

Методи декомпозиції гса на підграфи для синтезу автоматів Мілі на лічильнику

Баркалов А.А.
ун-т Зеленогурский (Польша)
A.Barkalov@iie.uz.zgora.pl

Красичков А.А., Выприцкая П.А.
Кафедра ЭВМ ДонНТУ

Аннотация

Баркалов А.А., Красичков А.А., Выприцкая П.А. Методы декомпозиции ГСА на подграфы для синтеза автоматов Мили на счетчике. В статье рассматриваются управляющие автоматы Мили на счетчике, оптимизация быстродействия и аппаратных затрат при их использовании, необходимость создания системы автоматизированного проектирования (САПР) этих автоматов. В качестве начального этапа создания САПР разработаны алгоритмы разбиения ГСА на ЛПС двух типов, также обнаружена закономерность, позволяющая применить алгоритм разбиения одного типа для получения разбиения другого.

Анотація

Баркалов О.О., Красичков О.О., Выприцька П.О. Методи декомпозиції ГСА на підграфи для синтезу автоматів Мілі на лічильнику. В статті розглядаються керуючі автомати Мілі на лічильнику, оптимізація швидкодії та апаратних затрат при їхньому використанні, необхідність створення системи автоматизованого проектування (САПР) цих автоматів. Як перший етап створення САПР розроблені алгоритми розбивання ГСА на ЛПС двох типів, також знайдена закономірність, що дозволяє використати алгоритм розбивання одного типу для отримання розбивки іншого.

Abstract

Barkalov A.A., Krasichkov A.A., Vypritskaya P.A. Methods of flowchart decomposition on sub-charts for Mealy FSM synthesis with counter. In article the Mealy FSM on counter, optimization of speed and hardware expenses are considered at their use, necessity of CAD creation for these automata's. As the initial stage of CAD algorithms creation the splitting flowchart on linear state sequences (LSS) of two types are developed, rule allowing to apply algorithm of splitting of one type for reception of splitting another also is found out.

Вступ

У наш час відбувається автоматизація процесів у всіх сферах знань навколишнього світу. У деяких сферах застосування ця автоматизація зводиться до створення програмного забезпечення для персонального комп'ютера, але в більшості випадків вимагає створення спеціалізованих ЕОМ. Для організації роботи ЕОМ, як одна з її складових, необхідний керуючий автомат. У його функції входить формування керуючих сигналів, на основі деякого алгоритму для операційної частини ЕОМ.

Алгоритм управління системи задається кодом управління, що надходить в КА із зовнішнього середовища. Алгоритм управління ОА називається мікропрограмою і реалізується керуючим автоматом. Однією з основних проблем при створенні спеціалізованої ЕОМ є пошук компромісу між швидкістю, вартістю і універсальністю КА. Також важливим є час проектування і реалізації схеми КА.

Як відомо, КА з жорсткою логікою виграють за швидкістю перед автоматами з програмованою логікою, але мають жорстку структуру і не можуть бути змінені. ПЛІС дозволяють замінити весь пристрій конфігурований на мікросхему. Таким чином з'являється можливість створювати перепрограмовані КА з жорсткою логікою [1].

Особливості синтезу автоматів Мілі з розділенням кодів

Доведено, що для алгоритмів, в яких не менше 75% вершин складають операторні, оптимізація апаратних витрат в схемі автомата можлива за рахунок заміни регістра пам'яті лічильником. [2]

Для використання лічильника запропоновано ГСА розподілити на підграфи – лінійні послідовності станів (ЛПС). Таке розбивання ГСА дозволяє використовувати подвійну нумерацію станів вигляду: <номер ЛПС> : <номер вершини усередині ЛПС>. Автомат Мілі, побудований з використанням розподілення ГСА на ЛПС, в роботі [3] запропоновано називати РС₁R-автоматом (рис. 1).

При переході до чергової ЛПС, по сигналу Inc=0 в регістр RG заноситься код Φ_C ЛПС α_g , а в лічильник СТ код $\Phi_{СТ}$ стану a_m усередині ЛПС α_g . Адресація чергового стану a_{m+1} усередині ЛПС α_g відбувається шляхом інкременту лічильника по сигналу Inc=1, при цьому значення $\Phi_{СТ}$ і Φ_C не визначені, а код стану автомата визначається парою <T_C, T_{СТ}>.

