

Лекція 11

Канонічний синтез цифрових автоматів.

Вступ

Математична модель цифрового пристрою представляється у вигляді абстрактного автомата (АА).

Абстрактний автомат (у теорії алгоритмів) - математична абстракція, модель дискретного пристрою, що має один вхід, один вихід і в кожний момент часу знаходиться в одному стані з множини можливих.

АА задається множиною: $S = \{A, Z, W, \delta, \lambda, a_0\}$, де

A - множина станів автомата;

Z - множина вхідних сигналів (вхідний алфавіт);

W - множина вихідних сигналів автомата (вихідний алфавіт);

δ - функція переходів автомата;

λ - функція виходів;

a_0 - вихідний стан автоматів.

Кінцевий автомат

Автомат, у якого множини A , Z , W обмежені, називається **кінцевим автоматом**.

Абстрактний автомат можна представити у вигляді "чорного ящика", що має один вхідний і один вихідний сигнал. Очевидно, що абстрактний автомат перетворить символи вхідного алфавіту в символи вихідного алфавіту.

Таким чином, абстрактний автомат є найбільш загальною моделлю пристрою обробки інформації.

Існує два основні способи завдання абстрактного автомата:

1. табличний;
2. графічний.

Автомати Мілі та Мура

На практиці найбільше широко поширені дві моделі автоматів:

1. **Автомат Мілі**, де

Функція переходу - $\delta = f(A, Z)$;

Функція вихідних сигналів - $\lambda = f(A, Z)$.

2. **Автомат Мура**, де

Функція переходу - $\delta = f(A, Z)$;

Функція вихідних сигналів - $\lambda = f(A)$.

Щоб оцінити стан автомата в будь-який момент часу достатньо знати вихідний стан і послідовність вхідних сигналів, які надійшли за цей відрізок часу.

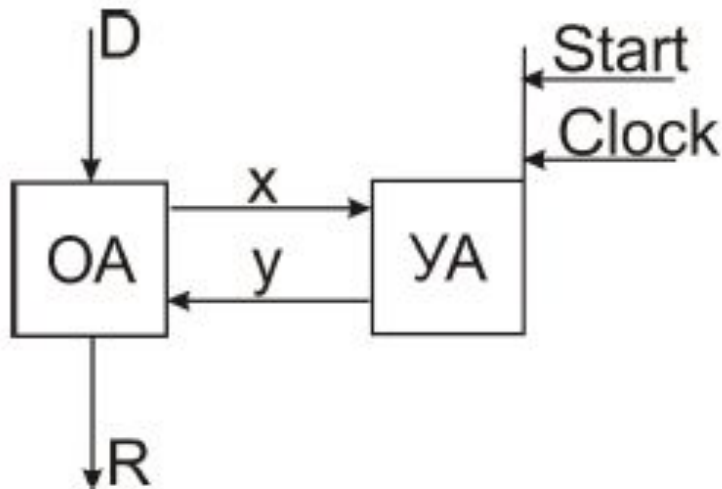
Під дією вхідних сигналів у кожен момент часу автомат переходить із попереднього в наступний стан.

Стан автомата - пам'ять про вхідні сигнали, які надійшли на вхід автомата в попередній момент часу. Стан зберігає трасу функціонування автомата.

Структурний автомат

Структурний автомат - пристрій, який реалізує закон поведінки абстрактного автомата. Являє собою схему, що складається з логічних елементів і елементів пам'яті.

В основі функціонування сучасних цифрових пристроїв лежить принцип мікропрограмного керування. Згідно із цим принципом будь-яке цифрове обладнання можна представити як композицію двох основних частин: операційного автомата і керуючого автомата. Загальна структура цифрового пристрою.



Операційний автомат (ОА) отримує на вхід дані D , обробляє їх і формує вихідний результат R .

Процес обробки даних відбувається згідно з алгоритмом функціонування ОА.

Робота КА

Процес обробки даних відбувається згідно з алгоритмом функціонування ОА. Цей алгоритм може бути представлений, наприклад графічно, у вигляді блок-схеми, в операторних вершинах якої вказані виконувані команди по обробці даних, а в умовних вершинах вказано, які необхідні перевірки умов, для визначення подальшого ходу алгоритму.

ОА формує набір запитів X , спрямованих у керуючий автомат (КА), ці запити X відображають умови, що перевіряються. КА відповідно реагує на результати перевірки умов X , і формує набір керуючих сигналів Y , спрямованих в ОА.

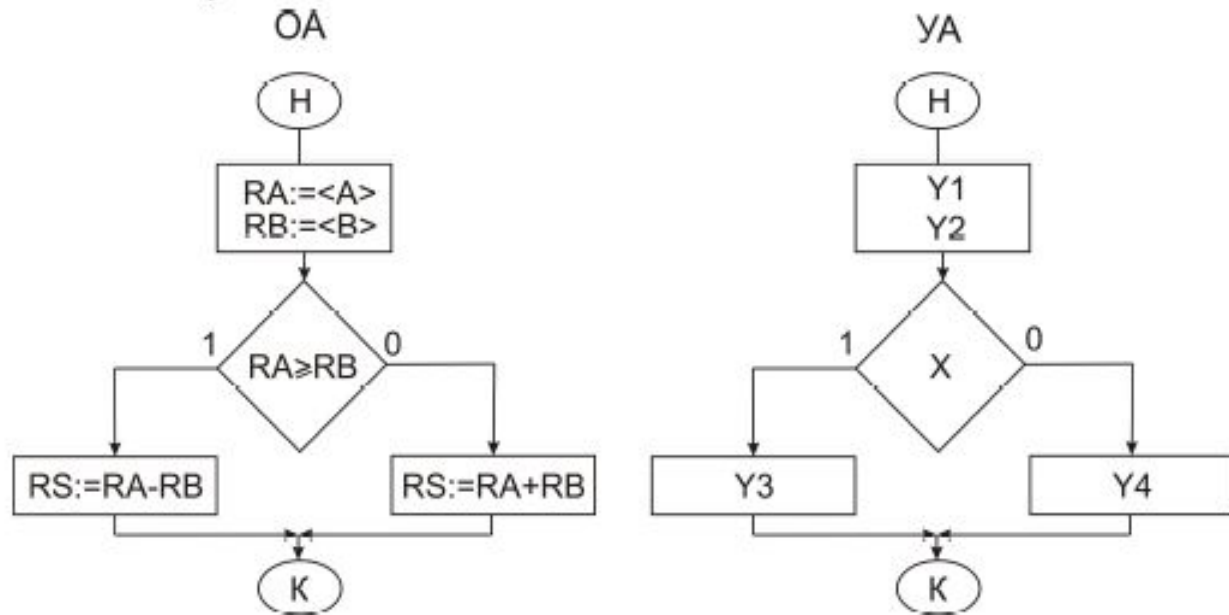
Цей набір керуючих сигналів Y дозволяє виконати певні дії в ОА. Так відбувається на кожному кроці функціонування. Очевидно, що між ОА і КА постійно відбувається діалог,

Робота КА

На лівому малюнку зображено функціональний змістовний алгоритм для ОА. На правому - зображено закодований алгоритм функціонування (граф схема алгоритму - ГСА).

