

Лабораторна робота №1

Ціль роботи: Вивчити основні методи множення чисел у прямих кодах і способи їх апаратної реалізації, одержати навички в проектуванні й налагодженні схем управління операційними пристроями з розподіленою логікою.

Основні способи множення чисел у прямих кодах

Принцип побудови пристроїв, що реалізують різні способи множення, показаний на рис. 2.1, де $RG3$ – регістр множеного, $RG1$ – регістр добутку, $RG2$ – регістр множника. Цифрами зазначені номери розрядів SM і регістрів, а стрілками показаний напрямок зсуву кодів у регістрах.

Цифри, що записані в молодших розрядах регістрів $RG3$ і $RG1$, при реалізації першого способу мають вагу 2^{-n} , а при реалізації інших способів – 2^{-2n} . Перед початком множення будь-яким способом регістр $RG1$ встановлюється в нульовий стан. Підрахунок кількості циклів множення забезпечують лічильники CT , відповідно з чим обирається його розрядність q .

Під час множення *першим способом* (рис. 2.1, а) в першому такті i -го циклу аналізується значення $RG2(n)$ – молодшого (n -го) розряду регістру $RG2$, в якому знаходиться чергова цифра множника. Вміст $RG3$ додається до суми часткових добутків, що знаходяться в регістрі $RG1$, якщо $RG2(n)=1$, або не додається, якщо $RG2(n)=0$. В другому такті здійснюється правий зсув в регістрах $RG1$ і $RG2$, що еквівалентно множенню їх вмісту на 2^{-1} . При зсуві цифра молодшого розряду регістру $RG1$ записується у вивільнюваний старший розряд регістру $RG2$. Після виконання n циклів молодші розряди $2n$ -розрядного добутку будуть записані в регістр $RG2$, а старші – у $RG1$.

Час множення, якщо не застосовуються методи прискорення операції, визначається виразом $t_m = n(t_n + t_3)$, де t_n і t_3 – тривалості тактів підсумовування і зсуву відповідно.

Перед початком множення *другим способом* (рис. 2.1, б) множник X записують в регістр $RG2$, а множене Y – в молодші розряди регістру $RG3$ (тобто в регістрі $RG3$ установлюють $Y_0 = Y \cdot 2^{-n}$). В кожному i -му циклі множення додаванням кодів $RG3$ і $RG1$ управляє цифра $RG2(n)$, а в регістрі $RG3$ здійснюється зсув вліво на один розряд, в результаті чого формується величина $Y_i = 2Y_{i-1}$. Оскільки сума часткових добутків в процесі множення нерухома, зсув в регістрі $RG3$ можна сполучити в часі з підсумовуванням (як правило, $t_n \geq t_3$). В цьому випадку $t_m = nt_n$. Завершення операції множення визначається за нульовим вмістом регістру $RG2$, що також приводить до збільшення швидкодії, якщо множник ненормалізований.

