Відокремлений підрозділ
Національного університету біоресурсів і природокористування України
“Ніжинський агротехнічний інститут”

Методичні рекомендації з дисципліни “Комп’ютерна логіка” для самостійної роботи студентів спеціальності “Обслуговування комп’ютерних систем і мереж”

Ніжин – 2012

ББК 32.973.26-04 Я73
М54

Рекомендовано вченою радою Відокремленого підрозділу Національного університету біоресурсів і природокористування України "Ніжинський агротехнічний інститут" протокол № 4 від 27 грудня 2012 року

Рецензенти: кандидат технічних наук, асистент кафедри прикладної математики, інформатики та освітніх вимірювань Ніжинського державного університету ім. Миколи Гоголя Чернишова Е.О.
к.т.н., старший викладач кафедри загальнотехнічних дисциплін ВП НУБіП України "Ніжинський агротехнічний інститут" Коровіна І.О.

Голуб Т.Б., Заболотній О.А. Методичні рекомендації з дисципліни "Комп'ютерна логіка" для самостійної роботи студентів спеціальності "Обслуговування комп'ютерних систем і мереж"

В методичних рекомендаціях наведено основи поняття теорії автоматів, способи побудови алгоритмів роботи таких автоматів.

Зміст

Основні поняття теорії абстрактних автоматів. Автомати Мілі та Мура.

Основні поняття та визначення.

Теорія автоматів має широкі можливості застосування:

Проектування систем логічного керування;

Обробка текстів та побудова компіляторів;

Специфікація і верифікація систем взаємодіючих процесів;

Мови опису документів і об'єктно-орієнтованих програм;

Оптимізація логічних програм ін

Найпростіший перетворювач інформації (рис.1.1, а) відображає деякий безліч елементів інформації X, що надходить на вхід, в деякий безліч на виході Y. Якщо безлічі X і Y є кінцевими і дискретними, тобто перетворення здійснюється в дискретні моменти часу, то такі перетворювачі інформації називаються кінцевими перетворювачами. Елементи множин X і Y в цьому випадку попередньо кодують двійковими кодами і будують перетворення одного безлічі в інше.

Результат перетворення $F: X \to Y$ найчастіше залежить не тільки від того, яка інформація в даний момент з'явилася на вході, але й від того, що відбувалося раніше, тобто від передісторії перетворення. Наприклад, один і той же вхід - вибачення сусіда після того, як він вам наступив на ногу в переповненому автобусі - викличе у вас одну реакцію в перший раз і зовсім іншу - в п’ятий раз.

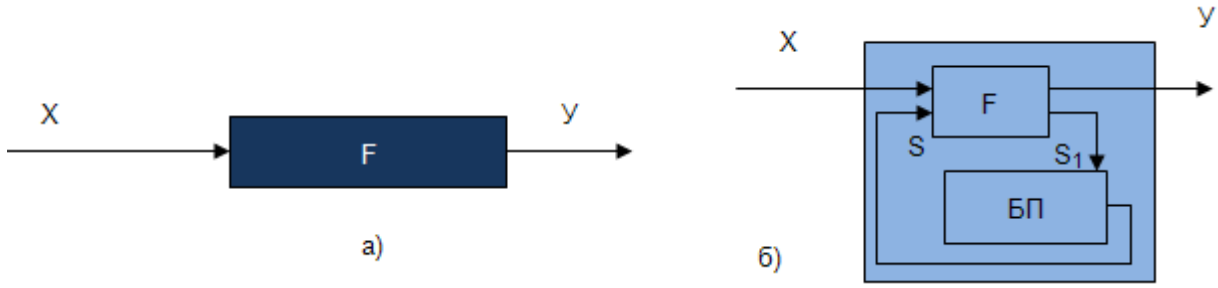


Рис. 1.1.

Таким чином, існують більш складні перетворювачі інформації (ПІ), реакція яких залежить не тільки від вхідних сигналів в даний момент, але і від того, що було раніше, від вхідних історії. Такі ПІ називаються автоматами (схемами з пам'яттю). У цьому випадку говорять про автоматному перетворенні інформації (рис.1.1, б). На один і той же вхідний сигнал автомат може реагувати по-різному, в залежності від стану, в якому він знаходився. Автомат змінює свій стан з кожним вхідним сигналом.

Розглянемо кілька прикладів.

Приклад 1. Автомат, що описує поведінку “розумного“ батька.

Опишемо поведінку батька, син якого навчається в школі і приносить двійки і п’ятірки. Батько не хоче хапатися за ремінь кожен раз, як тільки син отримає двійку, і вибирає більш тонку тактику виховання. Задамо такий автомат графом, у якому вершини відповідають станам, а дуга зі стану S в стан Q з позначенням X/Y, проводиться тоді, коли автомат з стану S під впливом вхідного сигналу X переходить у стан Q з вихідною реакцією Y. Граф автомата, моделює розумну поведінка батька, представлений на рис.1.2. Цей

автомат має чотири стани $\{S_0, S_1, S_2, S_3\}$, два вхідних сигналу $\{X_2, X_5\}$– оцінки та вихідні сигнали$\{Y_0, Y_1, Y_2, Y_3, Y_4\}$, які будемо інтерпретувати як дії батька таким чином:

Y_0 - брати ремінь

Y_1 - кричати на сина

Y_2 - заспокоювати сина

Y_3 - радіти

Y_4 - тріумфувати

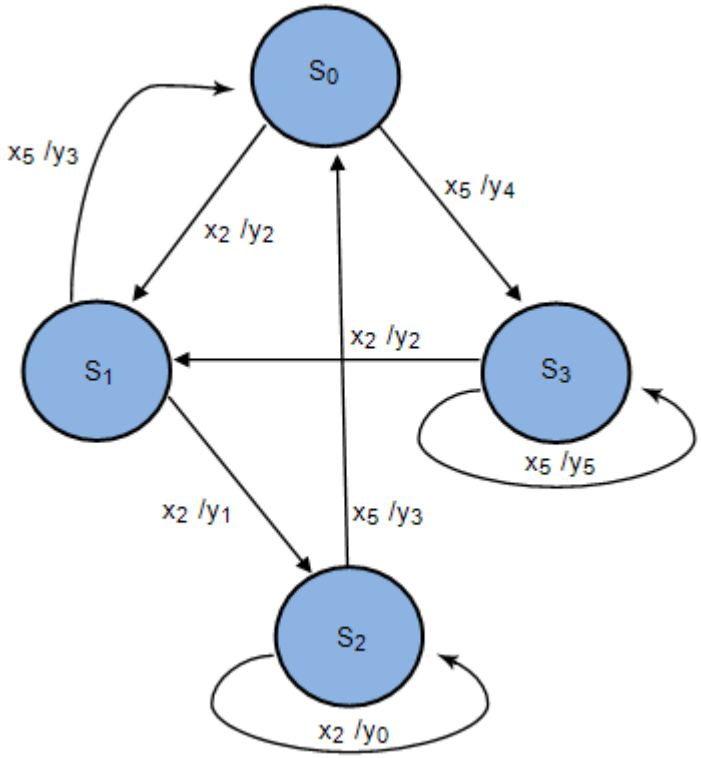


Рис. 1.2.

Сина, який отримав одну й ту ж оцінку – двійку, чекає вдома абсолютно різна реакція батька в залежності від передісторії його навчання. Наприклад, після третьої двійки $\{X_2, X_2, X_2\}$ в історії сина зустрінуть ременем, а в історії будуть $\{X_2, X_2, X_2, X_5\ X_2\}$заспокоювати і т.д.

Кінцевий автомат може бути реалізований як програмно, так і апаратно. Програмну реалізацію можна виконати на будь-якій мові високого рівня різними способами. На рис.1.3 представлена блок-схема програми, що реалізує поведінку автомата прикладу 1. Неважко побачити, що топологія блок-схеми програми повторює топологію графа переходів автомата. З кожним станом пов’язана операція NEXT, що виконує функцію очікування чергового події приходу нового вхідного сигналу і читання його в певний стандартний буфер x, а також подальший аналіз того, який це сигнал. В залежності від того, який сигнал прийшов на вхід, виконується та чи інша функція $Y_0 \ldots Y_5$ і відбувається перехід до нового стану.

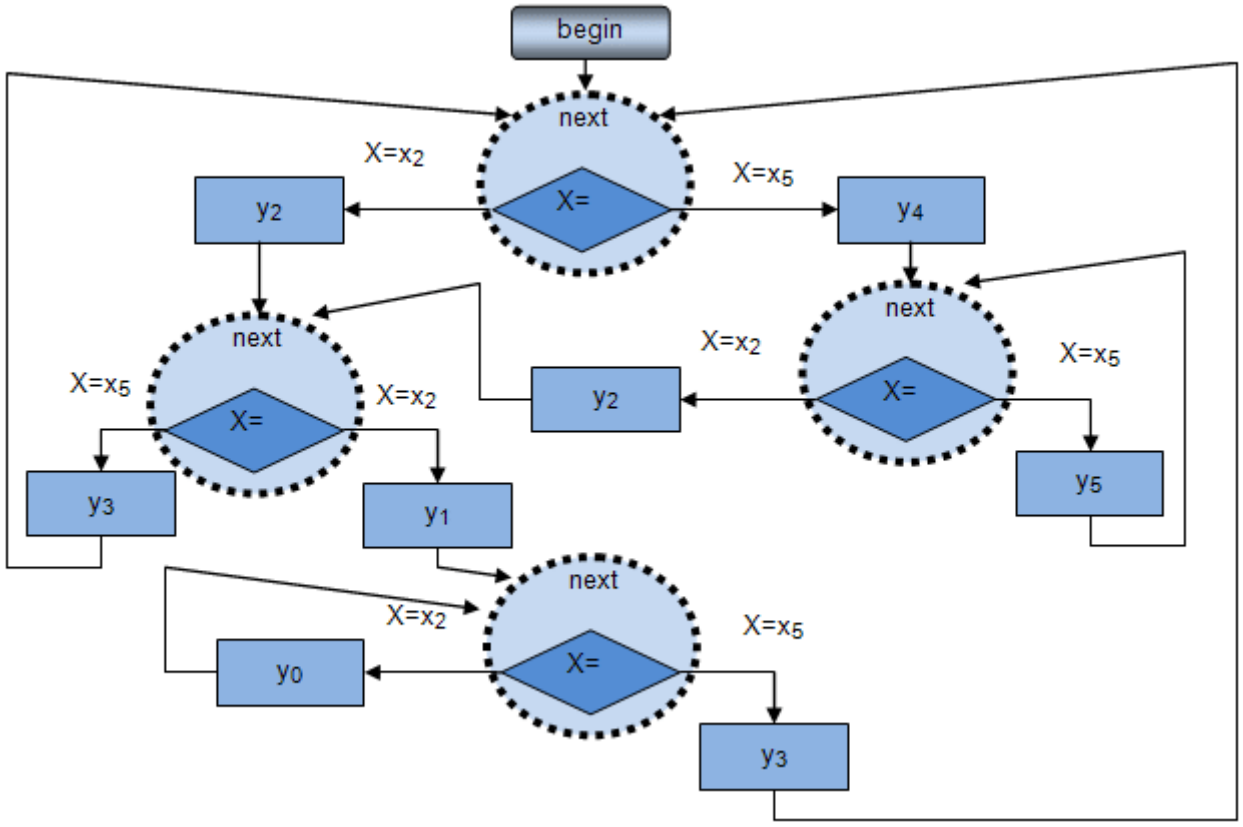


Рис. 1.3.

Апаратну реалізацію автомата розглянемо у другій частині цього розділу.

Приклад 2. “Студент”

Автомат, модель якого представлена на рис.1.4, описує поведінку студента і викладачів.

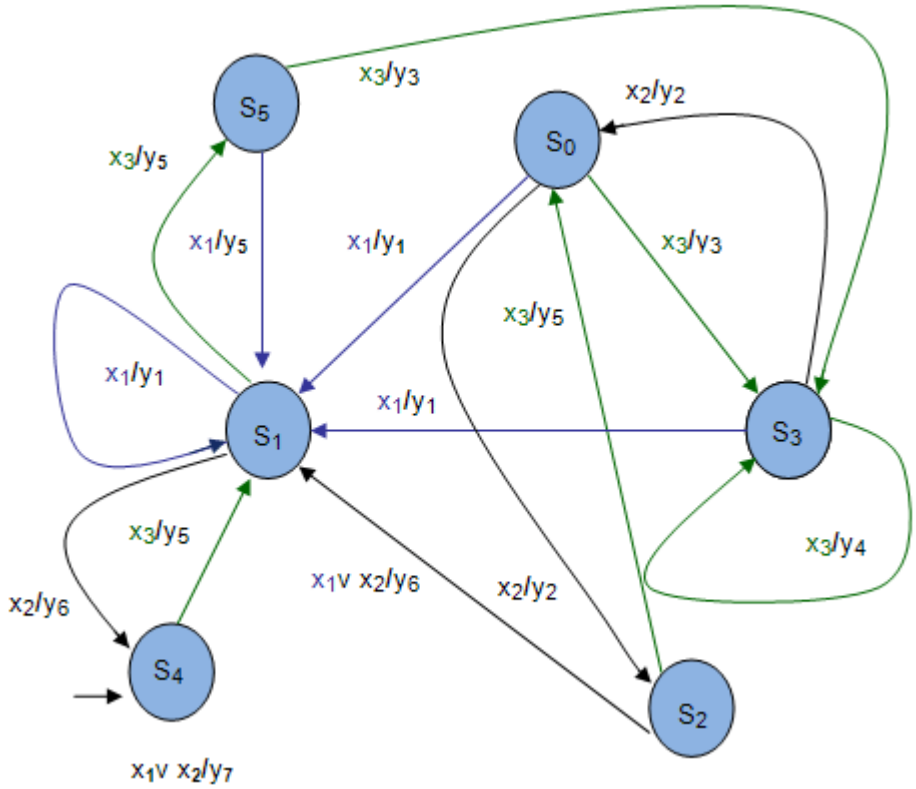


Рис. 1.4

$\{S_0, S_1,...,S_5\}$ - стани

$\{X_1, X_2, X_3\}$ - вхідні сигнали “н”, “2”, “5”

$\{Y_1, Y_2,...,Y_7\}$ - вихідні реакції

Y_1 - відмічаємо “н”

Y_2 - заспокоювати

Y_3 - хвалити студентів

Y_4 - заохочуємо

Y_5 - сподіваємося

Y_6 - попереджаємо

Y_7 - відраховуємо

Приклад 3. Електронні годинники (рис.1.5).

Електронні годинники різноманітних функціональних можливостей є одним з найбільш широко застосовуваних у побуті електронних приладів, управління якими побудовано на основі кінцевого автомату моделі. Вони зазвичай показують: час, дату, у них є функції такі як “установка часу і дати”, “Секундомір”, “Будильник” і т.п. Спрощена структурна схема електронного годинника показана на рис.1.5

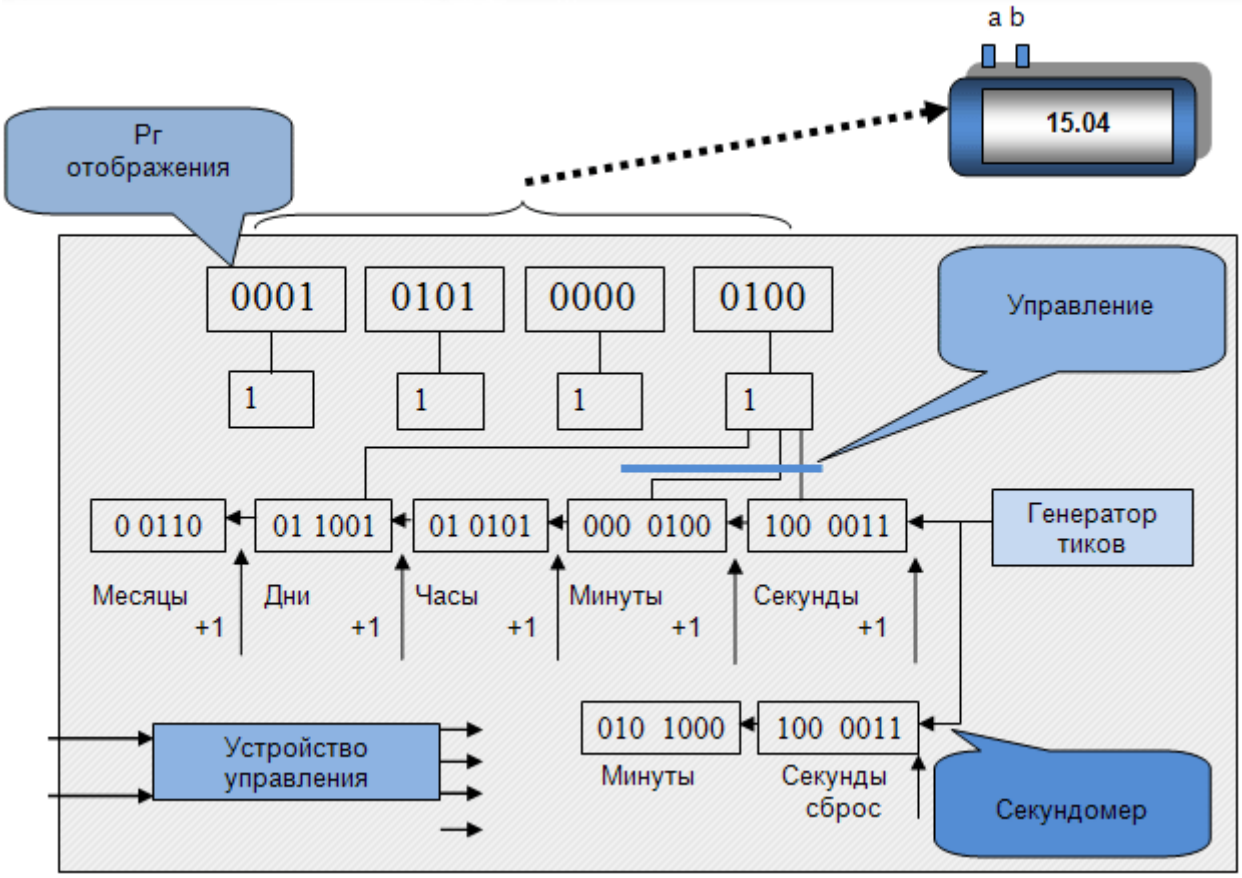


Рис. 1.5.