У прикладі для КА у вершинах ГСА записані закодовані мікрокоманди, позначувані Y_i . Процес виконання кожної мікрокоманди (МК) Y_i включає низку елементарних дій по обробці інформації, безпосередньо пов'язаної з апаратурою. Ці елементарні дії називаються мікрооперацією (МО), позначаються y_j . Таким чином, МК складається з набору МО.

$$S = \left\{ \begin{array}{l} A - B, A \geq B \\ A + B, A < B \end{array} \right\}$$



Робота КА

Наприклад. Виконання команди $RS := RA + RB$ складається з наступних МО:

- у1: записати вміст регістру А на шину операнда А1 ($RA \rightarrow$ шина А1);
- у2: записати вміст регістру В на шину операнда А2 ($RB \rightarrow$ шина А2);
- у3: на суматорі скласти вміст шин А1 і А2 ($S_{шин} := A1 + A2$);
- у4: записати результат суми на шині результатів Z1 ($S_{шин} \rightarrow Z1$);
- у5: записати вміст шини Z1 у регістр результат RS ($Z1 \rightarrow RS$);

$Y = \{y_1, \dots, y_5\}$, у такий спосіб команда складається з п'яти МО.

Кожна мікрокоманда y_i виконується тільки при наявності спеціального сигналу дозволу. Цей сигнал формується у КА і позначається також, як і відповідна МО, тобто y_i .

Таким чином, фізично y_i - це сигнали, формовані в КА. Складність схеми ОА в сотні раз перевищує складність КА.

Приклад МО

Прикладом МО може бути керуючий сигнал який:

- Встановлює або очищує прапорець стану МП
- Керуючий сигнал запису до регістра
- Керуючий код на вході мультиплексора .

Для реалізації команди необхідно на відповідні керуючі входи подати розподілену в часі послідовність керуючих сигналів.

Синтез керуючих автоматів

Цифровий автомат – пристрій, що характеризується набором внутрішніх станів у які він перейде під впливом команд програми, яка була закладена в нього.

Перехід автомата з одного стану в інший здійснюється в певний момент часу. Математичною моделлю ЦА (а в загальному випадку будь-якого дискретного пристрою) є так званий абстрактний автомат, визначений як 6- компонентний кортеж: $S=(A,Z,W,\delta,\lambda,a_1)$, розшифрування якого представлено нижче.

1. $A=\{a_1, a_2, \dots, a_m\}$ – множина станів (внутрішній алфавіт)
2. $Z=\{z_1, z_2, \dots, z_f\}$ – множина вхідних сигналів (вхідний алфавіт)
3. $W=\{w_1, w_2, \dots, w_g\}$ – множина вихідних сигналів (вихідний алфавіт)
4. $\delta : A \cdot Z \rightarrow A$ – **функція переходів**, що реалізує відображення $D_\delta \subseteq A \cdot Z$ в A .
Іншими словами функція δ деяким парам «стан - вхідний сигнал» (a_m, z_f) ставить у відповідність стани автомата $a_s = \delta(a_m, z_f)$, $a_s \in A$.
5. $\lambda : A \cdot Z \rightarrow W$ - **функція виходів**, що реалізує відображення $D_\lambda \subseteq A \cdot Z$ на W .
Функція λ деяким парам «стан - вхідний сигнал» (a_m, z_f) ставить у відповідність вихідні сигнали автомата $Wg = \lambda(a_m, z_f)$, $Wg \in W$.
6. $a_1 \in A$ - початковий стан автомата.

Місце ПК

Взаємодія пристрою керування з іншими вузлами процесора



Синтез керуючих автоматів

Під алфавітом тут розуміється непуста множина попарно різних символів. Елементи алфавіту називаються буквами, а кінцева впорядкована послідовність букв - словом у даному алфавіті.

Абстрактний автомат, зображений на малюнку має один вхід і один вихід. Автомат працює в дискретному часі, що приймає цілі не негативні значення $t = 0, 1, 2, \dots$. У кожний момент t дискретного часу автомат знаходиться в деякому стані $a(t)$ з множини можливих станів автомата, причому в початковий момент $t = 0$ він завжди перебуває в початковому стані $a(0) = a_1$. У момент t , будучи в стані $a(t)$, автомат здатен сприйняти на вході букву вхідного алфавіту $z(t) \in Z$. Відповідно до функції виходів він видасть у той же момент часу t букву вхідного алфавіту $w(t) = \lambda(a(t), z(t))$ і відповідно до функції переходів перейде в наступний стан $a(t+1) = \delta[a(t), z(t)]$, $a(t) \in A$, $w(t) \in W$.

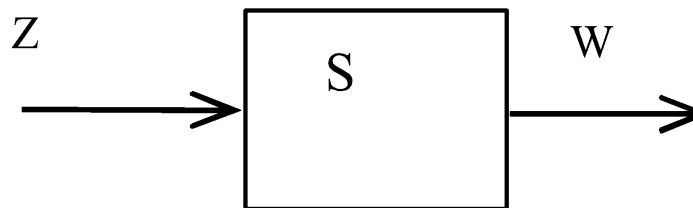


Рисунок – Абстрактный автомат

Синтез керуючих автоматів

На практиці найбільше поширення отримали два класи автоматів - автомати Мілі (Mealy) і Мура (Moore).

Закон функціонування автомата Мілі задається рівняннями:

$$a(t+1) = \delta(a(t), z(t)); w(t) = \lambda(a(t), z(t)), t = 0, 1, 2, \dots$$

Закон функціонування автомата Мура задається рівняннями:

$$a(t+1) = \delta(a(t), z(t)); w(t) = \lambda(a(t)), t = 0, 1, 2, \dots$$

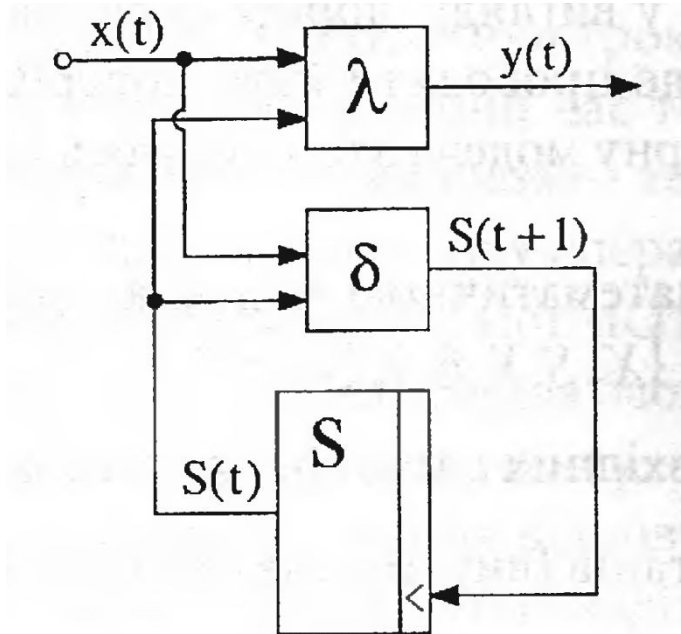
$w(t)$ -вихідні сигнали

$$a(t + 1) = f(a(t); x(t));$$

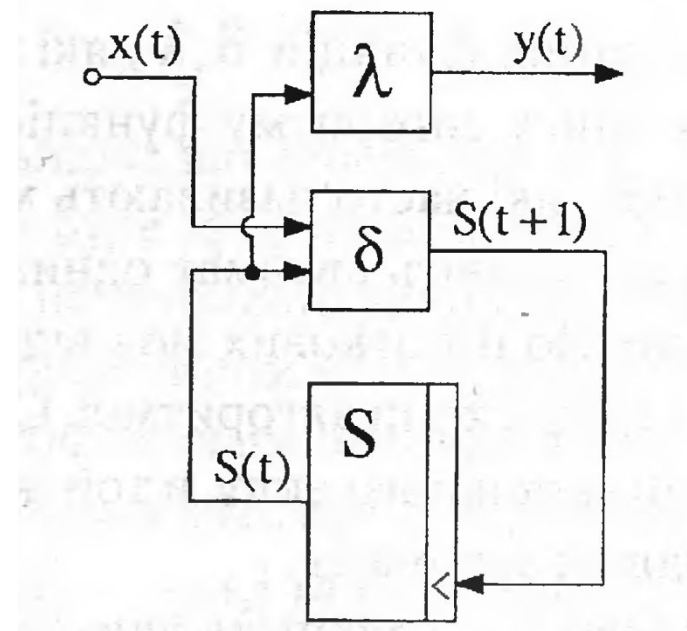
З порівняння законів функціонування видно, що, на відміну від автомата Мілі, вихідний сигнал в автоматі Мура залежить тільки від поточного стану автомата і у явному вигляді не залежить від вхідного сигналу. Для повного представлення автомата Мілі чи Мура додатково до законів функціонування, необхідно вказати початковий стан і визначити внутрішній, вхідний і вихідний алфавіти.

входное слово	x_2	x_1	x_2	x_2	x_1	x_2	x_1	x_1	x_2	...
состояние автомата	a_0	a_0	a_1	a_2	a_3	a_0	a_0	a_1	a_3	a_3
выходное слово	y_1	y_2	y_1	y_1	y_1	y_1	y_2	y_2	y_3	...

Структура автоматів Мілі та Мура



Автомат Мілі



Автомат Мура

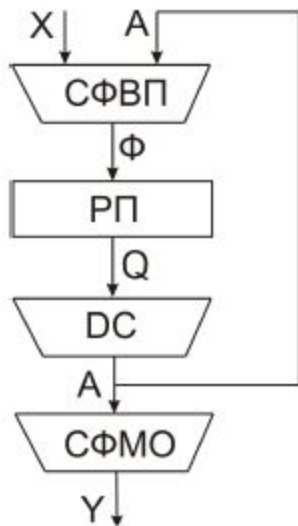
Синтез керуючих автоматів

Проектування керуючих автоматів Мура та Мілі за заданою граф-схемою алгоритму

Автомат Мура.

Для синтезу КА вихідними даними слугує алгоритм керування, представлений у графічному вигляді, тобто ГСА, в операторних вершинах якої записані набори МО.

Загальна структура автомата Мура наступна:



СФВП - схема функції збудження пам'яті автомата. Формує функції $\Phi = f(x, A)$;

РП - регістр пам'яті, запам'ятовує код стану;

ДС - дешифратор станів ;

СФМО - схема формування мікрооперацій, формує функцію $Y = f(A)$.

X-певні умови, що перевіряються автоматом ОА, які можуть змінити хід обчислень.

Φ-сформовані функції по зміні пам'яті.

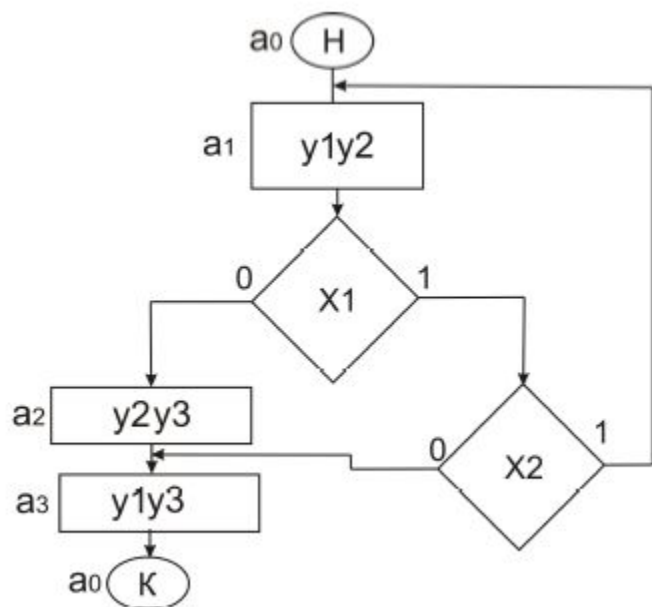
ДС- дешифратор, що визначає номер стану КА

Автомат Мура

Методика синтезу автомата Мура:

1. Відмічення станів автомата на ГСА.
2. Кодування станів.
3. Формування таблиці переходів автомата.
4. Формування функцій збудження пам'яті.
5. Формування функцій вихідних сигналів.
6. Синтез логічної схеми КА Мура.

Нехай заданий алгоритм керування у вигляді ГСА:



1. Відмічення станів.

Початкова й кінцева вершина відмічаються станом a_0 .

Станом a_i в автоматі Мура відмічаються кожна операторна вершина.

Множина станів у даному прикладі :
 $A = \{a_0, \dots, a_3\}$, $M = 4$.

Необхідна розрядність коду стану :

$R = \lceil \log_2 M \rceil$. У нашому випадку необхідно 2 біта і 2 тригери: $R = \lceil \log_2 4 \rceil = 2 \Rightarrow \{Q_1, Q_2\}$, тому що кожен біт коду формується на виході одного тригера.

Автомат Мура

2. Таблица кодов станів:

a_m	Q_1	Q_2
a_0	0	0
a_1	0	1
a_2	1	0
a_3	1	1

3. Таблица переходів автомата Мура:

Текущее состояние		Последующее состояние		X	Y	D_i
Состояние	Код	Состояние	Код			
a_0	00	a_1	01	1	-	D_2
a_1	01	a_1	01	$\overline{x_1} x_2$	$y_1 y_2$	D_2
		a_2	10	$\overline{x_1}$	$y_1 y_2$	D_1
		a_3	11	$x_1 x_2$	$y_1 y_2$	$D_1 D_2$
a_2	10	a_3	11	1	$y_2 y_3$	$D_1 D_2$
a_3	11	a_0	00	1	$y_1 y_3$	-

Автомат Мура

4. Формування функцій збудження пам'яті (ФВП):

$$D_1 = a_1 \cdot x_1 \vee a_1 \cdot x_1 \cdot x_2 \vee a_2$$

$$D_2 = a_0 \vee a_1 \cdot x_1 \cdot x_2 \vee a_1 \cdot x_1 \cdot \overline{x_2} \vee a_2 = a_0 \vee a_1 \cdot x_1 \vee a_2$$

5. Формування функцій вихідних сигналів (ФВС) (мікрооперацій).