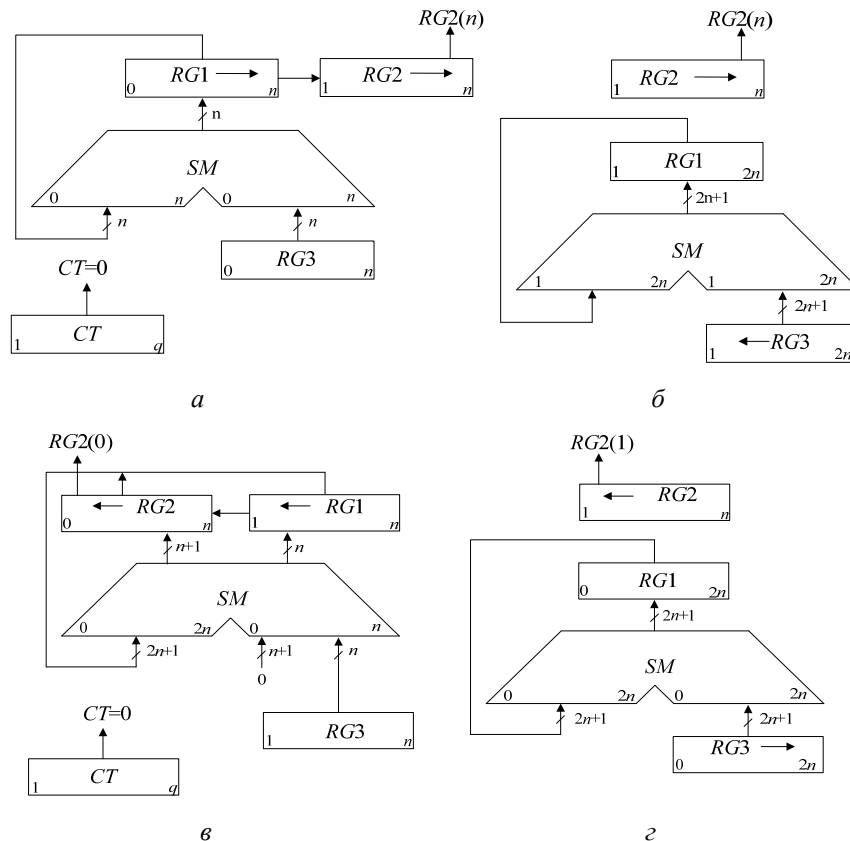


Рис. 2.1. Операційні схеми пристроїв для множення чисел:
а – перший спосіб; б – другий спосіб; в – третій спосіб; г – четвертий спосіб

Під час множення *третім способом* (рис. 2.1, в) вага молодшого розряду $RG3$ дорівнює 2^{-2n} , тому код в регістрі $RG3$ являє собою значення $Y2^{-n}$. На початку кожного циклу множення здійснюється лівий зсув в регістрах $RG1$ і $RG2$, а потім виконується додавання, яким управляє $RG2(1)$. В результаті підсумовування вмісту $RG3$ і $RG1$ може виникнути перенос в молодший розряд регістру $RG2$. У старшій частині суматора, на якому здійснюється підсумовування коду $RG2$ з нулями, відбувається поширення переносу. Збільшення довжини $RG2$ на один розряд усуває можливість поширення переносу в розряди множника. Після виконання n циклів молодші розряди добутку будуть знаходитися в регістрі $RG1$, а старші – в регістрі $RG2$. Час множення третім способом визначається аналогічно першому способу і дорівнює $t_m = n(t_n + t_3)$.

Перед множенням *четвертим способом* (рис. 2.1, г) множник записують в регістр $RG2$, а множене – в старші розряди регістру $RG3$ (тобто в $RG3$ установлюють $Y_0=Y2^{-1}$). В кожному циклі цифра $RG2(1)$, що знаходиться в старшому розряді регістру $RG2$, управляє підсумовуванням, а в $RG3$ здійснюється правий зсув на один розряд, що еквівалентно множенню вмісту цього регістра на 2^{-1} . Час виконання множення четвертим способом складає $t_m = nt_n$, визначається аналогічно другому способу.

В ЕОМ при роботі з дробовими числами часто потрібно обчислювати не $2n$, а тільки $(n+1)$ цифр добутку й округляти його до n розрядів. В цьому випадку при реалізації другого способу можна зменшити довжину SM і $RG1$, а при реалізації четвертого – зменшити довжину SM , $RG1$ і $RG3$. Для того, щоб похибка від відкидання молодших розрядів не перевищила половини ваги n -го розряду результату, в перерахованих вузлах досить мати тільки по l додаткових молодших розрядів, де l вибирається з умови

$$l \geq 1 + \log_2(n - l - 1).$$

Операція округлення здійснюється звичайно шляхом додавання одиниці до $n+1$ -го розряду результату і відкидання всіх розрядів, розташованих правіше n -го. При цьому похибка стає знакозмінною, а максимальне абсолютне її значення не перевищує половини ваги молодшого розряду. Додаткового такту підсумовування для округлення не потрібно. Досить записати одиницю перед початком множення в той розряд регістру $RG1$, що після виконання множення залишається старшим розрядом, який відкидається.