Регістри відображають або час, або дату, або секундомір в залежності від “Управління”. Пристрій управління побудовано на основі моделі кінцевого автомата. Кінцевий автомат реагує на натиснення кнопки “а” на корпусі переходом в стан “Установка хвилин”, в якому подія натиснення кнопки “б”

викличе збільшення числа. Пристрій управління побудовано на основі моделі кінцевого автомата, граф якого показаний наріс.1.6

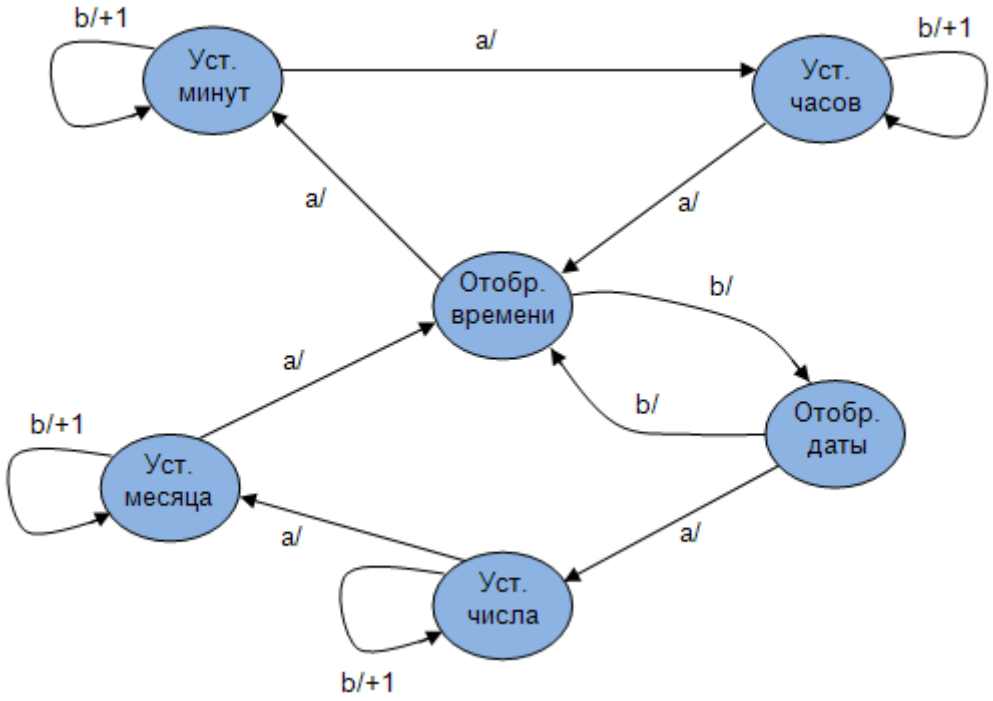


Рис. 1.6.

Отже. З одного боку АВТОМАТ – пристрій, що виконує деякі дії без участі людини. З іншого боку, АВТОМАТ - математична модель, що описує поведінку технічного пристрою. В даному випадку реальний пристрій, система і т.д. розглядається як деякий “ЧОРНИЙ ЯЩИК” (рис.1.7).

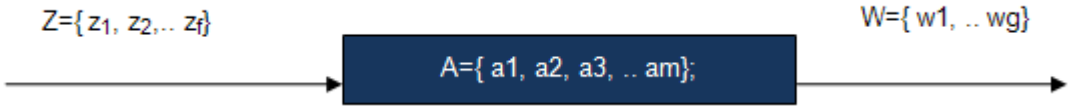


Рис. 1.7.

Абстрактна теорія автоматів — напрям в теорії автоматів, який характеризується тим, що при вивченні автоматів не беруть до уваги їх структурні особливості. За такого підходу, внутрішні стани автомата, його вхідні та вихідні сигнали розглядаються як деякі абстрактні символи, які утворюють відповідно алфавіти: Q (внутрішній), X (вхідний), Y (вихідний). X та Y вважаються скінченими алфавітами. Q може бути нескінченним.

Абстрактний автомат – це математична модель, що описує технічний пристрій сукупністю вхідних, вихідних сигналів і станів, крім того:

має безліч внутрішніх станів $A=\{a_1(t), a_2(t), \dots a_m(t)\}$, які називаються станами автомата;

на вхід автомата надходить кінцеве безліч вхідних сигналів $Z=\{z_1(t), z_2(t), \dots z_j(t)\}$, є кінцева множина вихідних сигналів $W=\{w_1, w_2, \dots w_q\}$;

задана функція переходу $\delta(a, z)$;

задана функція формування виходів автомата $\lambda(a, z)$;

визначено початковий стан автомата $a_1 \in A$.

Тобто для опису автомата потрібно використовувати шістку виду $S = \{A, Z, W, \delta, \lambda, a_1\}$, де

$A = \{a_1(t), a_2(t), \dots a_m(t)\}$

$Z = \{z_1(t), z_2(t), \dots z_j(t)\}$

$W = \{\omega_1, \omega_2, \dots \omega_q\}$

$\delta : A \cdot Z \to A(a_S = \delta(a_m, Z_i) \mid a_S \in A)$

$\lambda : A + Z \to W(w_S = \lambda(a_m, Z_i) \mid a_S \in A, w \in W)$

$a_1 \in A$

Автомат реалізує деяке відображення множини слів вхідного алфавіту Z в безліч слів вихідного алфавіту W.

Автомат називається **кінцевим**, якщо безліч його внутрішніх станів і безліч значень вхідних сигналів є кінцеві безлічі.

Автомат називається **синхронним**, якщо інтервал тимчасової дискретизації постійний, в іншому випадку говорять про асинхронному автоматі.

Автомат називається **детермінованим**, якщо поведінка автомата в кожний момент часу однозначно визначено (s_i, a_i)

В залежності від способу визначення вихідного сигналу в **синхронних** автоматах розрізняють:

Автомат першого роду (Автомат Мілі)

1. $a(t+1) = \delta(a(t), Z(t))$, де $t = 0, 1, 2, \dots$
2. $W(t) = \lambda(a(t), Z(t))$.

Автомат другого роду (Автомат Мура)

1. $A(t+1) = \delta(a(t), Z(t))$,
2. $W(t) = \lambda(a(t))$. де $t = 0, 1, 2, \dots$
3. $W(t+1) = \lambda(a(t+1), = \lambda(\delta(a(t),\ Z(t)))$.

Питання для самостійного опрацювання:

1. Методи завдання автоматів
2. Теоретико-множинне задання автоматів.
3. Таблична форма.
4. Графова форма завдання абстрактних автоматів
5. Матрична форма.

Еквівалентні автомати. Перетворення еквівалентних автоматів. Мінімізація автоматів.

Визначення скінченого автомата

Розглянемо абстрактну систему, традиційно називану математиками “чорна скриня” (рис. 1), що перетворює подавані на її вхід символи *x* певної абетки *X* на вихідні символи *y* абетки *Y*. Таким чином можна подати поводження більшості дискретних систем, застосовуваних у різних галузях техніки, надто обчислювальної.

Рисунок 1 – “Чорна скриня”

Побудуємо скінченний автомат, керуючий ліфтом у триповерховому будинку. Вхідна абетка автомата складається з натискання кнопки виклику відповідного поверху: $X = \{C1, C2, C3\}$; вихідна абетка складається з переміщень на один чи два поверхи нагору чи донизу, а також зупинки ліфта: $Y = \{U1, U2, D1, D2, S\}$; стан відповідає поверхові, на якому перебуває автомат: $Q = \{q_1, q_2, q_3\}$; для визначеності оберемо початковий стан q_1. Функцію переходів автомата зручно подавати діаграмою станів (рис. 2).

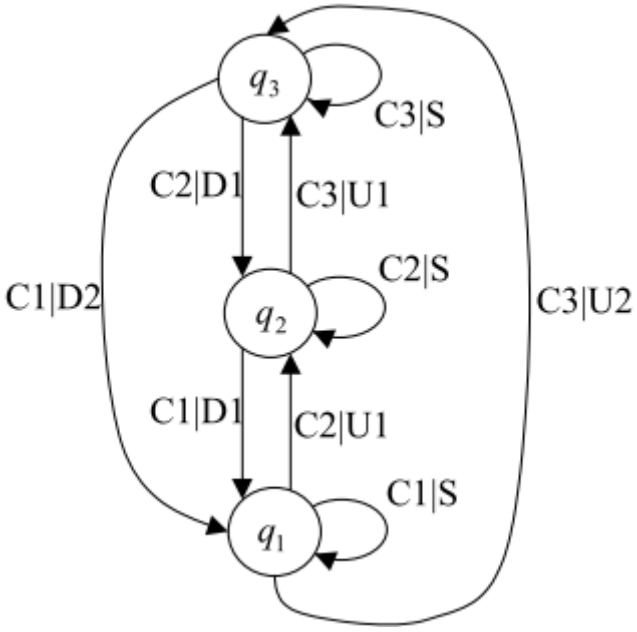


Рисунок 2 – Автомат для керування ліфтом

Отже, скінченний автомат – це п’ятірка $A = \{X, Q, Y, q_0, F\}$), де $X = \{x\}$ скінченна вхідна абетка, $Y = \{y\}-$ скінченна вихідна абетка, $Q = \{q\}$ }множина внутрішніх станів, q_0 – початковий стан, $F : X \cdot Q \to Y \cdot Q$ – функція переходів. Буває також матричне подання функції переходів. Наведемо матрицю переходів для побудованого раніше автомата.

Q / X	$C1$	$C2$	$C3$
q_1	$q_1 \mid S$	$q_2 \mid U1$	$q_3 \mid U2$
q_2	$q_1 \mid D1$	$q_2 \mid S$	$q_3 \mid U1$
q_3	$q_1 \mid D2$	$q_2 \mid D1$	$q_3 \mid S$

Під синтезом, як правило, розуміють побудову скінченного автомата за певною заданою його специфікацією. Специфікація, як у нашому прикладі з ліфтом, може бути словесною. Іноді використовують формальні специфікації, наприклад, у вигляді регулярних виразів. Синтез автомата полягає в діставанні його формалізованого опису – діаграми станів або матриці переходів. У

схемотехніці під синтезом автомата іноді розуміють реалізацію автомата на заданій елементній базі.

Розглядання простого прикладу, поданого на рис. 3, дозволяє зробити висновок, що той самий автомат може мати безліч подань з різною кількістю станів та переходів. Так, наприклад, стан s_4 є недосяжний з початкового стану і тому може бути вилучений, а стани s_2 та s_3 можна поєднати; у результаті одержимо автомат для розпізнавання цілих чисел. Дістати мінімальне подання автомата є надто важливо, якщо надалі його буде реалізовано апаратно. Окрім того, виникає задача для "схожих" автоматів визначити формально, чи є вони еквівалентні.

Надалі розглядатимемо автомати, які не мають недосяжних станів, оскільки такі стани може бути вилучено з діаграми за допомогою методів теорії графів. Методи мінімізації й визначення еквівалентності буде побудовано для автоматів Мура, що, як було відзначено в попередньому розділі, не обмежує їхньої загальності.

Два стани автомата s_1 та s_2 називатимемо *n-еквівалентними*, якщо для довільного вхідного ланцюжка σ довжиною n символів вихідні ланцюжки символів збігаються. Відношення *n-еквівалентності* позначимо $\underset{n}{\equiv}$.

Два стани автомата s_1 та s_2 називатимемо еквівалентними, якщо вони є n-еквівалентні для будь якого цілого n. Відношення еквівалентності позначимо $\equiv$.

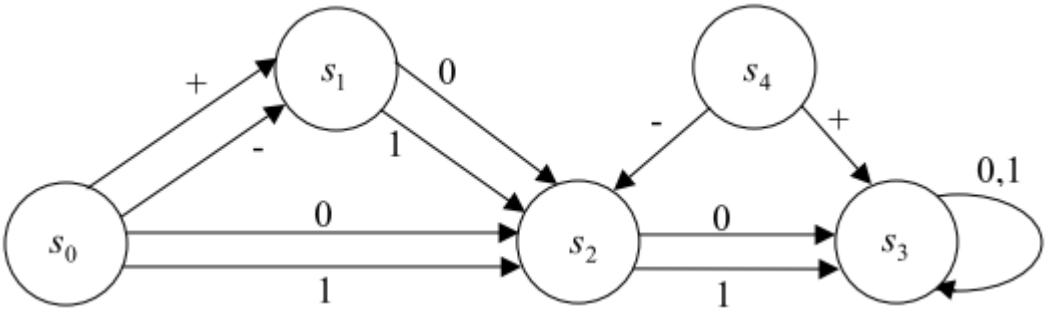


Рисунок 3 – Надлишковий автомат для розпізнавання цілих чисел

Теорема 1 У скінченному автоматі з m станами два довільних стани є еквівалентні тоді й лише тоді, коли $(m-2)$-еквівалентні.

Теорема дозволяє запропонувати простий алгоритм мінімізації скінченних автоматів. Алгоритм складається в послідовній побудові розбиття множини станів на одно-, дво-, три- й так далі еквівалентні. Якщо поточне розбиття збігається з попереднім, то дістані класи еквівалентності й визначають мінімальний автомат. У таблиці розбиття зазначено номер класу наступного стану. Первісне розбиття визначається функцією виходу: нульеквівалентні стани мають однакові вихідні символи.

Проілюструємо описаний процес для автомата, поданого на початку цього підрозділу (недосяжний стан s_4 не розглядається):

Клас	Стан	+	-	0	1
К0	s_0	К0	К0	К1	К1

	s_1			К1	К1
К1	s_2			К1	К1
	s_3			К1	К1

Клас К0 розщеплюється на два нових еквівалентних класи, тому що його рядки не збігаються:

Клас	Стан	+	-	0	1
К0	s_0	К1	К1	К2	К2
К1	s_1			К2	К2
К2	s_2			К2	К2
	s_3			К2	К2

Подальшого розщеплення класів еквівалентності не відбувається, тому мінімальний автомат містить три стани.

Алгоритм мінімізації автоматів дозволяє також визначити їхню еквівалентність. Дійсно, якщо необхідно довідатися, чи є еквівалентні автомати B_1 та B_2 , доволі виконати мінімізацію автомата $B_1 \cup B_2$ й визначити, чи попадають їхні початкові стани до того самого класу еквівалентності.

Отже, скінченні автомати є простим та ефективним засобом подання дискретних систем. Вони широко використовуються в різних галузях техніки. Слід пам’ятати, що будь-який, навіть надто потужний суперкомп'ютер являє собою скінченний автомат. Нехай кількість його станів обчислюється мільярдами, але вона завжди є скінченна.

Метод мінімізації абстрактних автоматів застосовується і для мінімізації повністю визначених мікропрограмних автоматів.

Якщо два стани автомати Мілі сумісні, то під дією одинаково вхідних сигналів вони видають однакові вихідні сигнали. З цієї причини для виявлення сумісних станів микропрограммного автомата Мілі необхідно порівняти безліч мікрокоманд, які видаються на переходах з цих станів.

Для мінімізації мікропрограмного автомата Мура необхідно передньо знайти розбиття станів на класи 0 - сумісності. У кожен такий клас потраплять стану, відзначені однаковою мікрокоманд.

Далі процес мінімізації мікропрограмного автомата повністю збігається з процесом мінімізації абстрактних автоматів Мілі і Мура.

Таким чином, методи мінімізації мікропрограмних і абстрактних автоматів дуже близькі. Різниця полягає в тому, що замість порівняння абстрактних вхідних/вихідних сигналів порівнюються конкретні булеві функції, відповідні вихідним сигналам мікропрограмних автоматів.

Питання для самостійного опрацювання:

1. Зазначте основні області застосовування скінченних автоматів.
2. Назвіть скінченні автомати в оточуючих нас штучних об'єктах.
3. У чому полягає доцільність використання двох різних типів скінченних автоматів?
4. Які автомати називають еквівалентними?

5. Схарактеризуйте основні кроки алгоритму мінімізації скінченного автомата?

6. Як визначити, чи є скінченні автомати еквівалентними?

Способи опису роботи дискретних пристроїв. Способи представлення мікропрограм.

Всі пристрої, які оперують з двійковою (дискретною) інформацією, поділяються на великий клас: комбінаційні схеми (дискретні автомати без пам'яті) і послідовні пристрої (дискретні автомати з пам'яттю).

Комбінаційні схеми.

Комбінаційною схемою чи логічним пристроєм називають такий пристрій, який має сигнали на виходах в будь-який момент часу, які однозначно визначаються поєднанням сигналів на входах і залежить від попередніх станів даного пристрою.

Схемною ознакою таких пристроїв є відсутність ланцюгів зворотнього зв'язку, тобто замкнутих петель для проходження сигналів з виходів пристрою в його входи.

Прикладом комбінаційних схем можуть бути окремі логічні елементи, набори електронних ключів, шифратори, дешифратори, мультиплексори, демультиплексори, більшість арифметичних пристроїв: суматори, напівсуматори, множники тощо.

Мультиплексори.

Призначення мультиплексора – комутація у бажаному порядку інформації, котра надходить з кількох вхідних ліній однією вихідному.

З допомогою мультиплексора здійснюється поділ у часі інформації, котра надходить через канали. Мультиплексор можна вважати як безконтактний багатопозиційний перемикач.

Мультиплексор "два до одного".

Для перемикання вхідних сигналів використовується один зовнішній сигнал.

Мультиплексори мають двома групами входів і одного, рідше двома – взаємодоповнюючими виходами.

Одні входи інформаційні, інші – керуючі. До керуючих відносяться адресні і які дозволяють входи.

Набір сигналів на адресних входах визначає конкретний інформаційний вхід, який з'єднується після виходу.

Дозволяючий вхід керує одночасно усіма інформаційними входами, незалежно від стану адресних входів. Заборонені сигнал у цьому вході блокують дію всього пристрою.

Мультиплексор "чотири до одного".

Містить чотири інформаційних входи D0 ..D3, два адресних входи A і B, що дозволяють вхід V.

Двійкові числа, що характеризують сигнали на входах A і B, еквівалентні індексу задіяного інформаційного входу.

Таблиця істинності.

Входи			Вихід F
V	A	B	
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	X	X	

Демультиплексори.

Демультиплексори в функціональному відношенні протилежні мультплексорам.

Сигнали з одного інформаційного входу розподіляються в бажаній послідовності з кількома виходами. Обрання потрібної вихідної лінії забезпечується кодом на адресних входах.

При адресних входах демультиплексор може мати до 2m виходів.

Послідовність мікрокоманд, що забезпечує виконання даної операції (наприклад, операції нормалізації числа у формі з плаваючою точкою (коми)), називається **мікропрограмою** даної операції.

Таким чином, функціонування пристрою або системи обробки цифрової інформації може бути описано сукупністю реалізованих в ньому мікропрограм.

Всі найскладніші операції, виконувані цифровими автоматами і можуть бути реалізовані послідовним виконанням деяких елементарних операцій:

- Пересилання інформації з будь-якої комірки пам’яті в будь-яку іншу ячей-ку;
- Додавання коду з +1 (інкремент) або з -1 (декремент);
- Умовний перехід по збігу кодів;
- Безумовна зупинка.

З цієї причини операційні автомати (блоки) мають фактично уніфіковану структуру для різних цифрових пристроїв і систем обробки цифрової інформації і фіксований набір виконуваних елементарних операцій (наприклад, арифметико-логічні пристрої (АЛП) мікропроцесорів 580ВМ80, і80, і86, 1804ВС1 і т. д.). Таким чином, операційні автомати при проектуванні цифрових пристроїв і систем вибираються з серійно випускаються промисловістю.

