступним правилом:

а) якщо i -е розряди для станів B и C збігаються, те (корекція функцій збудження не потрібно); $f_i^* = f_i$

б) якщо i -ий розряд при природному переході в стан B переключується, а для штучного переходу в стан C переключення не повинне відбуватися, те $f_i^* = f_i \cdot \bar{f}_A$, де f_A - функція, що приймає одиничне значення на наборі A и, можливо, на виключених наборах (здійснюється заборона переключення тригера);

в) якщо необхідно викликати переключення i -го розряду при штучному переході в стан C , що не переключується при природному переході в стан U , те $f_i^* = f_i \vee f_A$ (Забезпечується примусове переключення тригера);

одержати операторні форми функцій f_i^* у заданому елементному базисі і побудувати схему лічильника.

Якщо в лічильнику виключаються останні 2^{n-k} станів (наприклад, у збільшуючому лічильнику рахунок починається з 0 і закінчується числом $K-1$), то такі лічильники є лічильниками з природним порядком рахунку по модулю K . При виключенні інших станів природний порядок рахунку порушується. У цьому випадку (наприклад, для збільшуючого лічильника) число рахункових сигналів, що надійшли, не відповідає чисельному еквіваленту коду суми по модулю K даної кількості одиниць.

Нехай необхідно побудувати лічильник на T -тригерах з коефіцієнтом

перерахунку $K = 6$. Визначаємо розрядність лічильника по формулі $n = \lceil \log_2 K \rceil = \lceil \log_2 6 \rceil = 3$ і будуємо таблицю переходів 3-розрядного лічильника з природним порядком рахунку по модулі $2^n = 2^3 = 8$ (табл. 1.21).

Таблиця 1.21

Состояния счетчика			
q_2	q_1	q_0	
0	0	0	← C
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	← A
1	1	0	← B
1	1	1	

Вибираємо паралельний спосіб формування переносів і записуємо функції T_i ($i = 1, 2, 3$) для збільшуючого лічильника з природним порядком рахунку

$$T_1 = 1, T_2 = Q_1, T_3 = Q_1 Q_2$$

Виключаємо стани 110 і 111, відмічені в табл. 1.21 пунктиром, і позначимо три стани через A , B і C . Відповідно до приведеного вище правилом знаходимо

$$T_1^* = T_1; T_2^* = T_2 \cdot \bar{f}_A; T_3^* = T_3 \vee f_A.$$

Оскільки функція f_A на наборах 110 і 111 не визначена, одержуємо $f_A = Q_3 Q_1$. З обліком цього знаходимо

$$T_1^* = 1; T_2^* = Q_1 \cdot \overline{Q_3 Q_1} = \bar{Q}_3 Q_1; T_3^* = Q_2 Q_1 \vee Q_3 Q_1.$$

Схема лічильника показана на рис. 1.35. Установка лічильника в нульовий стан здійснюється за допомогою асинхронних входів \bar{R} тригерів.

Кільцеві лічильники будуються на базі зсуваючих регістрів (рис. 1.36). Виходи Q_i регістра підключаються до входів комбінаційної схеми (КС), а вихід останньої - до входу DR занесення в регістр інформації послідовним кодом. Рахункові сигнали X керують зрушенням у регістрі.

Кільцеві лічильники можуть забезпечити будь-який заданий коефіцієнт K

перерахунку, але не можуть, у загальному випадку, забезпечити довільний порядок зміни станів.

Основна задача проектування таких лічильників складається в синтезі КС, що забезпечує необхідний коефіцієнт перерахунку. У процесі проектування необхідно виконати наступне:

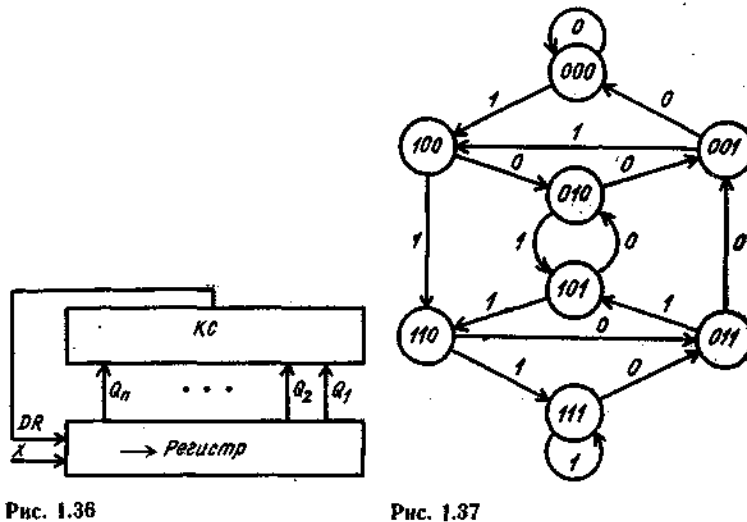


Рис. 1.36

Рис. 1.37

визначити розрядність регістра по формулі $n = \lceil \log_2 K \rceil$ і побудувати граф переходів зсуваючого регістра;

виділити на графі всі циклічні шляхи, що проходять через К вершин (можливі періоди лічильника з коефіцієнтом перерахунку К);

одержати операторні форми перемикальних функцій, реалізованих КС, відповідно до заданого елементного базису для кожного періоду лічильника;

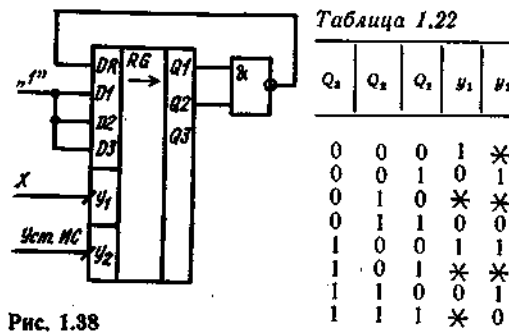
оцінити складність і швидкодія КС для кожної з перемикальних функцій, вибрати КС, що відповідає заданим параметрам, і побудувати схему лічильника.

Розглянемо синтез кільцевого лічильника по модулю 5. Для побудови такого лічильника досить трьохрозрядного зсуваючого регістра, тому що $n = \lceil \log_2 K \rceil = \lceil \log_2 5 \rceil = 3$. Нехай КС повинна бути реалізована на елементах 2І-НІ. Припустимо, що регістр знаходиться в стані 000, тоді в залежності від сигналу на вході DR (0 чи 1) після зсуву регістр перейде в стан 100 або залишиться в

попередньому стані. Зазначений і наступний переходи можна зобразити у виді графа (рис. 1.37), вершини якого відповідають станам регістра, а дуги - переходам, здійснюваним при 0 чи 1 на вході DR .

На графі можна виділити два циклічних шляхи, що проходять через п'ять вершин. Один з них проходить через вершини 000, 100, 110, 011 і 001, а інший - через вершини 111, 011, 001, 100 і 110.

У першому випадку на вході DR регістра повинні по черзі бути присутнім значення 1, 1, 0, 0 і 0, а в другому - 0, 0, 1, 1 і 1. Перемикальну функцію, що повинна реалізувати КС, у першому випадку позначимо через f_1 , а в другому - через f_2 . У відповідності є таблицею істинності функцій (табл. 1.22) одержуємо операторні форми $f_1 = \overline{\overline{Q_2 Q_1}}$ і $f_2 = \overline{Q_2 Q_1}$. Функція f_2 при реалізації на елементах І-НІ забезпечує меншу складність і більшу швидкодію КС. Таким чином, остаточно вибираємо $DR = f_2 = \overline{Q_2 Q_1}$. Схема лічильника на основі асинхронного зсуваючого регістра представлена на рис.1.38.



При позитивному перепаді сигналу на вході v_1 здійснюється зсування вмісту регістра (цей вхід використовується в якості рахункового), а на вході v_2 - паралельне занесення інформації. За допомогою мікрооперації занесення здійснюється установка регістра у початковий стан 111.

ПІДГОТОВКА ДО РОБОТИ

1. На синхронних JK-тригерах і елементах І розробити схеми збільшуючого, що віднімає і реверсивного лічильників з наскрізним переносом і природним порядком рахунку по модулю 16.

2. Використовуючи синхронні Т-тригери й елементи ІЛИ-НІ, побудувати збільшуючий, що віднімає і реверсивний лічильники з паралельним переносом і природним порядком рахунку по модулю 16.

3. Побудувати збільшуючий і що віднімає лічильники з груповим переносом, що містять по чотирьох m-розрядні групи тригерів (за схемою групи тригерів позначити прямокутником). У збільшуючому лічильнику перенос між групами паралельний, а в що віднімає - наскрізний. Для реалізації ланцюгів переносу використовувати елементи І. Записати перемикальні функції для виходів груп тригерів, підключених до ланцюгів поширення переносів.

4. Для збільшуючих лічильників, отриманих при виконанні пунктів 1, 2 і 3 завдання, побудувати часові діаграми роботи з урахуванням затримки сигналів на логічних елементах ($t_{\text{и}}$ і тили-ні і часу переключення тригерів ($\tau_{\text{т}}$). На основі діаграм визначити для кожного лічильника параметри f і t .

5. Використовуючи JK-, T- і D-тригери, а також елементи ЗІ-НІ, побудувати лічильник з періодом, зазначеним у табл. 1.23, де $\alpha_4, \alpha_3, \dots, \alpha_1$ - шість молодших двійкових розрядів номера варіанту завдання. Забезпечити мінімальну складність КС вибором відповідного типу тригера.

6. На Т-тригерах і елементах І, АБО і НІ побудувати лічильник по модулю $K=31$. Період проєктованого лічильника одержати, склавши таблицю станів збільшуючого лічильника з природним порядком рахунку по модулю 32 і виключивши з неї стан

$$\alpha_3\alpha_4\alpha_3\alpha_2\alpha_1$$

7. Побудувати кільцевий лічильник з коефіцієнтом перерахунку $K = 4 + 2\alpha_2 + \alpha_1$ на базі регістра, показаного на рис. 1.38. Для КС використовувати елементи, зазначені в табл. 1. 24.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

Зібрати, налагодити і дослідити схем лічильників, отриманих при виконанні завдання.

ЗМІСТ ЗВІТУ

Звіт повинен містити короткі теоретичні відомості, необхідні для виконання завдання і відповіді на контрольні питання, усі схеми, формули, таблиці і діаграми, отримані при виконанні лабораторної роботи, а також висновки по роботі.

Таблиця 1.23

Состояния счетчика			
Q_4	Q_3	Q_2	Q_1
0	0	0	α_1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	α_2
1	0	0	α_3
1	0	1	α_4
1	1	0	α_5
1	1	1	α_6

Таблиця 1.24

α_4	α_5	α_6	Елементи
0	0	0	2И-НЕ
0	0	1	3И-НЕ
0	1	0	2ИЛИ-НЕ
0	1	1	3ИЛИ-НЕ
1	0	0	2И, 2ИЛИ, НЕ
1	0	1	2И, 3ИЛИ, НЕ
1	1	0	3И, 2ИЛИ, НЕ
1	1	1	3И, 3ИЛИ, НЕ

КОНТРОЛЬНІ ПИТАННЯ

1. По яких ознаках можна класифікувати лічильники?
2. Напишіть вирази для функцій формування переносів (позик) збільшуючого, що віднімає, реверсивного лічильників з наскрізним і паралельним переносом і природним порядком рахунку по модулі $2n$.
3. Як одержати Т-тригер на основі RS-, D- і JK-тригерів?
4. Як враховуються набори, що відповідають забороненим станам лічильників, при мінімізації функцій збудження тригерів?
5. Складіть таблицю переходів RS-, JK-, D- я Т-тригерів. б. Як можна установити лічильник у вихідний стан?
7. Охарактеризуйте основні етапи проектування лічильників з довільною зміною станів і з виключенням надлишкових станів, що впливають підряд. Укажіть переваги і недоліки зазначених методів.

8. Який порядок рахунку називається природним, а який штучним?
9. Побудуйте лічильник на RS-, JK-, D- і T-тригерах по заданій таблиці станів.
10. Назвіть основні часові характеристики лічильників.
11. Як забезпечити заданий коефіцієнт перерахунку кільцевого лічильника?
12. Як побудувати часову діаграму роботи лічильника з урахуванням часу затримки сигналів логічними елементами і часу переключення тригерів?
13. Як визначити час переходу лічильника з одного стану в інше і максимальну частоту надходження рахункових сигналів?
14. Яка внутрішня організація тригерів дозволяє використовувати на один розряд двійкового лічильника один тригер?
15. У яких випадках доцільна групова організація ланцюгів переносів у лічильниках?
16. Укажіть переваги і недоліки кільцевих лічильників.

