

1. Базилевич Р. П. Декомпозиционные и топологические методы автоматизированного конструирования электронных устройств. – Львов: Вища школа, 1981. – 168с.
2. Селютин В. А. Машинное конструирование электронных устройств. – М.: Сов. радио, 1977. – 384с.

**Л.Лукащук, Н.Кустра**

Національний університет "Львівська політехніка"

УДК 681.142

## **СИНТЕЗ РЕГІСТРІВ ТА ЛІЧИЛЬНИКІВ ЯК СИНХРОННИХ ПОСЛІДОВНІСНИХ СХЕМ**

© Лукащук Л., Кустра Н., 2003

*Розглядається методика синтезу схем, різноманітних за своїми функціональними можливостями регістрів та лічильників на двоступінчастих тригерах.*

*The methodology of synthesis of the various functional possibilities of registers and counters on twolevel triggers schemes is considered in this article.*

### **Аналіз останніх досліджень**

У численних науково-технічних публікаціях розглядаються різні за своїм призначенням та способом побудови такі широко і давно застосовувані вузли цифрової техніки, як регістри та лічильники. Зокрема, схемотехніці цих вузлів приділяється певна увага [1]. В цей же час в цій, як і в багатьох публікаціях, не в достатній мірі використовується той факт, що схеми регістрів з додатковими функціями і схеми різноманітних лічильників, побудовані на двоступінчастих тригерах, є синхронними послідовнісними схемами (СПС) і тому для синтезу їх схем з різними функціональними можливостями доцільно використовувати методику проектування СПС, наведену в ряді науково-технічних публікацій[2]. При цьому в навчальному процесі зникає потреба в розгляді численних схем, тому що, користуючись вищезгаданою методикою, відносно легко і просто синтезувати будь-яку з них. Це не тільки поглиблює можливі способи побудови схем вищезгаданих цифрових вузлів, але й робить їх схеми достатньою мірою обґрунтованими і зрозумілими.

Не менш важливим є й те, що синтез схеми регістра чи лічильника із заданими функціональними можливостями стає простою задачею, що розв'язується відповідно до методики проектування СПС. До складу синхронної послідовнісної схеми входять двоступінчасті тригери. Їх кількість відповідає співвідношенню (1)

$$2^{n-1} + 1 \leq N \leq 2^n, \quad (1)$$

де  $N$ ,  $n$  – кількість станів і кількість двійкових елементів пам'яті послідовної схеми.

Тип тригера вибирається з міркувань зменшення складності схем СПС. Априорі при проектуванні регістрів можна використовувати  $D$ -тригер, а при проектуванні лічильників –  $T$ -тригер. Використання одноходових тригерів в загальному дає з самого початку перевагу у зменшенні кількості зв'язків. Проте не гарантує її в будь-якій ситуації, тому може виявитись доцільним на стадії синтезу схеми перевірити і застосування інших типів тригерів.

### Постановка задачі

Схеми регістрів з додатковими функціями, а також лічильники на двоступінчастих тригерах є синхронними послідовнісними схемами і тому для їх побудови доцільно використовувати загальну методику проектування таких схем. Метою даної статті є формалізація процедури синтезу подібних схем.

### Синтез схем регістрів

Якщо регістр виконує тільки свою основну функцію, яка полягає в прийомі, збереженні і видачі інформації, то його проектування не потребує використання якоїсь спеціальної методики. Найпростіша вхідна і вихідна логіка є очевидною. Інша справа, коли на регістр покладаються деякі додаткові функції, наприклад, зсув інформації – в такому випадку може виявитись корисним його проектування як СПС.

