

ПРОЕКТУВАННЯ КІНЦЕВИХ АВТОМАТІВ

ВСТУП

Конструкції кінцевих автоматів широко використовуються у послідовних керуючих логічних системах, які формують ядро багатьох цифрових систем. Кінцеві автомати застосовуються у безлічі пристроїв широкого діапазону дії та складності, наприклад: у пристроях керування низького рівня "мікропроцесор-VLSI-периферійні інтерфейси", у арбітражах шин та пристроях генерації часу у традиційних мікропроцесорах, у секційних мікропроцесорах, системах кодування та декодування даних, у протоколах передачі даних - це лише кілька прикладів систем, де потрібне застосування кінцевих автоматів.

Як правило, система логічного керування - це останнє, що повинне бути розроблене у циклі проектування пристрою, тому що конструкція керуючого автомата залежить від особливостей побудови пристрою. Застосування програмувальної логіки - оптимальне рішення для проектування систем логічного керування, тому що це дозволяє робити модифікації у проекті, не порушуючи при цьому фізичної структури комп'ютерного пристрою. Застосування елементів програмувальної логіки дозволяє швидко виправляти та модифікувати проекти, практично не впливаючи при цьому на час їхнього виходу на ринок.

У більшості проектів систем логічного керування зручно та доцільно використати для їхньої реалізації мікросхеми ПЛМ типу PAL. Зараз є можливість реалізовувати високошвидкісні та складні проекти пристроїв керування на мікросхемах програмувальної логіки. При проектуванні систем логічного керування стало можливим застосування широкого діапазону різних функціональних рішень. У цій статті ми розглянемо функції, які виконують системи логічного керування, реалізацію цих систем на різних пристроях та вибір конструкції систем.

ЩО ТАКЕ КІНЦЕВИЙ АВТОМАТ?

Кінцевий автомат - це цифровий пристрій, робота якого полягає у проходженні через послідовність попередньо завданих станів певним чином. Стани - це набір сигналів, що реєструються у різних частинах схеми. Традиційна структура системи керування складається з PAL-елементів, що реалізують комбінаторну логіку, вихідних реєстрів і реєстрів станів. У такій структурі стани послідовно зберігаються у реєстрі станів і вихідному реєстрі.

Загальна структура системи керування зображена на рис. 1. Крім входів і виходів у структурі системи керування виділяють два основних структурних елементи - схему комбінаторної логіки та схему пам'яті (реєстр). Подібна структура є базовою і найпростішою зі структур систем керування. Елементи пам'яті у системі використовуються для зберігання станів автомата. Блок комбінаторної логіки може розглядатися у вигляді композиції двох функціонально різних частин: блоку формування наступного стану та блоку формування вихідних сигналів (рис. 2). Блок формування наступного стану відповідає за формування сигналів, що необхідні для переходу автомата у наступний стан, а блок формування вихідних сигналів відповідає за формування слів керування на виході керуючого автомата. Хоча ці два блоки і виконують зовсім різні функції, вони традиційно логічно поєднуються у єдиний блок комбінаторної логіки, як зображено на рис. 1.