ПРОЕКТУВАННЯ І ДОСЛІДЖЕННЯ СУМАТОРІВ

Мета роботи: вивчення методів проектування суматорів, одержання навичок у зборці і налагодженні однорозрядних суматорів на потенційних елементах, оцінка якості досліджуваних суматорів і їхня оптимізація.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Суматор - операційний вузол, що виконує мікрооперацію арифметичного додавання (підсумовування) двох чисел (слів). Додавання n-розрядних чисел зводиться до виконання поразрядних операцій:

$$\left. \begin{array}{l} s_i = x_i + y_i + z_i \\ p_i = 0 \end{array} \right\} \text{ при } x_i + y_i + z_i < k;$$
$$\left. \begin{array}{l} s_i = x_i + y_i + z_i \\ p_i = 1 \end{array} \right\} \text{ при } x_i + y_i + z_i \geq k,$$

де s_i - значення суми в i -му розряді; z_i - перенос з молодшого розряду; p_i - перенос у старший розряд; k - основа системи числення; $x_i, y_i \in \{0, 1, \dots, k-1\}$ порозрядні значення доданків

$$X = \sum_{i=1}^n x_i k^{i-1} \text{ и } Y = \sum_{i=1}^n y_i k^{i-1}.$$

У залежності від основи системи числення і прийнятого кодування k -їчних цифр розрізняють *двійкові, трійкові, десяткові, двійково-десяткові* й інші суматори.

По способу організації процесу додавання однорозрядних доданків розрізняють *комбінаційні, накопичуючі і комбінаційно-накопичуючі суматори*.

По способу обробки багаторозрядних чисел суматори розділяють на *паралельні, послідовні і послідовно-паралельні*.

Організація ланцюгів переносу між розрядами визначає структуру суматора з *послідовним, наскрізним, груповим і одночасним (паралельним) переносом*.

Схемне рішення суматора при його проектуванні залежить від використовуваної системи елементів (серії мікросхем), вимог до тривалості операції додавання і припустимих витрат.

Комбінаційні однорозрядні суматори. Однорозрядним суматором називають перемикальну схему, що за розрядним значенням x_i і y_i доданків і за значенням переносу z_i з молодшого розряду формує значення розрядної суми s_i і перенос у старший розряд p_i . Робота такого суматора може бути описана в табл. 1.25 (при $k = 2$).

Таблиця 1.25

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

Мінімальні диз'юнктивні нормальні форми (МДНФ) функцій s_i і p_i мають вигляд

$$s_i = x_i y_i \bar{z}_i \vee x_i \bar{y}_i z_i \vee \bar{x}_i y_i z_i \vee x_i y_i z_i; \quad (4)$$

$$p_i = x_i y_i \vee x_i z_i \vee y_i z_i; \quad (5)$$

а їх мінімальні кон'юнктивні нормальні форми (МКНФ) -

$$s_i = (x_i \vee y_i \vee z_i)(x_i \vee \bar{y}_i \vee \bar{z}_i)(\bar{x}_i \vee y_i \vee \bar{z}_i)(\bar{x}_i \vee \bar{y}_i \vee z_i); \quad (6)$$

$$p_i = (x_i \vee y_i)(x_i \vee z_i)(y_i \vee z_i). \quad (7)$$

Нормальні форми (4)-(7) функцій s_i і p_i можуть бути перетворені до операторного вигляду, зручного для реалізації суматора на елементах типу І-НІ, І-АБО-НІ і т.п. Наприклад, при використанні елементів І-АБО-НІ однорозрядний комбінаційний суматор можна побудувати по виразах

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

$$p_i = \overline{x_i y_i \vee x_i z_i \vee y_i z_i}.$$

Схема такого суматора приведена на рис. 1.39.

Напівсуматором називають комбінаційну схему, що реалізує функції q_i суми по $\text{mod } k$ і переносу c_i при додаванні двох змінних, тобто ($k = 2$).

$$q_i = x_i \bar{y}_i \vee \bar{x}_i y_i = (x_i \vee y_i)(\bar{x}_i \vee \bar{y}_i) = (x_i \vee y_i) \bar{c}_i, \quad c_i = x_i y_i.$$

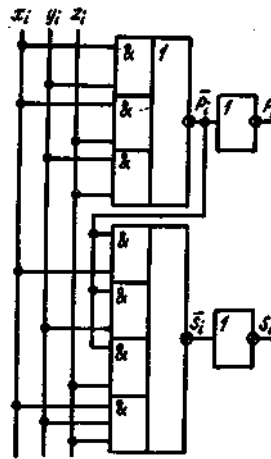


Рис. 1.39

Повний однорозрядний суматор може бути побудований із двох напівсуматорів у відповідності зі схемою на рис. 1.40.

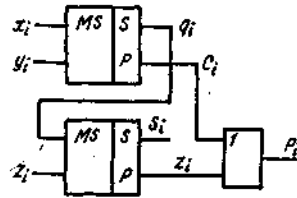


Рис. 1.40

Комбінаційні двійково-десяткові суматори. При побудові двійково-десяткових суматорів (ДДС) десяткові цифри, як правило, кодують за допомогою чотирьохрозрядних двійкових чисел - тетрад.

У цьому випадку ДДС повинен реалізувати п'ять перемикальних функцій: чотири з них відповідають двоїчно-кодованій десятковій сумі - $S_i, S_{i+1}, S_{i+2}, S_{i+3}$ і одна - переносу в старший десятковий розряд p_{i+3} . Ці функції залежать від десяткових цифр $(x_i, x_{i+1}, x_{i+2}, x_{i+3})$ і $(y_i, y_{i+1}, y_{i+2}, y_{i+3})$ і переноси з молодшої тетради z_i . Нормальні форми функцій, реалізованих ДДС, дуже громіздкі і важко мінімізуємі. Тому додавання двійково-десяткових кодів (ДДК) виконують у відповідності зі схемою на рис.1.41.

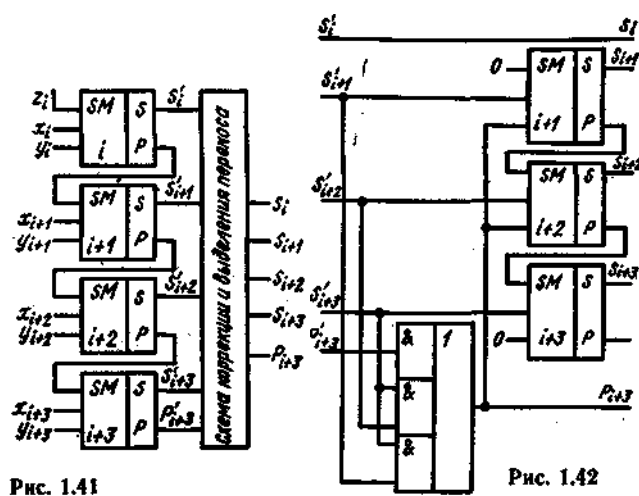
Тут на першому етапі складають ДДК десяткових цифр за правилами двійкової арифметики. Потім на другому етапі за допомогою суматорів роблять корекцію отриманого результату шляхом додавання чи віднімання деякого виправлення, величина якого залежить від некоректованого результату і виду

ДДК, а також виділяють десятковий перенос у старшу тетраду.

Використовуваний ДДК повинен мати властивість аддитивності, тобто ДДК суми десяткових цифр повинен дорівнювати сумі ДДК доданків. Такою властивістю володіють, наприклад, ДДК 8, 4, 2, 1 і 8, 4, 2, 1 + Δ , де Δ - ціле число, що називають *надлишком*. Якщо ж використовуваний ДДК не має властивість аддитивності, то цифри доданків необхідно попередньо (до додавання) перетворити в аддитивний ДДК. Структура схеми корекції і виділення переносу може бути визначена шляхом порівняння слова $p_{i+3}^i s_{i+3}^i s_{i+2}^i s_{i+1}^i s_i^i$, отриманого при додаванні цифр доданків і необхідного результату $p_{i+3} s_{i+3} s_{i+2} s_{i+1} s_i$ на виходах схеми корекції (рис. 1.41). Нехай, наприклад, у якості ДДК використовується код 8, 4, 2, 1. Тоді стани виходів суматорів за схемою на рис.1.41 можна описати табл.1.26.

З табл. 1.26 видно, що в залежності від суми, отриманої на першому етапі, корекція результату для ДДК 8, 4, 2, 1 складається в додатку 0 чи 6. Вважаючи функції s_{i+3}'' , s_{i+2}'' , s_{i+1}'' , s_i'' і p_{i+3} нецілком визначеними функціями аргументів p_{i+3} , $s_{i+3}^i s_{i+2}^i$, s_{i+1}^i , s_i^i , неважко переконатися, що $s_{i+3}'' = s_i'' = 0$ а $s_{i+2}'' = s_{i+1}'' = p_{i+3} = p_{i+3} \vee s_{i+3}^i s_{i+1}^i \vee s_{i+3}^i s_{i+2}^i$

Схема виділення переносу і корекції показана на рис. 1.42.



Таблиця 1.26

Десятична сумма	Сумма до корекції					Сумма после корекції					Коррекция			
	p'_{i+3}	s'_{i+3}	s'_{i+2}	s'_{i+1}	s'_i	p_{i+3}	s_{i+3}	s_{i+2}	s_{i+1}	s_i	s''_{i+3}	s''_{i+2}	s''_{i+1}	s''_i
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
2	0	0	0	1	0	0	0	0	0	1	0	0	0	0
3	0	0	0	1	1	0	0	0	1	1	0	0	0	0
4	0	0	1	0	0	0	0	1	0	0	0	0	0	0
5	0	0	1	0	1	0	0	1	0	1	0	0	0	0
6	0	0	1	1	0	0	0	1	1	0	0	0	0	0
7	0	0	1	1	1	0	0	1	1	1	0	0	0	0
8	0	1	0	0	0	0	1	0	0	0	0	0	0	0
9	0	1	0	0	1	0	1	0	0	1	0	0	0	0
10	0	1	0	1	0	1	0	0	0	0	1	1	0	0
11	0	1	0	1	1	1	0	0	0	1	0	1	1	0
12	0	1	1	0	0	1	0	0	1	0	0	1	1	0
13	0	1	1	0	1	1	0	0	1	1	0	1	1	0
14	0	1	1	1	0	1	0	1	0	0	1	1	1	0
15	0	1	1	1	1	1	0	1	1	0	1	1	1	0
16	1	0	0	0	0	1	0	1	1	0	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1	0	1	1	0
18	1	0	0	1	0	1	1	0	0	0	1	1	1	0
19	1	0	0	1	1	1	1	0	0	1	0	1	1	0

Накопичуючі однорозрядні суматори. Накопичуючим однорозрядним суматором називають схему з пам'яттю, що здійснює арифметичне додавання цифр що складаються x_i , y_i і переносу z_i при подачі їхній на суматор послідовно в часі, незалежно від того, яким кодом (послідовним чи паралельною) представлені слова, що складаються, X і Y , а також запам'ятовуючий результат додавання. Такі суматори будують на основі тригерів Т-типу. Одна зі схем однорозрядного накопичуючого суматора представлена на рис.1.43. Тут на тригері 1 спочатку формується сума $q_i = x_i \bar{y}_i \vee \bar{x}_i y_i$, а потім $s_i = q_i \bar{z}_i \vee \bar{q}_i z_i$.

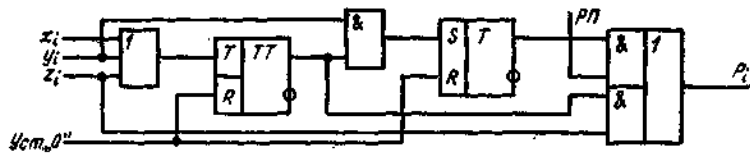


Рис. 1.43

Перенос при додаванні x_i і y_i формується елементом 4 і запам'ятовується тригером 2 до подачі зовнішнього сигналу дозволу переносу (РП), синхронного із сигналом z_i (відмінність між z_i і РП у тім, що, коли РП = 1, z_i може бути дорівнює і 0, і 1). Перенос при додаванні q_i і z_i формується елементом 7. Час додавання на такому суматорі дорівнює трьом тактам. Вхід x_i може бути

відсутнім. При цьому цифри доданків надходять послідовно на вхід y_i

У комбінаційно-накопичуючому суматорі (рис.1.44) сигнал переносу p_i формується по МДНФ цієї функції, а сума s_i - на тригері Т-типу. На вхід цього тригера подають сигнал, дорівнює сумі по mod 2 y_i і z_i і формований комбінаційною схемою. Час додавання тут складає два такти.