**Приклад 1.** Допустимо, що потрібно розробити трьохрозрядний двійковий регістр з циклічним зсувом ліворуч і праворуч на один розряд. Завдання очевидне і не потребує складання діаграми станів, тому зразу можна приступити до складання таблиці переходів (рис. 1). При відсутності зсуву

Стани		Умови переходів	
		$\overline{L} \bullet$	$L \bullet$
$\overline{A} \overline{B} \overline{C}$	S0	S0	S0
$\overline{A} \overline{B} C$	S1	S2	S4
$\overline{A} B \overline{C}$	S2	S4	S1
$\overline{A} B C$	S3	S6	S5
$A \overline{B} \overline{C}$	S4	S1	S2
$A \overline{B} C$	S5	S3	S6
$A B \overline{C}$	S6	S5	S3
$A B C$	S7	S7	S7

Рис. 1. Таблиця переходів реверсивного зсуваючого регістра ( $L$  – сигнал зсуву ліворуч,  $r$  – праворуч,

$\overline{L}$  – синхросигнал;  
A, B, C – вторинні змінні)

$$L = r = 0,$$

а при його наявності

$$L = 1, r = 0 \quad \text{або} \quad L = 0, r = 1.$$

Кількість можливих станів СПС дорівнює кількості можливих комбінацій двійкового числа, кількість розрядів якого дорівнює кількості тригерів. В тому випадку, якщо деякі комбінації не використовуються, то тоді відповідно зменшується кількість станів. У даному випадку використовуються всі вісім можливих комбінацій, тому таблиця переходів має  $2^3$  рядків. При циклічному зсуві двійкового числа ліворуч чи праворуч його значення, а відповідно і стан СПС змінюються, що відображено в таблиці переходів наявністю двох колонок. Наявність умовного значення імпульсу " $\overline{L}$ " показує, що зсув відбувається при дії активної фази синхросигналу. На основі таблиці переходів (рис. 1) складаємо вхідні рівняння для тригерів  $D$ ,  $T$ ,  $RS$  і  $JK$  типів. Кожна колонка таблиці переходів відображає ті стани, в які проходить регістр із відповідних станів, що зображені зліва таблиці, при зсуві ліворуч ( $\overline{L} \bullet L$ ) або праворуч

імпульсу " $\overline{L}$ " показує, що зсув відбувається при дії активної фази синхросигналу. На основі таблиці переходів (рис. 1) складаємо вхідні рівняння для тригерів  $D$ ,  $T$ ,  $RS$  і  $JK$  типів. Кожна колонка таблиці переходів відображає ті стани, в які проходить регістр із відповідних станів, що зображені зліва таблиці, при зсуві ліворуч ( $\overline{L} \bullet L$ ) або праворуч

( $\bar{L} \bullet r$ ). Ще лівіше записані комбінації вторинних змінних і їх доповнень, які забезпечують стани регістра, що записані поряд. Кожному тригеру відповідає одна вторинна змінна. Так, наприклад, якщо всі тригери ( $A, B$  і  $C$ ) в стані нуля, то реалізується стан  $S_0$ , а відповідна комбінація  $\overline{ABC}$  вторинних змінних дорівнює одиниці. При наявності одного із сегментів зсуву і активної фази синхросигналу здійснюється перехід в стан, записаний в одній із двох колонок таблиці – ліворуч при  $L = 1$  і праворуч при  $r = 1$ . При цьому необхідно забезпечити встановлення кожного тригера в положення, яке відповідає тому стану, в який переходить регістр.

Спосіб складання вхідних рівнянь залежить від типу тригера. При використанні  $D$ -тригера – це сума всіх кон'юнкцій – встановлення цього тригера в одиницю,  $T$ -тригера – це сума кон'юнкцій, при яких змінюється стан тригера. При використанні  $RS$ -тригера це сума кон'юнкцій, при яких тригер з "0" переходить в "1" для  $S$ -входу, а для  $R$ -входу – це сума кон'юнкцій, при яких тригер переходить з "1" в "0". Аналогічно  $RS$ -тригеру складаються рівняння для  $JK$ -тригерів, тільки при їх мінімізації потрібно враховувати структурну наявність на схемах збігу  $J$  і  $K$  - входів відповідно інверсного і прямого значення вторинної змінної, що відповідає даному тригеру. Тому її з отриманого рівняння опускають. Відповідно до вищевикладеного складені рівняння для вхідних логічних схем тригерів  $A, B, C$  розглядуваного прикладу 1