$$y_1 = a_1 \vee a_3$$

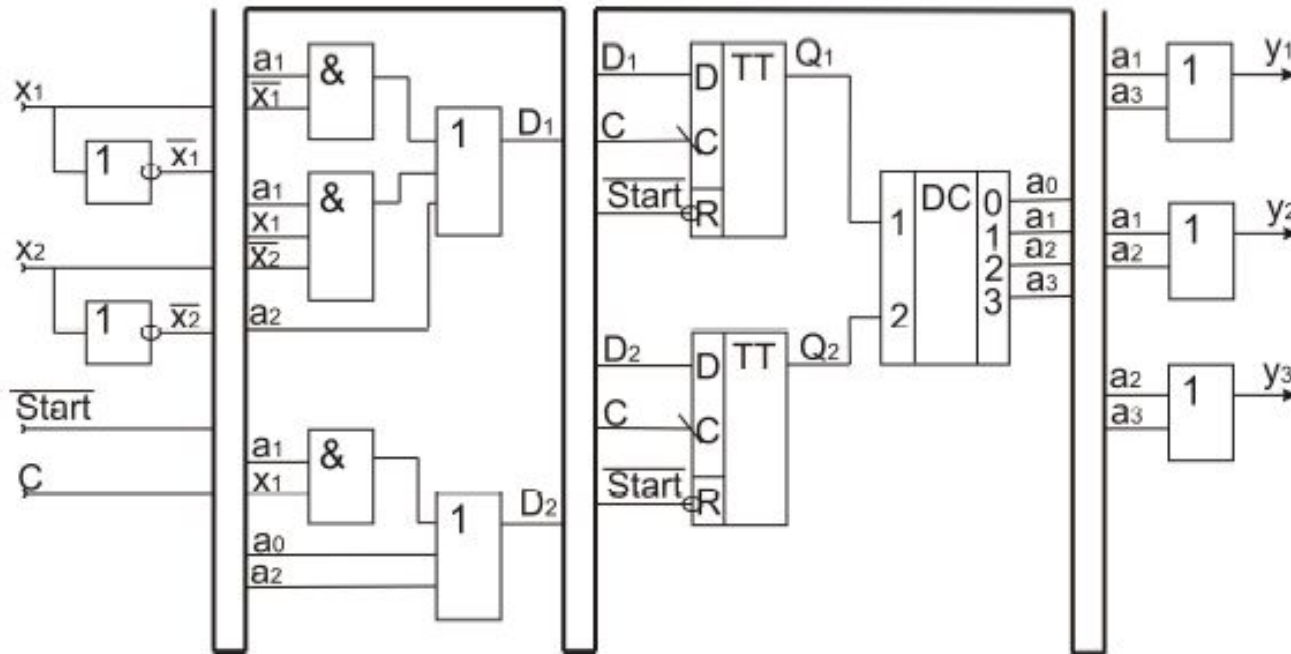
$$y_2 = a_1 \vee a_2$$

$$y_3 = a_2 \vee a_3$$

Автомат Мура

6. Логічна схема автомата Мура:

Код нульового стану задається на виходах тригера по сигналу Start.



Автомат Мілі

Розрізняють три основні типи схемного пристроїв:

1. **Структурна схема** - найбільш загальне позначення складових частин схеми й зв'язків між ними.
2. **Функціональна схема**
3. **Принципова схема**.

Функціональна схема показує логічну послідовність і взаємозв'язок елементів пристрою. Ступінь подробиць функціональної схеми може бути різним (у порядок від структурної до принципової).

Керуючий автомат Мілі характеризується наступними функціями:

1. **Функція переходу** $\delta = f(A, Z)$;
2. **Функція вихідних сигналів** $\lambda = f(A, Z)$.

A – множина станів (внутрішній алфавіт)

Z – множина вхідних сигналів (вхідний алфавіт)

1. Автомат Мура, де

Функція переходу - $\delta = f(A, Z)$;

Функція вихідних сигналів - $\lambda = f(A)$.

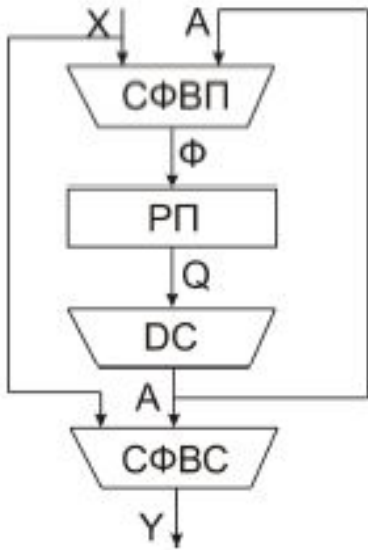
2. Автомат Мілі, де

Функція переходу - $\delta = f(A, Z)$;

Функція вихідних сигналів - $\lambda = f(A, Z)$.

Автомат Мілі

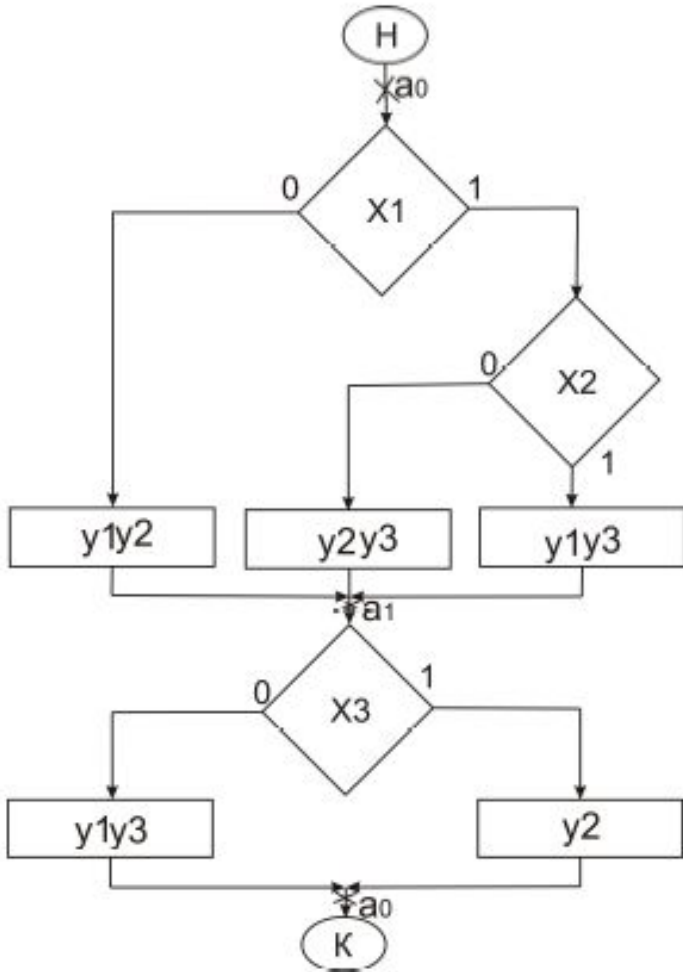
Загальна структура автомата Мілі наступна:



СФВС - схема формування вихідних сигналів.

Методика синтезу автомата Мілі включає ті ж пункти, що й методика синтезу Автомата Мура.

Автомат Мілі



Наприклад, маємо алгоритм.

Крок 1. Відмітка станів автомата Мілі.

Стан a_0 відмічається до початку та після кінця виконання алгоритму.

Станом a_i автомата Мілі відмічається вхід кожної вершини, яка іде за операторною (або навпаки операторних вершин).

У загальному випадку число станів автомата Мілі менше числа станів автомата Мура.

Множина станів у цьому прикладі: $A = \{a_0, a_1\}$, $M = 2$.

Необхідна розрядність коду стану:

$R = \lceil \log_2 2 \rceil = 1 \rightarrow \{Q_1\}$, тобто потрібно один тригер .