Питання для самостійного опрацювання:

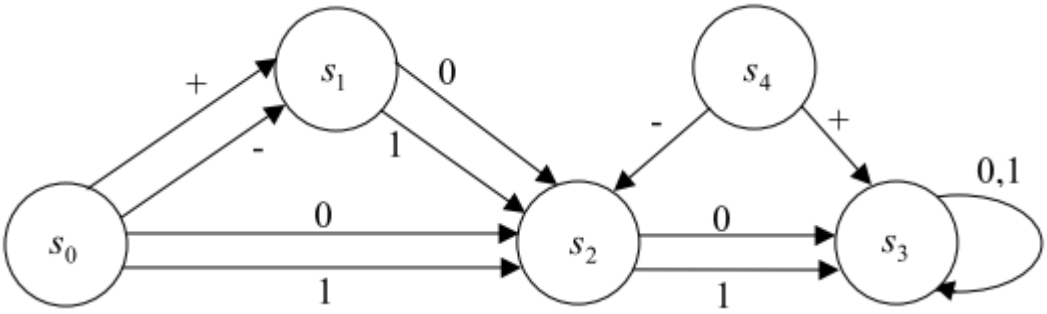
1. Коли доцільно задавати автомат графом переходів, а коли структурними таблицями?
2. Які типи мов мікропрограмування використовуються для представлення мікропрограм виконання операцій?
3. Що називається мікрооперацією, мікрокомандою, логічною умовою?
4. У чому полягає зміст функцій операційного і керуючого автоматів?
5. Чим відрізняється обернена структурна таблиця автомата від прямої?
6. Які умови коректності ГСА?

7. У якого з автоматів (Мілі або Мура) кількість внутрішніх станів більше (для однієї і тієї ж ГСА)?

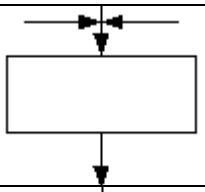
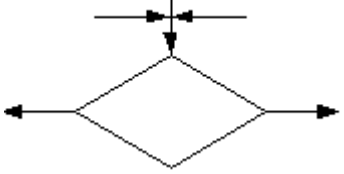
8. У чому відмінність мікропрограмного автомата від цифрового?

Побудова абстрактних автоматів за граф-схемою мікропрограми. Суміщені автомати.

Граф автомата - орієнтований граф, вершини крторого відповідають станам, а дуги - переходам між ними, наприклад:



Поряд з графом автомата, можна запропонувати граф-схему автомата, у якій (дивись таблицю)

Початок з одним виходом і без входу	
Кінець	
Виконання дії	
Перевірка умов	

Правила складання ГСА:

1. Кожен вихід з’єднується тільки з одним входом.

2. Кожен вхід з’єднується принаймні з одним виходом.

3. Через кожну вершину проходить як мінімум один шлях з початкової вершини в кінцеву.

4. У кожній операторній вершині записується мікрокоманда, тобто під безліч безлічі мікрооперацій: $Y = \{y_1, ..., y_n\}$.

5. У кожній умовній вершині записується логічна умова – один з елементів множини $X = \{x_1, ..., x_L\}$.

6. Дозволяється запис однакових мікрооперацій, і однакових логічних умови в різних операторних і умовний вершинах відповідно.

7. Один з виходів умовної вершини може бути з'єднаний з її входом.

Виконання ГСА на певній послідовності наборів виконується шляхом блукання по ГСА.

Принципово ці набори можуть і збігатися.

$Y_0, Y_1, Y_2, Y_3, Y_2, Y_4, Y_k$

Те, що ми отримали, називається значенням ГСА на заданій послідовності наборів.

Вибирається початковий оператор Y0 і перший набір w. З урахуванням цього набору проводиться "блукання по ГСА". Отримуємо значення Y_0 Y_1. Потім набір міняється на наступний і т.д. Процедура "блукання" закінчується, коли:

1) вичерпані всі набори;

2) дійшли до оператора Y_k.

Якщо, дійшовши до Y_k, ми не вичерпали наборів, то треба починати з оператора Y_0 спочатку.

Поєднувати в одній структурі декілька моделей різних класів кінцевих автоматів можна тільки в разі збігу часових параметрів вихідних сигналів кожної моделі. В іншому випадку, може скластися ситуація, коли значення вихідних сигналів, сформованих в одному і тому ж стані кінцевого автомата, на виході схеми будуть з'являтися в різних тактах автоматного часу. З аналізу тимчасового функціонування автоматів класів випливає, що припустимі наступні чотири поєднуючи нині моделі кінцевих автоматів: ADE, AD, AE і BF. На підставі цих чотирьох поєднуючих моделей установки регістрів на входи та/або виходи, а також замків на входи, можна побудувати ще 20 поєднуючих моделей кінцевих автоматів.

Структури поєднуючих моделей показані на рис. 1. Тут комбінаційна схема CL реалізує вихідні функції і функції збудження елементів пам'яті; регістр RGEF служить для зберігання кодів станів, що визначаються вхідними наборами (автомати класів E і F); регістр RGD – для зберігання кодів станів, що визначаються вихідними наборами (автомат класу D), і регістр RGAB – для зберігання кодів станів автоматів класів A і B. У поєднуючи ╥ нних моделях вміст вхідного регістра RGEF визначається значеннями вхідних змінних множини $X = \{x_1, ..., x_L\}$, вміст вихідного регістра RGD визначається значеннями вихідних змінних безлічі $Y = \{y_1, ..., y_N\}$ і вміст внутрішнього регістра RGAB визначається значеннями функцій переходів безлічі $D = \{d_1, ..., d_R\}$. Коди внутрішніх станів кінцевого автомата визначаються значеннями змінних множин $G == \{g_1, ..., g_L\}$, $Z = \{z_1, ..., z_N\}$ і $E = \{e_1, ..., e_R\}$, формованих на виходах регістрів RGEF, RGD і RGAB, відповідно.

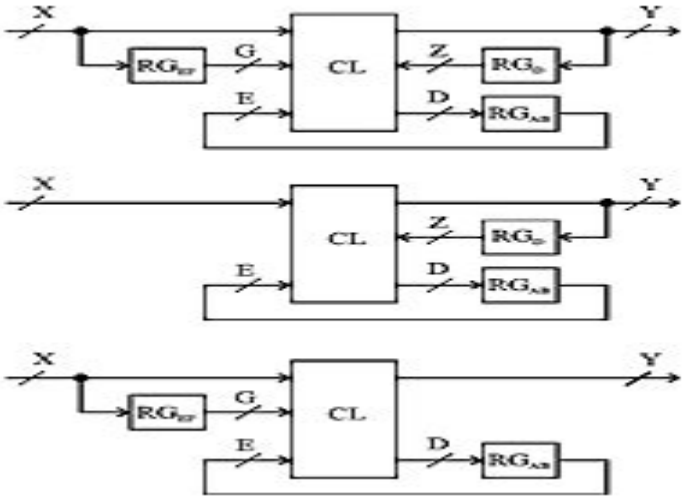


Рис. 1. Структури сполучених моделей кінцевих автоматів: ADE (а), AD (б), AE і BF (в).

Відзначимо, що в суміщених моделях кінцевих автоматів регістр RGEF реалізується вхідними буферами, регістр RGD – вих.ідними макрокомірками, а регістр RGAB – прихованими макрокомірками.

Кодування внутрішніх станів суміщених моделей кінцевих автоматів

Спочатку розглянемо кодування внутрішніх станів кінцевих автоматів для суміщеної моделі ADE, а потім уточнимо цей метод для моделей AD, AE і BF.

Суть кодування полягає в тому, щоб зробити всі внутрішні стани помітними між собою, тобто всі коди внутрішніх станів кінцевого автомата повинні бути взаємно ортогональні. Вхідні і вихідні набори кінцевого автомата, що визначаються значеннями змінних множин $G = \{g_1, ..., g_L\}$ і $Z = \{z_1, ..., z_N\}$, частково вирішують цю задачу. Подальше рішення полягає у введенні мінімального числа R додаткових змінних $e_1, ..., e_R$ множини E і кодуванні окремих груп станів двійковими кодами.

Нехай кінцевий автомат заданий таблицею переходів, стовпці якої мають позначення a_m, $a_и$, $X(a_m, a_и)$ і $Y(a_m, a_и)$. Для вирішення завдання кодування внутрішніх станів суміщеної моделі ADE будується матриця W третього порядку. Рядки матриці відповідають внутрішнім станам кінцевого автомата, а стовпці – змінними множинами G і Z. На перетині рядка, що відповідає деякому стану a_i, a_i € A, і стовпця, відповідного змінної g_j, g_j € G, ставиться одиниця, якщо вхідна змінна x_j входить в умову $X(a_i)$ в прямому значенні, нуль - в інверсному, і не визначене значення (дефіс), якщо змінна x_j не входить в умову $X(a_i)$, де $X(a_i)$ – умова переходу в стан a_i. Значення $X(a_i)$ визначається наступним чином: якщо всі переходи в стан a_i, a_i € A, здійснюються під впливом одного і того ж вхідного набору $X(a_m, a_i)$, то покладається $X(a_i):=X(a_m,a_i)$, інакше покладається $X(a_i)$: = порожня множина. На перетині рядків, відповідних стану a_i, a_i € A, і стовпця, відповідного змінної z_j, z_j € Z, ставиться одиниця, якщо y_j € $Y(a_i)$, і нуль в іншому випадку, де Y (ai) - підмножина вихідних змінних, приймають поодинокі значення на переходах в стан ai. Значення $Y(a_i)$ визначається наступним чином: якщо на всіх переходах в стан a_i, a_i € A, формується один і той же вихідний набір $Y(a_m, a_i)$, то

покладається Y (a_i): = Y (a_m, a_i), інакше покладається Y (ai): = порожня множина.

Питання для самостійного опрацювання:

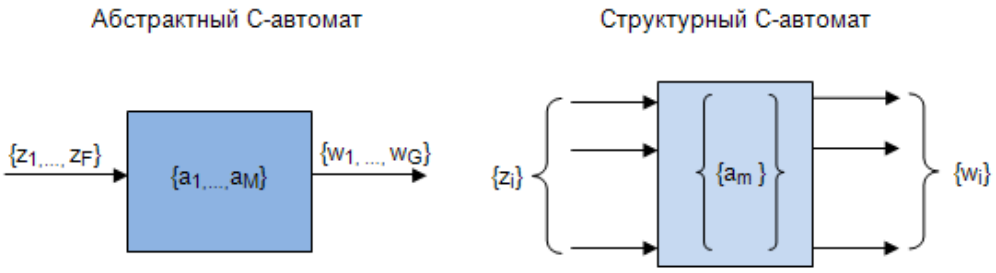
1. Метод синтезу сполучених моделей кінцевих автоматів

Синтез структурного автомата.

Процес абстрактного проектування полягає в переході від вихідної мікропрограми (або набору мікропрограм) до однієї з традиційних форм подання автомата: матрична, таблична або графічна (граф). Етап переходу до подання автомата також є необхідним, оскільки забезпечує реалізацію процесу структурного проектування шляхом використання досить, ефективного апарату теорії кінцевих автоматів.

Структурне проектування являє собою процес переходу від зазначених вище форм завдання до його функціональною схемою.

Отже, абстрактний автомат на вході має деяку послідовність вхідних сигналів, в залежності від яких переходить з одного стану в інший, видаючи деяку послідовність вихідних сигналів (рис.1).



Рисинок 1

У структурному автоматі враховується структура вхідних і вихідних сигналів, тобто їх конкретне уявлення у вигляді двійкових векторів. Стан автомата так само кодується двійковими векторами.

Розглянемо суміщений автомат (рис.2). Кожен стан абстрактного автомата кодується двійковим вектором:

$a_m = (\xi_{m1}, \xi_{m2}, \xi_{m3}, ...\xi_{mR}$ де $\xi_{mi} = 0,1$

$R \geq |Log2M|$, дужки показують, що береться найбільше ціле число.

M - кількість станів абстрактного автомату

R - кількість елементів пам'яті.

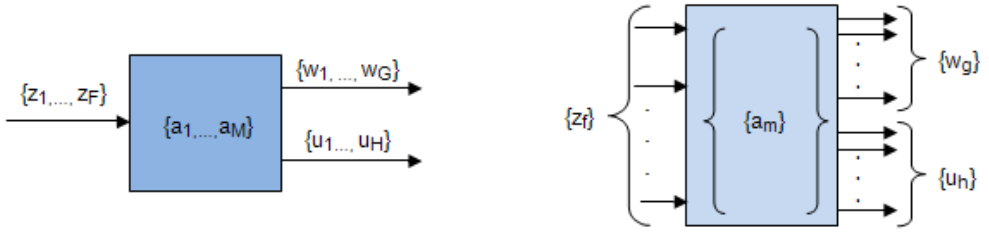


Рисунок 2.

Вхідний і вихідні представляються також двійковими векторами:^

$z_j = (\xi_{f1}, \xi_{f2}, ..., \xi_{fL})$ де $\xi_{fi} = 0, 1$, $L > |Log2F|$, F - кількість вхідних сигналів абстрактного автомату, L - кількість входів структурного автомату.

$w_g = (w_{g1}, w_{g2}, ..., w_{gN})$ де $w_{gi} = 0, 1$, $N \geq |Log2G|$ F - кількість вихідних сигналів 1 автомата, N - кількість виходів 1 структурного астомата.

Канонічний метод структурного синтезу автоматів

Схема структурного С-автомата при канонічному методі синтезу видається, що складається з трьох частин: двох комбінаційних схем і пам'яті автомата (рис. 3). Комбінаційна схема 1 призначена для формування функцій збудження надходять на входи елементів пам'яті, і вихідних сигналів 1 типу Y_n, що залежать від вхідних сигналів і сигналів з виходів елементів пам’яті R.

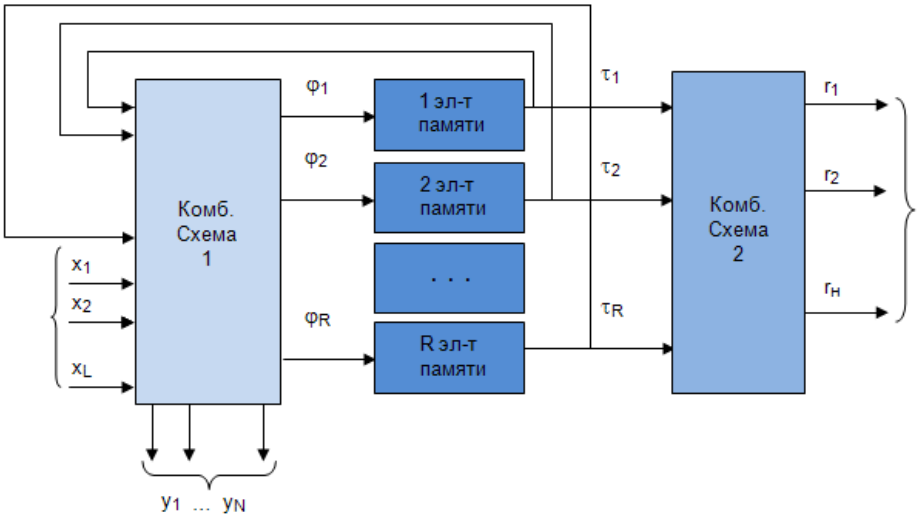


Рисунок 3.

Комбінаційна схема 2 призначена для формування вихідних сигналів 2 типу Th як функцій з виходів елементів пам’яті R.

Так як в автоматі Мілі сигнали 2 типу відсутні, то, відповідно до структурної схемою відсутня комбінаційна схема 2. Схема структурного автомата Мілі показана на рис.4

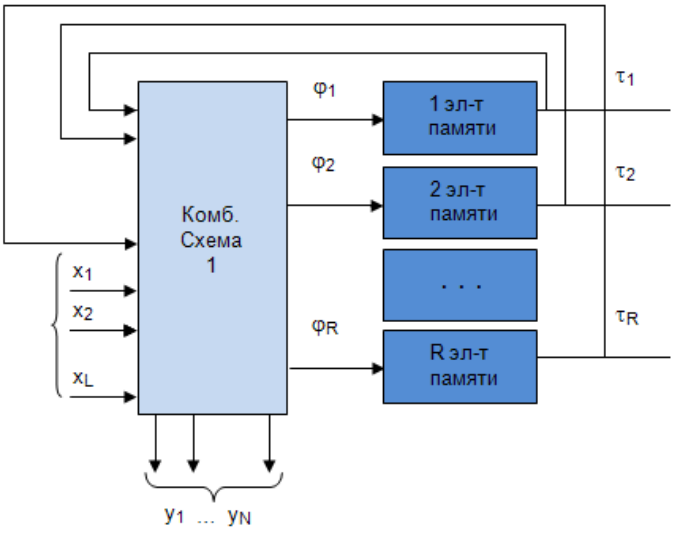


Рисунок 4.

В автоматі Мура сигнали 1 типу відсутні, отже, в структурній схемі в комбінаційної схемою 1 відсутні вихідні сигнали 1 типу Y_n. Схема структурного автомата Мура показана на рис.5.

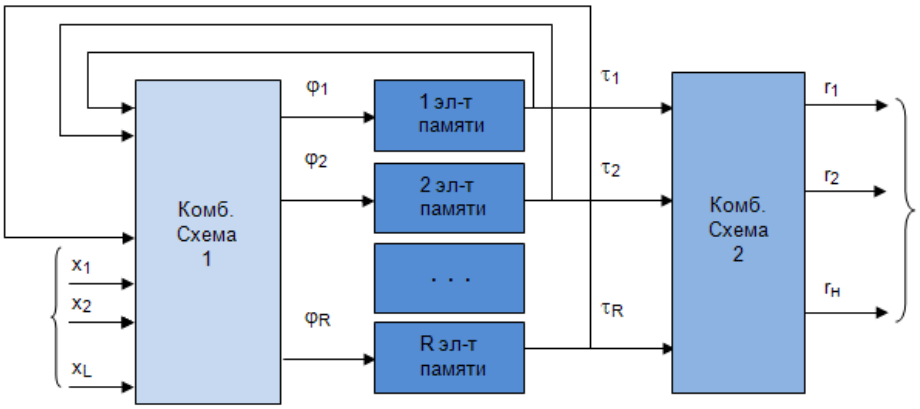


Рисунок 5.

Таким чином для того, щоб синтезувати структурний автомат, необхідно синтезувати дві комбінаційні схеми по системі канонічних рівнянь. Система канонічних рівнянь для С-автомата виглядає наступним чином:

$\phi_1 = \phi_1(r_1,\ r_2,\ldots,r_R,\ x_1,\ x_2,\ \ldots,\ x_L)$

$\phi_2 = \phi_2(r_1,\ r_2,\ldots,r_R,\ x_1,\ x_2,\ \ldots,\ x_L)$

.....

$\phi_R = \phi_R(r_1,\ r_2,\ldots,r_R,\ x_1,\ x_2,\ \ldots,\ x_L)$

$y_1 = y_1(r_1,\ r_2,\ldots,r_R,\ x_1,\ x_2,\ \ldots,\ x_L)$

$y_2 = y_2(r_1,\ r_2,\ldots,r_R,\ x_1,\ x_2,\ \ldots,\ x_L)$

.....

$y_R = y_R(r_1,\ r_2,\ldots,r_R,\ x_1,\ x_2,\ \ldots,\ x_L)$

$r_1 = r_1(r_1,\ r_2,\ldots,r_R)$

$r_2 = r_2(r_1,\ r_2,\ldots,r_R)$

.....

$r_R = r_R(r_1,\ r_2,\ldots,r_R)$

Етапи синтезу

1. Знаходимо кількість елементів пам'яті $L \geq \lvert Log 2F \rvert$, ($F$ - число станів абстрактного автомата) і кодуємо стану абстрактного автомата (табл.1).