У процесі формування суми часткових добутків код з регістру $RG1$ видається на суматор SM , а з виходів SM знову записується в регістр $RG1$. У зв'язку з цим при використанні потенційних елементів регістр $RG1$ будують на тригерах із внутрішньою затримкою. Характер управляючих сигналів і ланцюга, на який вони впливають, визначається конкретно теоретичною реалізацією вузлів і використовуваною елементною базою.

У операційних пристроях, що реалізують другий і четвертий способи множення, можна без пересилань кодів між регістрами обчислювати вирази вигляду $\sum X_i Y_i$, де $(i = \overline{1, n})$ для чого досить черговий результат операції залишати в регістрі $RG2$, який в цьому випадку повинен мати додаткові старші розряди.

У операційному пристрої, що реалізує третій спосіб, можна без пересилань обчислювати, наприклад, функції вигляду X_i . Для цього множник X перед початком обчислення записується в регістр $RG3$ і в молодші розряди регістру $RG2$, а потім $(i-1)$ раз виконується операція множення з округленням проміжних результатів до n розрядів. Після кожної чергової операції регістр $RG1$ встановлюється в нульовий стан. Остаточний результат буде знаходитися в n молодших розрядах регістру $RG2$. 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, крім того, вимагає великих апаратних витрат при реалізації.

АЛП з розподіленою логікою застосовуються в спеціалізованих та проблемно-орієнтованих ЕОМ. Відрізняються від АЛП інших типів високою швидкодією, але мають досить обмежені функціональні можливості. Структура таких АЛП залежить від операцій, що вони виконують, причому для кожної системи операцій необхідно будувати окремий АЛП.

АЛП з розподіленою логікою складаються з двох функціональних частин (рис. 2.2):

- управляючий пристрій, що забезпечує формування всіх управляючих сигналів;
- операційний пристрій, забезпечує перетворення інформації та виконує мікрооперації над машинними словами.

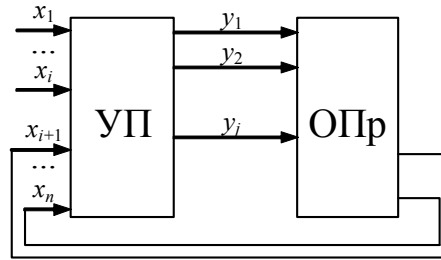


Рис. 2.2. Загальна структура АЛП

Побудова таких АЛП відбувається за наступними етапами:

1. Для кожної операції будується операційна схема та функціональний мікроалгоритм (Ф-мікроалгоритм). Рекомендується обирати такі мікроалгоритми виконання операцій, що краще сполучаються, тобто вимагають однакового напрямку зсувів в регістрах, однакову розрядність регістрів, одні й ті самі джерела операндів суматорів і таке інше.
2. Обирається розрядність регістрів, лічильників. Виконується логічне моделювання роботи ОПр, наприклад, із застосуванням діаграми стану регістрів при виконанні МА з критичними значеннями операндів.
3. Розробляється функціональна та принципова схеми ОПр із зазначенням управляючих сигналів для кожного вузла пристрою.
4. Складається закодований структурний мікро алгоритм (С-мікроалгоритм) виконання заданих операцій.
5. Виконується синтез управляючого пристрою.
6. Складається функціональна та принципова схеми АЛП.

Приклад 2.1. Побудувати схему АЛП для реалізації операції множення чисел за першим способом.

Синтезувати схему, що дозволяє обчислити добуток $Z=Y \times X$ двох правильних дробів $Y = 0, y_1, y_2 \dots y_n$ та $X = 0, x_1, x_2 \dots x_n$. Вважати, що розрядність дробів $n = 16$.