Автомат Мілі

Крок 2. Таблица кодів станів :

a_m	Q_1
a_0	0
a_1	1

Крок 3. Таблица переходів автомата Мілі:

Текущее состояние		Последующее Состояние		X	Y	T_i
Состояние	Код	Состояние	Код			
a_0	0	a_1	1	$\overline{x_1}$	$y_1 y_2$	T_1
		a_1	1	$x_1 \overline{x_2}$	$y_2 y_3$	T_1
		a_1	1	$x_1 x_2$	$y_1 y_3$	T_1
a_1	1	a_0	0	$\overline{x_3}$	$y_1 y_3$	T_1
		A_0	0	x_3	y_2	T_1

Автомат Мілі

Крок 4. Формування сигналів для збудження пам'яті ФВП:

$$T_1 = a_0 \cdot \overline{x_1} \vee a_0 \cdot x_1 \cdot \overline{x_2} \vee a_0 \cdot x_1 \cdot x_2 \vee a_1 \cdot \overline{x_3} \vee a_1 \cdot x_3 = a_0 \cdot \overline{x_1} \vee a_0 \cdot x_1 \vee a_1 = a_0 \vee a_1$$

Крок 5. Формування вихідних сигналів (мікрооперацій) ФВС (МО).

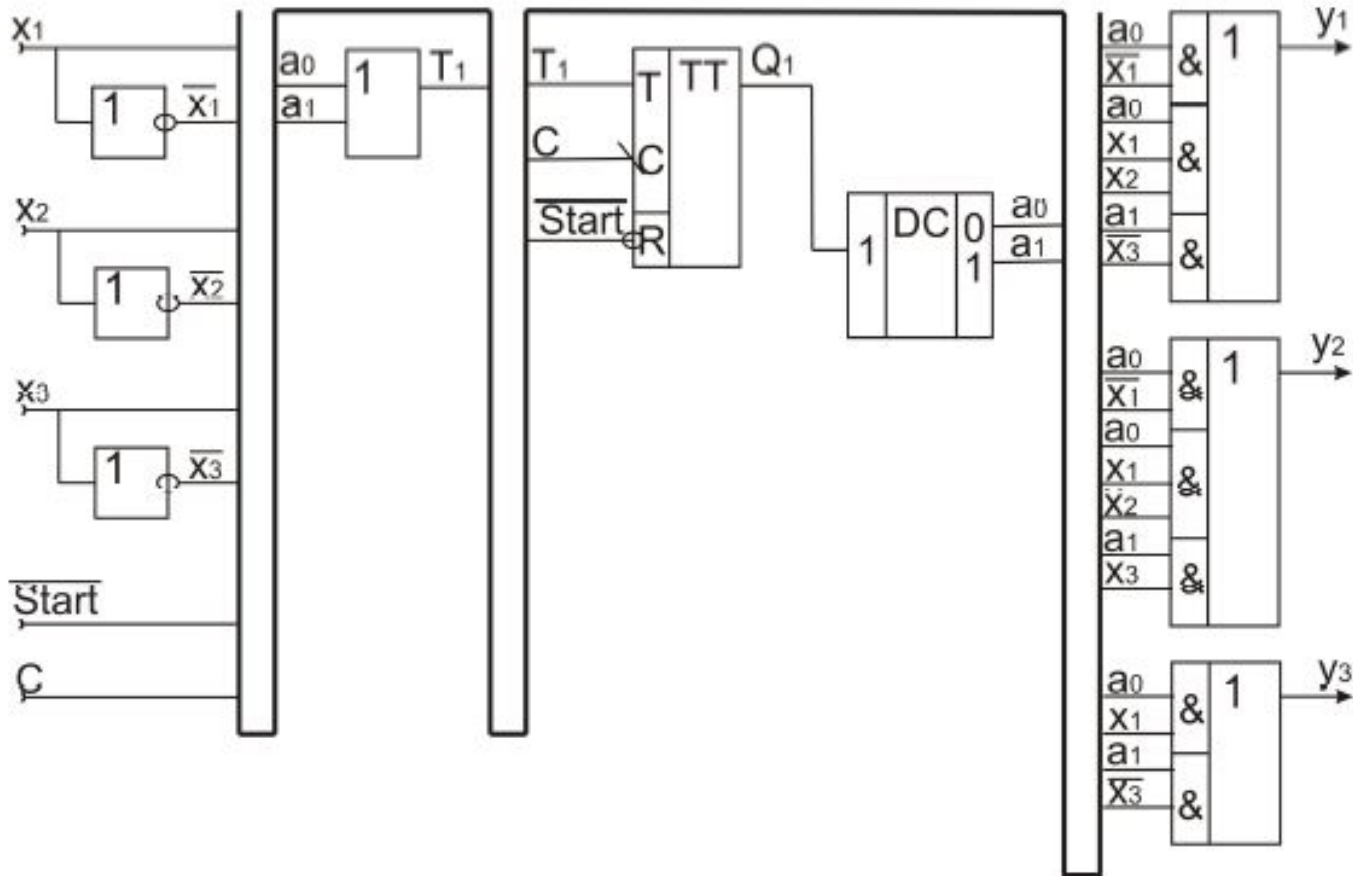
$$y_1 = a_0 \cdot \overline{x_1} \vee a_0 \cdot x_1 \cdot \overline{x_2} \vee a_1 \cdot \overline{x_3}$$

$$y_2 = a_0 \cdot \overline{x_1} \vee a_0 \cdot x_1 \cdot \overline{x_2} \vee a_1 \cdot x_3$$

$$y_3 = a_0 \cdot x_1 \cdot \overline{x_2} \vee a_0 \cdot x_1 \cdot x_2 \vee a_1 \cdot \overline{x_3} = a_0 \cdot x_1 \vee a_1 \cdot \overline{x_3}$$

Автомат Мілі

Крок 6. Логічна схема автомата Мілі:

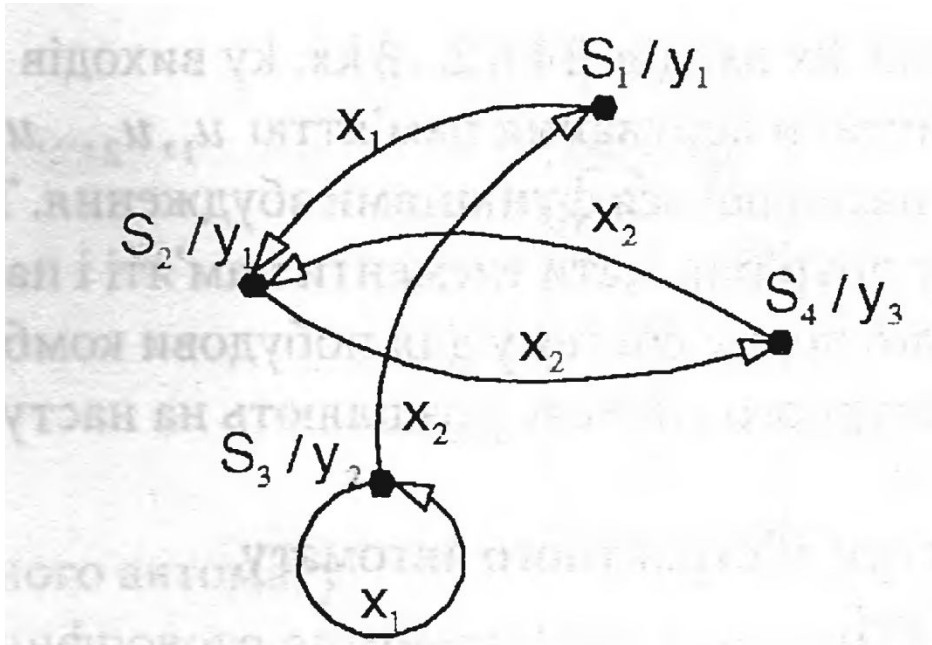


Автомати Мілі чи автомати Мура?