Таблиця 1

$a_m \backslash\ r_1, r_2,\ldots,r_R$	$r_1, r_2,\ldots,r_R$
a_1	
a_2	
…	
a_M	

2. Кодуємо вхідні і вихідні сигнали, тобто

- знаходимо кількість входів структурного автомата $L \geq \lfloor Log2F \rfloor$, ($F$ - число вхідних сигналів абстрактного автомата);
- кількість виходів 1 типу $N \geq \lfloor Log2G \rfloor$; ($G$ - число вихідних сигналів 1 типу);
- кількість виходів 2 типу $D \geq \lfloor Log2H \rfloor$, ($H$ - число вихідних сигналів 2 типу) та кодуємо вхідні (табл.2) і вихідні сигнали (табл.3) і (табл.4) абстрактного автомата.

Таблиця 2

z_f / x_1	x_L $x_1x_2...x_1$
z	
z	
...	
z	

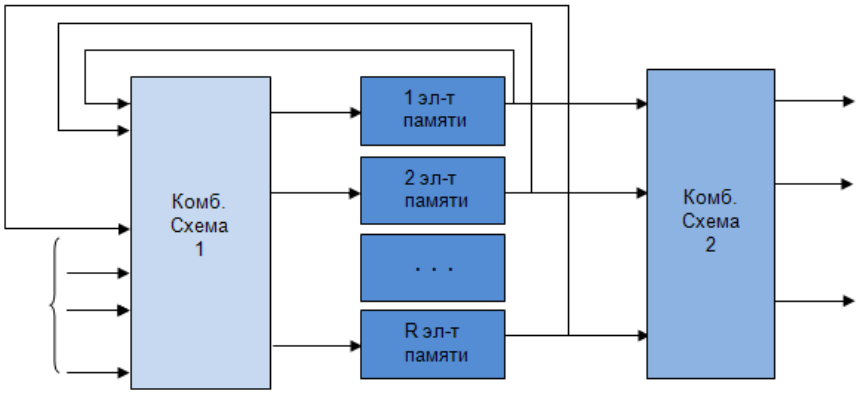
Таблиця 3

$w_f/y_1y_2...y_N$	$y_1y_2...y_N$
w_1	
w_2	
...	
w_G	

Таблиця 4

u_h/r_1 $r_2...r_D$	r_1 $r_2...r_D$
u_1	
u_2	
...	
u_H	

3. Структурний автомат представляємо узагальненою схемою (рис.6)



Рисьунок 6.

4. Складаємо закодовану таблицю виходів автомата і по ній записуємо рівняння виходів.

$y_1 = y_1(r_1, r_2,...,r_R, x_1, x_2, ..., x_L)$

$y_2 = y_2(r_1, r_2,...,r_R, x_1, x_2, ..., x_L)$

.....

$y_R = y_R(r_1, r_2,...,r_R, x_1, x_2, ..., x_L)$

$r_1 = r_1(r_1, r_2,...,r_R)$

$r_2 = r_2(r_1, r_2,...,r_R)$

.....

$r_R = r_R(r_1, r_2,...,r_R)$

5. Складаємо закодовану таблицю переходів автомата і по ній записуємо рівняння для функцій збудження, використовуючи таблиці переходів відповідних елементів пам'яті.

$\varphi_1 = \varphi_1(r_1, r_2,...,r_R, x_1, x_2, ..., x_L)$

$\varphi_2 = \varphi_2(r_1, r_2,...,r_R, x_1, x_2, ..., x_L)$

.....

$\varphi_R = \varphi_R(r_1, r_2,...,r_R, x_1, x_2, ..., x_L)$

6. Рівняння функцій збудження і виходів мінімізуються (по картах Карно, наприклад) і по ним будується схема в заданому функціонально - логічному базисі ({І, АБО, НЕ}, {І-НЕ}, {АБО-НЕ}).

Питання для самостійного опрацювання:

Поняття структурного автомату.

Пам'ять структурного автомата та її характеристики. Тригери.

Загальні положення

Пам'ять структурного автомата призначена для зберігання станів автомата (рис.1).

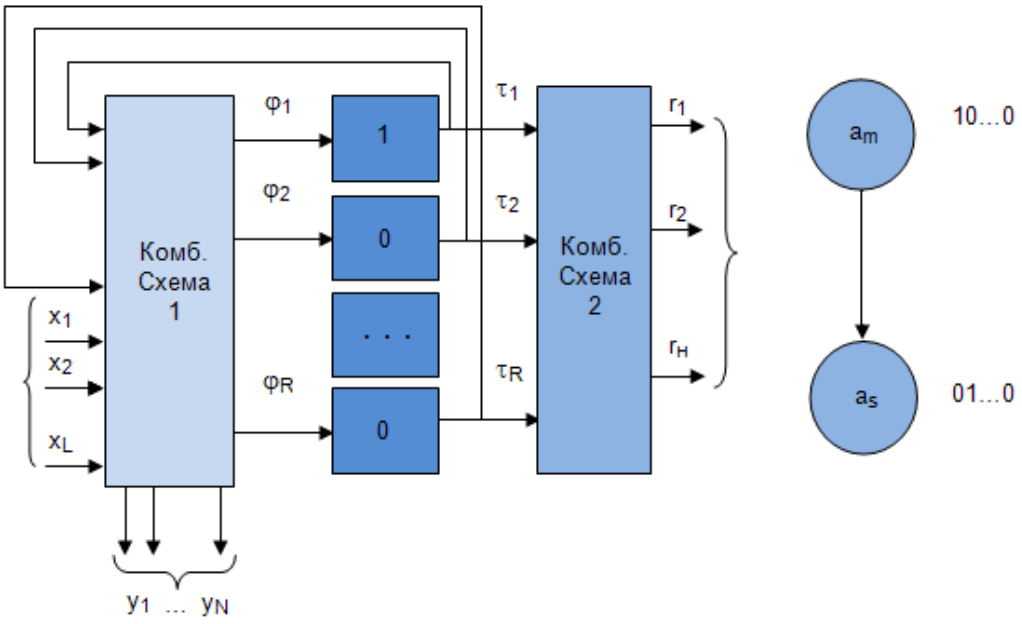


Рисунок 1.

Кількість елементів пам'яті обчислюється за формулою $R \geq \lfloor Log 2M \rfloor$, де M - число станів абстрактного автомата.

Якісна характеристика пам'яті грунтується на таких положеннях:

1. Так як для правильної роботи схеми неприпустимо участь вихідних сигналів запам'ятовуючих елементів у формуванні сигналів, які по ланцюгах зворотного зв'язку надходять на вхід запам'ятовуючих елементів в той же

момент часу, то в якості запам’ятовуючих елементів повинні бути використані абстрактні автомати Мура.

2. Автомат, як елемент пам’яті повинен бути з повною системою переходів і виходів для оптимального синтезу.

Повнота системи переходів означає, що для будь-якої пари станів (a_m, a_n) є свій вхідний сигнал, який переводить автомат з стану a_m в стан a_n.

Повнота системи виходів означає, що для кожного стану є свій вихідний сигнал. З цього випливає, що вихідні сигнали як би можуть бути ототожнені з станами автомата.

Приклад автомата з повною системою переходів і виходів приведений в табл.1.

Таблиця 1.

U_n	U_1	U_2	U_3
$Z_f \backslash A_m$	A_1	A_2	A_3
Z_1	A_1	A_3	A_1
Z_2	A_2	A_1	A_3
Z_3	A_3	A_2	A_2

Таблиця 2.

Исходное состояние	Входной сигнал	Состояние переходов
A1	Z1	A1
A1	Z2	A2
A1	Z3	A3
A2	Z2	A1
A2	Z3	A2
A2	Z1	A3
A3	Z1	A1
A3	Z3	A2
A3	Z2	A3

Розглянувши кожен перехід по табл. 1, можна цю інформацію представити в дещо іншій формі, так як показано в табл. 2.

Тригери.

В якості елементів пам’яті найчастіше використовуються тригери. Тригер - це елемент електронних схем, яка може знаходитися в будь-якому з двох стійких станів, а також багаторазово переходити з одного стану в інший. Стосовно до логічним схемам два стани тригера відповідають логічній “1” і логічній “0”. Таким чином, тригери є однорозрядними елементами пам’яті.

Розглянемо найбільш широко застосовуються тригери, такі як RS-тригери, Т-тригери, D - тригери і JK - тригери.

RS-тригери.

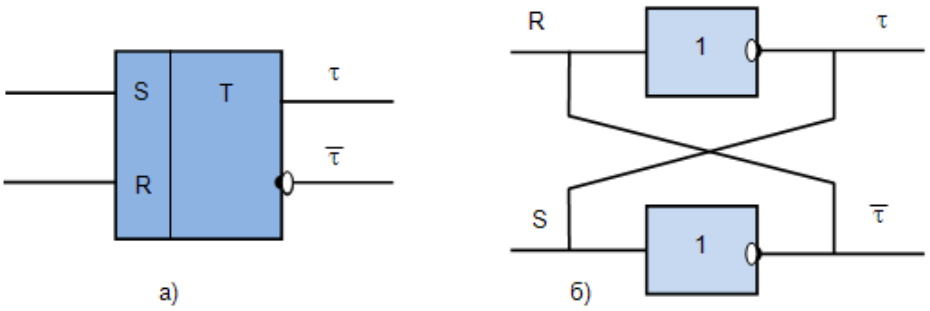


Рисунок 2.

На схемах тригери позначаються у вигляді прямокутника, розділеного на два поля. У лівому полі вказані назви входів тригера (рис.2, а), в правому буквою “Т” позначений тригер, що має прямий вихід T і інверсний $\overline{T}$.

На рис.2, б показана реалізація тригера за допомогою вентилів І-НЕ.

Робота RS-тригера представлена в табл. 3.

Таблиця 3

Входи		**Стани**		
R	S	0	1	Операція
0	0	0	1	Зберігання
0	1	1	1	Установка в 1
1	0	0	0	Установка в 0
1	1			Заборона

Якщо тригер встановлений в 1, то це значення зберігається в ньому до тих пір, поки не буде зроблене скидання (подача сигналу на вхід R-reset) або не буде вимкнено живлення. Якщо тригер встановлено у 0, то це значення зберігається в ньому до тих пір, поки не буде подано сигнал на вхід S-set. Одночасна подача сигналів на обидва входи тригера є забороненою, так як в цьому випадку ситуація виходить неоднозначною. У більш складних тригерах, наприклад в JK-тригерах, подібна ситуація виключається.

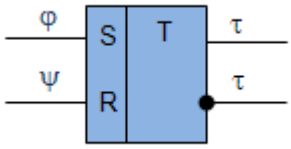


Рисунок 3.

Позначимо функції збудження φ і ψ, які надходять відповідно на R і S входи тригера (рис. 3) (табл. 4). Роботу тригера представимо таблицею переходів аналогічно табл. 2, тобто опишемо всі переходи з початкового стану тригера в можливі стани переходів (табл. 5)

Таблиця 4

R	**S**	**0**	**1**
0	0	0	1
0	1	1	1
1	0	0	0

1	1	- -

Таблиця 5.

$\tau_{исх.}$	$\varphi\psi$	$\tau_{пер.}$
0	00v10	0
0	0 1	1
1	0 1	0
1	00v01	1

Таблиця 6.

$\tau_{исх.}$	$\varphi\psi$	$\tau_{пер.}$
0	- 0	0
0	0 1	1
1	1 0	0
1	0 -	1

Аналізуючи табл. 5, бачимо, що тригер зі стану "0" у стан "0" переходить, коли на обидва входи подається "0" або на вході S "0", а на вході R може бути "1", тобто на вході S завжди при такому переході має бути "0", а на вході R будь сигнал. Таким чином, функції збудження при переході тригера з "0" в "0" такі: $\varphi = 0$, ψ ="-" (будь-який сигнал). Перехід тригера зі стану "0" у стан "1" відбувається, якщо на вході S "1", а на вході R повинен бути "0", тобто функції збудження при переході тригера з "0" в "1": $\varphi = 1$, $\psi = 0$ і т . д. Всі переходи і відповідні функції збудження RS-тригера показані у табл.6.6. Цю таблицю іноді називають таблицею функцій збудження RS-тригера.

Т - тригер (тригер з лічильним входом).

Таблиця 7.

T	**0**	**1**
0	0	1
1	1	0

Таблиця 8.

$\tau_{исх.}$	$\varphi\psi$	$\tau_{пер.}$
0	0	0
0	1	1
1	1	0
1	0	1

Таблиця 9.

D	0	1
0	0	0
1	1	1

Таблиця 10.

$\tau_{исх.}$	$\varphi\psi$	$\tau_{пер.}$
0	0	0

0	1	1
1	0	0
1	1	1

Тактований тригер, вихід якого “перемикається”, тобто змінює поточний стан на протилежне при кожному надходженні активного сигналу “1”. Робота Т-тригера описана в табл. 7, подання якої для явного відображення функції збудження Т-тригера дано в табл. 8. $\varphi = 1$, Тільки тоді, коли стан автомата переходить з 0 в 1 або з 1 в 0.

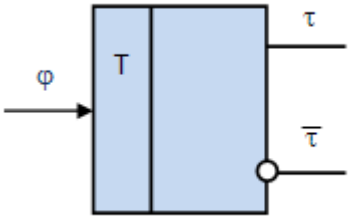


Рисунок 4.

D -тригер (елемент затримки)

D-тригер (рис. 5) має режими установки “1” і “0” і реалізує функцію тимчасової затримки (табл. 8). Як бачимо з табл. .9 функція рис. 5 збудження D-тригера ($\varphi = r_{11}ep$) збігається зі станом, у яке перемикається тригер.

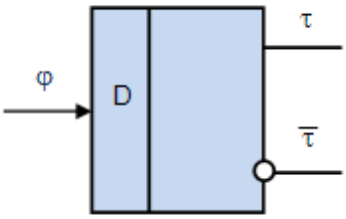


Рисунок 5.

JK-тригер

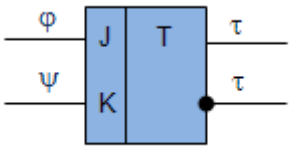


Рисунок 6.

Найбільш широко використовуваним є універсальний JK-тригер (рис. 6). Робота JK-тригера представлена в табл. 11. Одночасна подача сигналів на обидва входи тригера змушує його працювати як Т - тригер, тобто якщо тригер був встановлений в “0”, то він переключається в “1” і навпаки.

Таблиця 11.

Входи	**Стани**		
J K	0	1	Операція
0 0	0	1	Зберігання
0 1	0	0	Установка в “0”

1 0	1	1	Установка в “1”
1 1	1	0	Переключення

Позначимо функції збудження φ і ψ, які надходять відповідно на J і K входи тригера (рис. 6) і (табл. 11). Роботу тригера представимо таблицею (табл. 12) і таблицями функцій збудження (табл. 13) і (табл. 14).

Таблиця 12.

J K	0	1
0 0	0	1
1 0	1	1
1 1	1	0
0 1	0	0

Таблиця 13.

$\tau_{исх.}$	$\varphi\psi$	$\tau_{пер.}$
0	00v01	0
0	10v11	1
1	01v11	0
1	00v10	1

Таблиця 14.

$\tau_{исх.}$	$\varphi\psi$	$\tau_{пер.}$
0	0 -	0
0	1 -	1
1	- 1	0
1	- 0	1

Аналізуючи табл. 11, бачимо, що тригер зі стану “0” у стан “1” переходить, коли на вхід К подається “1”, а на вході J може бути будь-який сигнал. Перехід тригера зі стану “1” у стан “0” відбувається, якщо на вході J “1”, а на вході К будь-який сигнал. Таким чином, функції збудження така: $\varphi = 0$, при переході тригера з “1” в “0” і $\psi = 0$ при переході тригера з “0” в “1” .

Питання для самостійного опрацювання

1. Опишіть положення характеристики пам’яті.
2. Опишіть роботу тригерів.

Синтез структурних автоматів на тригерах.

Коротко відзначимо основні етапи синтезу автомата:

1. Знаходимо кількість елементів пам’яті (М - число станів абстрактного автомата) і кодуємо стану абстрактного автомата.

2. Кодуємо вхідні і вихідні сигнали.

3. Структурний автомат представляємо узагальненою схемою.

4. Складаємо закодовану таблицю виходів автомата і по ній записуємо рівняння виходів.

5. Складаємо закодовану таблицю переходів автомата і по ній записуємо рівняння для функцій збудження.

6. Рівняння функцій збудження і виходів мінімізуються (по картах Карно) і по ним будується схема в заданому функціонально - логічному базисі базисі ({І, АБО, НЕ}, {І-НЕ}, {АБО-НЕ}).

Розглянемо синтез структурного автомата Мура, заданого табл. 1, на D-тригерах в елементному базисі {І, АБО, НЕ}.

Таблиця 1.

u	**u_1**	**u_2**	**u_3**	**u_1**
z\a	a_1	a_2	a_3	a_4
z_1	a_1	a_3	a_1	-
z_2	-	a_1	a_4	a_2
z_3	a_4	a_2	a_3	a_3

1. Знаходимо кількість елементів пам’яті і кодуємо стан абстрактного автомата, наприклад так, як показано в табл. 2.

Таблиця 2.

	τ_1	τ_2
a_1	0	0
a_2	0	1
a_3	1	0
a_4	1	1

2. Кодуємо вхідні і вихідні сигнали абстрактного автомата, наприклад так, як показано в табл. 3 і табл. 4

3. Структурний автомат представляємо узагальненою схемою (рис. 1)

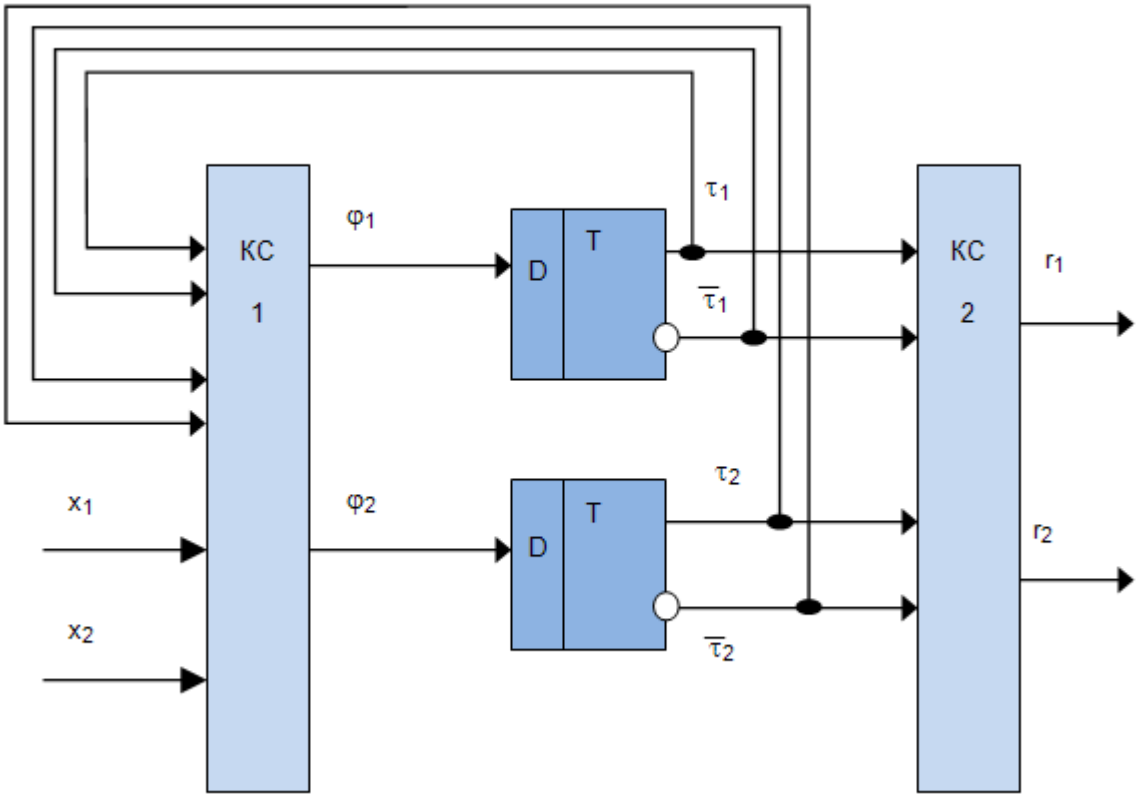


Рисунок 1.

