

## 8 Лабораторна робота № 8. Регістри

### 8.1 Мета

Засвоїти принцип роботи та способи використання регістрів в цифрових схемах.

### 8.2 Завдання

Побудувати генератор М-послідовності для заданого вектора зворотніх зв'язків і початкової комбінації на основі регістра зсуву.

Таблиця 8.1 – Варіанти індивідуальних завдань

Варіант	Вектор зворотніх зв'язків	Початкова комбінація
1	10010	11001
2	10100	01101
3	10111	01111
4	11011	00010
5	11101	01010
6	11110	00111
7	100001	001100
8	100100	001101
9	101011	111000
10	101101	001000
11	110000	001100
12	110011	011100
13	110110	011101
14	111001	110000
15	111010	010000

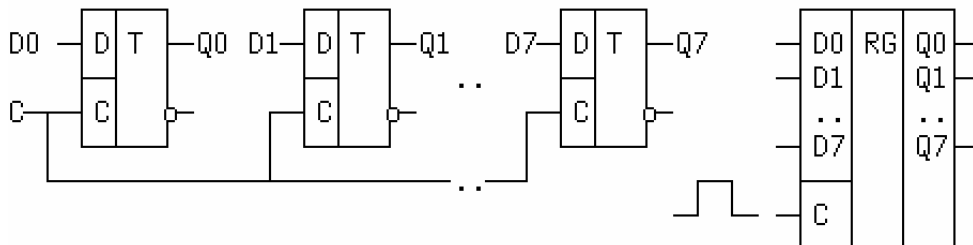
*Примітка.* Крайній правий розряд відповідає наймолодшому (верхньому) розряду регістра.

### 8.3 Короткі теоретичні відомості

#### 8.3.1 Основні відомості про регістри

*Регістри* - послідовні схеми з різними комбінаціями послідовного і паралельного способів запису і зчитування інформації. Виконуються на основі тригерів.

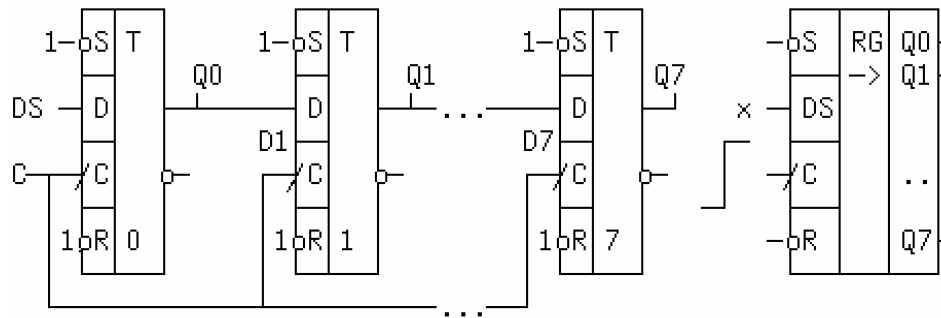
*Регістри з паралельним записом* називаються також регістрами пам'яті. У них можуть використовуватися як статичні D-тригери, так і тригери з динамічним управлінням. На рис.8.1 приведена схема 8-ми розрядного регістра пам'яті із загальним входом управління записом інформацією і її умовне позначення. Високий рівень на вході С переписує інформацію з входу на вихід ( $Q_i=D_i$ ), а низький рівень – переводить регістр в стан зберігання.



**Рисунок 8.1 - Паралельний регістр**

*Послідовні регістри* або, як їх ще називають, *регістри зсуву* (рис.8.2) виконуються на основі тригерів з динамічним синхровходом (справедливо для регістрів з одним тактуючим сигналом, у двотактних можна використовувати і "прозорі" D-тригери). Функція запису в n-

розрядному регістрі зсуву на D-тригерах задається у вигляді умов:  $D_0=DS=x$ ,  $D_i=Q(i-1)$ , де  $i=1,2,\dots,n-1$ , DS - вхід для послідовного запису.