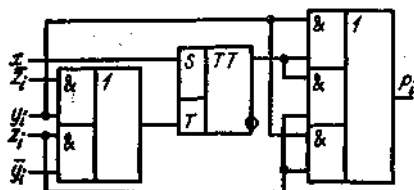


Рис. 1.44

ПІДГОТОВКА ДО РОБОТИ

1. Побудувати операторні форми функцій, що реалізуються однорозрядним повним суматором на елементах І-НІ, АБО-НІ, АБО-І-НІ, АБО-І-НІ. Відповідні схеми замальовати в протоколі, а також оцінити швидкодію і складність (у числі елементів і по Квайну) отриманих схем.

2. Використовуючи елементи, що маються в макеті, побудувати і замальовати до протоколу схему одного розряду двійково-десятькового суматора і цифрову діаграму станів його виходів (за аналогією з табл.1.26). Використовуваний ДДК визначається номером бригади і зазначений у табл.1.27.

3. Якщо ДДК не має властивість аддитивності, то необхідно також побудувати схеми перетворювачів заданого ДДК в аддитивний ДДК.

4. У відповідності з наявними в завданні тригерами і логічними елементами побудувати і замальовати до протоколу схеми однорозрядного накопичуючого і комбінаційно-накопичуючого суматорів. Побудувати часові діаграми роботи цих суматорів при різних значеннях цифр доданків.

Таблиця 1.27

Номер бригады	1, 11, 21	2, 12, 22	3, 13, 23	4, 14, 24	5, 15, 25	6, 16, 26
ДДК	7, 4, 2, 1	6, 4, 2, 1	5, 4, 2, 1	4, 4, 2, 1	4, 3, 2, 1	2, 4, 2, 1
Номер бригады	7, 17, 27	8, 18, 28	9, 19, 29	10, 20	30	31
ДДК	8, 4, 2, 1+1	8, 4, 2, 1+2	8, 4, 2, 1+3	8, 4, 2, 1+4	8, 4, 2, 1+5	8, 4, 2, 1+6

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Зібрати і налагодити однорозрядний комбінаційний суматор і напівсуматор на елементах І-НІ по схемах, отриманим при виконанні п. 1 теоретичного завдання.

2. Зібрати і налагодити заданий варіант двоїчно-десятькового суматора по схемах, отриманим при виконанні п. 2.

3. Зібрати і налагодити однорозрядний накопичуючий і комбінаційно-накопичуючий суматори по схемах, що були отримані при виконанні п. 3.

ЗМІСТ ЗВІТУ

Звіт повинен містити короткі теоретичні відомості, необхідні для виконання завдання і відповіді на контрольні питання, усі схеми, формули, таблиці і діаграми, отримані при підготовці і виконанні лабораторної роботи, а також висновки по роботі.

КОНТРОЛЬНІ ПИТАННЯ

1. По яких ознаках класифікують суматори?
2. Чому дорівнює мінімальна складність (у числі елементів і по Квайну) однорозрядного комбінаційного суматора і напівсуматора?
3. Яким вимогам повинні задовольняти ДДК, використовувані в ДДС?
4. У чому сутність властивості аддитивності ДДК і до чого може привести відсутність такої властивості в ДДК?

5. Приведіть приклади ДДК, що володіють і не володіють властивістю аддитивності.

6. В скількох разів число заборонених входних наборів для ДДК більше числа дозволених наборів?

7. Скільки розрядів повинен мати накопичуючий суматор, якщо на ньому послідовно підсумовують M чисел довжиною в n розрядів?

8. Чи можна змінювати місцями позначення входів (тобто x_i і y_i , x_i і z_i і т.д.) у комбінаційному суматорі? У накопичуючому? У комбінаційно-накопичуючому?

ПРОЕКТУВАННЯ І ДОСЛІДЖЕННЯ КЕРУЮЧИХ І ОПЕРАЦІЙНИХ АВТОМАТІВ

Мета роботи: вивчення методів синтезу керуючих автоматів із твердою логікою, одержання навичок у їхньому налагодженні й експериментальному дослідженні.

ОСНОВНІ ПОЛОЖЕННЯ

Арифметико-логические устройства (АЛУ) предназначены для выполнения операций над машинными словами.

Машинные слова задаются перечислением их разрядных значений (разрядов) в двоичном представлении или описанием, которое состоит из идентификатора (последовательности букв и цифр, начинающейся с буквы), справа от которого в скобках указываются номера старшего и младшего разрядов слова. Например, слова можно задать следующим образом: $X(1...32)$, $P2(1...8)$, $CT2(1...4)$, $Z(0...n + 1)$ и т. д.

Если разрядность (длина) слова предварительно оговорена, то оно может быть представлено одним идентификатором, т. е. в виде X , $P2$, $CT2$, Z и т. д.

Поле (часть слова) обозначается идентификатором слова с указанием номера старшего и младшего из разрядов, принадлежащих полю. Например, слово $P1(1...32)$ можно разделить на поля

$P1(1...8)$, $P1(9...16)$, $P1(17...32)$.

Слово может быть определено последовательностью описаний полей и разрядов, отделенных друг от друга точками. При этом любое поле может быть представлено двоичными цифрами. Возможны, например, следующие записи: $11.X(2...8).Z(10...12)$, $X(1...4).1011$, $Y(2)$, $P3(2...6).00\overline{P3}(1)$ и т. д. Такие слова называются составными.

Выполнение какой-либо операции в АЛУ сводится к выполнению последовательности микроопераций.

Микрооперацией называется элементарная операция, соответствующая элементарному машинному действию, в результате которого могут изменяться значения слов

Наиболее часто при выполнении арифметических операций используются микрооперации пересылки, инвертирования, сдвига суммирования и счета.

Микрооперация пересылки применяется для передачи слова (операнда) из одного узла (регистра, счетчика и т. д.) в другой и записывается с помощью оператора присваивания. Например, микрооперации пересылки могут быть представлены так:

$P1 := P2$ - передача слова из регистра $P2$ в регистр $P1$

$PA := 11.X.PY(I...8)$ - запись в регистр PA составного слова:

$CT := 1011$ - запись в счетчик CT двоичного числа 1011;

$PZ(I...8) := PX(9...16)$ - передача 8-разрядного поля регистра PX в старшие разряды регистра PZ . Микрооперация инвертирования заключается в замене единичных значений разрядов слова на нулевые, а нулевых - на единичные. Например, могут иметь место такие микрооперации: $\bar{P} := \bar{P}$ - инвертирование всех разрядов регистра P ;

$PA(1...18) := PA(1...2).PA(3...18)$ - образование обратного кода основных разрядов регистра PA (знаковые разряды не инвертируются).

Инвертирование может осуществляться в процессе пересылки слов либо с помощью специальных цепей, входящих в состав регистра.

Микрооперация сдвига слова выполняется на регистре (счетчике), имеющем цепи сдвига. Слово X , сдвинутое на i разрядов вправо, обозначается как $R_i X$, а влево - как $L_i X$. Для уточнения порядка заполнения освобождающихся разрядов используются составные слова. Микрооперации сдвига можно записать, например, в виде

$P1 := R1P1$ - сдвиг слова в регистре $P1$ на один разряд вправо;

$P2(1...32) := P2(32).R1P2$ - циклический сдвиг на один разряд вправо;

$PY := L2PY.11$ - сдвиг на два разряда влево с записью единиц в освобождающиеся разряды. Сдвиг на различное число разрядов можно также выполнять в процессе пересылки слов с использованием сдвигателя.

Микрооперация суммирования записывается в виде операторов: - суммирование прямых кодов слов, находящихся $P1 := P1 + P2$ в регистрах $P1$ и $P2$ (результат, формирующийся на сумматоре, помещается в $P1$);

$P2(1...8) := 00.P1(1...16) + 11.P3(1...16)$ - суммирование прямого и обратного кодов $P1$ и $P3$, дополненных знаковыми разрядами:

$PZ := PX + PY + 1$ - суммирование прямого кода слова, находящегося в PX с дополнительным кодом слова, записанного в PY (единица подается на вход переноса младшего разряда сумматора).

Микрооперация счета состоит в увеличении или уменьшении содержимого счетчика на единицу. Для этой микрооперации используются записи вида $C := C + 1$; $CT := CT - 1$.

Слева от микроопераций могут присутствовать метки (идентификаторы микроопераций). Метки и микрооперации разделяются двоеточием, например, $A1: P1 := P2$, $M: CT := CT + 1$ и т. д.

Способ выполнения микроопераций существенно зависит от структуры операционного устройства и используемой элементной базы.

Лабораторный макет содержит четыре реверсивных сдвигающих регистра (рис. 2.1, а), два комбинационных сумматора (рис. 2.1, б), четыре коммутатора (рис. 2.1, в) и два счетчика (рис. 2.1, е), имеющих цепи сдвига слова на один разряд влево (в сторону старших разрядов).

Запись слова на регистр с информационных входов $D1...D8$ осуществляется по заднему фронту (отрицательному перепаду) сигнала на входе V . Сдвиг влево и вправо также производится по заднему фронту сигналов соответственно на входах CL и CR . При сдвиге слова освободившиеся разряды регистра заполняются цифрами, поступающими последовательным кодом на входы DL или DR (в зависимости от направления сдвига). Установка нулевого состояния

регистра осуществляется единичным уровнем сигнала на входе R .

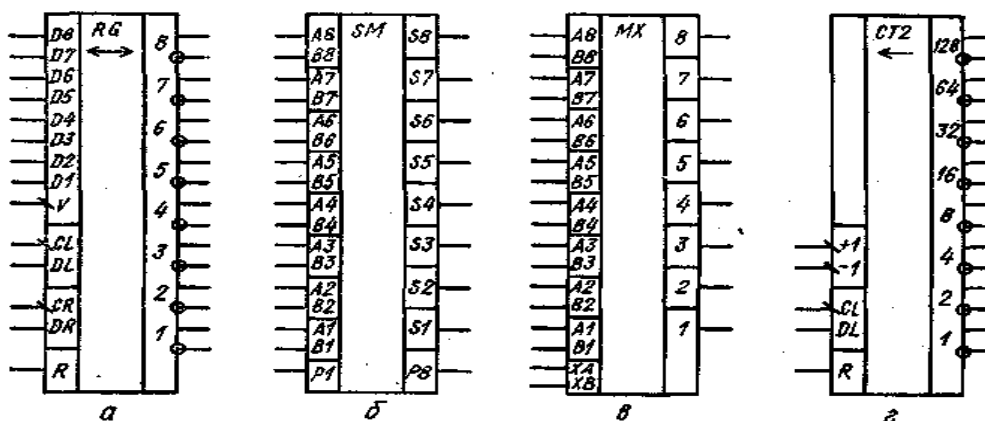


Рис. 2.1

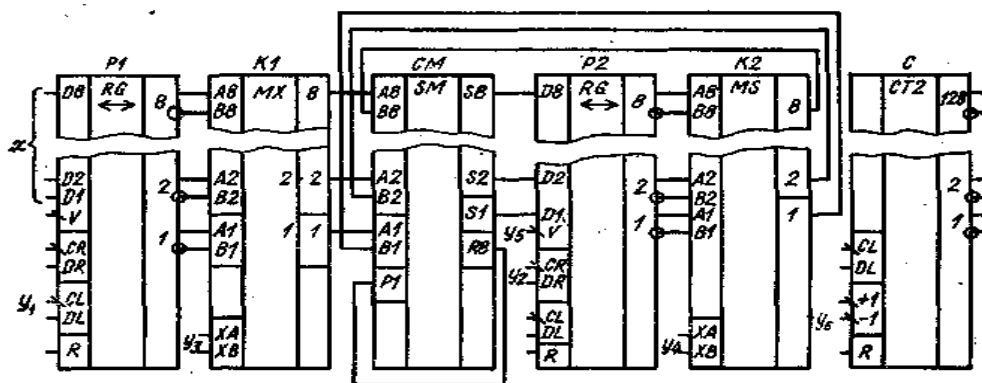


Рис. 2.2

Прибавление и вычитание единицы на счетчике осуществляется по заднему фронту сигналов соответственно на входах +1 и -1.