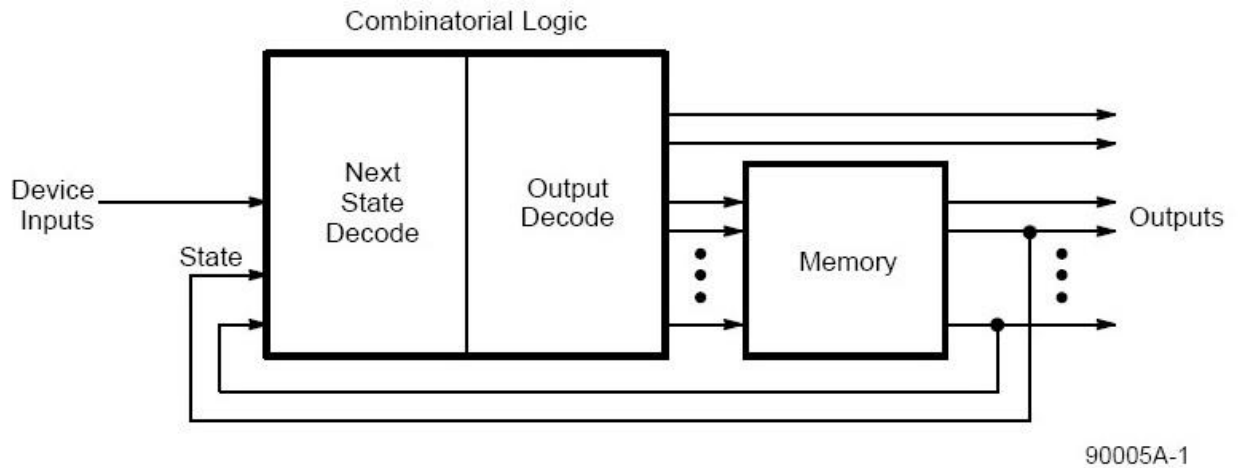


Рисунок 1. Структурна схема тривіального кінцевого автомата

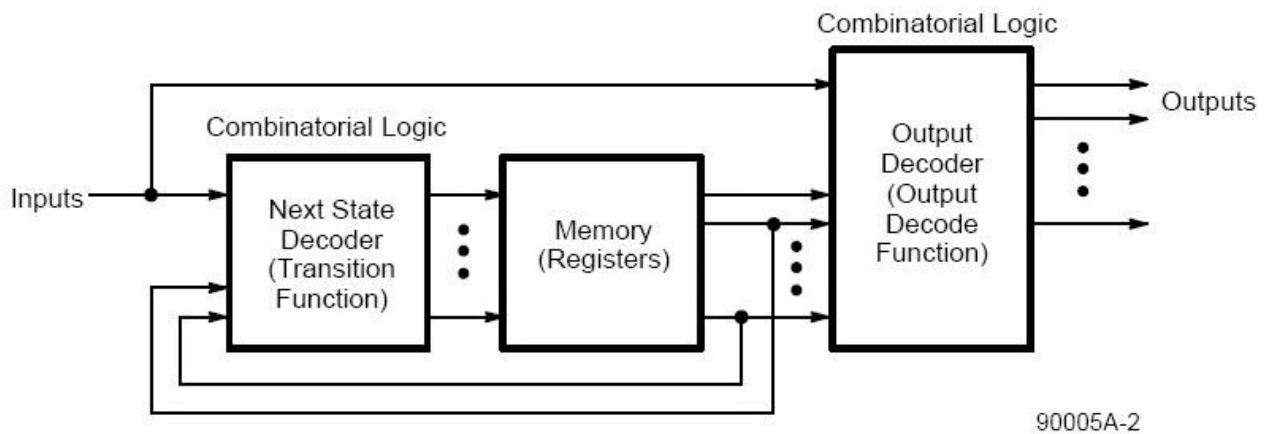


Рисунок 2. Структура кінцевого автомата з окремою схемою формування наступного стану автомата та окремою схемою формування вихідних слів

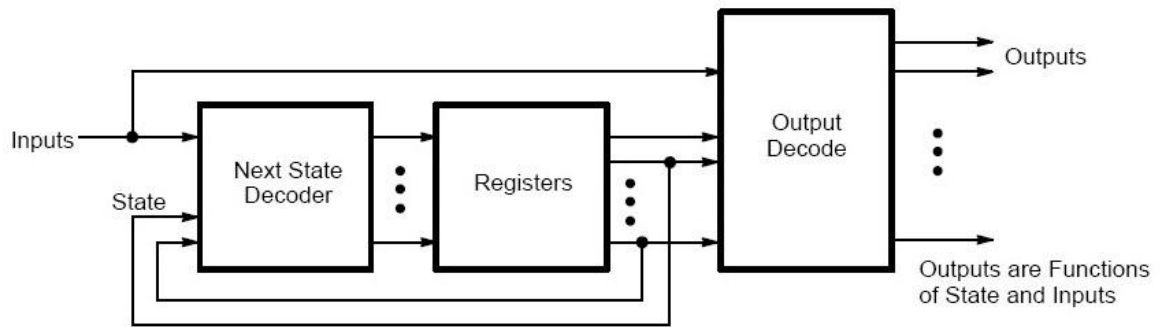
Керуючі автомати виконують дві наступні основні операції:

1. Перехід у наступний стан, що здійснюється за допомогою схеми формування функцій збудження пам'яті (Next State Decoder (Transition Function)), яка виробляє свої вихідні сигнали з урахуванням вхідних сигналів умов та попереднього стану автомата.
2. Формування вихідних сигналів керування, що формуються на виході схеми формування вихідних сигналів (Output Decoder (Output Decode Function)) з урахуванням поточного стану, у якому перебуває автомат, та вхідних сигналів умов.