Виконання завдання

Операційна схема, що реалізує перший спосіб множення, подана на рис. 2.3, де $RG1$ – регістр накопичення суми часткових добутків, $RG2$ – регістр множника, $RG3$ – регістр множеного, $RG4$ (CT) – лічильник циклів, TC – тригер переносу, SM – комбінаційний суматор. Регістри $RG1$ та $RG2$ реалізують мікрооперації зсуву, лічильник $RG4$ дозволяє формувати ознаку нуля – що визначає закінчення обчислення добутку. За нульовим вмістом регістру $RG4$ результат обчислення формується в регістрах $RG1$ та $RG2$.

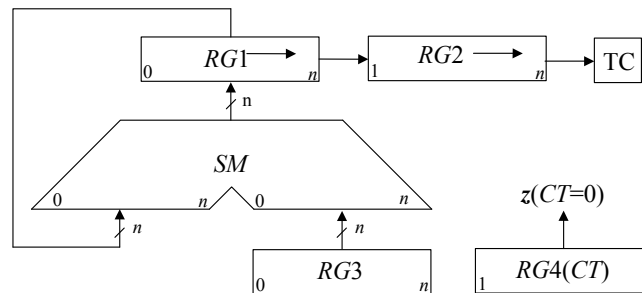


Рис. 2.3. Операційна схема множення

Зауваження. Операційні схеми застосовують для відображення апаратури, що застосовується для виконання послідовності заданих мікрооперацій. ОС містить всі функціональні частини операційного пристрою із зазначенням зв'язків між ними. За ОС виконання операції будують структурну схему ОПр.

Для розробленої операційної схеми побудуємо Ф-мікроалгоритм. Припустимо, що ОПр входить до складу АЛП із централізованим управлінням, отже робота цього блоку розпочинається із надходження сигналу “Пуск” від центрального блоку управління. Функціональний мікроалгоритм зображений на рис. 2.4, де TC – стан тригера переносу, z – значення ознаки нуля в лічильнику циклів $RG4$.

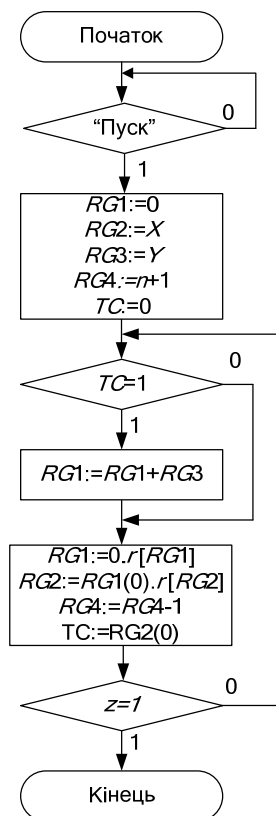


Рис. 2.4. Ф-мікроалгоритм множення чисел

Зауваження. Мікроалгоритми можна розглядати на функціональному та структурному рівнях. На *функціональному* рівні розглядають узагальнені МО, які не суперечать операційній схемі пристрою. При цьому можна не враховувати кількість тактів, необхідних для виконання МО. На *структурному* рівні операційна вершина відповідає одному такту перетворення інформації. С-мікроалгоритми повністю відповідають схемі пристрою з урахуванням елементної бази та тривалості управляючих сигналів. Для побудови С-мікроалгоритму необхідно отримати перелік МО в АЛП, що розробляється.

Логічне моделювання потактової роботи ОПр приведене в табл. 2.1

Значення операндів:

$$Y = 5_{10} = 0101_2;$$

$$X = 7_{10} = 0111_2;$$

$$Z = 35_{10} = 00100011_2.$$

Розрядність дробів $n = 4$.