Сдвиг и установка нулевого состояния производится так же, как и на регистре.

Кроме того, макет содержит четырехвходовый дешифратор, синхронные D - и JK -триггеры (срабатывающие по заднему фронту тактирующего сигнала) с разделенными \bar{R} и \bar{S} входами, логические элементы И-НЕ (четырёх- и восьмивходовые), а также элементы индикации (сигнальные ячейки) и управления (кнопки, переключатели).

Сборка схемы осуществляется с помощью внешних проводников. В макете предусмотрена возможность установки регистров и счетчиков в исходное состояние, не используя внешнюю коммутацию.

Выполнение микроопераций рассмотрим на примере конкретного операционного устройства (рис. 2.2).

Значения сигналов, которые необходимо подать на управляющие входы

узлов ($P1, P2, K1, K2, C$) при выполнении некоторых микроопераций, показаны в табл. 2.1. Регистры и счетчик изменяют свое состояние после снятия единичных управляющих сигналов со входов $V, CL, CR, +1, -1$ (в процессе выполнения микроопераций управляющий сигнал для каждого узла может подаваться только на один из указанных входов). Знаком * в табл. 1.2 обозначены произвольные значения сигналов (0 или 1), которые не влияют на выполнение заданной микрооперации. Благодаря этому некоторые микрооперации можно выполнять одновременно. Например, можно совмещать микрооперацию $A1$ с любой другой, кроме $A5$. Нельзя выполнять одновременно микрооперации $A10$ и $A11$, но каждая из них может быть совмещена с другими микрооперациями.

Время выполнения микроопераций зависит от пути прохождения сигналов в схеме. Требуемая длительность управляющих сигналов определяется с помощью временной диаграммы, построенной с учетом задержек сигналов в элементах и узлах операционного устройства. Обычно наиболее длительной является микрооперация суммирования.

Таблица 2.1

Микрооперации	Значения сигналов на управляющих входах													
	P1			K1			P2			K2		C		
	CL	CR	V	R	XA	XB	CL	CR	V	R	XA	XB	+1	-1
$A1: P1 := X$	0	0	1	0	*	*	*	*	*	*	*	*	*	*
$A2: P2 := P1$	*	*	*	0	1	0	0	1	0	0	0	0	0	*
$A3: P2 := \overline{P2}$	*	*	*	*	0	0	0	1	0	0	1	*	*	*
$A4: P2 := L1P2$	*	*	*	*	*	*	1	0	0	0	*	*	*	*
$A5: P1 := R1P1$	0	1	0	0	*	*	*	*	*	*	*	*	*	*
$A6: P2 := P2 + P1$	*	*	*	0	1	0	0	1	0	1	0	*	*	*
$A7: P2 := P2 + \overline{P1}$	*	*	*	0	0	1	0	0	1	0	1	*	*	*
$A8: P2 := \overline{P2} + P1$	*	*	*	0	1	0	0	1	0	0	1	*	*	*
$A9: P2 := \overline{P2} + \overline{P1}$	*	*	*	0	0	1	0	0	1	0	0	1	*	*
$A10: C := C + 1$	*	*	*	*	*	*	*	*	*	*	*	*	1	0
$A11: C := C - 1$	*	*	*	*	*	*	*	*	*	*	*	*	0	1

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Послідовність мікрооперацій, виконання яких приводить до виконання заданої операції (команди), називають *мікроалгоритмом цієї операції*.

Мікроалгоритм може бути реалізований за допомогою *керуючого автомата*, узагальнена структура якого, показана на рис.2.3. Автомат містить

комбінаційну схему (КС) і пам'ять (П), що складається з тригерів T (елементарних автоматів). Входами КС є виходи Q_1, \dots, Q_m тригерів і вхідні сигнали (логічні умови) x_1, \dots, x_k , що формуються в операційному пристрої. КС виробляє керуючі сигнали y_1, \dots, y_p для операційного пристрою і функції збудження тригерів q_1, \dots, q_m , що визначають перехід автомата з одного стану в інше. Кожному з множини станів $\{a_1, \dots, a_m\}$ відповідає визначений набір значень Q_L .

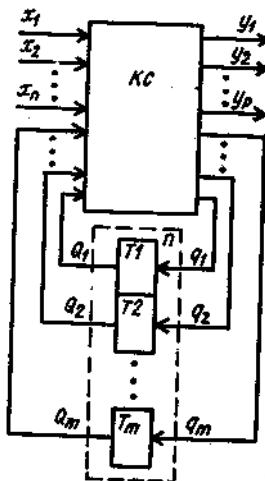


Рис. 2.3

Якщо вихідні сигнали залежать тільки від стану, у якому знаходиться автомат, то його називають автоматом Мура. Закон функціонування такого автомата визначається виразами

$$a^{S+1} = \delta(a^S, x^S), \quad y^S = \lambda(a^S),$$

де $S = 0, 1, 2, \dots$ - моменти автоматного (дискретного) часу; δ - функція переходів; λ - функція виходів; $a \in \{a_1, \dots, a_n\}$, $a^0 = a_1$, $x = (x_1, x_2, \dots, x_k)$ - вектор значень вхідних сигналів;

$y = (y_1, y_2, \dots, y_p)$ - вектор вихідних сигналів автомата.

Автомат, вихідні сигнали якого залежать як від стану, так і від вхідних сигналів, називають автоматом Мілі. Його функціонування визначається виразами

$$a^{S+1} = \delta(a^S, x^S);$$

$$y^S = \lambda(a^S, x^S).$$

Вхідними даними для синтезу автомата є схема операційного пристрою і

змістовний мікроалгоритм операції, що включає опису мікрооперацій і логічних умов. Етапи побудови принципової схеми і змістовного мікроалгоритму є взаємозалежними. Процес синтезу схеми автомата включає наступні етапи:

1. Складання списку керуючих сигналів, що забезпечують виконання кожної мікрооперації.
2. Визначення тривалості кожного керуючого сигналу (у числі тактів) і періоду тактируючих сигналів автомата.
3. Одержання закодованого мікроалгоритму.
4. Відмітка станів автомата.
5. Створення графа автомата.
6. Кодування станів автомата.
7. Створення структурної таблиці автомата.
8. Одержання МДНФ функцій збудження тригерів і керуючих сигналів.
9. Представлення функцій збудження тригерів і керуючих сигналів в операторній формі.
10. Побудова схеми керуючого автомата.

Розглянемо приклад синтезу автомата Мілі для керування виконанням операції $D = 2A^2 + 0, 5B$ на операційному пристрої (рис. 2.2).

Змістовний мікроалгоритм виконання операції представлений на рис. 2.4. У вихідному стані операнд B записаний у $P2$, а операнд A - у $P1$ і C . У першому такті шляхом зрушення слів здійснюється подвоєння A і $P1$ і розподіл B на 2 у $P2$. Далі до вмісту $P2$ A раз додається слово, записане в $P1$. Після кожного додатка вміст C зменшується на 1. Обчислення закінчуються при виконанні умови $C = 0$. Відповідний інформаційний сигнал можна одержати, наприклад, дешифруванням нульового стану C . Результат операції формується в $P2$.

Для виконання мікрооперацій $Y1$, $Y2$, $Y3$ і $Y4$ (рис. 2.4) необхідно використовувати керуючі входи операційного пристрою, позначені на рис. 2.2 через y_1 , y_2 , y_3 , y_4 , y_5 і y_6 . На всі інші керуючі входи повинний подаватися нульовий потенціал.

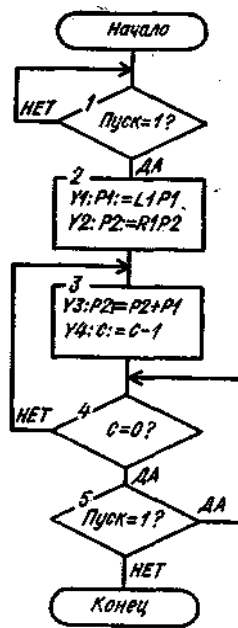


Рис. 2.4

Мікрооперації $Y1$, $Y2$ і $Y4$ керуються відповідно сигналами y_1 , y_2 і y_6 . Для виконання мікрооперації $Y3$ необхідно подати три сигнали: y_3 , y_4 , y_5 .

Необхідна тривалість керуючих сигналів визначається за допомогою часової діаграми, побудованої з урахуванням затримок сигналів в елементах операційного пристрою. Період Δt тактуючих сигналів, як правило, вибирається або рівним максимальній тривалості керуючих сигналів, або мінімальній. При цьому величина Δt повинна бути не менше часу переключення автомата з одного стану в інший. У першому випадку всі мікрооперації виконуються в синхронному режимі (за однаковий проміжок часу), а в другому - в асинхронному, причому тривалості керуючих сигналів кратні величині Δt . Асинхронний режим можна забезпечити, наприклад, введенням у мікроалгоритм додаткових операторних вершин з керуючими сигналами, тривалості яких перевищують Δt .

Будемо вважати, що з урахуванням швидкодії елементів для розглянутого приклада керуючі сигнали y_3 і y_4 повинні мати тривалість $2\Delta t$, а інші - Δt . Результати виконання перших двох етапів відбиті в табл. 2.2.

Для одержання закодованого мікроалгоритму складаємо таблицю позначень логічних умов (табл. 2.3) і заміняємо в змістовному мікроалгоритмі (рис. 2.4) опису логічних умов їх позначеннями, а опису мікрооперацій

відповідними керуючими сигналами (рис. 2. 5). Оскільки керуючі сигнали y_3 і y_4 , що відповідають мікрооперації Y_3 , записаній у вершині 3 на рис. 2.4, повинні мати тривалість 2τ , то на закодованому мікроалгоритмі (рис. 2.5) вводимо додаткову операторну вершину (4) з цими керуючими сигналами.

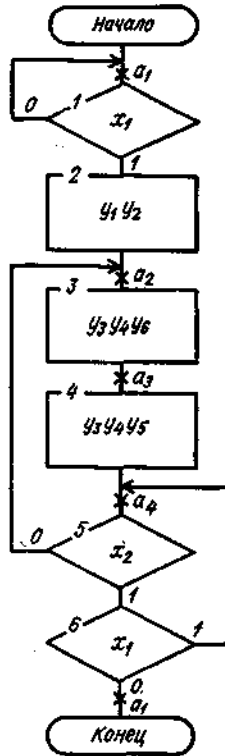


Рис. 2.5

Відмітка станів автомата Мілі здійснюється в такий спосіб:

символом a_1 відмічається вхід вершини (логічної чи операторної), що слідує за початковою, а також вхід кінцевої вершини;

входи усіх вершин, що сліднують за операторними, повинні бути відмічені різними символами a_j .

Таблиця 2.2

Мікрооперації	Управляючі сигнали	Длительність управляючого сигнала
$Y1: P1 = LIP1$	y_1	τ
$Y2: P2 = RIP2$	y_2	τ
$Y3: P2 = P2 + P1$	y_3, y_4, y_6	$2\tau, 2\tau, \tau$
$Y4: C = C - 1$	y_6	τ

Таблиця 2.3

Логические условия	Обозначение логических условий	Состояния	Код состояний	
			q_1	q_2
Пуск-1	x_1	a_1	0	0
$C = 0$	x_2	a_2	0	1
		a_3	1	1
		a_4	1	0

Крім станів, визначених таким чином, може виникнути необхідність введення додаткових станів. Наприклад, якщо мікроалгоритм має петлю, що

охоплює операторну вершину, із сигналом, що змінює стан вузла в момент свого перепаду, чи кілька вершин, що слідують підряд, то для забезпечення таких перепадів при кожному черговому виконанні мікрооперації необхідно вводити додаткові стани. У цих станах автомат не повинен виробляти керуючі сигнали. Додаткові стани можуть знадобитися для забезпечення протигоночного кодування (див. нижче).

Мікроалгоритм на рис. 2.5 відмічений чотирма різними станами (a_1, a_2, a_3, a_4).

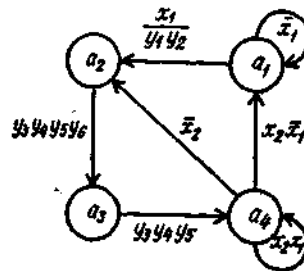


Рис. 2.6

Граф розглянутого автомата представлений на рис. 2.6. Число вершин графа дорівнює числу станів a_j автомата. Кожному переходу автомата з одного стану в інший відповідає дуга графа. Дугі приписується набір логічних умов, при якому здійснюється перехід автомата з одного стану в інший, а також набір керуючих сигналів, що відповідають даному переходу.