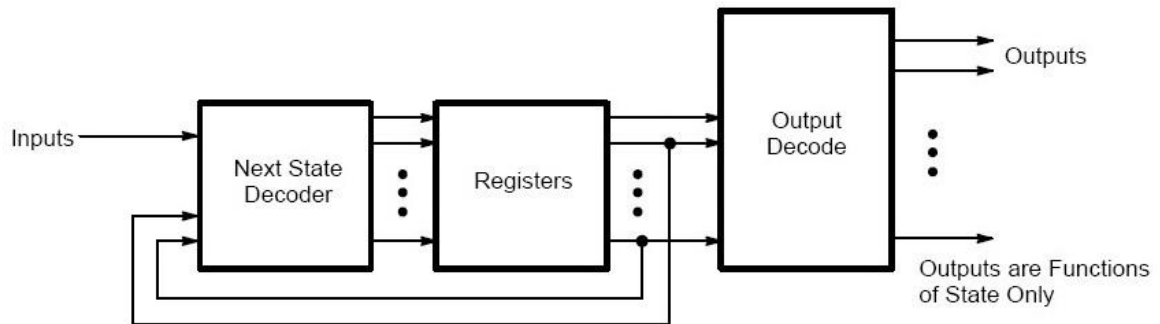
Використання вхідних сигналів для формування наступного стану автомата називають переходом. Крім переходів у наступний стан, у керуючому автоматі можлива організація повторення послідовностей станів (цикли) та організація підпрограм. Переходи з одного стану в інше називаються "керуючою послідовністю", а логічні функції, які потрібно вирішити для переходу в наступний стан, називаються "функціями переходу" (Transition Function) (рис. 2).

Спосіб використання вхідних умовних сигналів для формування вихідних керуючих сигналів визначає тип керуючого автомата. Є два широко відомих типи керуючих кінцевих автоматів: автомат Мілі та автомат Мура (рис. 3). В автоматі Мура вихідні сигнали залежать тільки від поточного стану, у якому перебуває автомат. У більш розповсюджених автоматах Мілі вихідні сигнали залежать як від поточного стану автомата, так і від вхідних сигналів умов. Логічні функції, які визначають значення вихідних сигналів, називаються "вихідними функціями". Для обох типів кінцевих автоматів керуючі послідовності залежать від попереднього та наступного стану автомата і від вхідних сигналів.

Більшість керуючих автоматів, що широко застосовуються на практиці, діють по синхронному принципу по колу, вони здійснюють перехід у наступний стан за активним фронтом синхросигналу. Генератор синхросигналів зв'язаний з усіма станами та із тригерами-засувками регістру пам'яті, ці тригери запам'ятовують вхідні сигнали за переднім фронтом синхросигналу. Також можлива побудова і асинхронного керуючого автомата, робота якого заснована на затримці поширення сигналу в логічних елементах схеми комбінаторної логіки, ця затримка використовується для організації пам'яті автомата. У таких автоматах є висока імовірність помилок, їх важко проектувати і тому вони не знайшли широкого застосування на практиці. У цій статті ми зосередимо увагу тільки на послідовних керуючих автоматах.



a. Mealy State Machines



b. Moore State Machines

90005A-3

Рисунок 3. Два стандартних типи керуючих автоматів

ЗАСТОСУВАННЯ КЕРУЮЧИХ АВТОМАТІВ

Керуючі автомати застосовуються в основному для контролю функціонування цифрових систем. Здійснення вибірки команд у таких системах та застосування для цього керуючих автоматів описано нижче.