Таблиця 3.

	X_1	X_2
z_1	0	1
z_2	1	0
z_3	1	1

Таблиця 4.

	r_1	r_2
u_1	0	0
u_2	0	1
u_3	1	0

4. Табл. 1 представляємо, використовуючи коди станів, вхідних і вихідних сигналів (табл.5), і по ній записуємо рівняння виходів.

Таблиця 5.

r_1r_2	00	01	10	00
x_1 x_2\ $\tau_1\tau_1$	00	01	10	11
01	00	10	00	-
10	-	00	11	01
11	11	01	10	10

5. Функція виходу r_1, залежна для автомата Мура тільки від стану, приймає одиничне значення на єдиному наборі r_1r_2 рівному 10, тобто $r_1 = r_1r_2$.

Функція виходу r_2, приймає одиничне значення так само на єдиному наборі $r_1 r_2$ рівному 01, тобто $r_2 = r_1 r_2$.

6. Таким чином, рівняння виходів:

7. $r_1 = r_1 r_2$

8. $r_2 = r_1 r_2$

9. Записуємо рівняння для функцій збудження. Так як в D-тригері функція збудження збігається зі станом переходу, то функцію збудження можна записати по табл. 5. Знаходимо одиничні стани першого тригера. Їх усього п'ять, вони виділені у табл. 5. Функція збудження залежить від вхідних сигналів і стану автомата, з якого був переключений даний тригер:

$\varphi R = \varphi R(r_1, r_2, ..., r_R, x_1, x_2, ..., x_L)$.

Перше одиничне значення функція збудження приймає при переході зі стану a_0, закодованого як 00, тобто $a_0 = r_1 r_2$ при надходженні одиничних вхідних сигналів $x_1 x_2 = 11$. Аналогічно знаходимо інші терми і записуємо функцію збудження першого тригера:

$$\varphi_1 = \bar{r}_1 \bar{r}_2 x_1 x_2 \vee \bar{r}_1 r_2 \bar{x}_1 x_2 \vee r_1 \bar{r}_2 x_1 \bar{x}_2 \vee r_1 r_2 x_1 x_2 \vee r_1 \bar{r}_2 x_1 x_2$$

Для другого тригера станів переходу, в яких він бере поодинокі значення, чотири. За ним записана функцію збудження другого тригера:

$$\varphi_2 = \bar{r}_1 \bar{r}_2 x_1 x_2 \vee \bar{r}_1 r_2 x_1 x_2 \vee r_1 \bar{r}_2 x_1 \bar{x}_2 \vee r_1 r_2 x_1 \bar{x}_2$$

10. Рівняння функцій збудження і виходів мінімізуються по картах Карно

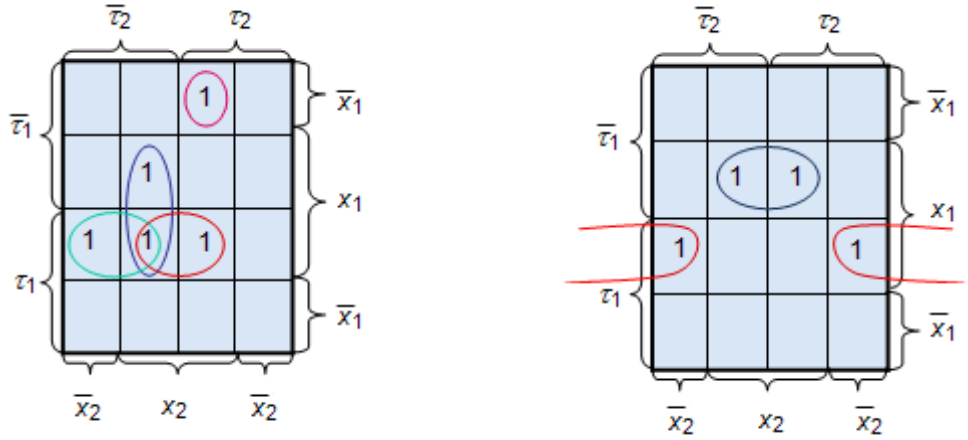


Рисунок 2.

$$\varphi_1 = \bar{r}_1 r_2 \bar{x}_1 x_2 \vee r_1 \bar{r}_2 x_1 \vee \bar{r}_2 x_1 x_2 \vee r_1 x_1 x_2$$

$$\varphi_2 = \bar{r}_1 x_1 x_2 \vee r_1 x_1 \bar{x}_2$$

За отриманими рівняннями функцій збудження і виходів будуємо функціонально - логічну схему

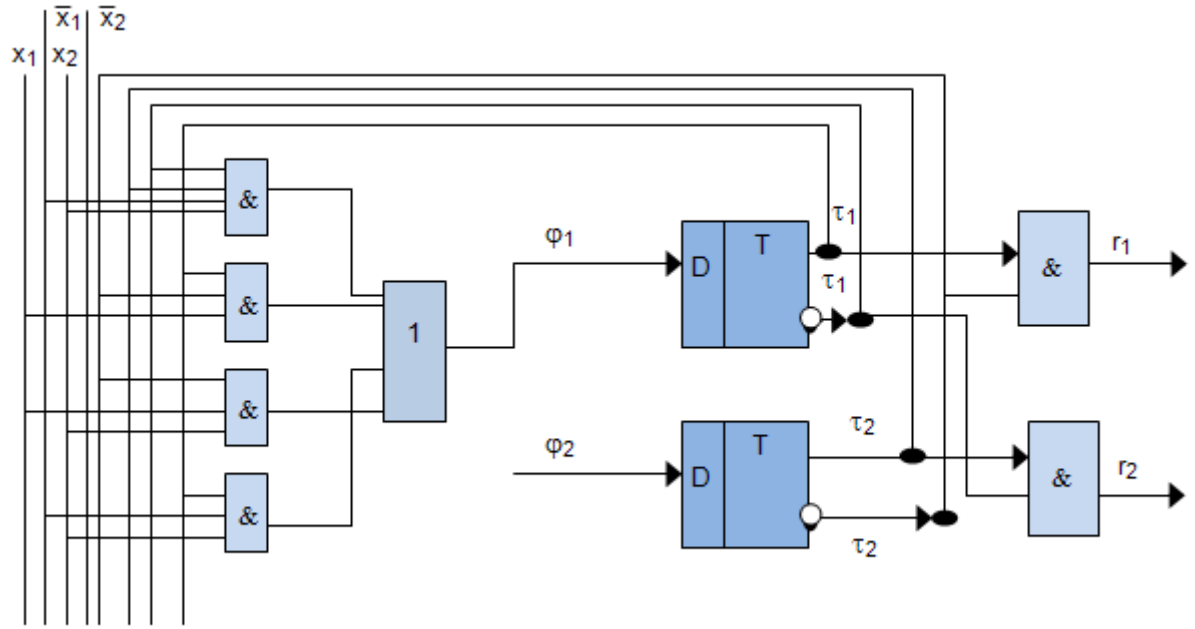


Рисунок 3.

Синтез структурного автомата Мура на Т-тригерах

Етапи кодування, побудови узагальненої схеми, побудови рівнянь виходів збігаються з етапами синтезу на D-тригерах. Розглянемо побудову рівнянь функцій збудження, тобто починаючи з п’ятого етапу.

Таблиця 6.

$\tau_{исх.}$	φ	$\tau_{пер.}$
0	0	0
0	1	1
1	1	0
1	0	1

Таблиця 7.

$\mathbf{r_1 r_2}$	**00**	**01**	**10**	**00**
$x_1 x_2 \ \tau_1 \tau_2$	00	01	10	11
01	00	10	00	-
10	-	00	11	01
11	11	01	10	10

Таблиця 8.

$x_1 x_2 \tau_1 \tau_2$	**00**	**01**	**10**	**11**
01	00	11	10	-
10	-	01	01	10
11	11	00	00	01

Оскільки функція збудження Т-тригера (табл. 6) $\varphi = 1$, тільки тоді, коли стан автомата переходить з 0 в 1 або з 1 в 0, то по закодованої рис. 7 переходів вихідного автомата Мура знаходимо такі перемикання тригерів, при яких вони змінювали свої стани. Складаємо таблицю функцій збудження, яка має в якості заголовків стовпців коди станів, а рядки позначені кодами вхідних сигналів

(табл. 8). У кожній клітці таблиці записані функції збудження Складаємо для них рівняння:

$$\varphi_1 = \bar{r}_1\bar{r}_2x_1x_2 \vee \bar{r}_1r_2\bar{x}_1x_2 \vee r_1\bar{r}_2x_1\bar{x}_2 \vee r_1r_2x_{12}$$

$$\varphi_2 = \bar{r}_1\bar{r}_2x_1x_2 \vee \bar{r}_1r_2\bar{x}_1x_2 \vee \bar{r}_1r_2x_1\bar{x}_2 \vee r_1\bar{r}_2x_1\bar{x}_2 \vee r_1r_2x_1x_2$$

Далі рівняння мінімізуються і по ним будується схема в заданому базисі.

Синтез структурного автомата Мура на RS-тригерах

Розглянемо синтез структурного автомата Мілі, заданого табл. 9 і табл. 10, на RS-тригерах в елементному базисі {І, АБО, НЕ}.

Таблиця 9.

z\a	**a1**	**a2**	**a3**	**a4**
z1	a1	a3	a1	-
z2	-	a1	a4	a2
z3	a4	a2	a3	a3

Таблиця 10.

z\a	**a1**	**a2**	**a3**	**a4**
z1	w1	w3	w1	-
z2	-	w1	w2	w2
z3	w2	w2	w3	w3

1. Знаходимо кількість елементів пам’яті R=2 і кодуємо стан абстрактного автомата, наприклад, так, як показано в табл. 11.

Таблиця 11.

	t 1	t 2
a1	0	0
a2	0	1
a3	1	0
a4	1	1

2. Кодуємо вхідні і вихідні сигнали абстрактного автомата, наприклад, так, як показано в табл. 12 і табл. 13

Таблиця 12.

	1	2
1		
2		
3		

Таблиця 13.

	1	2
1		
2		
3		

3. Структурний автомат представляємо узагальненою схемою (рис. 4).

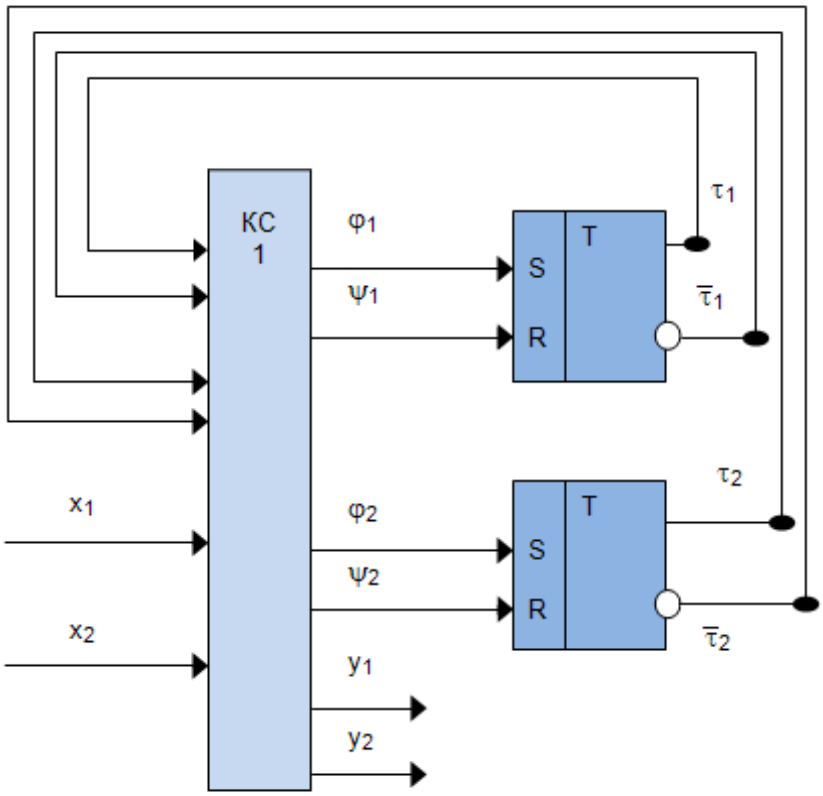


Рисунок 4.

4. Табл. 10 представляємо, використовуючи коди станів, вхідних і вихідних сигналів (табл. 14), і по ній записуємо рівняння виходів.

$$y_1 = \bar{r}_1 r_2 \bar{x}_1 x_2 \vee r_1 \bar{r}_2 x_1 x_2 \vee r_1 r_2 x_1 \bar{x}_2$$

$$y_2 = \bar{r}_1 \bar{r}_2 x_1 x_2 \vee \bar{r}_1 r_2 x_1 x_2 \vee r_1 \bar{r}_2 x_1 \bar{x}_2 \vee r_1 r_2 x_1 \bar{x}_2$$

Таблиця 14.

$x_1x_2\backslash t_1t_2$	**00**	**01**	**10**	**11**
01	00	10	00	-
10	-	00	01	01
11	01	01	10	10

5. Складаємо закодовану таблицю переходів автомата (табл. 15) і по ній записуємо рівняння для функцій збудження.

Таблиця 15.

r_1r_2	**00**	**01**	**10**	**00**

$x_1x_2\backslash t_1t_2$	00	01	10	11
01	00	10	00	-
10	-	00	11	01
11	11	01	10	10

Функція збудження RS-тригера представлена в табл. 16. Переглядаючи кожен перехід тригерів за таблицею переходів автомата (табл. 15), складаємо таблицю функцій збудження (табл. 17), яка має в якості заголовків стовпців коди станів, а рядки позначені кодами вхідних сигналів. У кожній клітці таблиці записані функції збудження для першого тригера $\varphi_1\psi_1$ і для другого тригера $\varphi_2\psi_2$. Складаємо для них рівняння:

Таблиця 16.

		$\tau_{\text{пер.}}$
	-	
	0	
	1	
	0	

Таблиця 17.

x_1x_2	0	1	0	1
01	- 0-	0 01	1 0-	
10		- 01	0 10	1 -0
11	0 10	- -0	0 0-	0 01

$$\varphi_1 = \bar{r}_1\bar{r}_2x_1x_2 \vee \bar{r}_1r_2\bar{x}_1x_2$$

$$\psi_1 = \bar{r}_1\bar{r}_2\bar{x}_1x_2 \vee r_1\bar{r}_2x_1\bar{x}_2$$

$$\varphi_2 = \bar{r}_1\bar{r}_2x_1x_2 \vee r_1\bar{r}_2x_1\bar{x}_2$$

$$\psi_2 = \bar{r}_1r_2\bar{x}_1x_2 \vee \bar{r}_1r_2x_1\bar{x}_2 \vee r_1r_2x_1x_2$$

Далі рівняння мінімізуються і по ним будується схема в заданому базисі. Аналогічно проводиться синтез і на JK-тригерах.

Питання для самостійного опрацювання

1. Опишіть синтез структурного автомата Мура на RS-тригерах
2. Опишіть синтез структурного автомата Мура на JK-тригерах

Графічний метод синтезу структурних автоматів та тригерах. Дискретні пристрої.

Етапи графічного методу синтезу структурного автомата

Перші три етапи графічного методу синтезу збігаються з табличним методом. Абстрактний автомат представлений у вигляді графа.

1. Знаходимо кількість елементів пам'яті $R \geq \left| Log_2 M \right|$, (M - число станів абстрактного автомата) і кодуємо стан абстрактного автомата.

2. Кодуємо вхідні і вихідні сигнали.

3. Структурний автомат представляємо узагальненою схемою.

4. Складання рівнянь вихідних функцій.

Представляємо закодований граф абстрактного автомата, тобто замість станів автомата вказуються відповідні кодові комбінації, а вхідні сигнали вказуються на переходах своїми логічними кодовими комбінаціями. Логічні кодові комбінації вихідних сигналів 1 роду записуються на переходах, а сигнали 2 роду записуються як мітки станів (або всередині вершини графа). Причому для вихідних функцій слід вказувати тільки ті значення функцій, які приймають дійсні значення, за якими складаються рівняння виходів.

5. Складання рівнянь функцій збудження.

На закодованому графі на дугах переходу вказуємо функції збудження, які відповідають переключенню тригерів, причому слід вказувати тільки ті значення функцій, які приймають дійсні значення, за якими складаються рівняння функцій збудження.

6. Рівняння функцій збудження і виходів мінімізуються (по картах Карно, наприклад) і по ним будується схема в заданому функціонально - логічному базисі ({І, АБО, НЕ}, {І-НЕ}, {АБО-НЕ}).

Приклад графічного методу синтезу структурного автомата

Нехай дано автомат Мілі (рис. 1). Виконаємо синтез структурного автомата на RS - тригерах.

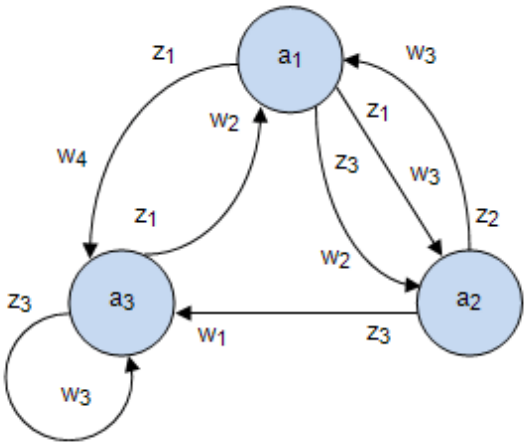


Рисунок 1.

1. кількість тригерів дорівнює $2(R \geq \left| Log_2 3 \right|)$. Стани абстрактного автомата закодуємо так, як показано в табл. 1

Таблиця 1.

$a_i \tau_1 \tau_2$	$\tau_1 \tau_2$
$\mathbf{a_1}$	00
$\mathbf{a_2}$	01
$\mathbf{a_3}$	11

2. Кодуємо вхідні і вихідні сигнали, наприклад, так як показано в табл. 2 і табл. 3.

Таблиця 2.

$Z_1 \backslash x_1 x_2$	$x_1 \backslash x_2$
z_1	**00**
z_2	**01**
z_3	**10**

Таблиця 3.

$w_i \backslash y_1 y_2$	$y_1 \backslash y_2$
w_1	**10**
w_2	**00**
w_3	**11**
w_4	**01**

3. Структурний автомат представляємо узагальненою схемою (рис. 2).