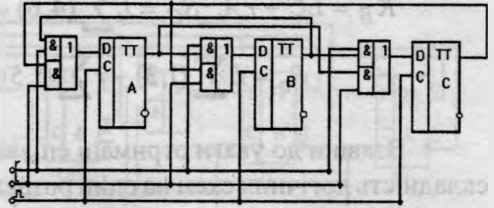


Рис. 2 Функціональна схема реверсивного зсуваючого регістра (приклад 1)

$$\begin{aligned} D_a &= r(S_1 + S_3 + S_5 + S_7) + L(S_2 + S_3 + S_6 + S_7) = LB + rC; \\ D_b &= L(S_1 + S_3 + S_5 + S_7) + r(S_4 + S_5 + S_6 + S_7) = LC + rA; \\ D_c &= L(S_4 + S_5 + S_6 + S_7) + r(S_2 + S_3 + S_6 + S_7) = LA + rB. \end{aligned} \quad (2)$$

Далі рівняння для вхідних схем записуються спрощено і більш компактно. Наприклад, вхідне рівняння для  $A$ -тригера при цьому буде мати такий вигляд:

$$D_A = r \sum (1, 3, 5, 7) + L \sum (2, 3, 6, 7). \quad (2a)$$

Функціональна схема (рис.2) відображає тільки взаємозв'язки між тригерами регістра, пов'язані зі зсувом інформації. Хоч цей приклад розглядався тільки для трирозрядного регістра, отримані результати легко розповсюдити на регістр будь-якої розрядності.

З метою дослідження впливу типу тригера на складність отриманої схеми розглянемо той же приклад у випадку вибору  $T$  та  $RS$  ( $JK$ ) -тригерів. На основі тієї ж таблиці переходів складаються для них вхідні логічні функції:

$$\begin{aligned} T_A &= L \sum (2, 3, 4, 5) + r \sum (1, 3, 4, 6) = L(\bar{A} B + \bar{A} \bar{B}) + r(\bar{A} C + \bar{A} \bar{C}); \\ T_B &= L \sum (1, 2, 5, 6) + r \sum (2, 3, 4, 5) = L(\bar{B} C + \bar{B} \bar{C}) + r(\bar{A} B + \bar{A} \bar{B}); \end{aligned}$$

$$T_C = L \sum (1, 2, 5, 6) + r \sum (1, 2, 5, 6) = L(\overline{BC} + \overline{BC}) + r(\overline{BC} + \overline{BC}) = (L + r)(\overline{BC} + \overline{BC});$$

$$J_A = LB + rC; \quad S_A = L \sum (2, 3) + r \sum (1, 3) = L\overline{AB} + r\overline{AC};$$

$$R_A = L \sum (5, 7) + r \sum (4, 6) = L\overline{AB} + r\overline{AC}; \quad K_A = L\overline{B} + r\overline{C}; \quad J_B = LC + rA;$$

$$S_B = L \sum (1, 5) + r \sum (4, 5) = L\overline{BC} + r\overline{AB}; \quad R_B = L \sum (2, 6) + r \sum (2, 3) = L\overline{BC} + r\overline{AB};$$

$$K_B = L\overline{C} + r\overline{A}; \quad S_C = L \sum (4, 6) + r \sum (2, 6) = L\overline{AC} + r\overline{BC}; \quad J_C = LA + rB;$$

$$R_C = L \sum (1, 3) + r \sum (1, 5) = L\overline{AC} + r\overline{BC}; \quad K_C = L\overline{A} + r\overline{B}. \quad (3)$$

Взявши до уваги отримані співвідношення (2–3), можна наближено визначити складність логічних схем на один розряд проєктованого регістра. Ця складність є така:

Тип тригера	Кількість використаних логічних схем на 1 розряд	Оцінка складності, кількість р-п переходів
D	2xJ2, 1xАБО-2	6
T	4xJ2, 1xАБО-4	12
RS	4xJ3, 2xАБО-2	16
JK	4xJ2, 2xАБО-2	12

Примітка. За одиницю складності приймається один р-п перехід.

Користуючись вищевикладеною методикою, можна синтезувати регістр з будь-якими іншими додатковими функціями.