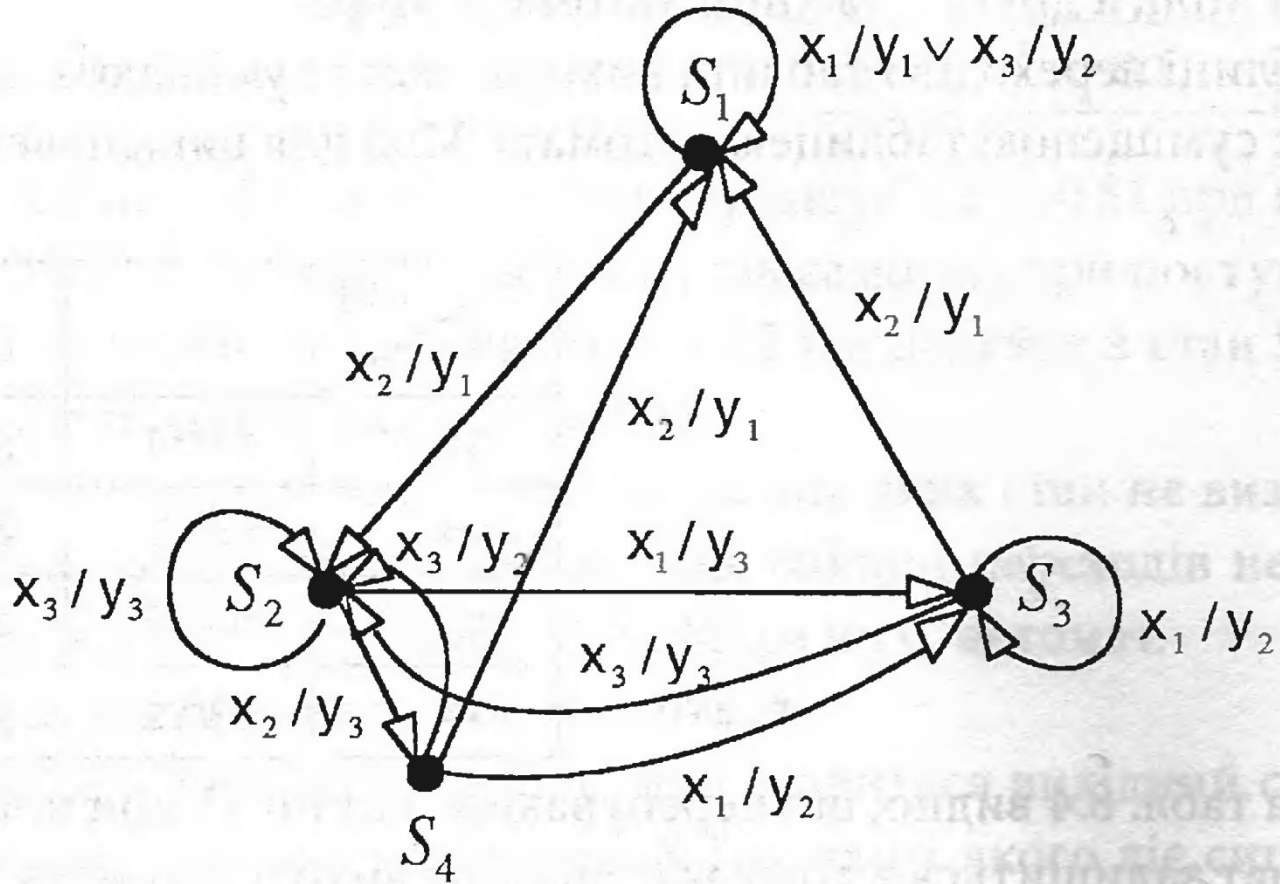
Для правильної роботи схем, напевне, не варто дозволяти, щоб сигнали на вході запам'ятовуючих елементів безпосередньо брали участь в утворенні вихідних сигналів, які по ланцюгах зворотного зв'язку подавалися б у той же самий момент часу на ці входи.

У зв'язку із цим запам'ятовуючими елементами повинні бути не автомати Мілі, а автомати Мура.