Кількість m тригерів, необхідних для організації пам'яті автомата (рис. 2.3), визначається з умови $m \geq \lceil \log_2 M \rceil$, де M - число станів автомата. Кожному стану a_j повинна відповідати одна визначена комбінація значень Q_1, \dots, Q_m . Для розглянутого приклада вибираємо коди станів відповідно до табл. 2.4. Для організації пам'яті будемо використовувати JK -тригери.

Помітимо, що спосіб кодування впливає на правильність формування керуючих сигналів і складність автомата. Можливість формування сигналів, не передбачених графом автомата при неоптимальному кодуванні станів, обумовлена появою «гонок», що зв'язано з розкидом часу переключення окремих тригерів автомата. Наприклад, у розглянутому випадку при переході автомата зі стану a_1 (10) у стан a_2 (01) під час переключення тригерів можлива поява станів 00 чи 11 (у залежності від того, який із тригерів раніше

переключиться). Ці проміжні стани при використанні тригерів із внутрішньою затримкою не впливають на правильність переключення автомата, однак можуть привести до появи помилкових керуючих сигналів (наприклад, y_1 і y_2 при виконанні умови $x_1=1$). Для усунення цього недоліку можна використовувати протигоночне кодування, одним з варіантів якого є сусіднє кодування. При сусіднім кодуванні перехід автомата з одного в будь-яке інше припустиме для даного автомата стан здійснюється переключенням тільки одного тригера, унаслідок чого «гонки» не виникають. В автоматах, що не допускають сусіднього кодування, необхідно вводити додаткові стани.

Структурна таблиця автомата будується по його графу. Кожен рядок (табл. 2.5) відповідає визначеному переходу автомата з одного стану в інший. У ній записують поточний стан, стан переходу, коди цих станів, значення логічних умов, що забезпечують перехід, відповідні значення керуючих сигналів і функцій збудження тригерів. Значення функцій збудження визначаються відповідно до таблиці переходів тригера відповідного типу. У кожному рядку для i -го тригера розглядаються переходи $Q_i^s \rightarrow Q_i^{s+1}$. Довільні значення (0 чи 1) сигналів позначаються в таблиці знаком *.

На основі структурної таблиці автомата визначаємо МДНФ функцій збудження і керуючих сигналів. Аргументами перемикальних функцій J_i , K_i і y_i є значення Q_1 , Q_2 , x_1 і x_2 . Використовуючи діаграми Вейча (рис. 2.7), отримуємо

$$y_1 = y_2 = \bar{Q}_1 \bar{Q}_2 x_1;$$

$$y_5 = y_6 = J_1 = Q_2;$$

$$K_1 = \bar{Q}_2 \bar{x}_2 \vee \bar{Q}_2 \bar{x}_1 = \bar{Q}_2 (\bar{x}_2 \vee \bar{x}_1);$$

$$K_2 = Q_1$$

$$J_2 = \bar{Q}_1 x_1 \vee Q_1 \bar{x}_2$$

Крім того, з табл. 2.5 випливає, що

$$y_3 = y_4 = 1$$

Функціональна схема автомата представлена на рис. 2.8, де $УИС$ - установка вихідного стану і $ТС$ - тактуючі сигнали.

Таблиця 2.5

Исходное состояние	Код исходного состояния		Состояние перехода	Код состояния перехода		Логические условия		Управляющие сигналы						Функции возбуждения триггеров			
	Q_1^S	Q_2^S		Q_1^{S+1}	Q_2^{S+1}	x_1	x_2	y_1	y_2	y_3	y_4	y_5	y_6	J_1	K_1	J_2	K_2
a_1	0	0	a_1	0	0	0	*	0	0	*	*	0	0	0	*	0	*
a_1	0	0	a_2	0	1	0	1	0	1	*	*	0	0	0	*	1	*
a_2	0	1	a_3	1	1	*	*	0	0	1	1	*	1	1	*	0	0
a_2	1	1	a_4	1	0	*	*	0	0	1	1	1	1	1	*	0	1
a_4	1	0	a_4	1	0	1	1	*	0	*	*	0	0	0	*	0	*
a_4	1	0	a_2	0	1	*	0	0	0	*	*	0	0	*	1	1	*
a_4	1	0	a_1	0	0	0	1	*	0	*	*	0	*	*	1	0	*

Відмінність синтезу автомата Мура складається в способі одержання відміченого мікроалгоритму і графа автомата.

Відмітка станів автомата в цьому випадку здійснюється так: символом a_1 відмічаються початкова і кінцева вершини; всі операторні вершини відмічаються різними символами a_j .

У даному випадку, як і при відмітці станів автомата Мілі, може виникнути необхідність уведення додаткових станів.

На графі автомата Мура дугам приписують тільки набори логічних умов, що забезпечують відповідний перехід автомата. Керуючі сигнали записують у вершинах графа, тому що вони не залежать від логічних умов. Відмічений мікроалгоритм і граф автомата Мура для розглянутого приклада представлені відповідно на рис. 2.9 і 2.10.

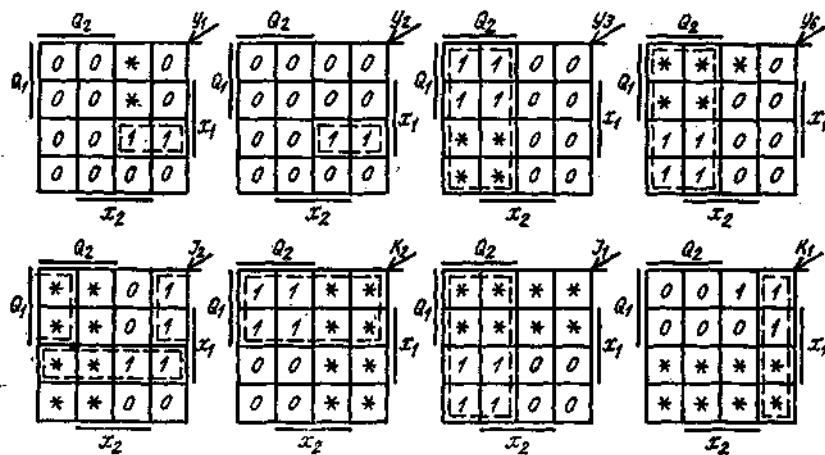


Рис. 2.7

Кодування станів автомата Мура можна виконувати так само, як і для автомата Мілі. Однак при відповідному кодуванні керуючі сигнали можна знімати і безпосередньо з виходів тригерів автомата Мура (комбінаційна схема для формування функцій y_j не потрібна). При цьому число тригерів повинне бути не менше числа керуючих сигналів, що не повторюють один одного і не є константними. У процесі переключення автомата Мура в цьому випадку відповідний рівень керуючих сигналів установлюється швидше, ніж в автомата Мілі. Крім того, при будь-якому переході виключається можливість короткочасного формування помилкових керуючих сигналів. Приклад такого кодування станів автомата, що відповідає графу на рис. 2.8, приведений у табл. 2.6.

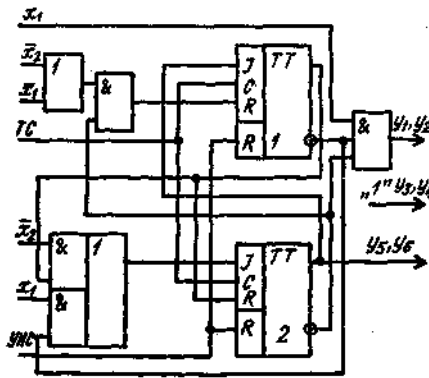


Рис. 2.8

У даному випадку автомат має три тригери, причому 1

$$y_1 = Q_1, y_2 = Q_1, y_3 = Q_2, y_6 = Q_2$$

Для формування константних керуючих сигналів $y_3 = y_4 = 1$ тригери не вимагаються. Співпадаючі сигнали y_1 і y_2 знімаються з одного тригера.

ПІДГОТОВКА ДО РОБОТИ

1. Скласти змістовний мікроалгоритм обчислення функції D на операційному пристрої (рис. 2.2). Вид функції D визначається по табл. 2.7 у відповідності зі значеннями альфа₅ і альфа₄ двох цифр номера варіанта, записаного у виді двоичного слова, обмеженого п'ятьма молодшими розрядами $\alpha_5\alpha_4\alpha_3\alpha_2\alpha_1$. Вважати, що у вихідному стані операнди A, B к C записані відповідно в $P1, C$ и $P2$. Для опису мікрооперацій використовувати символіку, приведену в

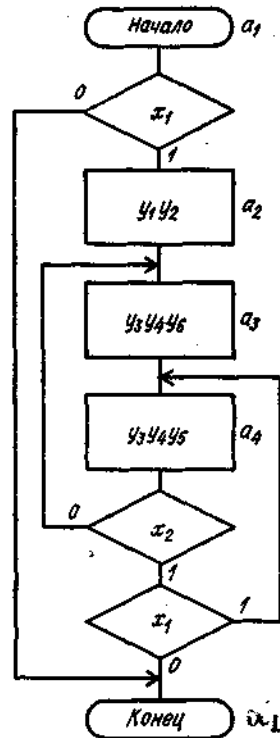


Рис. 2.9

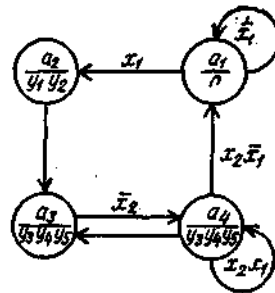


Рис. 2.10

2. Відповідно до отриманого мікроалгоритму і рис. 2.2 побудувати керуючий автомат для випадку виконання мікрооперацій у синхронному режимі. Тип тригерів визначається по табл. 2.7 і залежності, від значень α_3 і α_2 , а тип автомата - за значенням α_1 . Для побудови комбінаційної схеми використовувати елементи І-НІ.

3. Побудувати часову діаграму роботи автомата для кожної комбінації значень логічних умов.

4. Проілюструвати прикладом обчислення результату D для одного довільного набору значень операндов A , B я C , представлених трьома двоичними розрядами.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Зібрати схему автомата і налагодити її. Логічні умови задаються за допомогою тумблерів.

2. Зібрати схему операційного пристрою (рис.2.2) і підключити до його керуючих входів виходи автомата. Записати в $P1$, C и $P2$ трьохрозрядні операнди, що відповідають п. 4 завдання, подати сигнал ПУСК і перевірити правильність одержання результату $D \text{ } 9 \text{ } P2$.

ЗМІСТ ЗВІТУ

Звіт повинен містити короткі теоретичні відомості, необхідні для виконання лабораторної роботи, усі схеми, формули, таблиці і графіки, отримані при виконанні теоретичного завдання й у процесі експериментального дослідження схем, а також висновки по роботі.

КОНТРОЛЬНІ ПИТАННЯ

1. Що таке мікроалгоритм операції?
2. Намалюйте узагальнену структуру схеми керуючого автомата.
3. Напишіть вирази, що визначають закон функціонування автоматів Мілі і Мура.
4. У чому складається відмінність автоматів Мілі і Мура?
5. Охарактеризуйте основні етапи проектування керуючого автомата.
6. Як перейти від змістовного мікроалгоритму до закодованого мікроалгоритму?

Таблиця 2.6

Состояние	Код состояния		
	Q_1	Q_2	Q_3
a_1	0	0	0
a_2	1	0	0
a_3	1	0	1
a_4	1	1	0

Таблиця 2.7

α_1	α_2	Функция	α_3	α_4	Тип триггера	α_5	Тип автомата
0	0	$D = 4AB + 2C$	0	0	JK	0	Мили
0	1	$D = A(B-1) + 0,5C$	0	1	T	1	Мура
1	0	$D = 2A(B+1) + 0,5C$	1	0	RS	—	—
1	1	$D = A(B+1) + 2C$	1	1	D	—	—

7. Як побудувати граф автомата?
8. Як здійснюється оцінка станів автомата?
9. Як визначити необхідну тривалість керуючих сигналів?
10. Від чого залежить кількість тригерів, необхідне для побудови автомата?
11. У чому суть протигончного кодування станів автомата?
12. Як скласти структурну таблицю автомата?
13. Складіть таблицю переходів для *JK*-, *RS*-, *T*- і *D*-тригерів.
14. Чи можливий перехід автомата в стан, не передбачене графом, при використанні тригерів із внутрішньою затримкою (тригерів, керованих рівнем сигналів)?
15. Коли можливе виникнення помилкових керуючих сигналів (не передбачених графом автомата) і чим визначається їхня тривалість?
16. Як визначити час переходу автомата з одного стану в інше?
17. Як одержати *T*- і *RS*-тригери на основі *JK*-тригерів?
18. Як побудувати часову діаграму роботи автомата?