Застосування керуючих автоматів у цифрових сигнальних мікроконтролерах (DSP) дозволяє досягти максимальну швидкість та широкі функціональні можливості функціонування цих мікроконтролерів без застосування складних мікропроцесорів. У простих алгоритмах, наприклад в алгоритмі швидкого перетворення Фур'є (FFT), керуючі автомати можуть керувати множенням та додаванням у мікроконтролер наборів векторів. Для виконання складних операцій у DSP краще застосовувати програмувальні мікроконтролери. Але з іншого боку,

програмувальні DSP уступають за швидкістю функціонування спеціалізованим системам.

Розглянемо функціонування відеоконтролера. У цих системах потрібно формувати адреси для вибірки та обчислення операндів різної довжини у різноманітних послідовностях. Ці послідовності обчислень контролюються та здійснюються за допомогою переходів керуючого кінцевого автомата. Застосування автомата дозволяє помітно скоротити апаратні витрати у системі. Стан лічильників може бути встановлений або відновлений керуючим автоматом, за рахунок чого ресурси мікропроцесора звільняються для виконання інших операцій.

У сфері периферійного контролю застосування керуючих автоматів може бути дуже ефективним. Розглянемо випадок RLL кодування. Задачі кодування та декодування можуть бути покладені на керуючий кінцевий автомат, який буде читати вхідний потік даних і формувати вихідний потік.

Застосування керуючих автоматів також можливо у таких областях техніки, як індустріальний контроль та робототехніка. Рішення таких завдань, як розташування механічного маніпулятора та обчислення тригонометричних функцій, звичайно не вимагає застосування потужних мікропроцесорів зі стеками та покажчиками. Скоріше потрібен пристрій, який здатний зберігати обмежену кількість станів і здійснює прості умовні переходи.

Кодування та декодування даних - це ті проблеми, які потрібно вирішувати при шифруванні та дешифруванні інформації у сфері технічного забезпечення засобів масової інформації, у цій сфері потрібно застосовувати системи, що надійно захищені від злому. Для реалізації цих завдань ідеально підходять керуючі кінцеві автомати, тому що їхні елементи пам'яті програмуються під час розробки автомата і потім неможливо буде одержати доступ до цих елементів ззовні.

ФУНКЦІЇ, ЯКІ ВИКОНУЮТЬ КІНЦЕВІ АВТОМАТИ

Усі функції, які виконує кінцевий керуючий автомат, можуть бути виділені у наступні основні категорії:

- арбітраж;
- контроль подій;
- багаторазове тестування умов;
- вибір часу затримок;
- контроль генерації сигналів керування.

Нижче ми розглянемо приклад проекту і покажемо на ньому, як можна використати функції, які були перераховані, при проектуванні керуючого автомата.

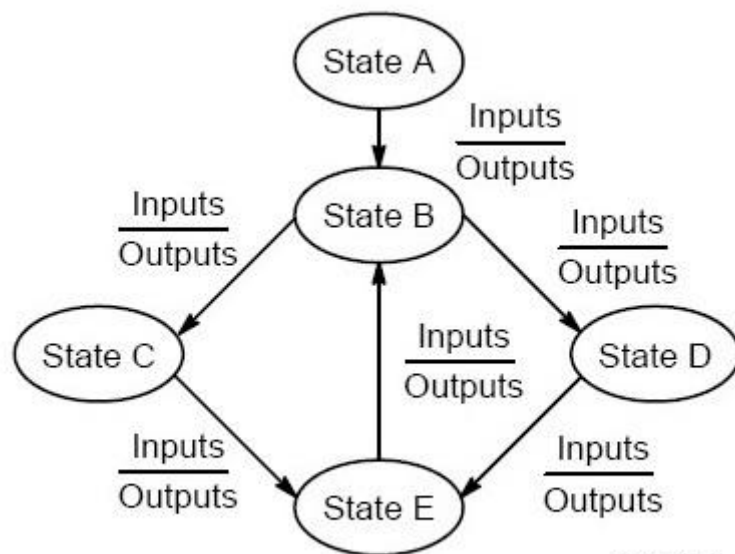
ТЕОРІЯ КЕРУЮЧИХ АВТОМАТІВ

Давайте швидко розглянемо теоретичні основи послідовних логічних схем, кінцевих керуючих автоматів (ККА) або просто керуючих автоматів.