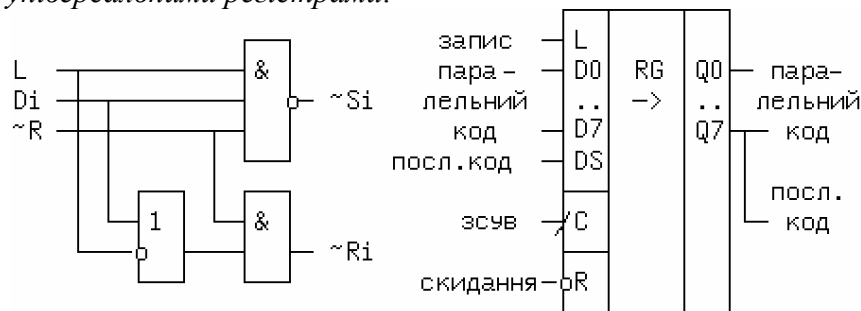
**Рисунок 8.2 - Регістр зсуву**

З приходом чергового позитивного фронту синхроімпульса С сигнал з входу і-го тригера через час  $t_{з.р.}$  з'явиться на його виході і поступить на вхід наступного (і+1)-го тригера. Проте на його вихід ця інформація не переписеться, оскільки тривалість активного фронту  $t_{0,1}$  менше  $t_{з.р.}$ . На цьому процес зсуву даних на один розряд закінчиться до приходу наступного позитивного фронту тактового сигналу. Звідси зрозуміло, чому не можна використовувати тригери із статичним управлінням. Кожного разу при  $C = 1$  весь ланцюжок виявиться прозорим від входу DS до виходу Q7 і значення  $DS = x$  буде записане у всі тригери.

У позначеннях регістрів зсуву напрям стрілки, що вказує зсув, умовний. У різних довідниках її напрям різний. Умовно приймається, що зсув здійснюється від молодшого розряду до старшого.

Регістри зсуву, як і лічильники, можна каскадувати, тобто якщо необхідний регістр зсуву більшої розрядності, ніж забезпечують мікросхеми регістрів, необхідно просто з'єднати вихід останнього тригера першого регістра із входом послідовних даних наступного регістра, а також об'єднати входи тактування і скидання.

Практичні схеми регістрів доповнюються схемами, що підключаються до кожного тригера і мають вхід паралельного запису  $D_i$ , загальний вхід дозволу запису L і загальний асинхронний вхід скидання  $\sim R$  всіх тригерів (рис.3). Ці схеми підключаються до незадіяних входів  $\sim R_i$ ,  $\sim S_i$  тригерів. Регістри зсуву з паралельним завантаженням коду називаються також *універсальними регістрами*.



**Рисунок 8.3 - Універсальний регістр**

Вхід  $\sim R$  має найвищий пріоритет: якщо  $\sim R=0$ , то  $\sim R_i=0$ , а  $\sim S_i=1$  і всі тригери обнулюються, незалежно від сигналів L, Di і C. Менший пріоритет мають входи L і Di. Якщо  $\sim R=1$ , то при  $L=1$ , здійснюється паралельний запис інформації і  $Q_i = D_i$  незалежно від сигналу C. І, нарешті, якщо на входах  $\sim R$  і L пасивні рівні, то  $\sim R_i = \sim S_i = 1$  - теж пасивний рівень і регістр зберігає інформацію або здійснює її зсув.

### 8.3.2 Використання універсальних регістрів (редагувати)

На основі універсальних регістрів можна побудувати цифрові схеми з різною функціональністю, наприклад, кільцеві лічильники, лічильники Джонсона, генератори

цифрових послідовностей з різними властивостями.