РЕАЛІЗАЦІЯ ДОДАВАННЯ ЧИСЕЛ З ФІКСОВАНОЮ КОМОЮ В ПРЯМИХ І ДОДАТКОВИХ КОДАХ

Мета роботи: вивчення способів виконання операцій додавання і віднімання в цифрових ЕОМ, оволодіння методами проектування пристроїв, що підсумовують, в потенційній елементній базі, отриманих навиків в їх відлагодженні.

ОСНОВНІ ПОЛОЖЕННЯ

В ЗУ ЕОМ числа, як правило, зберігаються в прямому або додатковому коді. В таких же кодах операнди витягаються із ЗУ і пересилаються в регістри АУ. Тому розрізняють операції в прямих і додаткових кодах. Виконувати операції алгебраїчного додавання і віднімання чисел в прямих кодах можна за допомогою суматора і віднімача. Проте з метою зменшення витрат апаратури ці операції доцільно виконувати на суматорі, використовуючи додаткові або обернені коди, в які (при необхідності) перетворюються прямі коди операндів безпосередньо в АЛП. Результат операції також виходить в додатковому або оберненому коді. Якщо результат від'ємний, то виникає необхідність в преобразованні додаткового або оберненого коду результату в прямий. Хай перший доданок (зменшуване) X і другий доданок (вычитаемое) Y є двійковими n -розрядними числами з фіксованою комою, представленими в прямому коді. Тоді алгоритм додавання-віднімання чисел в прямих кодах з використанням обратних кодів полягає в наступному:

1. Якщо знаковий розряд X_0 операнда X рівний 1, то сформувати обернений код операнда X шляхом інвертування основних розрядів, в осоружному випадку залишити прямий код операнда Y без зміни.

2. Якщо знак Y_0 другого операнда і знак S_0 операції совпадають ($S = 0$ відповідає складанню, а $S_0 = 1$ - відніманню), то записати в знаковий розряд слова Y цифру 0, в осоружному випадку записати в знаковий розряд слова Y

цифру 1 і здійснити інвертування його основних розрядів.

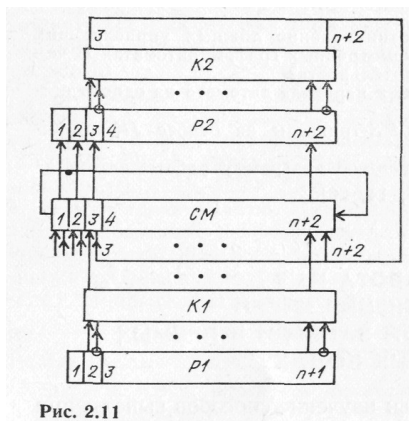


Рис. 2.11

3. Виконати підсумовування модифікованих кодів, отриманих в п. п. 1 і 2 по всіх розрядах (включаючи знакові) з циклическим перенесенням із старшого знакового розряду в молодший розряд результату.

4. Якщо в знакових розрядах результату отримана комбінація цифр 11 (отриманий від'ємний результат), то здійснити інвертування основних розрядів результату.

5. Якщо в знакових розрядах отримана комбінація цифр 00, то результат позитивний і перетворювати його не потрібен.

6. Якщо в знакових розрядах результату отримана комбінація цифр 01 або 10, то відбулося переповнювання розрядної сітки.

Структурна схема операційної частини підсумовує пристрої показана на рис. 2.11. Перед початком операції операнд Y знаходиться в регістрі $P1$, а X - в $P2$, причому знак Y_0 записаний в першому розряді $P1$, а X_0 - в другому розряді $P2$. Аналіз знакових розрядів операндів і тригера знака S_0 операції, а також формування управляючих сигналів здійснює блок управління. Мікроалгоритм операції представлений на рис. 2.12. Якщо $X_0 = 1$, то на суматор CM подається слово $11.(3... n+2)$, що формується за допомогою комутатора $K.2$. В осоружному випадку на CM видається слово $00.P2(3... n + 2)$. Одночасно з цим перевіряється умова $Y_0 = S_0$. Якщо воно виконується, то з виходів $Do.1$ на CM поступає слово $00.P1(2... n + 1)$, а якщо не виконується - слово $11.(2... n+1)$. Результат підсумовування записується в $P2$, який повинен бути побудований на тригерах з внутрішньою затримкою. Далі аналізуються 1-й і 2-й розряди $P2$. Якщо в них

записана комбінація цифр 10 або 01, то це свідчить про переповнювання розрядної сітки. Якщо ж у вказаних розрядах записані нулі - результат позитивний і не вимагає перетворень, а за наявності комбінації цифр 11 - відмінний. Для його перетворення в прямий код виконується мікро операцій інвертування $P2 := 11.P2 (3...n + 2)$. Якщо P2 не містить спеціальних ланцюгів інвертування, то цю мікро операцією можна виконати, наприклад, шляхом пересилки інформації з інверсних виходів P2 на його входи через K2 і Див. Хоча при цьому інформація проходить через СМ, тривалість інвертировання менше тривалості підсумовування, оскільки перенесення в СМ не розповсюджується.

Якщо операції додавання - віднімання алгебри чисел, представлених прямим кодом, виконувати з використанням додаткових кодів, то при отриманні від'ємного результату необхідно перетворити його додатковий код в прямий шляхом порозрядного інвертування і додавання 1 в молодший розряд. Для цього потрібне повторне підсумовування, що приводить до звеличенню часу виконання операції.

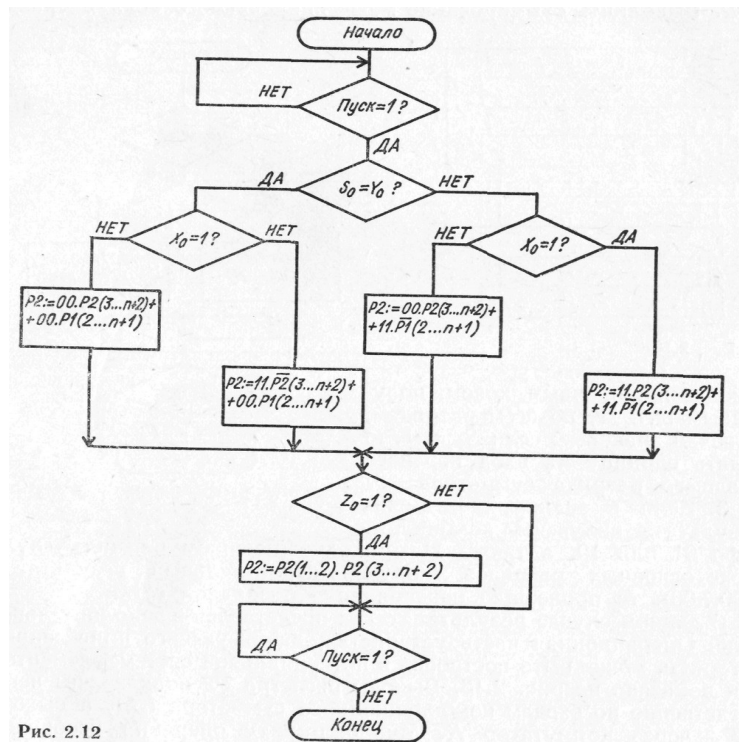


Рис. 2.12

Операції додавання алгебри і віднімання в дополнительных кодах (коли

числа в ЗУ зберігаються в додатковому коді) можна виконувати відповідно до наступного алгоритму.

1. Якщо $S_0 = 1$, то здійснити інвертування всіх розрядів (включаючи знакові) операнда V (від'ємника); в осоружному випадку залишити операнд U без змін.

2. Виконати підсумовування модифікованого коду операнда з модифікованим кодом, отриманим в п. 1, по всіх розрядах, включаючи знакові; якщо $S_0 = 1$, то подати одиницю на вхід перенесення молодшого розряду суматора.

3. Якщо в знакових розрядах результату отримані комбінації цифр 01 або 10, а також 11 при одночасній рівності нулю всіх основних розрядів (отриманий від'ємний результат - 1,00...00), то відбулося переповнювання розрядної сітки. В даному випадку результат завжди представлений в додатковому коді. Операційна частина пристрою, що реалізує приведений алгоритм, може бути побудований в потенційній системі елементів, як показано на рис. 2.13. Виходи регістра P_2 підключені непосредственно до входів комбінаційного суматора Див., а виходи P_2 - через комутатор K . Мікроалгоритм операції зображений на рис. 2.14. В даному випадку блоком управління аналізується цифра S_0 , записана в тригері знака. Якщо $S_0 = 0$, то виконується мікро операція $P_2 := P_2(2).P_2(2...n+2) + P_1(1).P_1(1...n+1)$, а при $S_0 = 1$ - мікро операція $P_2 := P_2(2).P_2(2...n+2) + P_1(1).P_1(1...n+1) + 1$ (тут 1 має вагу 2^{-n}). Результат завжди представлений в додатковому коді, тобто перетворювати його не вимагається.

У зв'язку з цим середній час додавання (віднімання) в додаткових кодах менше ніж в прямих.

В пристроях послідовного типу операції додавання і віднімання виконуються за тими ж правилами, що і в пристроях паралельного типу.

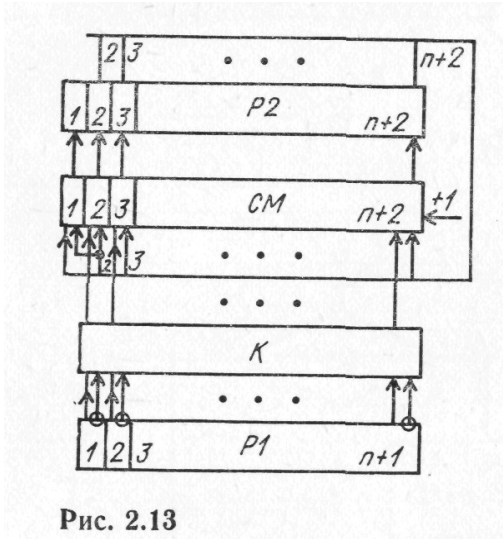


Рис. 2.13

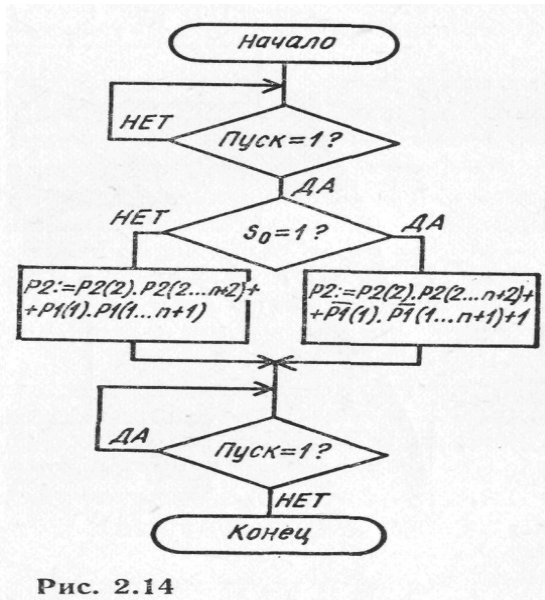


Рис. 2.14

В пристроях послідовного типу операції додавання і віднімання виконуються за тими ж правилами, що і в пристроях паралельного типу. Проте підсумовування кодів виконується послідовного в часі, починаючи з молодшими розрядами. Для реалізації операцій в прямих кодах можуть бути використані як допоміжний, так і обернені коди. У будь-якому випадку з деякою ймовірністю може вийти від'ємний результат. Для преобразовання додаткового коли від'ємного результату в прямій необхідно додати 1 в молодший розряд про інвертованого коли результату, для чого необхідно повторити цикл підсумовування.