Таблиця 2.1. Логічне моделювання роботи ОПр

№ такту	RG1	RG2	TC	RG3	RG4	z	МО
ПС	0000	0101	0	0111	0101	0	Початковий стан
1	0000	0010	1	0111	0100	0	$RG1 \rightarrow, RG2 \rightarrow,$ $RG4 - 1; z = 0$
2	0000 <u>+0111</u> 0111 0011	0010 1001	1 0	0111 0111	0100 0011	0 0	$RG1 + RG3$ $RG1 \rightarrow, RG2 \rightarrow,$ $RG4 - 1; z = 0$
3	0001	1100	1	0111	0010	0	$RG1 \rightarrow, RG2 \rightarrow,$ $RG4 - 1; z = 0$
4	0001 <u>+0111</u> 1000 0100	1100 0110	1 0	0111 0111	0010 0001	0 0	$RG1 + RG3$ $RG1 \rightarrow, RG2 \rightarrow,$ $RG4 - 1; z = 0$
5	0010	0011	0	0111	0000	1	$RG1 \rightarrow, RG2 \rightarrow,$ $RG4 - 1; z = 1$

На підставі ОС множення та Ф-мікроалгоритму складемо перелік управляючих сигналів для всіх функціональних частин ОПр та побудуємо функціональну схему.

Перелік управляючих сигналів наведений в табл. 2.2, функціональна схема ОПр зображена на рис. 2.5.

Таблиця 2.2. Таблиця управляючих сигналів

Елемент	Мікрооперація	Управляючий сигнал
RG1	Скидання	R
	Запис	W
	Зсув вправо	SR
	Заповнення старшого розряду при зсуві вправо	DR
RG2	Запис	W
	Зсув вправо	SR
	Старший розряд при зсуві вправо	DR
RG3	Запис	W
RG4	Запис	W
	Декремент лічильника	dec
TC	Скидання	R
	Запис молодшого розряду множника у тригер переносу	C

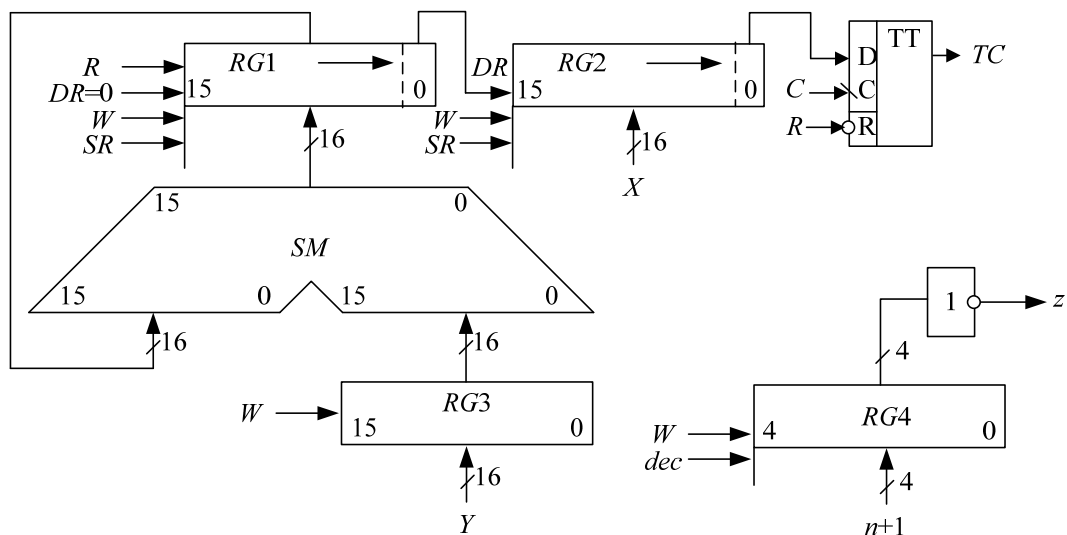


Рис. 2.5. Функціональна схема операційного пристрою

За побудованою функціональною схемою будуємо функціонально-структурний мікроалгоритм (ФС-мікроалгоритм), що зображений на рис 2.6. Індекс указує до якої з функціональних частин пристрою множення належить управляючий сигнал.

Кодування сигналів управління та логічних умов наведене в табл. 2.3 – 2.4.

Для забезпечення перепаду сигналів управління SR_1 , SR_2 , dec , C_{TC} (вершину з цими сигналами охоплює петля рис. 2.6) необхідно ввести порожню додаткову вершину.