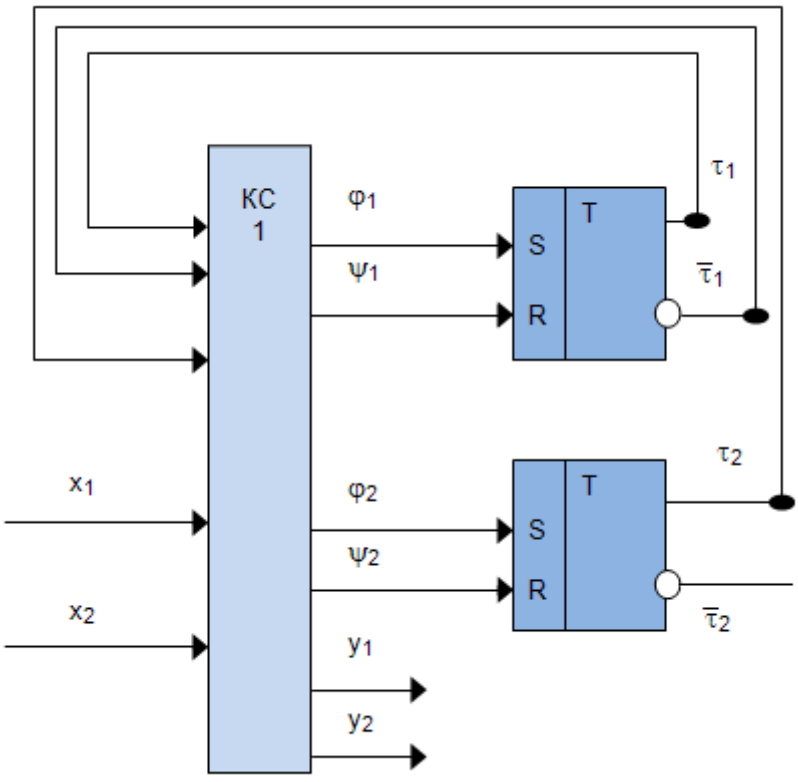


Рисунок 2.

4. Представляємо закодований граф абстрактного автомата (рис. 3) тобто замість станів автомата вказуються відповідні кодові комбінації, а вхідні сигнали вказуються на переходах своїми логічними кодовими комбінаціями.

Логічні кодові комбінації вихідних сигналів 1 роду записуються на переходах, а сигнали 2 роду записуються як мітки станів (або всередині вершини графа). Причому для вихідних функцій слід вказувати тільки ті значення функцій, які приймають дійсні значення, за якими складаються рівняння виходів.

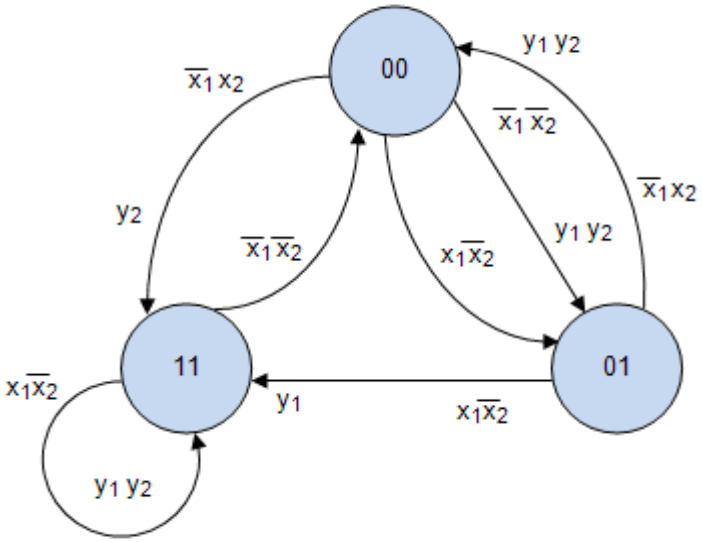


Рисунок 3.

$$y_1 = \overline{r_1}\,\overline{r_2}\,x_1\overline{x_2} \vee r_1\overline{r_2}\,\overline{x_1}\,\overline{x_2} \vee \overline{r_1}\,r_2\,x_1\overline{x_2} \vee r_1 r_2 x_1 x_2$$

$$y_2 = \overline{r_1}\,\overline{r_2}\,x_1\overline{x_2} \vee r_1\overline{r_2}\,\overline{x_1}\,\overline{x_2} \vee \overline{r_1}\,r_2\,x_1\overline{x_2} \vee r_1 r_2 x_1 \overline{x_2}$$

5. Складаємо рівняння функцій збудження для RS-тригера. На закодованому графі на дугах переходу вказуємо функції збудження: φ_1 якщо перший тригер переключився з 0 в 1; ψ_1 якщо другий тригер переключився з 0 в 1; φ_2 якщо перші тригер переключився з 1 в 0; ψ_2 якщо другий тригер переключився з 1 в 0; (рис. 4).

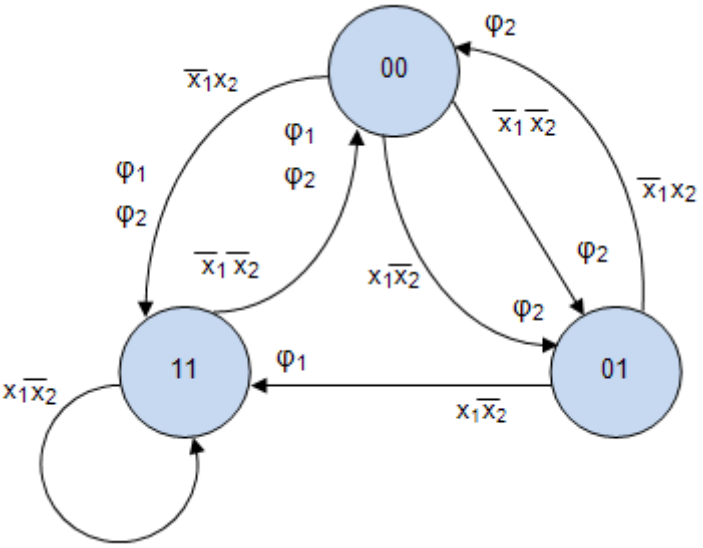


Рисунок 4.

Рівняння функцій збудження будуть мати вигляд:

$$\varphi_1 = \overline{r_1}\,\overline{r_2}\,x_1 x_2 \vee r_1 r_2 x_1 x_2$$

$$\psi_1 = r_1 r_2 \overline{x_1} x_2$$

$$\varphi_2 = r_1 r_2 x_2 \vee r_1 r_2 x_2$$

$$\psi_2 = \overline{r_1} r_2 \overline{x_1} x_2 \vee r_1 r_2 \overline{x_1} x_2$$

6. Останній етап мінімізації рівнянь, побудова схеми виконується як і в попередніх випадках синтезу.

Приклад графічного методу синтезу структурного автомата Мура

Нехай дано автомат Мура (рис. 5). Виконаємо синтез структурного автомата на JK-тригерах.

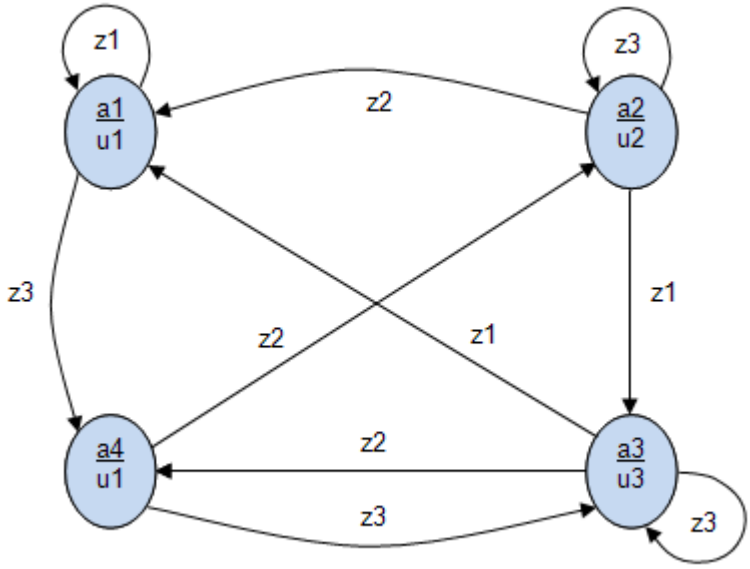


Рисунок 5.

1. кількість тригерів дорівнює $2\,(R \geq Log_2 4)$. Стани абстрактного автомата закодуємо так, як показано в табл. 4.

Таблиця 4.

	τ_1	τ_2
$\mathbf{a_1}$	0	0
$\mathbf{a_2}$	0	1
$\mathbf{a_3}$	1	0
$\mathbf{a_4}$	1	1

2. Кодуємо вхідні і вихідні сигнали, наприклад, так як показано в табл. 5 і табл. 6

Таблиця 5.

	$\mathbf{x_1}$	$\mathbf{x_2}$
$\mathbf{z_1}$	0	1
$\mathbf{z_2}$	1	0
$\mathbf{z_3}$	1	1

Таблиця 6.

	$\mathbf{r_1}$	$\mathbf{r_2}$
$\mathbf{u_1}$	0	0

$\mathbf{u_2}$	0	1
$\mathbf{u_3}$	1	0

3. Структурний автомат представляємо узагальненою схемою (рис. 6).

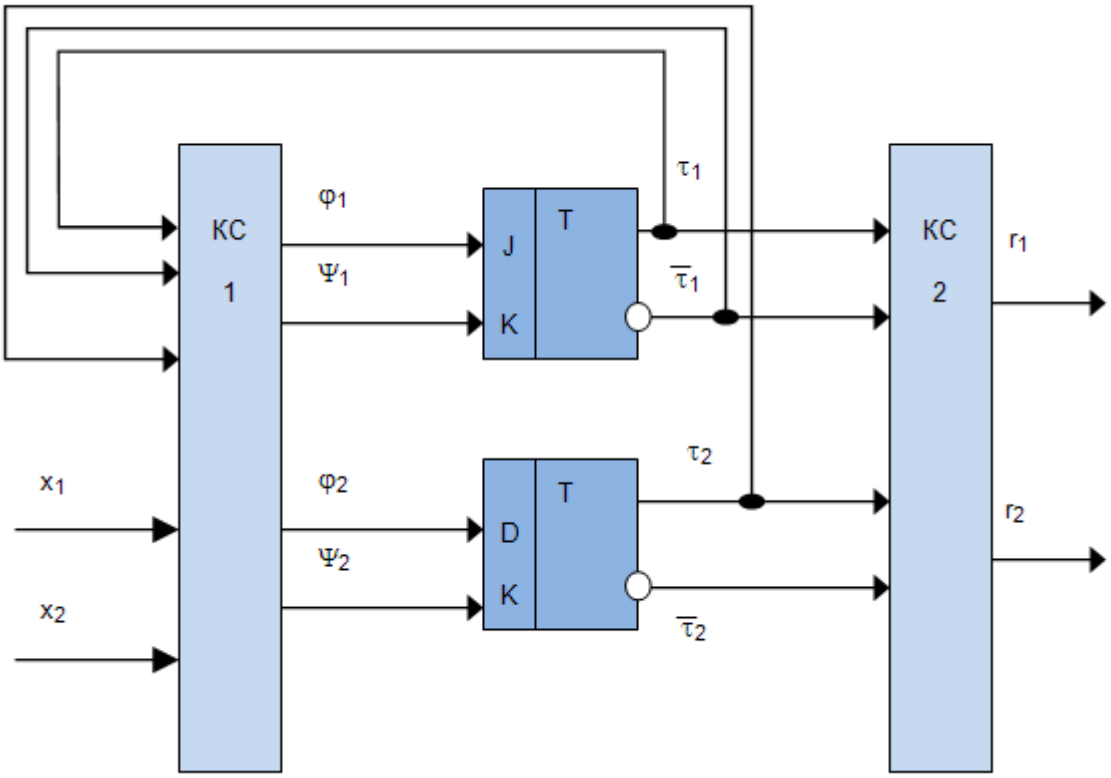


Рисунок 6.

4. Представляємо закодований граф абстрактного автомата (рис. 7) і (рис. 8). У вершинах вказуємо тільки ті значення функцій, які приймають дійсні значення, за якими складаються рівняння виходів:

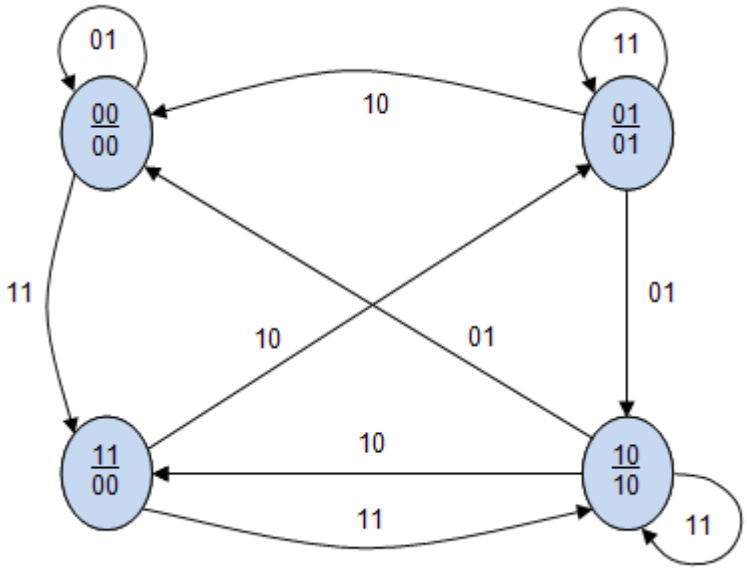


Рисунок 7.

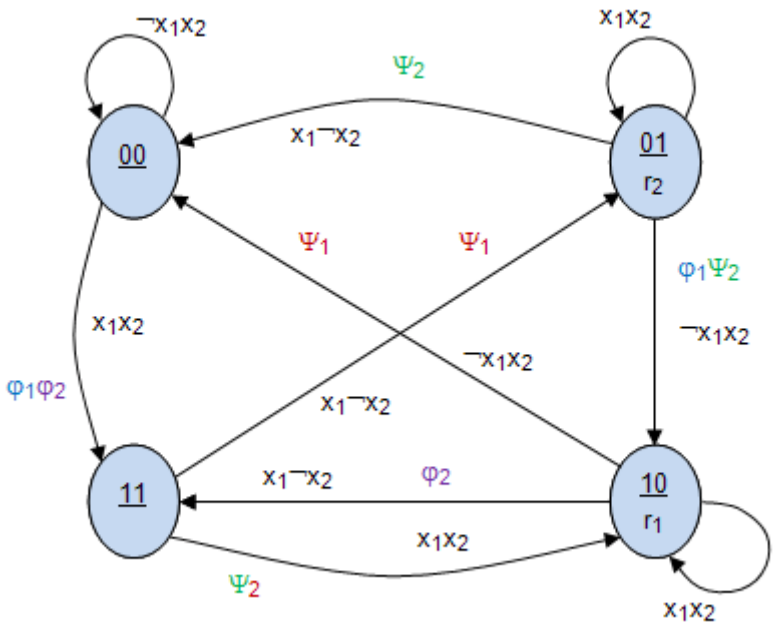


Рисунок 8.

$r_1 = r_1 r_2$

$r_2 = r_1 r_2$

5. Складання рівнянь функцій збудження для JK-тригера. На закодованому графі на дугах переходу вказуємо функції збудження: φ_1 якщо перший тригер переключився з 0 в 1; φ_2 якщо 2-ий. тригер переключився з 0 в 1; ψ_1 якщо перший тригер переключився з 1 в 0; ψ_2 якщо другий тригер переключився з 1 в 0; (рис. 4).

Рівняння функцій збудження будуть мати вигляд:

$\varphi_1 = \overline{r_1}\,\overline{r_2} x_1 x_2 \vee r_1 r_2 \overline{x_1} x_2$

$\psi_1 = r_1 \overline{r_2}\,\overline{x_1} x_2 \vee r_1 r_2 x_1 \overline{x_2}$

$\varphi_2 = \overline{r_1}\,\overline{r_2} x_1 \overline{x_2} \vee r_1 \overline{r_2} x_1 \overline{x_2}$

$\psi_2 = \overline{r_1} r_2 x_1 \overline{x_2} \vee r_1 \overline{r_2} \overline{x_1} x_2 \vee r_1 r_2 x_1 x_2$

6. Останній етап мінімізації рівнянь, побудова схеми виконується як і в попередніх випадках синтезу.

Питання для самостійного опрацювання

1. Етапи графічного методу синтезу структурного автомата

Синтез комбінаційних схем. Асинхронні схеми.

Синтез комбінаційної схеми з одним виходом

Синтез комбінаційної схеми можна поділити на три етапи.

Перший етап:

- Складають *таблицю істинності*, в якій фіксується склад і значення вхідних та вихідних логічних змінних і яка відображає задану логіку роботи комбінаційної схеми: - в такій таблиці для кожного можливого *набору значень* (далі просто *набору*) вхідних логічних змінних вказують значення логічної

функції: “**1**”, “**0**”, або “*****” (в останньому випадку значення функції невизначене). Можна також задати логічну функцію в іншій формі, наприклад в досконалій диз’юнктивній нормальній формі (*ДДНФ*).

- На основі таблиці істинності, застосовуючи ті чи інші методи мінімізації логічних функцій, знаходять логічне рівняння в *мінімальній диз’юнктивній нормальній формі* (*МДНФ*). При цьому якщо логічна функція визначена не на всіх *наборах* вхідних змінних, здійснюють її оптимальне довизначення (таке довизначення, при якому функція буде мати простішу МДНФ).

Другий етап:

Отримане на першому етапі логічне рівняння заданої функції (у МДНФ) записують в *операторній формі*, тобто у вигляді суперпозиції операторів логічних елементів (*оператором логічного елемента* називають функцію, яку реалізує цей елемент). Якщо обмежитися операторами *І, АБО, І-НЕ, АБО-НЕ* і припустити, що число входів відповідних логічних елементів є достатньо великим, то операторний запис функції зводиться до її подання в одній із восьми стандартних канонічних *нормальних форм* (див. ***Приклад 1***). Нормальні форми дозволяють будувати комбінаційні схеми з двома *рівнями (каскадами)* логічних елементів.

Якщо число входів ***p*** логічних елементів менше, ніж вимагається для реалізації рівняння в нормальній формі, то змінні об’єднуються в групи (не більше ***p*** змінних в кожній). Причому число таких груп також не повинне перевищувати ***p***, інакше така сукупність груп в свою чергу розбивається на групи по ***p*** елементів і так далі. Такі перетворення дозволяють подати задану функцію в операторній формі з врахуванням числа входів елементів. Але в цьому випадку операторна форма *не буде нормальною*, бо за рахунок додаткового каскадування елементів комбінаційна схема буде мати більше ніж два рівні.

Третій етап:

На основі операторного представлення логічної функції будують комбінаційну схему. При цьому враховують задану систему логічних елементів. Якщо задана система логічних елементів дозволяє реалізувати (або взяти за основу при реалізації з врахуванням числа входів елементів) *декілька* нормальних операторних форм, всі можливі варіанти реалізації комбінаційної схеми порівнюють за заданими параметрами і вибирають оптимальний варіант реалізації. Найчастіше такими параметрами є складність і швидкодія схеми. Розглянемо зміст етапів синтезу схеми на прикладі.