Ті частини цифрових систем, вихідні сигнали яких залежать від минулих та поточних входних сигналів, можуть бути представлені та промодельовані у вигляді кінцевих керуючих автоматів. "Історія" станів автомата впливає на його поточний внутрішній стан. Коли на який-небудь зі входів ККА надходить активне значення сигналу, це змушує здійснити перехід автомата у новий стан та генерувати нові вихідні сигнали. У свою чергу цей новий стан автомата також залежить від значень входних сигналів та від поточного стану керуючого автомата. Структура ККА зображена на рис. 2. Внутрішній стан керуючого автомата зберігається у блоці пам'яті (регістрі). Як було сказано раніше, в автоматі присутні дві комбінаційні логічні схеми: схема формування функцій переходу та схема формування вихідних сигналів керуючого автомата.

ПОБУДОВА ДІАГРАМ СТАНІВ

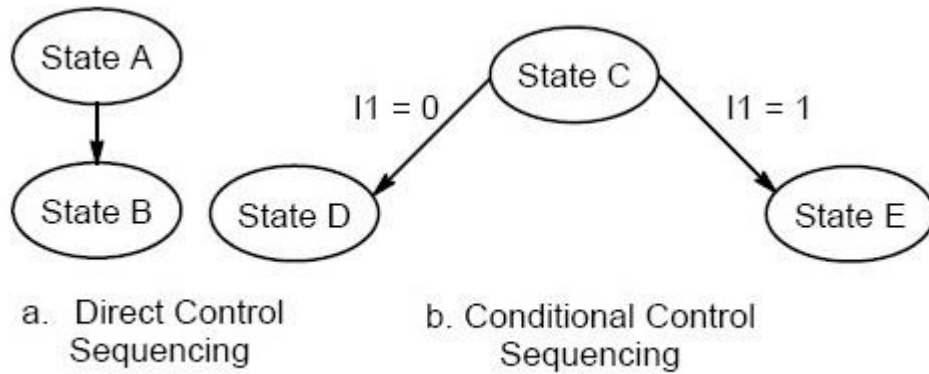
Поведінка кінцевих керуючих автоматів може бути представлена у графічному вигляді, як показано на рис. 4. Подання функціонування автомата у такому виді називають діаграмою станів або діаграмою переходів. Кожне коло являє собою окремий стан, а стрілки - це переходи між станами. Умови, які викликають той чи інший перехід, зображуються біля відповідної стрілки переходу.



90005A-4

Рисунок 4. Графічне подання функціонування керуючого автомата

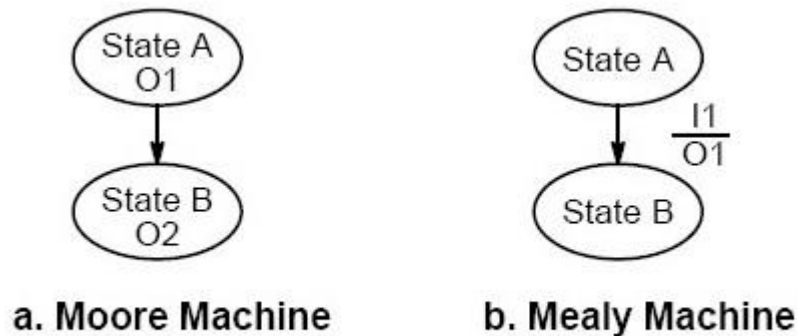
Упорядковані переходи між станами автомата представлені на рис. 5. Прямий перехід зображується як перехід зі стану А у стан В. Умовний перехід зі стану С здійснюється залежно від вхідного сигналу I1 або у стан D, або у стан Е.



90005A-5

Рисунок 5. Види переходів між станами

Перехід між станами в автоматі Мура зображений на рис. 6. В автоматі Мілі перехід між станами здійснюється так, як показано на рис. 4. Більш докладний опис процесу переходу між станами в автоматах Мілі та Мура буде здійснений пізніше.



90005A-6

Рисунок 6. Організація переходів у автоматах Мура та Мілі

У наведених прикладах може виникнути плутанина у процесі визначення вхідних та вихідних сигналів, що ініціюють перехід між станами та що позначені біля відповідних стрілок переходів. Вхідні та вихідні сигнали пишуться через дріб для того, щоб їх не переплутати (рис. 4 і 6). Іноді також використовуються більш детальні написи, які описують вхідні та вихідні сигнали переходів.