В цій лабораторній роботі універсальний регістр використовується для побудови генератора М-послідовності. В даному випадку нас цікавлять лише особливості функціонування регістрів, тому, не заглиблюючись у вищу алгебру, відмітимо лише, що М-послідовність (послідовність максимальної довжини) – це псевдовипадкова послідовність нулів та одиниць, яка повторюється з періодом  $2^{n-1}$  на виході n-розрядного регістру зсуву із зворотніми зв'язками. При цьому в кожному такті на виходах регістра з'являється новий n-розрядний код, що не повторюється в межах періоду, тобто на протязі періоду формуються всі можливі n-розрядні комбінації (окрім нульової), починаючи із початкової комбінації, яка до початку зсувів записується в регістр. Вид послідовності визначається структурою зворотніх зв'язків (тобто з яких саме виходів беруться сигнали, що об'єднуються через "виключне АБО" та подаються на вхід), причому тільки деякі варіанти призводять до формування саме М-послідовності – в більшості випадків період буде коротшим, ніж  $2^{n-1}$ . На підтвердження цього факту в роботі пропонується сформувати М-послідовність на основі "правильного" вектору зворотніх зв'язків, а потім інвертувати його.

Псевдовипадковою ця послідовність називається тому, що хоча спосіб її формування і є детермінованим, однак статистичні властивості наближаються до властивостей дійсно випадкових послідовностей. Ця властивість є корисною при застосуванні, зокрема, в системах передавання даних (щоб зробити корисний сигнал "схожим" на шум – це дозволяє підвищити конфіденційність передачі та зробити сигнал більш стійким до завад при поширенні в безпроводному середовищі на фоні інших подібних сигналів).

#### 8.4 Порядок виконання роботи

1. Побудувати 8-розрядний регістр зсуву на основі 2 мікросхем регістрів IP1. На входи тактування обох мікросхем подати імпульсну послідовність частотою 100 кГц (компонент Stim1) для забезпечення тактування зсувів. На вхід послідовних даних другої мікросхеми подати сигнал з останнього виходу даних першої мікросхеми.

2. Реалізувати паралельне завантаження заданої початкової комбінації: встановити режим паралельного завантаження, подавши короткочасний активний сигнал на вхід дозволу паралельного завантаження PE, на паралельні входи подати задану початкову комбінацію (використовуючи компоненти Fixed Digital), для її паралельного запису в регістр подати на С2 короткий одиничний імпульс (компонент Stim1).

3. Для побудови генератора М-послідовності ввести зворотній зв'язок згідно заданого вектора: сигнали з виходів, яким відповідає "1" у векторі зворотнього зв'язку, об'єднати через елементи "Виключне АБО" і результируючий сигнал подати на послідовний вхід даних першої мікросхеми. *Вважати, що наймолодший (крайній правий) розряд відповідає наймолодшому (верхньому) входу мікросхеми регістра.*

4. Вивести часові діаграми на входах та на виходах регістра. Визначити період М-послідовності (тобто інтервал часу між двома однаковими комбінаціями на виходах регістра).

5. Скласти аналогічну схему для інвертованого вектора зворотніх зв'язків. Повторити п.4.

#### 8.5 Звіт повинен містити:

1. Завдання (з вказанням номеру варіанта)
2. Принципові схеми
3. Часові діаграми на входах та виходах регістрів для обох схем. Вказати період М-послідовності для обох випадків.
4. Висновки.

#### 8.6 Контрольні питання

1. Паралельні і послідовні регістри: будова, принцип дії, призначення виводів,

застосування.

2. Вказати сукупність сигналів, що необхідно подати на входи послідовного або паралельного регістра для завантаження в нього заданого двійкового коду.
3. Як функціонує кільцевий лічильник, лічильник Джонсона? Як їх побудувати, використовуючи універсальні регістри?
4. На яких тригерах можна побудувати регістр зсуву?
5. Навести структуру регістра зсуву або паралельного регістра на основі тригерів.
6. Визначити стан виводів регістра в довільний момент часу при заданих часових діаграмах сигналів на входах.