Граф автомату Мура

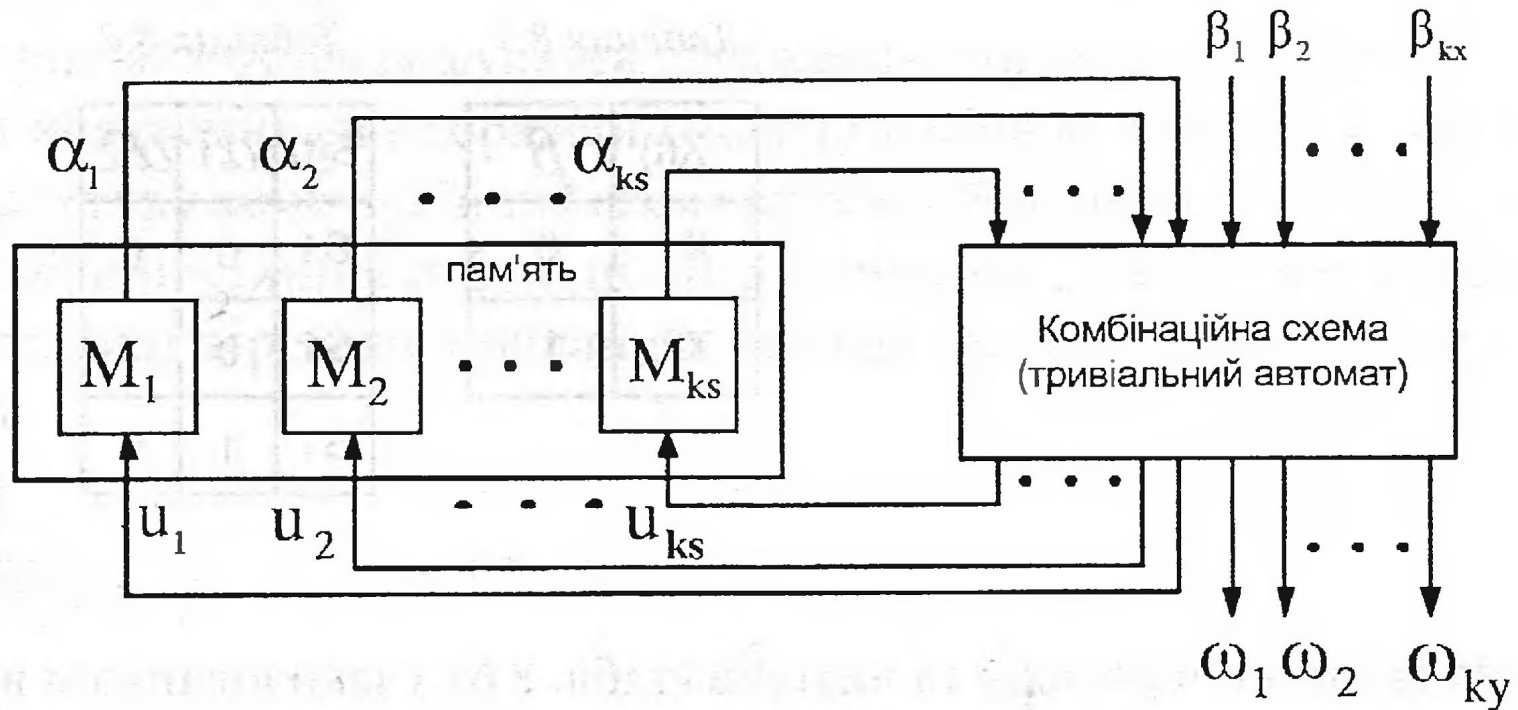


Граф автомату Мілі



Вихідні сигнали ставляться на дугах графа разом із вхідними сигналами

Загальна структура елементарного автомату

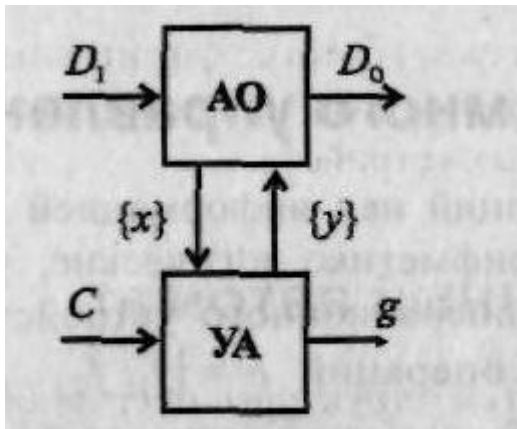


Синтез автомата Мура на ПЗУ

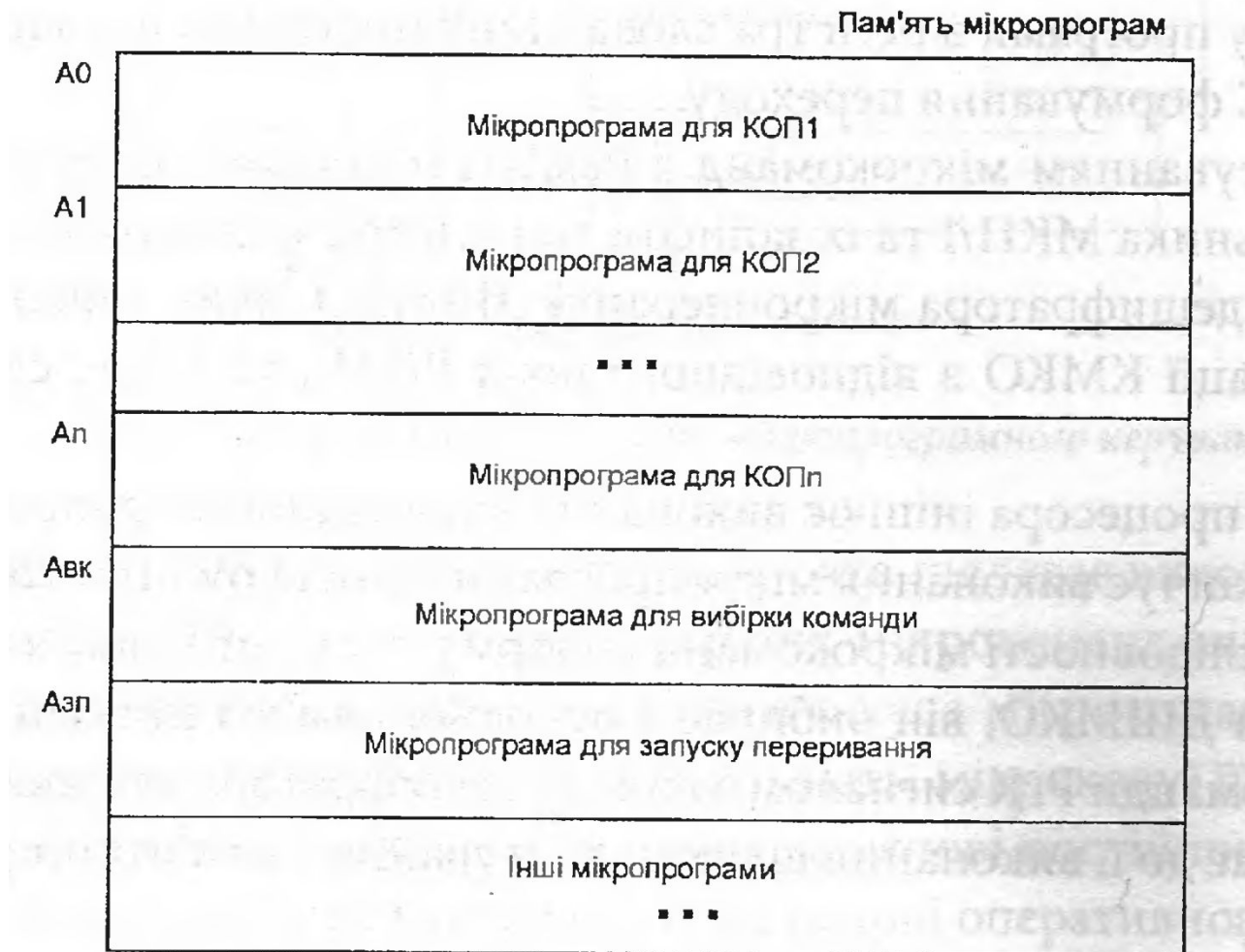
Принцип мікропрограмного керування (Жмакін)

Синтез керуючого автомата на основі програмованої логіки

(стр.107)



Розміщення мікрокоманд послідовно в пам'яті



Організація пам'яті мікрокоманд



МКПЛ-
мікропрограмний
лічильник.
РгМК- реєстр
мікрокоманд

*Пам'ять мікрокоманд з реєстром адреси мікрокоманди на вході
та реєстром мікрокоманди на виході*

Одновібратори й генератори

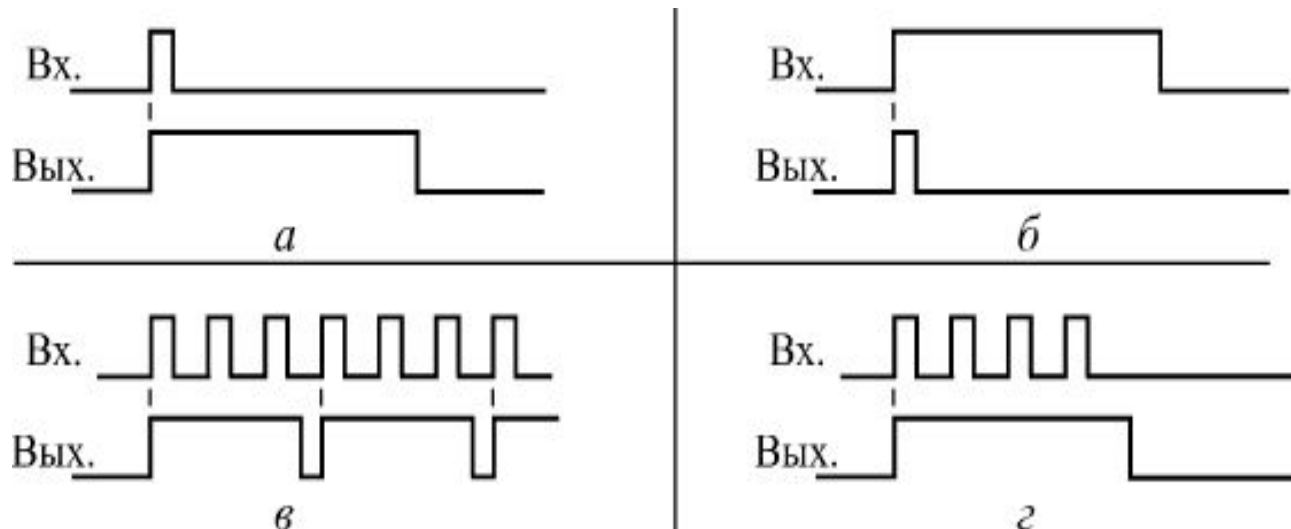
2. Що таке мікрооперація?
3. Що таке мікронаказ?
4. Що таке мікрокоманда?
5. Що таке мікропрограма?