Приклад 1: нехай потрібно побудувати комбінаційну схему, яка реалізує логічну функцію ***y***. Причому $y = 1$, якщо мають місце дві з чотирьох можливих подій. Якщо ж одночасно мають місце три події, логічна функція ***y*** може мати будь-яке значення. В інших випадках $y = 0$.

Перший етап:

Таблиця 1.

№ набору	x_1	x_2	x_3	x_4	y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	*
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	*
12	1	1	0	0	1
13	1	1	0	1	*
14	1	1	1	0	*
15	1	1	1	1	0

Позначимо згадані в умові задачі чотири події як *вхідні логічні змінні* x_1, x_2, x_3, x_4. Домовимося, що коли деяка і-та подія має місце, то $x_i = 1$. Інакше $x_i = 0$.

За словесним описом заданої логічної функції складаємо таблицю істинності (Таблиця 1). В першому стовпчику таблиці вказуємо *номер набору* вхідних логічних змінних. В наступних чотирьох стовпчиках - власне *набір* вхідних логічних змінних. Іншими словами в рядках Таблиці 1 (в межах стовпчиків x_1, x_2, x_3, x_4) перебираємо поєднання можливих подій: - починаючи від випадку коли не має місця жодна з подій і закінчуючи випадком коли одразу всі чотири події мають місце. Останній стовпчик таблиці (значення логічної функції ***y***) заповнюємо у відповідності з заданим словесним описом цієї функції. Тобто значення функції будь-яке (*) в тих рядках таблиці, в яких є три одиниці в межах стовпчиків x_1 - x_4 (набори 7, 11, 13, 14). Значення функції дорівнює одиниці в тих рядках таблиці, в яких є дві одиниці в межах стовпчиків x_1 - x_4 (набори 3, 5, 6, 9, 10, 12). Для інших наборів значення функції дорівнює нулю.

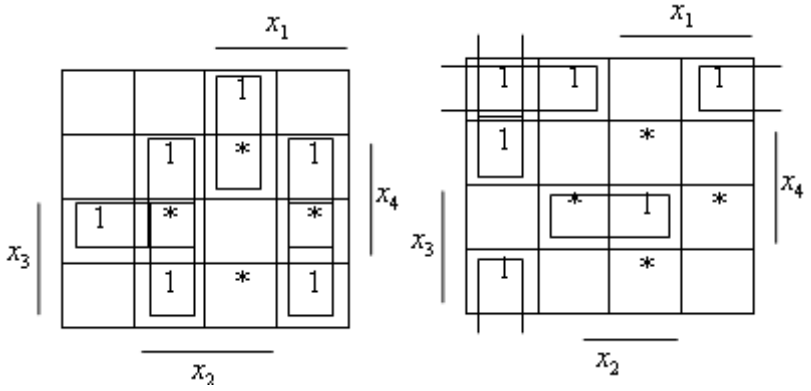


Рисунок 1. Рисунок 2.

Далі мінімізуємо логічну функцію, задану Таблицею 1, методом карт Карно. Причому використовуємо карту Карно для чотирьох змінних (Рис.1). У відповідності з таблицею істинності наносимо одиничні і невизначені значення логічної функції на карту Карно (Рис.1) і здійснюємо їх накриття мінімальним числом правильних прямокутників (контурів склеювання) максимальної площі. При цьому невизначені значення функції довизначаємо так, щоб отримана внаслідок мінімізації МДНФ була якомога простішою (в нашому прикладі невизначені значення функції, які відповідають наборам 7,11,13 ми визначаємо як одиничні, а значення функції, яке відповідає 14-ому набору довизначаємо як нульове). Кожному контуру склеювання в МДНФ функції відповідає одна кон'юнкція. Причому в цю кон'юнкцію входять ті змінні, які в однаковій формі (прямій або інверсній) входять в ті мінтерми, чиї клітинки охоплені даним контуром. Наприклад, значення функції, охоплені крайнім лівим контуром на Рис.1, будуть представлені в МДНФ кон'юнкцією $\overline{x_1}x_3x_4$, оскільки клітинкам карти Карно, охопленим даним контуром, відповідають мінтерми $\overline{x_1}\,\overline{x_2}x_3x_4$ і $\overline{x_1}x_2x_3x_4$, які відрізняються тільки формою входження в них змінної x_2.

Аналогічно мінімізуємо заперечення функції ***y*** ($\overline{y}$ - Рис.2). При заповненні цієї карти (порівняно з Рис.1) невизначені значення функції так і залишаються невизначеними, одиниці заміняються нулями, а нулі - одиницями.

Результатом мінімізації є МДНФ функції y та її заперечення $\overline{y}$:

$$y = x_1x_2\overline{x_3} + \overline{x_1}x_2x_4 + \overline{x_1}x_2x_3 + x_1\overline{x_2}x_4 + x_1\overline{x_2}x_3 + \overline{x_1}x_3x_4 \qquad (1)$$

$$\overline{y} = x_2x_3x_4 + \overline{x_1}\,\overline{x_2}\,\overline{x_4} + \overline{x_2}\,\overline{x_3}\,\overline{x_4} + \overline{x_1}\,\overline{x_2}\,\overline{x_3} + \overline{x_1}\,\overline{x_3}\,\overline{x_4} \qquad (2)$$

Другий етап:

Знаходимо представлення логічної функції ***y*** у восьми стандартних канонічних нормальних формах. Позначаються нормальні форми вказівкою на *внутрішні* та *зовнішні* функції розкладу. Наприклад, для МДНФ (1) внутрішньою логічною функцією є функція І (тобто кон'юнкції, що відповідають контурам склеювання). Зовнішньою функцією розкладу є функція АБО. Отже МДНФ (1) логічної функції ***y*** є формою І/АБО:

$$y = x_1x_2\overline{x_3} + \overline{x_1}x_2x_4 + \overline{x_1}x_2x_3 + x_1\overline{x_2}x_4 + x_1\overline{x_2}x_3 + \overline{x_1}x_3x_4 \qquad \text{(форма І/АБО)}$$

Наступну форму знаходимо, два рази проінвертувавши форму І/АБО і застосувавши один раз правило де Моргана:

$$y = \overline{\overline{x_1x_2\overline{x_3} + \overline{x_1}x_2x_4 + \overline{x_1}x_2x_3 + x_1\overline{x_2}x_4 + x_1\overline{x_2}x_3 + \overline{x_1}x_3x_4}} =$$

$$= \overline{\overline{x_1x_2\overline{x_3}} \cdot \overline{\overline{x_1}x_2x_4} \cdot \overline{\overline{x_1}x_2x_3} \cdot \overline{x_1\overline{x_2}x_4} \cdot \overline{x_1\overline{x_2}x_3} \cdot \overline{\overline{x_1}x_3x_4}} \qquad \text{(форма І-НЕ/І-НЕ)}$$

Застосувавши правило де Моргана до кожної з внутрішніх функцій розкладу форми І-НЕ/І-НЕ, отримують форму АБО/І-НЕ:

$$y=\overline{\left(\overline{x_1}+\overline{x_2}+x_3\right)\left(x_1+\overline{x_2}+\overline{x_4}\right)\left(x_1+\overline{x_2}+\overline{x_3}\right)\left(\overline{x_1}+x_2+\overline{x_4}\right)\left(\overline{x_1}+x_2+\overline{x_3}\right)\left(x_1+\overline{x_3}+\overline{x_4}\right)}$$

(форма АБО/І-НЕ)

Нарешті, застосувавши правило де Моргана до зовнішньої функції розкладу форми АБО/І-НЕ, отримують форму АБО-НЕ/АБО:

$$y=\overline{\left(\overline{x_1}+\overline{x_2}+x_3\right)}+\overline{\left(x_1+\overline{x_2}+\overline{x_4}\right)}+\overline{\left(x_1+\overline{x_2}+\overline{x_3}\right)}+\overline{\left(\overline{x_1}+x_2+\overline{x_4}\right)}+\overline{\left(\overline{x_1}+x_2+\overline{x_3}\right)}+\overline{\left(x_1+\overline{x_3}+\overline{x_4}\right)}$$

(форма АБО-НЕ/АБО)

Для отримання наступних чотирьох нормальних форм за основу беруть МДНФ заперечення функції ***y*** (2). Тоді форму І/АБО-НЕ можна отримати, проінвертувавши (2):

$$y=\overline{\overline{y}}=\overline{x_2x_3x_4+\overline{x_1}\,\overline{x_2}\,\overline{x_4}+\overline{x_2}\,\overline{x_3}\,\overline{x_4}+\overline{x_1}\,\overline{x_2}\,\overline{x_3}+\overline{x_1}\,\overline{x_3}\,\overline{x_4}}$$ (форма І/АБО-НЕ)

Застосувавши до зовнішньої функції розкладу попередньої форми правило де Моргана, отримаємо форму І-НЕ/І:

$$y=\overline{x_2x_3x_4}\cdot\overline{\overline{x_1}\,\overline{x_2}\,\overline{x_4}}\cdot\overline{\overline{x_2}\,\overline{x_3}\,\overline{x_4}}\cdot\overline{\overline{x_1}\,\overline{x_2}\,\overline{x_3}}\cdot\overline{\overline{x_1}\,\overline{x_3}\,\overline{x_4}}$$ (форма І-НЕ/І)

Далі застосовуємо правило де Моргана до внутрішніх функцій розкладу попередньої форми - отримуємо форму АБО/І:

$$y=\left(\overline{x_2}+\overline{x_3}+\overline{x_4}\right)\cdot\left(x_1+x_2+x_4\right)\cdot\left(x_2+x_3+x_4\right)\cdot\left(x_1+x_2+x_3\right)\cdot\left(x_1+x_3+x_4\right)$$ (форма АБО/І)

Проінвертувавши форму АБО/І два рази і застосувавши один раз правило де Моргана, отримаємо восьму нормальну форму - АБО-НЕ/АБО-НЕ:

$$y=\overline{\overline{\left(\overline{x_2}+\overline{x_3}+\overline{x_4}\right)\cdot\left(x_1+x_2+x_4\right)\cdot\left(x_2+x_3+x_4\right)\cdot\left(x_1+x_2+x_3\right)\cdot\left(x_1+x_3+x_4\right)}}=$$

$$=\overline{\left(\overline{x_2}+\overline{x_3}+\overline{x_4}\right)}+\overline{\left(x_1+x_2+x_4\right)}+\overline{\left(x_2+x_3+x_4\right)}+\overline{\left(x_1+x_2+x_3\right)}+\overline{\left(x_1+x_3+x_4\right)}$$

(форма АБО-НЕ/АБО-НЕ)

Третій етап:

За операторними представленнями функції, з врахуванням наявних логічних елементів, будують комбінаційну схему. Наприклад, якщо в нашому розпорядженні є логічні елементи І та АБО, ми можемо побудувати дворівневі комбінаційні схеми тільки на основі форм І/АБО, АБО/І. Порівнюючи ці дві схеми, приходимо до висновку, що схема за формою АБО/І буде простішою, бо вимагає менше тривходових елементів на першому рівні і елемента з меншою кількістю входів на другому рівні. Комбінаційну схему, побудовану за формою АБО/І у відповідності з Прикладом 1, подано на Рис.3.

Якщо кількість входів наявних логічних елементів обмежена, це може призвести до збільшення кількості рівнів в схемі. Приклад комбінаційної схеми, побудованої також за формою АБО/І, але з врахуванням числа входів логічних елементів (максимум чотири входи) подано на Рис.4.

Якщо ж наявна бібліотека логічних елементів дозволяє реалізувати не одну, а декілька нормальних форм логічної функції, необхідно порівняти можливі варіанти реалізації з метою вибору *оптимального*. Таке порівняння здійснюється, як правило, за двома показниками: *складністю та швидкодією*.

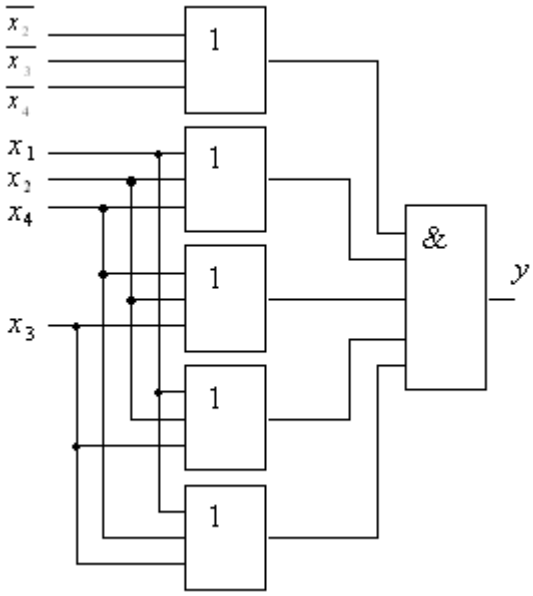


Рисунок 1.

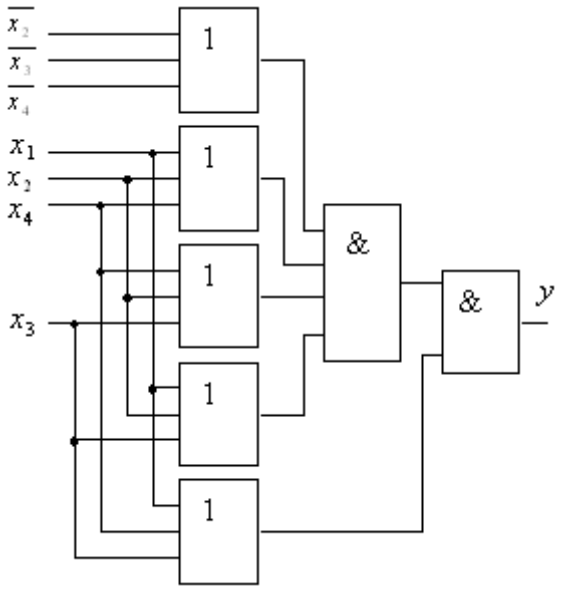


Рисунок 2.

Оцінка складності та швидкодії комбінаційних схем

Сучасний стан елементної бази, яка дозволяє реалізувати комбінаційні схеми, визначає різноманітність підходів до оцінки складності та швидкодії таких схем. В зв'язку з цим вкажемо на основні тенденції розвитку таких підходів. *Перша* з них - зменшення ваги такого показника, як складність (апаратні затрати). Ця тенденція обумовлена стрімким прогресом технології виготовлення інтегральних схем. Натомість постійно зростає роль швидкодії. Проте два згадані показники взаємопов'язані і складніша схема за деяких умов менш швидкодіюча. В складнішої схеми як правило більшою є загальна навантаженість входів і довшими є зв'язки між елементами. Це приводить до сповільнення розповсюдження сигналу від входу до виходу схеми.

Друга тенденція - різноманітність способів фізичної реалізації комбінаційних схем. Причому ці способи відрізняються мірою залежності складності і швидкодії від варіанту комбінаційної схеми (в припущенні, що ці варіанти реалізують одну й ту саму логічну функцію). При виготовленні комбінаційних схем на ПЗП такої залежності взагалі не існує. Якщо схему виготовляють на ПЛІС (програмована логічна інтегральна схема), або ПЛМ, швидкодія схеми залежить від варіанту реалізації комбінаційної схеми за умови, що кількість входів такої схеми є більшою, ніж кількість входів логічного блоку ПЛІС. Нарешті, якщо схему виготовляють на БМК (базовий матричний кристал), або складають з логічних ІМС середнього ступеня інтеграції, рівень залежності швидкодії від варіанту реалізації комбінаційної схеми ще більше зростає. Нарешті, технологічний прогрес в області виготовлення інтегральних схем обумовлює постійне підвищення частоти переключення таких схем (тобто зменшується затримка базових елементів). Відповідно збільшується доля затримки сигналу в провідниках у загальній

затримці розповсюдження сигналу, і оцінка швидкодії виключно за структурними і принциповими схемами втрачає свою повноту. Тому має місце *третя тенденція* - вдосконалення систем автоматичного проектування інтегральних схем, сучасні зразки яких надають засоби оптимального проектування з врахуванням всіх згаданих особливостей нової елементної бази. Сказане необхідно враховувати, користуючись *спрощеними* методами оцінки складності і швидкодії, що наведені нижче.

Існують різні способи оцінки складності: складність за Квайном (К), яка визначається як сумарне число входів усіх логічних елементів; складність за числом транзисторів (Т), що використовуються при реалізації схеми; складність за числом еквівалентних вентилів (логічних елементів 2І-НЕ), що йдуть на реалізацію схеми (В); складність за числом логічних елементів (блоків) (Л); складність за числом умовних корпусів мікросхем (N), визначена за формулою:

$$N = \sum_{i=1}^{r} \frac{m_i \cdot n_i}{14} \quad , \qquad (3)$$

де r - число типів мікросхем; m_i = кількість мікросхем і - ого типу; n_i - число виводів мікросхеми і - ого типу (умовною є мікросхема з n=14).

Параметри К, Т, В доцільно використовувати при проектуванні інтегральних схем, Л - при проектуванні ПЛІС. Оцінка N зручна при порівнянні складності пристроїв, побудованих на логічних мікросхемах середнього ступеня інтеграції.

Швидкодія комбінаційних схем залежить (спрощено) від часових параметрів логічних елементів t^{01} і t^{10} , які характеризують затримку сигналів елементом (час переходу вихідного сигналу із одного логічного рівня до другого). Оскільки вказані часові параметри залежать не тільки від виду логічного елементу, але й від конфігурації зв'язків даної схеми (від навантаженості виходів логічного елементу), на практиці може використовуватися середнє значення часу затримки t. Тоді для комбінаційних схем на однотипних елементах середній час затримки сигналів визначається як $T = L \cdot t$, де L - рівень схеми, визначений як число елементів, що входять в максимальної довжини ланцюг елементів. Якщо використовуються елементи з різною затримкою, то в схемі визначається шлях, який вимагає максимального часу проходження сигналу.

Питання для самостійного опрацювання

Імпульсні автомати. Синхронні схеми.

Електронні імпульсні пристрої з тимчасово стійкими станами. Електронні імпульсні пристрої з тимчасово стійкими станами є джерелами імпульсів напруги, значення, тривалість і частота повторення яких можуть регулюватися в широких межах. Мультивібратором називається пристрій із двома тимчасово стійкими станами, що представляє собою генератор імпульсів напруги прямокутної форми. Звичайно він служить для запуску в роботу інших

імпульсних пристроїв при їх спільній синхронній роботі. Найпоширенішийо мультивібратори на основі ОУ. Розрізняють симетричні й несиметричні мультивібратори

Тривалості прямокутних імпульсів і інтервали часу між ними в симетричних мультивібраторів рівні, у несиметричних різні. Розглянемо один період роботи мультивібратора. Такий стан ланцюга мультивібратора, хитливо. Дійсно, напруги на виході мультивібратора й на конденсаторі різні. Тому конденсатор буде розряджатися через резистор R ланцюга негативного зворотного зв'язку й ланцюг, підключений до виходу ОУ, а напруга на ньому змінюватися экспоненциально, прагнучи до значення ЭДС.

У момент часу обумовлений умовою, напруга на вході ОУ змінить своє позитивне значення на негативне. У результаті цього відбудеться його перемикання по передатній характеристиці й стрибком зміняться напруги. У несиметричному мультивібраторі інтервали часу зарядки й розрядки конденсатора різні. Це досягається включенням у ланцюг негативного зворотного зв'язку двох з'єднаних паралельно резисторів: один для зарядки, а іншої для розрядки конденсатора.