умова	Л. р <sub>-1</sub>		Л. z <sub>-1</sub>	
	Л. р <sub>-1</sub>	Л. z <sub>-1</sub>	Л. р <sub>-1</sub>	Л. z <sub>-1</sub>
АВС	S0	S1	S7/z <sub>2</sub> =1	
АВС	S1	S2	S0	
АВС	S2	S3	S1	
АВС	S3	S4	S2	
АВС	S4	S5	S3	
АВС	S5	S6	S4	
АВС	S6	S7	S5	
АВС	S7	S0/p <sub>2</sub> =1	S6	

Рис.3 Таблиця станів для двійкового трирозрядного реверсивного лічильника (р<sub>-1</sub> і z<sub>-1</sub> – вхідні сигнали відповідно додавання і позички проєктованого лічильника, р<sub>2</sub> і z<sub>2</sub> – аналогічні сигнали з виходу лічильника)

## Лічильники

Важливими вузлами цифрових пристроїв є різні типи лічильників та перерахункових схем, які також належать до СПС. Як і в попередньому випадку, методика їх проєктування розглядатиметься в прикладах.

**Приклад 2.** Розробити функціональну схему трирозрядного двійкового реверсивного лічильника, що працює в доповнювальному коді.

Як і у випадку регістрів, почнемо з таблиці станів (рис.3). За основу розробки можна взяти будь-який тригер. Очевидно, найбільш логічний вибір – це T-тригер, адже при T = 1 він підраховує вхідні сигнали, що надходять на його синхровхід, а при T = 0 – не реагує на них.

Вхідні логічні функції:

$$T_A = p_{-1} \sum (3, 7) + z_{-1} \sum (0, 4); \quad T_C = p_{-1} \sum_{i=0}^7 S_I + z_{-1} \sum_{i=0}^7 S_I = p_{-1} + z_{-1};$$

$$T_B = p_{-1} \sum (1, 3, 5, 7) + z_{-1} \sum (0, 2, 4, 6) = p_{-1} C + z_{-1} C. \quad (4)$$

Вихідні функції

$$p_2 = S_7 p_{-1} \bullet \text{Л}; \quad z_2 = S_0 z_{-1} \text{Л}. \quad (5)$$

Відповідно до отриманих співвідношень (4 і 5) побудована функціональна схема (рис.4). Очевидно, аналогічно кількість розрядів може нарощуватись. Такий лічильник є найбільш швидкодіючим, бо зв'язки старшого розряду з молодшим прями. Можна зменшити складність схеми, використовуючи кон'юнкції молодших розрядів, утворивши таким чином ланку послідовного або паралельно-послідовного перенесення. Але ж це приведе до зменшення швидкодії лічильника.

Значне застосування знаходять лічильники, що працюють в системах числення, основи яких не дорівнюють цілому степеню двійки. До таких перш за все слід віднести десяткові лічильники.

**Приклад 3.** Розробити декаду реверсивного десяткового лічильника, яка працює в коді з надлишком "3".

Вибираємо двотактний Т-тригер і на основі таблиці переходів (рис.5) складаємо вихідні логічні функції:

$$T_A = p_{k-1} \sum (4, 9) + z_{k-1} \sum (0, 5) = p_{k-1} (\bar{A} BCD + ABC\bar{D}) + z_{k-1} (\bar{A} BCD + ABC\bar{D});$$

$$T_B = p_{k-1} \sum (0, 4, 8, 9) + z_{k-1} \sum (0, 1, 5, 9) = p_{k-1} (CD + ABC\bar{D}) + z_{k-1} (\bar{C}D + \bar{A}BCD);$$

$$T_C = p_{k-1} \sum (0, 2, 4, 6, 9) + z_{k-1} \sum (0, 1, 3, 5, 7, 9) = p_{k-1} (D + ABC\bar{D}) + z_{k-1} (\bar{D} + \bar{A}BCD);$$

$$T_D = (p_{k-1} + z_{k-1}) \sum (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) = p_{k-1} + z_{k-1} = 1. \quad (6)$$

$$p_k = ABC\bar{D} p_{k-1} \bullet \text{Л}, \quad z_k = \bar{A}BCD z_{k-1} \bullet \text{Л}. \quad (7)$$