Застосування одновібраторів

Найпоширеніші застосування одновібраторів наступні (рис):

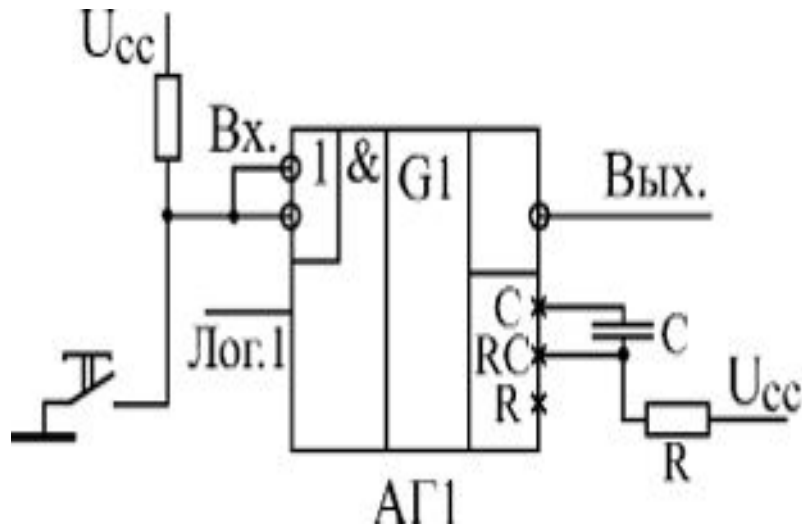
1. збільшення тривалості вхідного імпульсу;
2. зменшення тривалості вхідного імпульсу;
3. ділення частоти вхідного сигналу в задане число разів;
4. формування сигналу послідовності, що огинає, вхідні імпульси.

Для збільшення або зменшення тривалості вхідного сигналу (а і б) треба усього лишень вибрати опір резистора і ємність конденсатора, виходячи з необхідної тривалості вихідного сигналу. У цьому випадку можна використовувати одновібратор будь-якого типу: як з перезапуском, так і без перезапуску.



Застосування одновібраторів

Ще одне важливе застосування одновібратора полягає в придушенні дребезга контактів кнопки. Одновібратор з великим часом затримки (порядку декількох десятих часток секунди) надійно пригнічує паразитні імпульси, що виникають через дребезг контактів, і формує ідеальні імпульси при будь-якому натисканні кнопки

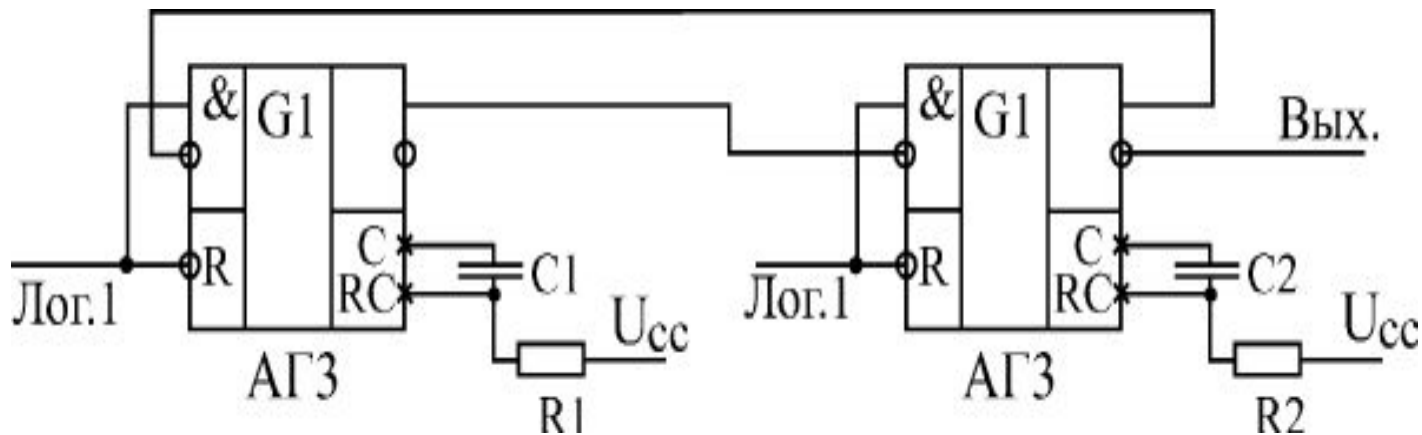


Можна використовувати як одновібратор з перезапуском, так і одновібратор без перезапуску (на малюнку). Можна також підібрати час затримки так, що одновібратор буде видавати один імпульс при натисканні кнопки, а інший імпульс - при відпусканню кнопки. Іноді це буває зручніше.

Застосування одновібраторів

Одновібратори можна також застосовувати для побудови **генераторів (мультивібраторів) прямокутних імпульсів** з різними значеннями тривалості імпульсів і паузи між ними.

При цьому два одновібратори замикаються в кільце так, що кожен з них запускає інший після закінчення свого вихідного імпульсу (рис.). Один одновібратор формує тривалість імпульсу, а інший визначає паузу між імпульсами. Змінюючи номінали резисторів і конденсаторів, можна одержати потрібні співвідношення імпульсу й паузи.



Генератор імпульсів на двох одновібраторах

Застосування одновібраторів

Таким чином, одновібратори досить легко дозволяють вирішувати самі різноманітні задачі. Однак, застосовуючи одновібратори, треба завжди пам'ятати, що тривалість їх вихідних імпульсів **не можна задати дуже точно** - адже одновібратор має аналогові ланки. На тривалість вихідного імпульсу одновібратора впливають номінали резисторів і конденсаторів, температура навколишнього середовища, старіння елементів, перешкоди в ланках живлення й інші фактори. Тому застосування одновібраторів потрібно по можливості обмежувати тільки тими випадками, коли час затримки можна задавати з не занадто високою точністю (погрішність не менш 20-30%).

Будь-яку функцію одновібратора може виконати синхронні пристрої (на основі кварцового генератора, тригерів, регістрів, лічильників), причому виконати набагато точніше й надійніше. І їм не потрібно ніяких додаткових елементів, що задають час (резисторів і конденсаторів).

Генератори з автопідстройкою частоти

Окрім одновібраторів, розробники використовують також спеціалізовані генератори ("мультивібратори", англ. "multivibrator"). Позначаються вони на схемах буквою **G**.

Наприклад, мікросхема ГГ1 являє собою два генератори в одному корпусі. Мікросхеми генераторів використовують досить рідко, частіше застосовують генератори на інверторах або на тригерах Шмітта.

Однак у деяких випадках генератори ГГ1 не можуть бути замінені нічим. Справа в тому, що вони допускають зміну частоти вихідних імпульсів за допомогою рівнів двох вхідних керуючих напруг. Тому вони називаються також "генератори, що керовані напругою". Ефект зміни частоти можна використовувати, наприклад, у системах автопідстроювання частоти (АПЧ) або в пристроях із частотною модуляцією (ЧМ).

Схема включення генератора ГГ1