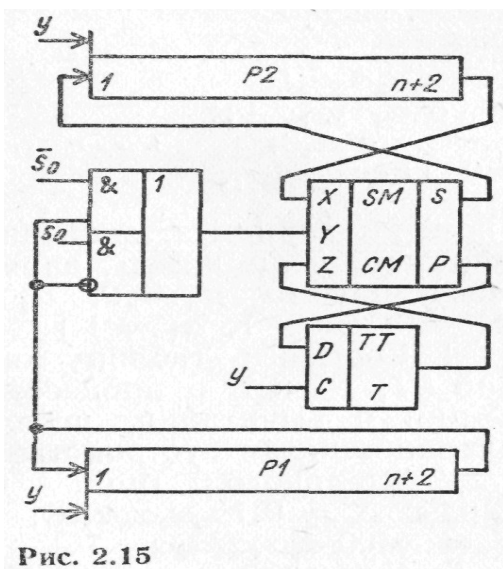


Рис. 2.15

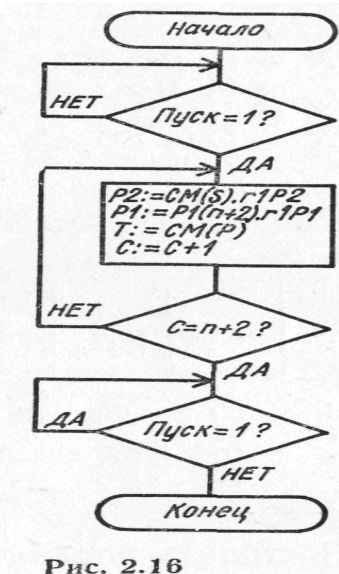


Рис. 2.16

При використуванні обернених кодів додатковий цикл

підсумовування необхідний для реалізації циклічного перенесення.

Внаслідок цього максимальний час виконання операцій в прямих кодах більше часу виконання операцій в додаткових кодах.

Структура операційної частини пристрою, що реалізовує операції додавання і віднімання в послідовних додаткових кодах, ілюструється рис. 2.15. В початковому стані операнди X і Y знаходяться в $P2$ і $P1$, а в тригері T записана цифра $S0$, оскільки при відніманні в першому такті на вхід 2, одно розрядного суматора необхідно подати одиницю. Мікроалгоритм виконання операції ілюструється рис. 2.16. Кінець операції визначається по вмісту лічильника Z , що входить до складу блоку управління.

ПІДГОТОВКА ДО РОБОТИ

1. На елементах, що є в макеті (див. п. 7), побудуйте функціональні схеми операційних пристроїв в соответствии із структурними схемами на рис. 2.11, 2.13 і 2.15 для $n = 6$. Що використовуються при виконанні операції управляючі входи вузлів позначити буквами u_i .

2. Для кожної з трьох отриманих схем визначити час виконання операції з урахуванням затримки сигналів в елементах і вузлах.

3. Виконати приклади додавання і віднімання на операційних пристроях паралельного типу для наступних значень операндів:

- 1) $X = 0, 0 \alpha_5 \alpha_4 \alpha_3 \alpha_2 \alpha_1, Y = 0, \alpha_5 \alpha_4 \alpha_3 \alpha_2 \alpha_1 1;$
- 2) $X = 0, 0 \alpha_5 \alpha_4 \alpha_3 \alpha_2 \alpha_1, Y = -0, \alpha_5 \alpha_4 \alpha_3 \alpha_2 \alpha_1 0;$
- 3) $X = -0, 0 \alpha_5 \alpha_4 \alpha_3 \alpha_2 \alpha_1, Y = 0, \alpha_5 \alpha_4 \alpha_3 \alpha_2 \alpha_1 0;$
- 4) $X = -0, 1 \alpha_5 \alpha_4 \alpha_3 \alpha_2 \alpha_1, Y = -0, 11 \alpha_4 \alpha_3 \alpha_2 \alpha_1,$

де $\alpha_5 \alpha_4 \alpha_3 \alpha_2 \alpha_1$ - останні цифри номера варіанту, записаного в двійковій системі числення. (Наприклад, якщо номер варіанту рівний 11, то $\alpha_5 = 0, \alpha_4 = 1, \alpha_3 = 0, \alpha_2 = 1, \alpha_1 = 1$.)

4. Скласти таблицю состояний $P1, P2$ і T в процесі додавання і віднімання чисел на операційному пристрої послідовного типу. Операнди $X = 0, 1 \alpha_5 \alpha_4 \alpha_3 \alpha_2 \alpha_1; Y = -0, 0 \alpha_5 \alpha_4 \alpha_3 \alpha_2 \alpha_1.$

5. Для кожного пристрою розробити закодований мікроалгоритм.

6. Побудувати управляючий автомат і тимчасову діаграму його роботи для одного операційного пристрою. Початкові дані виявляються номером бригади відповідно до табл. 2.8.

Таблиця 2.8

α_4	Операцій- ное устрой- ство	α_3	α_2	Тип тригг- геров	α_1	Тип автоматов
0	I	00		T	0	Мура
1	II	01		D	1	Миля
—	—	10		JK	—	—
—	—	11		DC	—	—

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Перевірити справність елементів макета, необхідних для побудови досліджуваних схем.

2. Зібрати і відлагодити схеми операційних пристроїв (останньою збирається схема, для якої розроблений управляючий автомат). Виконати на них операції додавання і віднімання в по-шаговому режимі з операндами, вказаними в пунктах 3 і 4 завдання. Управляючі сигнали при цьому формують за допомогою кнопок.

3. Підключити входи і виходи управляючого автомата до відповідаючих виходів і входів операційного пристрою і виконати операції додавання і віднімання в автоматичному режимі. Зміст звіту. Звіт повинен містити короткі теоретичні відомості, всі схеми, таблиці, графіки і формули, отримані в процесі підготовки і виконання роботи, а також висновки по роботі.

КОНТРОЛЬНІ ПИТАННЯ

1. Якими чинниками визначається розділення операцій на операції в прямих і додаткових кодах?

2. За допомогою яких засобів можна реалізувати додавання (віднімання) в прямих (додаткових) кодах?

3. Опишіть алгоритм додавання-віднімання чисел в прямих (додаткових) кодах.

4. Як класифікуються операційні пристрої в залежності від способу обміну інформацією між вузлами?

5. Які мікро операції виконуються в операційному пристрої при складанні (відніманні) чисел в прямих (додаткових) кодах?

6. В яких кодах (прямих або додаткових) операції виконуються швидше і чому?

7. В скільки разів пристрої послідовного типу, що підсумовують, поступаються в швидкодії пристроям паралельного типу?

8. Як визначите тривалість виконання мікрооперацій?

9. Охарактеризуйте призначення управляючих входів вузлів, що використовуються для побудови операційних пристроїв.

10. Визначити необхідні значення сигналів на управляючих входах вузлів при виконанні різних мікрооперацій (на прикладі конкретної схеми).

11. Охарактеризуйте етапи побудови блоків управління операційними пристроями.

12. Складіть таблицю переходів JK-, RS-, T- і D- тригерів.

13. Які тригери (по внутрішній організації) можна використовувати для побудови регістрів операційних пристроїв?

14. Охарактеризуйте особливості синтезу автоматів Милі і Мура.

Література: [4, 10, 14, 20, 23, 25].

РЕАЛІЗАЦІЯ МНОЖЕННЯ ЧИСЕЛ У ПРЯМИХ КОДАХ

Ціль роботи: вивчення нових методів множення чисел у прямих кодах та способів їх апаратної реалізації, отримання навиків відладки операційних пристроїв.

ОСНОВНІ ПОЛОЖЕННЯ.

При множенні чисел в прямих кодах знакові та основні розряди обробляються окремо. Для визначення знака результату існує додавання по модулю два цифр, записаних в знакових розрядах множників. Вважатимемо, що множене Y та множник X – правильні двійкові дроби виду

$$X=0, x_1x_2\dots x_n \text{ та } Y=0, y_1y_2\dots y_n, \text{ де } x_i, y_i \in \{0,1\}.$$

Тоді добуток Z модулей чисел дорівнює:

$$Z = YX = Yx_12^{-1} + Yx_22^{-2} + \dots + Yx_i2^{-i} + \dots + Yx_n2^{-n} \quad (8)$$

Множення Y на X може бути реалізовано шляхом виконання деякого циклічного процесу, характер якого залежить від контекстної форми виразу (8). Один цикл множення складається з додавання чергового часткового добутку, який є добутком множеного на одну цифру множника, до суми часткових добутків та взаємного зсуву множеного та суми часткових добутків. Розрізняють чотири основних способи множення.

Перший спосіб множення. Вираз (8) можна представити у вигляді

$$Z = YX = (\dots((0 + Yx_n)2^{-1} + Yx_{n-1}2^{-1} + \dots + Yx_i)2^{-1} + \dots + Yx_1)2^{-1}$$

Звідки видно, що отримання суми Z часткових добутків у i -му циклі ($i = \overline{1, n}$)

$$Z = (Z^{i-1} + Yx_{n-i+1})2^{-1}$$

зводиться к обчисленню з початковими значеннями $t = 1, Z_0 = 0$, при чому $Z_n - Z = YX$. Множення виконується з молодших розрядів множника, сума часткових добутків зсувається вправо, а множене лишається нерухомим.

Другий спосіб множення. Запишемо (8) у вигляді

$$Z = (\dots((0 + Y2^{-n}x_n) + Y2^{-n+1}x_{n-1}) + \dots + Y2^{-1}x_1)$$

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

$$Z_i = Z_{i+1} + Yx_{n-i+1}; Y_i = 2Y_{i-1}, \text{ з початковими значеннями } i = 1, Y_0 = Y2^{-n}, Z_0 = 0.$$

Множення виконується з молодших розрядів, множене зсувається вліво, а сума часткових добутоків лишається нерухомою.

Третій спосіб множення. Представимо (8) у вигляді

$$Z = (\dots((0 + Y2^{-n}x_1)2 + Y2^{-n}x_2)2 + \dots + Y2^{-n}x_i)2 + \dots + Y2^{-n}x_n$$

Звідки видно, що суму, часткових добутоків у n-му циклі ($i = \overline{1, n}$) можна отримати по формулі

$$Z_i = 2Z_{i-1} + Y2^{-n}x_i.$$

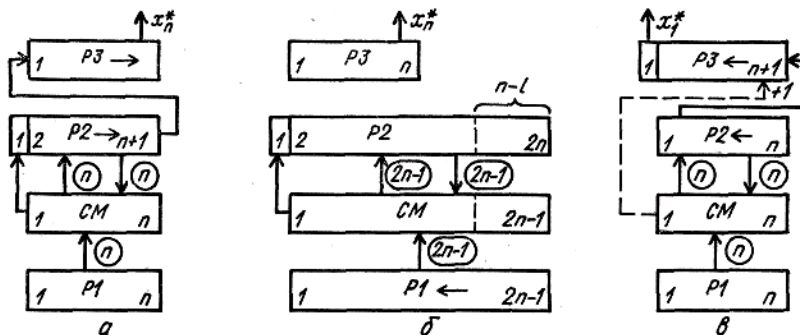
Початковими значеннями являються $i = 1, Z_0 = 0$. Множення виконується зі старших розрядів множника, сума часткових добутоків зсувається вліво, а множене нерухомим.

Четвертий спосіб множення. Вираз (8) можна визначити як:

$$Z = (0 \dots ((0 + Y2^{-1}x_1) + Y2^{-2}x_2) + \dots + Y2^{-i}x_i) + \dots Y2^{-n}x_n.$$

Процес множення може бути зведений до n-кратного виконання цикла $Z_i = Z_{i-1} + Y_{i-1}x_i, Y_i = Y_{i-1}2^{-1}$ з початковими значеннями $i = 1, Y_0 = Y2^{-1}, Z_0 = 0$.

Множення виконується зі старших розрядів множника, а сума часткових добутоків лишається рухомою, а множене зсувається вправо.



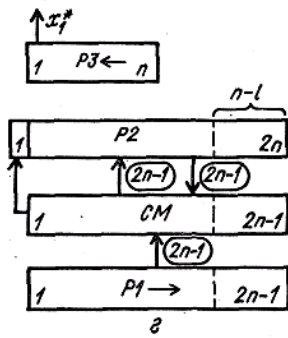


Рис. 2.17

Для формування і збереження суми часткових добутоків можна використовувати комбінаційний суматор та регістр добутку, або лише накопичуючий суматор, який у функціональному відношенні можна розглядати як композицію комбінаційного суматора та регістра. Принцип побудови пристроїв, що реалізують різні способи множення, пояснено на рис. 2.17, де $P1$ - регістр множеного; $P2$ - регістр добутку; $P3$ - регістр множника. Цифрами вказані номери порядків CM та регістрів, а стрілками вказано напрям зсуву кодів в регістрах. До входу старшого розряду регістра $P2$ на рис. 2.17, *a*, *б*, *в* підключено вихід переносу P старшого розряду CM . Цифри, записані в молодших розрядах $P1$ та $P2$, при реалізації першого способу мають вагу 2 , а при реалізації інших способів - $\bar{2}^{2n}$. Перед початком множення за будь-яким з методів $P2$ встановлюється в нульовий стан.