Мінімізація логічних функцій проводилась з врахуванням невикористаних в коді з надлишком "3" комбінацій (0, 1, 2, 13, 14, 15). Тільки завдяки цьому отримані відносно прості логічні вирази. Такі ж результати можна отримати при безпосередньому аналізі таблиці переходів з врахуванням прийнятих кодів, що значно простіше, але потребує певного досвіду. Структурна схема лічильника наведена на рис.6. До її складу входять такі ЛЕ: І5 – 8шт, І3 – 2шт, І2 – 2шт, І6 – 2 шт, АБО4 – 3шт. Загальна складність логічних схем лічильника, зображеного на рис.6, дорівнює

$$C_1 = 8 \cdot 5 + 3 \cdot 2 + 2 \cdot 2 + 6 \cdot 2 + 4 \cdot 3 = 74.$$

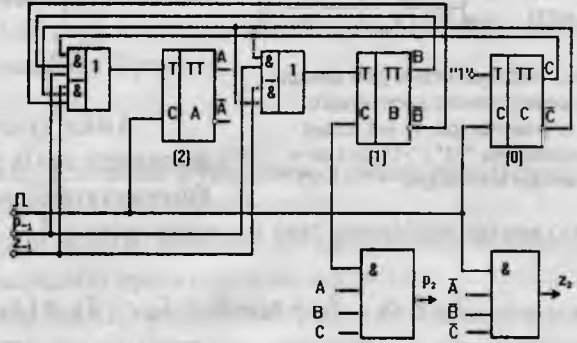


Рис.4. Функціональна схема реверсивного двійкового лічильника ( $p_{-1}$  – "+1",  $z_{-1}$  – "-1",  $p_2$  і  $z_2$  – відповідно перенесення і позичання із старшого розряду лічильника)

стани	умова	
	$p_{k-1} \text{ JL}$	$z_{k-1} \text{ JL}$
$\overline{A}BCD$	S0	S9/z <sub>k</sub>
$A\overline{B}CD$	S1	S0
$AB\overline{C}D$	S2	S1
$ABC\overline{D}$	S3	S2
$\overline{A}BCD$	S4	S3
$A\overline{B}CD$	S5	S4
$AB\overline{C}D$	S6	S5
$ABC\overline{D}$	S7	S6
$\overline{A}BCD$	S8	S7
$A\overline{B}CD$	S9	S8
$\overline{A}BCD$	S0 / p <sub>k</sub>	S8

Рис.5 Таблиця станів для декади реверсивного десятикового лічильника (pk-1, k-1 вхідні відповідно "+1" і "-1", pk і zk - вихідні відповідно "+1" і "-1")

цією метою здійснено таке їх перетворення:

$$T_A = p_{k-1} S_4 + z_{k-1} S_5 + L_k, \quad T_B = (p_{k-1} \overline{D}) \overline{C} + (z_{k-1} \overline{D}) C + L_k;$$

$$T_C = p_{k-1} D + z_{k-1} \overline{D} + L_k, \quad S_4 = \overline{A} B C D, \quad S_5 = \overline{A} \overline{B} \overline{C} D; \quad L_k = p_k^* + z_k^*, \quad p_k^* = S_9 p_{k-1},$$

$$S_9 = A B C D; \quad z_k^* = S_0 z_{k-1}, \quad S_0 = \overline{A} \overline{B} C D;$$

$$p_k = p_k^*, \quad z_k = z_k^*. \quad (10)$$

Якщо позначити затримку в передаванні сигналу на одному рівні логічної схеми, то затримка встановлення логічних сигналів на вході тригерів відносно переднього фронту синхросигналу дорівнює:

$$t_T^{(1)} = 2\tau, \quad (8)$$

а затримка в передаванні до наступної декади вихідних сигналів перенесення  $p_k$  чи позички  $z_k$  дорівнює:

$$t_b^{(1)} = \tau. \quad (9)$$

Аналізуючи логічні вирази (6), можна зауважити, що їх реалізація може бути спрощена, якщо використати більш ніж двоступеневі логічні схеми. З