Закодований ФС-мікроалгоритм зображений на рис. 2.7, де управляючі сигнали та сигнали логічних умов відповідають рис. 2.6 та табл. 2.2 – 2.4.

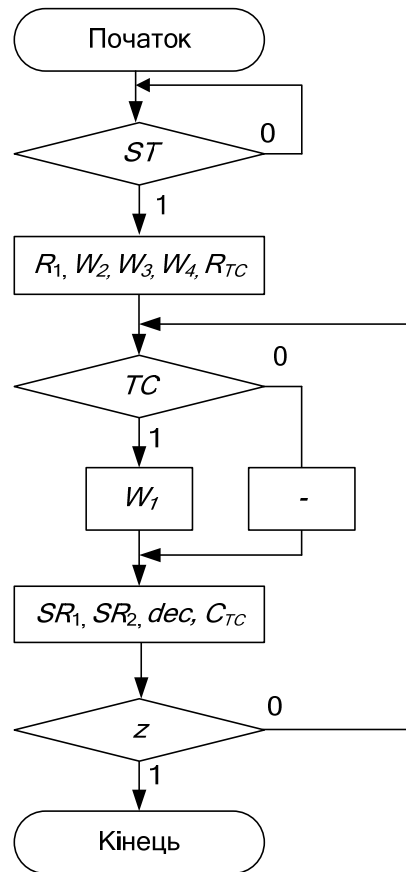


Рис. 2.6. Функціонально-структурний мікроалгоритм

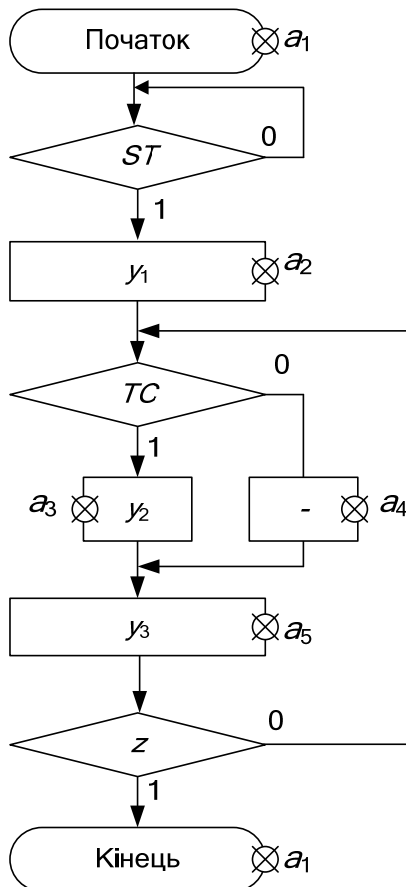


Рис. 2.7. Закодований функціонально-структурний мікроалгоритм

Таблиця 2.3. Кодування сигналів управління

Управляючі сигнали	Код
R_1	y_1
W_2	
W_3	
W_4	
R_{TC}	
W_1	y_2
SR_1	y_3
SR_2	
C_{TC}	
dec	

Отриманий закодований ФС-мікроалгоритм є вихідним для здійснення синтезу управляючого пристрою.

Таблиця 2.4. Кодування логічних умов

Логічні умови	Код
Пуск	ST
Аналіз молодшого розряду множника	TC
Нульовий вміст лічильника	z

Для управління роботою ОПр застосуємо *пристрій управління з жорсткою логікою*, який реалізуємо у вигляді цифрового автомата Мура.

Розмітка ФС-мікроалгоритма для автомата Мура наведена на рис. 2.7. Стани автомата позначені символами a_i . Часова діаграма роботи управляючого пристрою зображена на рис. 2.8. Часова діаграма відповідає потактовій роботі ОПр для прикладу, виконаного в табл. 2.1.