При множенні за першим методом (рис. 2.17, *a*) в першому такті i -го циклу аналізується значення x_n^* молодшого $(n-2i)$ розряду $P3$, в знаходиться чергова цифра x_{n-i+i} множника. Вміст $P1$ додається до суми часткових добутоків, що знаходяться у $P2$, якщо $x^* = 1$, чи не додається, якщо $x^* = 0$. У другому такті виконується зсув вправо у $P3$ та $P2$, що еквівалентно множенню вмістимого на 2^{-1} . При зсуві цифра молодшого розряду $P2$ переписується у старший розряд $P3$, який звільнився. Після виконання n циклів молодші розряди $2n$ -розрядного добутку будуть записані в $P3$, а старші - в $P2$.

Час множення, якщо не використовуються засоби прискорення операції, визначаються виразом $t_y = n(t_+ + tc)$, де t_+ та tc - тривалість тактів додавання та

зсуву відповідно.

Перед початком множення по другому способу (рис. 2.17, б), X записують у $P3$, а Y - в молодші розряди $P1$ (тобто у $P1$ встановлюють $Y_0 = Y \cdot 2^{-n}$). В кожному n -му циклі множення додаванням кодів $P1$ та $P2$ керує цифра x^* , а в $P1$ виконується зсув вліво на один розряд, в результаті чого формується величина $Y_i = 2Y_{i-1}$. Оскільки сума часткових добутоків в процесі множення нерухома, то зсув у $P1$ можна поєднати у часі з додаванням (як правило, $t_+ > tc$). В цьому випадку $t_y = nt_+$. Множення можна закінчувати за нульовим вмістом $P3$, що також призводить до збільшення швидкодії, якщо множник не нормалізовано.

При множенні по третьому способу. (рис. 2.17, б) вага молодшого розряду $P1$ дорівнює 2^{-2n} , тому код у $P1$ являє собою значення $Y2$. На початку кожного циклу множення виконується зсув вліво у $P2$ та $P3$, а далі виконується додавання, яким керує x^* . В результаті додавання вмісту $P1$ та $P2$ може виникнути перенос до молодшого розряду $P3$, який повинен мати кола для його додавання. Збільшення довжини $P3$ на один розряд виключає можливість поширення переносу в розряді множника. Після виконання n циклів молодші розряди добутку будуть знаходитись в $P2$, а старші - в $P3$. Час множення по третьому способу такий же, як і по першому.

Перед множенням по четвертому способу (рис. 2.17, з) множник записують в $P3$, а множене - в старші розряди $P1$ (тобто в $P1$ встановлюють $Y_0 = Y2^{-1}$). В кожному циклі цифра x^* , що знаходиться в старшому розряді $P3$, керує додаванням, а в $P1$ виконується перший зсув на один розряд, що еквівалентно множенню вмісту цього регістра на 2^{-1} . Час виконання множення за четвертим способом складає $t_y = nt_+$. В якості прикладу на рис. 2.18 і рис. 2.19 приведені мікроалгоритми множення чисел по третьому и четвертому способам. Для підрахунку циклів у пристроях використовуються лічильники C .

У ЕОМ часто необхідно обчислювати не $2n$, а тільки $(n + 1)$ цифр добутку й округлювати це до n цифр. В цьому випадку при реалізації другого способу можна зменшити довжину CM і $P2$, а при реалізації четвертого - зменшити

довжину CM , $P2$ й $P1$. Для того щоб похибка від відкидання молодших розрядів не перевищувала половини ваги n -го розряду результату, у перерахованих вузлах достатньо мати лише по l додаткових розрядів, де вибирається з умови : $l \geq 1 + \log_2(n - l - 1)$.

Операція округлення виконується зазвичай шляхом додавання одиниці до $(n + 1)$ -му розряду результату й відкидання всіх розрядів, що знаходяться правіше n -го. При цьому похибка стає знаковмінною, а максимальне абсолютне її значення не перевищує половини ваги молодшого розряду. Додаткового такту додавання для округлення не потрібно. Для цього достатньо записати одиницю перед початком множення в той розряд, $P2$, який після виконання множення залишається старшим розрядом, що відкидається.

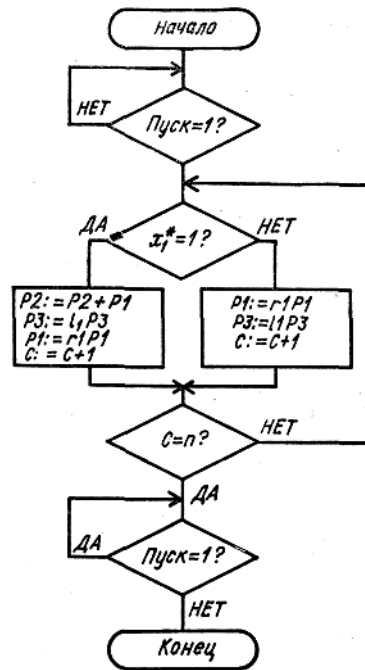
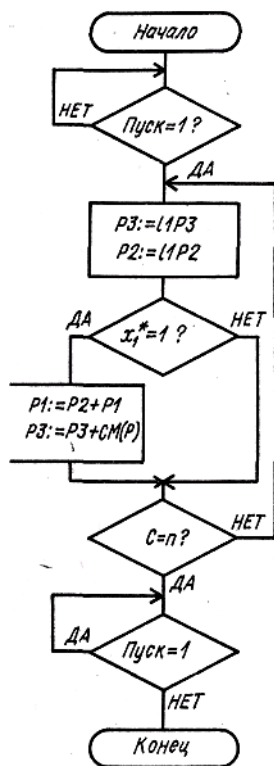


Рис. 2.19

Рис. 2.18

В процесі формування суми часткових добутоків код з $P2$ подається на CM , з виходів CM знову записується в $P2$. В зв'язку з цим при використанні потенційних елементів, $P2$ будують на триггерах із внутрішньою затримкою. Характер керуючих сигналів у колі, на які вони діють, визначаються

конкретною реалізацією вузлів та елементною базою.

У пристроях, що реалізують другий та четвертий способи множення, можна без пересилання кодів між регістрами обчислювати вираз вигляду

$$\sum_{i=1}^N X_i Y_i,$$

для чого достатньо черговий результат операції залишати в $P2$, який в цьому випадку повинен мати додаткові старші розряди.

В пристрої, що реалізує третій спосіб, можна без пересилок обчислювати, наприклад, функції вигляду X_i . Для цього X перед початком обчислення записується в $P1$ та в молодші розряди $P3$, а далі $(i - 1)$ раз виконується операція множення з округленням проміжних результатів до n розрядів. Після кожної чергової операції $P2$ встановлюється в нульовий стан. Кінцевий результат знаходиться в n молодших розрядах $P3$. Найбільш простими є пристрої, що реалізують перший спосіб, а найбільш швидкодіючими – другий і четвертий. Однак другий спосіб не має особливих переваг порівняно з четвертим і, окрім того, вимагає великих апаратних витрат.

ПІДГОТОВКА ДО РОБОТИ

1. Побудувати функціональні схеми операційних пристроїв для реалізації кожного з чотирьох способів множення. Довжина операндів вказана в табл. 2.9, де a_i – цифри номера варіанта, представленого п'ятирозрядним двійковим словом $a_6 a_4 a_8 a_2 a_1$. Схеми мають бути побудовані на елементах в макеті (гл. 2). Керуючі входи вузлів, використовувані при виконанні операції, позначити буквами u .

2. Побудувати закодовані мікро алгоритми множення чисел чотирма способами.

3. Для кожного способу множення визначити час виконання операції з урахуванням затримки в елементах і вузлах схеми, побудувати часові діаграми керуючих сигналів для кожної комбінації значень логічних умов.

4. Скласти таблиці станів регістрів в процесі множення для кожного операційного пристрою. Операнди приведені в табл. 2.9.

5. У відповідності з табл. 2.10 побудувати керуючий автомат для одного операційного пристрою. Цифрами I, II, III и IV позначені пристрої, що реалізують множення першим, другим, третім та четвертим способами відповідно.

Таблиця 2.9

$\alpha_2 \alpha_1$	Разрядность операндов	Операнды X и Y
0 0	8	$0,10\alpha_5\alpha_4\alpha_3\alpha_2\alpha_11$ $0,11\alpha_1\alpha_2\alpha_3\alpha_4\alpha_50$
0 1	7	$0,1\alpha_1\alpha_2\alpha_3\alpha_4\alpha_51$ $0,0\alpha_5\alpha_4\alpha_3\alpha_2\alpha_11$
1 0	6	$0,1\alpha_5\alpha_4\alpha_3\alpha_2\alpha_1$ $0, \alpha_1\alpha_2\alpha_3\alpha_4\alpha_51$
1 1	5	$0, \alpha_1\alpha_2\alpha_3\alpha_4\alpha_5$ $0, \alpha_5\alpha_4\alpha_3\alpha_2\alpha_1$

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Перевірити справність елементів макету.
2. Зібрати та відлагодити схеми операційних пристроїв (вказані викладачем). Виконати множення чисел, вказаних в табл. 2.9, у пошаговому режимі.
3. Зібрати схему керуючого автомата, отриманого при виконанні теоретичного завдання, відлагодити її в статичному режимі.
4. Підключити входи керуючого автомата до відповідних виходів операційного пристрою та виконати множення в автоматичному режимі.

Таблиця 2.10

$\alpha_3 \alpha_2$	Операционное устройство	$\alpha_5 \alpha_4$	Тип триггеров	α_1	Тип автомата
0 0	I	0 0	D	0	Мили
0 1	II	0 1	T	1	Мура
1 0	III	1 0	RS	—	—
1 1	IV	1 1	JK	—	—

КОНТРОЛЬНІ ПИТАННЯ

1. Запишіть формули обчислення суми часткових добутоків для кожного

способу множення.

2. Які мікрооперації використовуються при виконанні множення? На які керуючі входи вузлів необхідно подавати сигнали при виконанні цих сигналів?

3. Як визначити необхідну тривалість керуючих сигналів при виконанні мікро операцій?

4. В яких пристроях множення мікрооперації додавання та зсуву можна виконувати одночасно? Чому можливе таке поєднання?

5. Який пристрій множення є найбільш економічним (найбільш швидкодіючим)?

6. Охарактеризуйте призначення керуючих входів вузлів, що використовуються для побудови операційних пристроїв?

7. Як побудувати закодований мікроалгоритм операції?

8. Охарактеризуйте етапи синтезу керуючого автомата?

9. Чим відрізняються автомати Мура від автоматів Мілі?

10. Які особливості повинен мати регістр $P3$ пристрою, що реалізує множення третім способом?

11. Як визначити число додаткових молодших розрядів регістрів $P1$ та $P2$ при множенні четвертим способом?

12. Як виконується округлення результату множення?

13. Яким чином можна підрахувати суму добутків чисел в пристроях, що реалізують другий та четвертий спосіб множення?

14. Як за допомогою пристрою, що реалізує третій спосіб множення, обчислити ступеневу функцію?

15. Яким чином можна отримати сигнал, що свідчить про закінчення множення в кожному з чотирьох операційних пристроїв?

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Глушков В.М Синтез цифровых автоматов. - М.: ФМЛ, 1962
2. Капитонова Ю.В., Летичевский А.А. Математическая теория проектирования вычислительных систем. - М.: Наука, 1988. – 295 с.
3. Самофалов К.Г., Корнейчук В.Н., Тарасенко В.П. Цифровые ЭВМ. - К.: Вища шк., 1989. – 423 с.
4. Самофалов К.Г., Корнейчук В.Н., Тарасенко В.П., Жабин В.Н. Цифровые ЭВМ. Практикум. - К.: Вища шк., 1990. - 215с.
5. Майоров С.А., Новиков Г.И. Принципы организации цифровых машин. - Л.: Машиностроение, 1974. – 432 с.
6. Баранов С.И. Синтез микропрограммных автоматов. - Л.: Энергия, 1979.- 232с.