Діаграми станів можна зробити більш компактними, ставлячи на переходах не вхідні параметри, які викликають цей перехід, як на рис. 4,

а булеві функції, які викликають цей чи інший перехід. Наприклад, у діаграмі на рис. 7 вхідні сигнали, що ініціюють той чи інший перехід, позначені як Start, X1 та X2. У переході між станами 1 та 2 сигнали X1 та X2 не використовуються (тобто вони не впливають на перехід), і тому вони не позначені біля стрілки відповідного переходу на діаграмі. Таке представлення скорочує запис і робить діаграму наочніше.

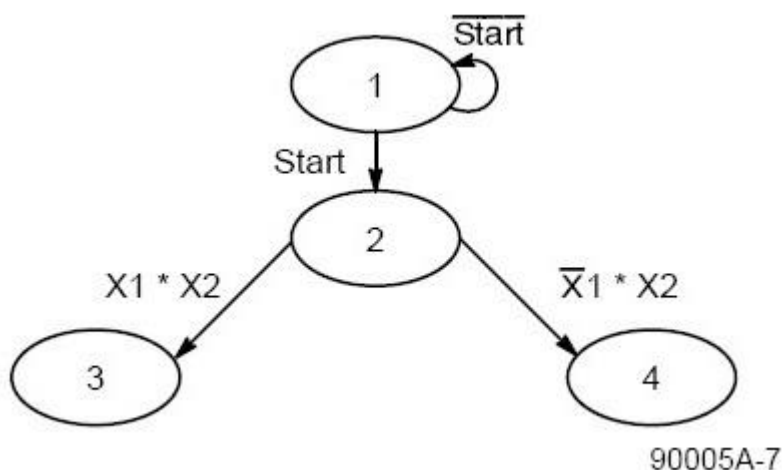


Рисунок 7. Діаграма станів із позначенням функцій переходів у вигляді булевих функцій

При застосуванні цього методу позначення переходів можуть виникнути невизначені ситуації. Переходи між станами діаграми на рис. 8 показують те, що може трапитися. Є три вхідні комбінації $(I0, I1, I2, I3) = \{1011\}, \{1101\}$ та $\{1111\}$, які роблять обидві функції $(I0 * I2 + I3)$ та $(I0 * I1 + I0 * I2)$ дійсними. Перехід же одночасно у два стани неможливий, ця ситуація – це помилка специфікації діаграми. У цьому випадку функції переходів повинні бути змінені або треба гарантувати, що три вищеописані комбінації ніколи не виникнуть. У цьому прикладі проблему можна вирішити, змінивши $(I0 * I1 + I0 * I2)$ на $(I0 * I1 + I0 * I2) * I3$.

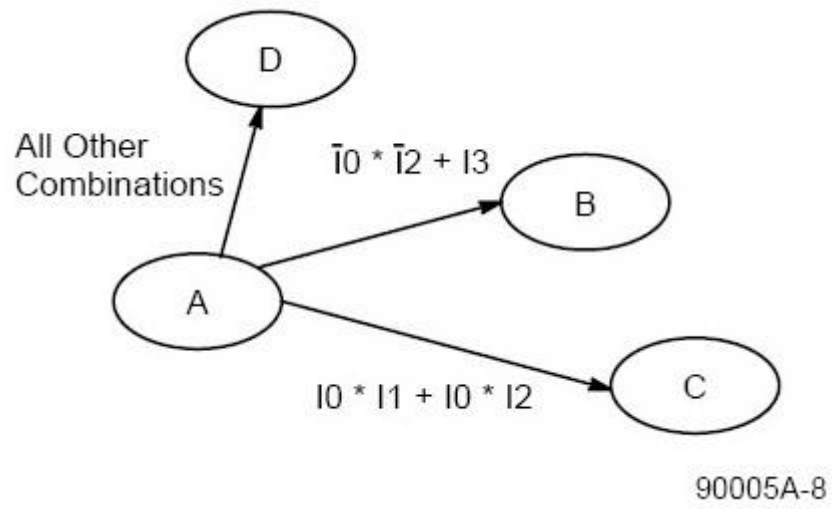


Рисунок 8. Діаграма станів із прикладом конфліктних переходів