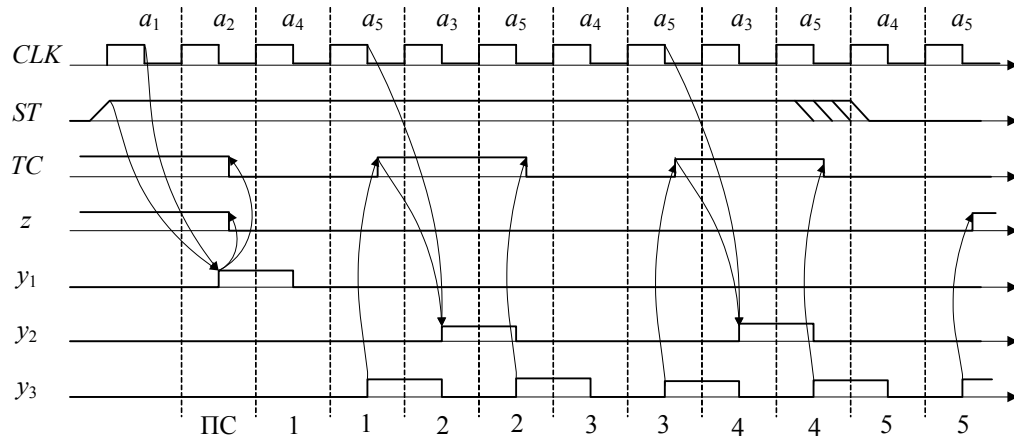


Рис. 2.8. Часова діаграма роботи пристрою управління

На рис. 2.9 зображена узагальнена структурна схема АЛП множення. Управляючі сигнали з виходів пристрою управління підключаються до входів відповідних функціональних частин ОПр.

Схема електрична функціональна АЛП для множення додатних чисел наведена у додатку А. Опис функціональної схеми наведений у прикладі 7.1.

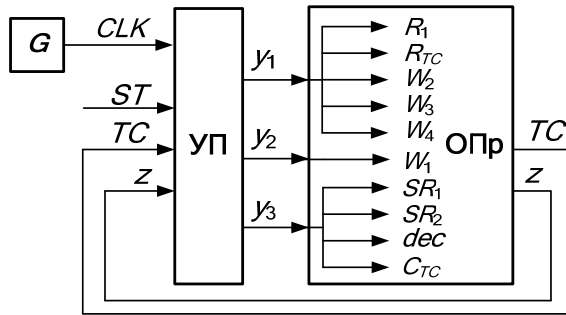


Рис. 2.9. Узагальнена структурна схема АЛП

Підготовка до лабораторного заняття

1. Розробити структурну схему операційного пристрою та змістовний мікроалгоритм відповідно до завдання наведеного у табл. 2.7), де a_7, \dots, a_1 – молодші розряди двійкового номера залікової книжки. Для побудови схеми використати комбінаційний суматор, регістр-лічильник циклів та асинхронні регістри, що мають входи управління зсувами і занесенням інформації. На схемі повинні бути зазначені розрядність регістрів та шин. Для обчислення функції використовувати множення накопиченням.
2. Розробити функціональну схему операційного пристрою.
3. Виконати логічне моделювання роботи операційного пристрою за допомогою цифрової діаграми із зазначеними викладачем значеннями операндів.
4. Здійснити синтез пристрою управління, тип управляючого автомата обрати із табл.2.9. Пам'ять автомата реалізувати на тригерах, тип яких обрати з табл. 2.8. Ураховувати, що мікрооперації на регістрах виконуються за зворотним перепадом управляючих сигналів.
5. Побудувати часові діаграми роботи автомата для кожної комбінації значень логічних умов.

Порядок виконання роботи

1. В моделюючій програмі AFDK побудувати схему операційного пристрою та доповнити її схемою управляючого автомата. На першому етапі виходи автомата до входів операційного пристрою не підключати. Налаштувати окремо схему операційного пристрою та схему управляючого автомата в синхронному режимі. Опис програмного комплексу AFDK наведений у додатку М.
2. Підключити до управляючих входів операційного пристрою виходи автомата. Зробити комплексне налагодження схеми в синхронному режимі й переконатися в правильності одержання результату.
3. Перейти до асинхронного моделювання. Дослідити зазначені викладачем часові параметри схеми.