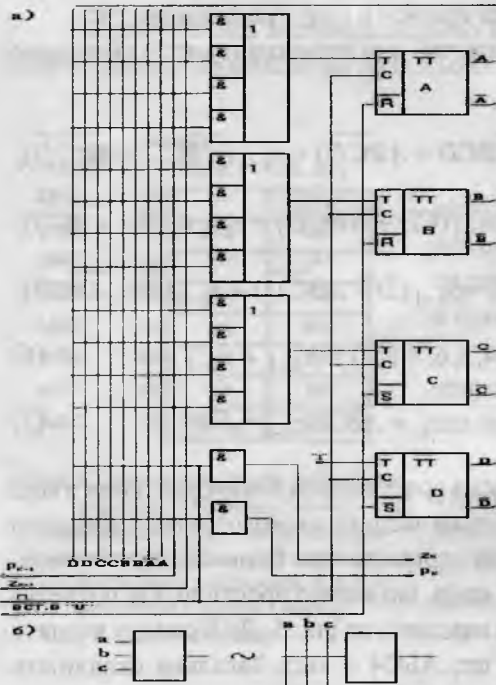


Рис.6 Структурна схема декади реверсивного десятикового лічильника, що працює в код з надлишком "3" (варіант 1)

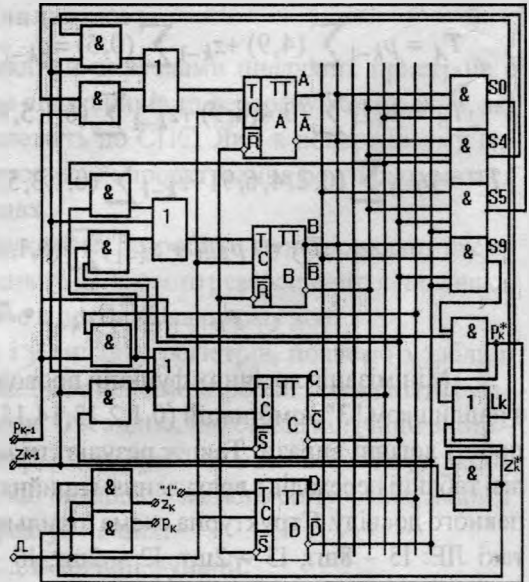


Рис.7 Структурна схема декади реверсивного десятикового лічильника, що працює в код з надлишком "3" (варіант 2)



$$p_k = p_k^* \text{Л}, z_k = z_k^* \text{Л}.$$

Загальна складність реалізації логічних схем (рис.7) дорівнює:

$$C_2 = 2 \cdot 10 + 3 \cdot 3 + 4 \cdot 4 + 2 \cdot 1 = 47,$$

а відповідні затримки сигналів:

$$t_T^{(2)} = 3\tau, \quad t_b^{(2)} = 2\tau. \quad (11)$$

Отже, складність зменшено в 1,57 раза, а час затримки сигналу на вході тригера збільшилась у 1,5 раза і на виході декади – у 2 рази.

### Висновки

1. Регістри цифрових вузлів з додатковими функціями, а також лічильники та перерахункові схеми доцільно буде проектувати як СПС.
2. Проектування цифрових вузлів як СПС дозволяє встановлювати найкращий баланс між складністю і швидкістю проектованого вузла.
3. Проектування цифрових вузлів як СПС зменшує потребу в детальному аналізі існуючих аналогічних вузлів, тому що забезпечує потрібну їх якість, виходячи з можливостей методики проектування.

1. Угрюмова Є. Цифрова схемотехніка. – М., 2001.
2. Лукашук Л. Проектування послідовних схем. – Львів, 1988

А. Батюк, А. Худий, І. Цмоць  
Національний університет “Львівська політехніка”

УДК 681.325

## КОНВЕЄРНИЙ ПАРАЛЕЛЬНО-ПОТОКОВИЙ ПРИСТРІЙ СОРТУВАННЯ ДАНИХ У РЕАЛЬНОМУ ЧАСІ МЕТОДОМ ЗЛИТТЯ

© Батюк А., Худий А., Цмоць І., 2003

*Розроблено на основі методу двошляхового злиття новий алгоритм сортування інтенсивних потоків даних, який ґрунтується на базовій операції пара-*