Послідовно з кожним резистором включається діод, прямий струм якого відповідає струму зарядки або току розрядки називається пристрій з одним стійким і одним тимчасово стійким станами, призначене для формування на своєму виході однократного прямокутного імпульсу напруги необхідної тривалості при впливі на вході імпульсу напруги від зовнішнього джерела.

Одновібратори застосовуються для стандартизації імпульсів напруги по тривалості, керування роботою електромагнітних реле, затримки імпульсів напруги й розподіли частоти їхнього повторення. Роботу одновібратора ілюструє сполучена тимчасова діаграма. По другому законі Кирхгофа для контуру, а в будь-який момент часу справедливе співвідношення. Стійкому стану одновібратора відповідає схема. Напруги на його елементах постійні й рівні тому що діод включений у прямому напрямку й напруга на конденсаторі.

Якщо в момент часу, на вхід одновібратора подати досить великий по амплітуді імпульс напруги позитивної полярності від джерела сигналів, то під дією ЭДС напруга на вході ОУ може стати негативним. У результаті відбудеться перемикання ОУ аналогічне описаному вище для мультивібратора й стрибком зміняться напруги тому що напруга на конденсаторі стрибком не змінюється. Це стан одновібратора, (ключ К у положенні, тимчасово стійке.

Дійсно, після перемикання ОУ починається зарядка конденсатора через резистор R ланцюга негативного зворотного зв'язку до моменту часу, обумовленого умовою, У цей момент часу відбудеться перемикання ОУ й стрибком зміняться напруги Одночасно процес зарядки конденсатора зміниться процесом його розрядки до моменту часу обумовленого умовою. Далі напруги на всіх елементах одновібратора постійні й відповідають його стійкому стану.

Тривалість позитивного прямокутного імпульсу напруги на виході одновібратора визначається формулами. Генератори лінійно, що змінюється напруги, входять до складу компараторів, пристроїв керування переміщенням електронного променя по екрані осцилографа й т.д. Найпростіший генератор

лінійно, що змінюється напруги, містить несиметричний мультивібратор, напругу якого використовується для керування роботою транзистора в ключовому режимі.

В інтервалах часу Аар транзистор виключений і відбувається зарядка конденсатора ємністю Із через резистор RK з великий постійної часу від джерела ЭДС Ек, в інтервалах часу Драз транзистор включений і відбувається розрядка конденсатора з малої постійної часу. При дотриманні умови напруга на конденсаторі при його зарядці змінюється практично лінійно.

Синхронні цифрові схеми складаються з комбінаційних вентилів (gates), ланцюгів (nets) і тригерів (flip-flops). У синхронній схемі є єдиний сигнал синхронізації, який управляє всіма елементами пам'яті (тригерами).

Формально, синхронну схему можна визначити наступним чином:

Синхронна схема - це цифрова схема C, яка складається з вентилів, тригерів і ланцюгів поширення сигналів, яка задовольняє таким умовам:

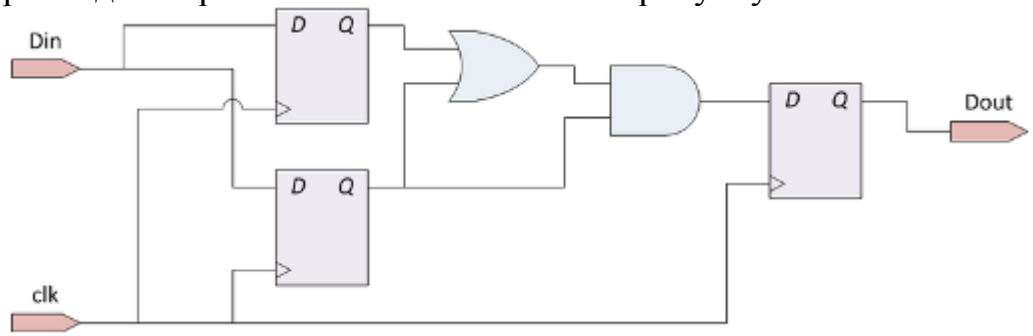
У схемі існує єдина ланцюг **clk**, по якій розповсюджується сигнал синхронізації (тактовий сигнал, clock signal)

Сигнал **clk** управляється вхідним портом схеми.

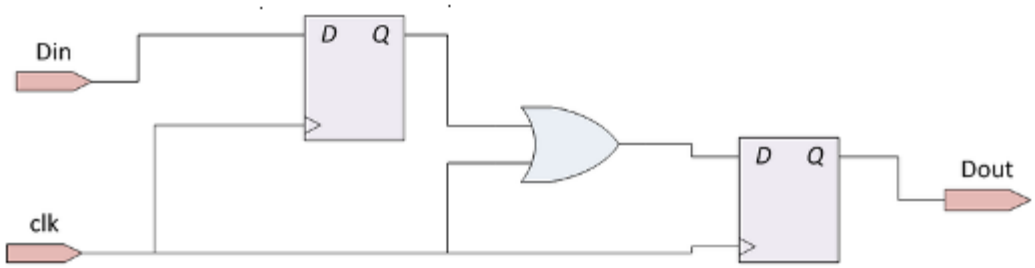
Безліч портів, керованих сигналом **clk**, еквівалентно безлічі входів синхронізації тригерів

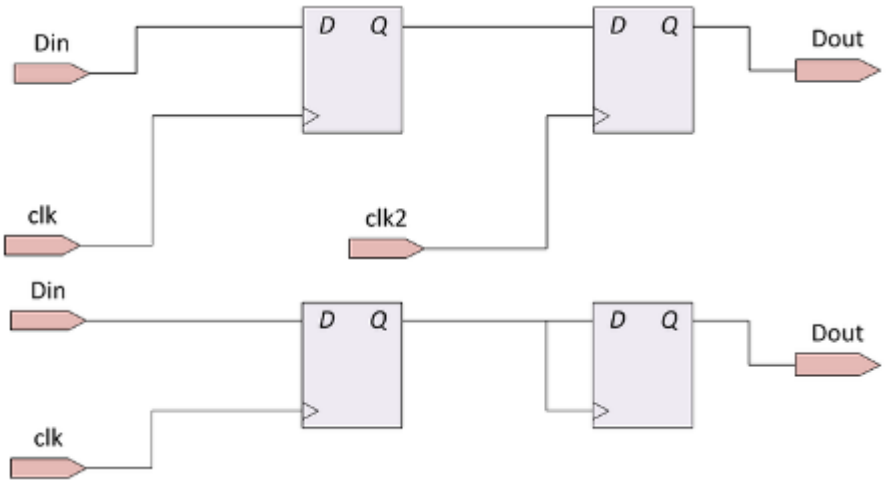
Визначимо схему C' таким чином: Схема C' виходить зі схеми C шляхом: (1) видалення ланцюга **clk**, (2) видалення вхідного порту, керуючого сигналом **clk**, (3) заміною всіх тригерів на вихідний порт (замість входу D) і вхідний порт (замість виходу Q). Отримана схема C' повинна бути комбінаційної

Приклад синхронної схеми показаний на рисунку:



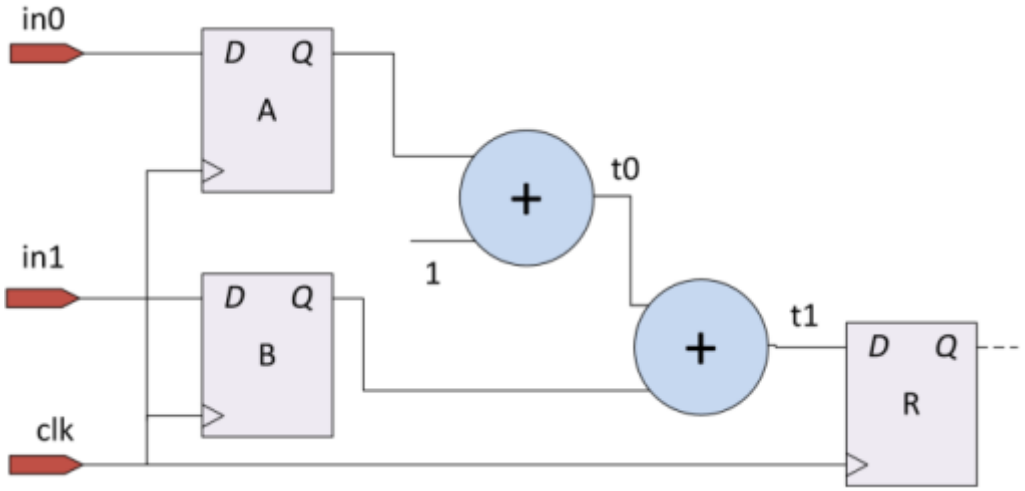
А ось кілька прикладів схем, які не є синхронними:





Практично всі існуючі цифрові схеми є синхронними, або складаються з декількох синхронних схем, взаємодіючих через асинхронні канали. Причина популярності синхронних схем - в простоті аналізу часу поширення сигналів.

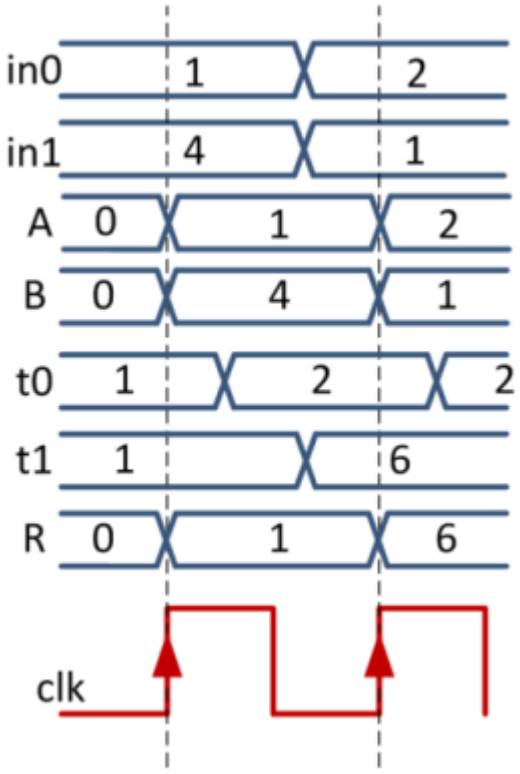
Розглянемо таку схему, що реалізує функцію R=A+B+1:



Регістри A, B і R зберігають значення на входах D по передньому фронту сигналу синхронізації (clk), тобто в той момент часу, коли значення clk змінюється з 0 на 1.

Сигнали поширюються через суматори (та інші комбінаційні елементи) не миттєво, а з затримкою, що залежить від довжини самого великого шляху з вентилів (критичного шляху), тобто від складності елемента. Приміром, критичний шлях в суматорі буде проходити через сигнали переносу в старший розряд (уявіть собі обчислення суми “стовпчиком”).

Припустимо, що спочатку у всіх регістрах був записаний 0. А на входи in0, in1 спочатку подаються значення 1 і 4, а потім 2 і 1. Тоді тимчасова діаграма для нашої схеми може виглядати наступним чином:



По першому фронту clk значення 1 і 4 будуть записані в регістри A і B. Після того як сигнал поширюється через суматори, значення результату 1+4+1=6 з'явиться на дроті t1. Потім, по другому фронту clk результат буде записаний в регістр R, а нові вхідні значення в регістри A і B.

Тепер уявімо, що період сигналу clk зменшився в два рази. Тоді другий фронт сигналу clk з'явиться на регістрі R до того, як на t1 з'являться правильні дані. Тобто схема буде працювати невірно!

Звідси випливає основне правило коректності роботи синхронної схеми:

Затримка через критичний шлях в схемі повинна бути менше періоду сигналу синхронізації.

Під критичним шляхом розуміється найдовший шлях в схемі, від виходу до входу регістра. З цього правила виводиться слідство, яке характеризує один з найбільших недоліків синхронних схем:

Синхронна схема працює на частоті, яка визначається критичним шляхом в схемі.

Уявіть, що в схемі тисячі комбінаційних шляхів із затримкою в 1 наносекунду. І один шлях із затримкою в 2 наносекунди. Через це єдиного шляху схема повинна тактіроватся на частоті в 500 МГц, хоча могла б працювати на гігагерц. Тому, при проектуванні синхронних схем довгі комбінаційні ланцюжка розбивають регістрами на конвеєрні стадії. Наприклад, в процесорі AMD Bulldozer середня довжина комбінаційного шляху - 12-14 FO4 еквівалентних вентилів (затримка, еквівалентна інвертору одиничного розміру, навантажених 4-ма інверторами).

Незважаючи на цей недолік, синхронні схеми стали дуже популярні. Синхронні схеми без труднощів піддаються автоматичному тимчасовому аналізу, тобто частота, на якій схема може коректно працювати, визначається програмою (тимчасовим аналізатором) автоматично. Коли розробник може

відсторонитися від цих деталей, синхронну схему можна специфікувати набором пересилань між регістрами. Такий підхід до опису схем - Register Transfer Logic (RTL) став мейнстримом в описі логіки роботи цифрових схем. До прикладу, схему з нашого прикладу можна описати наступними пересилками:

A = in0

B = in1

R=A+B+1

На кожному такті в регістр A записується in0, в регістр B записується in1, а в регістр R значення A+B+1. Ідея описувати схеми на RTL у вигляді тексту лежить в основі мов опису апаратури:

Питання для самоконтролю

1. З чого складається скінченний автомат?
2. Назвіть способи подання скінченного автомата.
3. Що зображують на діаграмі станів скінченного автомата?
4. Яка є структура матриці переходів скінченного автомата?
5. Чим відрізняється автомат Мілі від автомата Мура?
6. Сформулюйте основні дії з перетворення автомата Мілі на автомат Мура.
7. Сформулюйте основні дії з перетворення автомата Мура на автомат Мілі.
8. Зазначте основні області застосовування скінченних автоматів.
9. Назвіть скінченні автомати в оточуючих нас штучних об'єктах.
10. У чому полягає доцільність використання двох різних типів скінченних автоматів?
11. Які автомати називають еквівалентними?
12. Схарактеризуйте основні кроки алгоритму мінімізації скінченного автомата?
13. Як визначити, чи є скінченні автомати еквівалентними?
14. Перелічіть основні елементи сітки Петрі.
15. Назвіть основні області застосовування сіток Петрі.
16. Сформулюйте умову збудження переходу сітки Петрі.
17. У чому полягає спрацьовування переходу сітки Петрі?
18. Опишіть коротко процес функціонування сітки Петрі.
19. Зазначте способи наочного подання динаміки сітки Петрі.
20. Що являє собою граф досяжних маркувань сітки Петрі?
21. Які сітки Петрі називають ординарними?
22. Які матриці дозволяють подавати сітку Петрі?
23. Що являє собою рівняння станів сітки Петрі?
24. Перелічіть основні властивості сіток Петрі.
25. Які сітки називають обмеженими й безпечними?
26. У чому полягає властивість консервативності сітки Петрі?
27. У чому полягає властивість живості сітки Петрі?
28. Яке маркування сітки Петрі називають тупиковим?

29. Окресліть взаємозв'язок сіток Петрі й скінченних автоматів.

30. Схарактеризуйте структуру фундаментального рівняння сітки Петрі.

31. Що являє собою інваріант позицій сітки Петрі?

32. Що являє собою інваріант переходів сітки Петрі?

33. Схарактеризуйте властивості інваріантних сіток Петрі.

34. У чому полягає редукція сіток Петрі?

35. Зазначте основні правила редукції сіток Петрі.

36. Назвіть особливості побудови псевдомаркувань сітки Петрі.

37. Сформулюйте основні кроки алгоритму побудови дерева покривних маркувань сітки Петрі.

38. Які властивості сіток Петрі дозволяє визначити дерево покривних маркувань?

39. Назвіть переваги використання сіток Петрі для моделювання паралельних систем та процесів.

40. Сформулюйте інтуїтивне поняття алгоритму.

41. Назвіть основні риси, притаманні алгоритмові.

42. З яких елементів складається машина Тюринга?

43. Коротко опишіть процес функціонування машини Тюринга.

44. У чому полягає теза Чорча?

45. Сформулюйте проблему самозастосовності машини Тюринга.

46. Чи є проблема самозастосовності машини Тюринга алгоритмічно розв'язною?

47. У чому полягає метод зведення для доведення алгоритмічної нерозв'язності.

48. Які алгоритмічно нерозв'язні проблеми Вам відомі?

49. Назвіть алгоритмічно нерозв'язні проблеми в теорії сіток Петрі.

50. В чому полягає взаємозв'язок машини Тюринга й розширених сіток Петрі.

Список рекомендованої літератури

1. Глушков В.М. Синтез цифровых автоматов. – М.: Физматгиз, 1962. – 476 с.
2. Шоломов Л.А. Основы теории дискретных логических и вычислительных устройств. – М.: Наука, 1980. – 400 с.
3. Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984. – 264 с.
4. Котов В.Е. Сети Петри. – М.: Наука, 1984. – 160 с.
5. Грунский И.С., Козловский В.А., Пономаренко Г.П. Представление конечных
6. автоматов фрагментами поведения. – Киев: Наук. думка, 1990. – 232 с.
7. Слепцов А.И., Юрасов А.А. Автоматизация проектирования управляющих систем гибких автоматизированных производств / Под ред. Б.Н.Малиновского. – К. Техніка, 1986. – 160 с.
8. Ачасова С.М., Бандман О.Л. Корректность параллельных вычислительных процессов. – Новосибирск: Наука, 1990. – 254 с.
9. Марков А.А., Нагорный Н.М. Теория алгорифмов. – М.: Наука, 1984. – 432 с.
10. Успенский В.А., Семёнов А.Л. Теория алгоритмов: основные открытия и приложения. – М.: Наука, 1987.– 288 с.
11. Мурата Т. Сети Петри: Свойства, анализ, приложения // ТИИЭР. – т. 77, № 4, 1989. – с. 541-580.
12. Зайцев Д.А., Слепцов А.И. Уравнения состояний и эквивалентные преобразования временных сетей Петри // Кибернетика и системный анализ. – 1997, № 5. – с. 59-76.
13. Зайцев Д.А. Инварианты функциональных подсетей // Научные труды ОНАС им. А.С. Попова. – № 4, 2003. – с. 57-63.
14. Зайцев Д.А. Инварианты временных сетей Петри // Кибернетика и системный анализ. – № 2, 2004. – с. 92-106.
15. Зайцев Д.А., Шмелёва Т.Р. Моделирование коммутируемой локальной сети раскрашенными сетями Петри // Зв'язок. – № 2 (46), 2004. – с. 56-60.

Навчальне видання

КОМП’ЮТЕРНА ЛОГІКА

Методичні рекомендації
з дисципліни для самостійної роботи студентів спеціальності
“Обслуговування комп’ютерних систем і мереж”

Видання друкується за авторським редагуванням

Підписано до друку	Формат 60х54/18	Папір офсетний
Шрифт Times New Roman	Обл.-вид. арк..1,69	Тираж 50 пр.
Видання №	Ум. друк. арк..3,02	

Ніжинський державний університет
імені Миколи Гоголя
м. Ніжин, вул.. Воздвиженська,3/4
(04631)7-19-72
E-mail: vidavn_ndu@mail.ru
www.ndu.edu.ua

Свідоцтво суб’єкта видавничої справи
ДК № 2137 від 29.03.05 р.