Таблиця 2.7. Варіанти завдання

a_7	a_6	a_5	a_4	Спосіб множення	Розрядність операндів
0	0	0	0	1	16
0	0	0	1	2	8
0	0	1	0	3	16
0	0	1	1	4	8
0	1	0	0	1	8
0	1	0	1	2	16
0	1	1	0	3	8
0	1	1	1	4	16
Функція					
1	0	0	0	$D=2C+4AB$	32
1	0	0	1	$D=2C-2AB$	32
1	0	1	0	$D=2C-4AB$	32
1	0	1	1	$D=A(B-1)+0.5C$	32
1	1	0	0	$D=2A(B+1)+0.5C$	32
1	1	0	1	$D=A(B+1)+2C$	32
1	1	1	0	$D=0.5C+2AB$	32
1	1	1	1	$D=2A(B-1)+C$	32

Таблиця 2.8. Варіанти завдання

a_3	a_2	Тип тригера
0	0	JK
0	1	T
1	0	RS
1	1	D

Таблиця 2.9. Варіанти завдання

a_1	Тип автомата
0	Мілі
1	Мура

Зміст звіту

Звіт з лабораторної роботи повинен включати короткі теоретичні відомості, необхідні для виконання лабораторної роботи; структурні та функціональні схеми; таблиці та діаграми, отримані при виконанні теоретичного завдання, а також у процесі моделювання схем; висновки за роботою.

Контрольні питання

1. Охарактеризуйте чотири основних методи множення чисел.
2. Як розрахувати розрядність вузлів операційного пристрою?
3. Визначить поняття: операція, мікроалгоритм, мікрооперація.
4. Що таке мікроалгоритм операції?
5. Визначте основне призначення арифметико-логічного пристрою в ЕОМ.
6. Наведіть типи арифметико-логічних пристроїв, та їх основні відмінності.
7. Охарактеризуйте основні етапи проектування арифметико-логічного пристрою з розподіленою логікою.
8. Що відображує операційна схема виконання операції?
9. Що відображує функціональна схема пристрою?
10. В чому відмінність функціонального та структурного мікроалгоритмів?
11. Напишіть вирази, що визначають закони функціонування автоматів Милі та Мура.
12. У чому відмінність автоматів Милі та Мура?
13. Намалюйте узагальнену структурну схему управляючого автомата.
14. Охарактеризуйте основні етапи проектування управляючого автомата.
15. Як перейти від змістовного мікроалгоритму до закодованого мікроалгоритму?
16. Як побудувати граф автомата?
17. Як здійснюється оцінка станів автомата?
18. Як визначити необхідну тривалість управляючих сигналів?
19. Від чого залежить кількість тригерів, необхідних для побудови пам'яті автомата?
20. Як скласти структурну таблицю автомата?
21. Складіть таблицю переходів для JK-, RS-, T- і D-тригерів. Наведіть їх умовне графічне позначення.
22. Чи можливий перехід автомата в стан, що непередбачений графом, при використанні тригерів із внутрішньою затримкою (тригерів, керованих рівнем сигналів)?
23. Коли можливе виникнення помилкових управляючих сигналів (що непередбачені графом автомата) і чим визначається їх тривалість?
24. Наведіть способи усунення короточасних помилкових управляючих сигналів.
25. У чому суть «протигоночного» кодування станів автомата?
26. Як забезпечити перепад управляючого сигналу у випадку, коли операторну вершину з цим сигналом охоплює «петля»?
27. Як визначити час переходу автомата з одного стану в інший?

Література

1. Жабін В.І., Жуков І.А., Кліменко І.А., Стіренко С.Г. Арифметичні та управляючі пристрої цифрових ЕОМ. Навч. посібник. – К.: БЕК+, 2008. – С. 30-55.