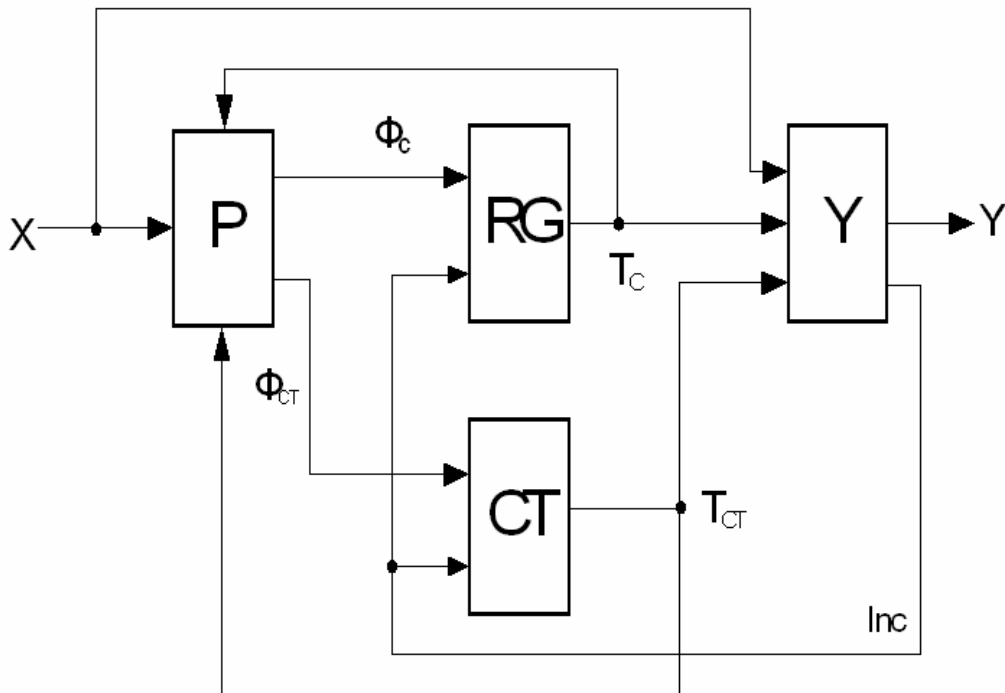


Рисунок 1 – Структура PC₁R-автомата

На реєстр і на лічильник поступають також сигнали синхронізації (Clock) і скидання (Reset), не вказані в структурі автомата.

При дослідженні PC₁R-автоматів було встановлено, що для кожного конкретного алгоритму можна домогтися економії апаратних витрат якщо використовувати один із чотирьох запропонованих варіантів ЛПС:

1. довільна ЛПС – більше одного входу, більше одного виходу (ЛПС¹);
2. ЛПС із одним входом (ЛПС²);
3. ЛПС із одним виходом (ЛПС³);
4. ЛПС із одним входом і одним виходом (ЛПС⁴).

Автомати, при синтезі яких використовується один із зазначених типів ЛПС в роботі [3] запропоновано називати відповідно PC₁R₁, PC₁R₂, PC₁R₃, PC₁R₄.

Для того щоб підібрати оптимальний варіант розбивки на ЛПС для конкретного алгоритму необхідно синтезувати досить велику кількість варіантів і проаналізувати отримані результати.

Зробити це «вручну» навіть для порівняно невеликих ГСА вимагає значного часу й не гарантує відсутності помилок, тому актуальним є створення програмного засобу автоматизації синтезу PC₁R-автоматів.

Алгоритм розбивки ГСА на ЛПС

Першим кроком створення системи синтезу автоматів буде розробка методів розбивки ГСА на ЛПС різних типів.

Очевидно, що для кожного типу ЛПС, методи розбивки ГСА будуть різними. Найпростішим буде розбивання ГСА на ЛПС⁴.

Для того, щоб розбити ГСА на ЛПС інформація про керуючі сигнали, що формуються при переходах, та про зовнішні умови, за якими відбуваються переходи, неважливі. Важливою є лише топологія вихідної ГСА. Для полегшення сприйняття ГСА, пропонується замінити її графом переходів, де вершини – це стани ГСА, ребра – переходи між ними.

Як було зазначено вище, ЛПС⁴ може мати лише один вхід, та один вихід. Слід пам'ятати, що вхід та вихід – це вершини графу, а не переходи. На рис. 2 наведено граф переходів, поділений на ЛПС⁴.

Розглянемо детальніше отриману розбивку.

Існує 4 типи вершин графу переходів:

1. вершина, в яку відбувається один перехід, та з якої відбувається теж один перехід;
2. вершина, в яку відбувається один перехід, та з якої відбувається декілька переходів;
3. вершина, в яку відбувається декілька переходів, та з якої відбувається один перехід.
4. вершина, в яку та з якої відбувається декілька переходів.

Вони наведені на рис. 3.

Як видно з рисунку 2, вершина графу переходів типу 4 може бути в ЛПС⁴ лише єдиною, тому що одночасно є і входом і виходом ЛПС. Вершина типу 3 може бути в ЛПС лише першою, а вершина типу 2 лише останньою. Таким чином, усі вершини типу 4 автоматично стають окремими ЛПС. Усі вершини, до яких здійснюється перехід з вершин типів 2 та 4 стають початком ЛПС. Формування поточної ЛПС закінчується тоді, коли відбувається перехід з вершини вже включеної до поточної ЛПС у вершини типів 2, 3 або 4, при цьому вершина типу 2 включається до поточної ЛПС, а вершини типів 3 та 4 – ні.

З усього вищезазначеного слідує, що розбивка на ЛПС⁴ є однозначною і може бути виконана лише одним способом, а це означає, що алгоритм розбивки буде послідовним.

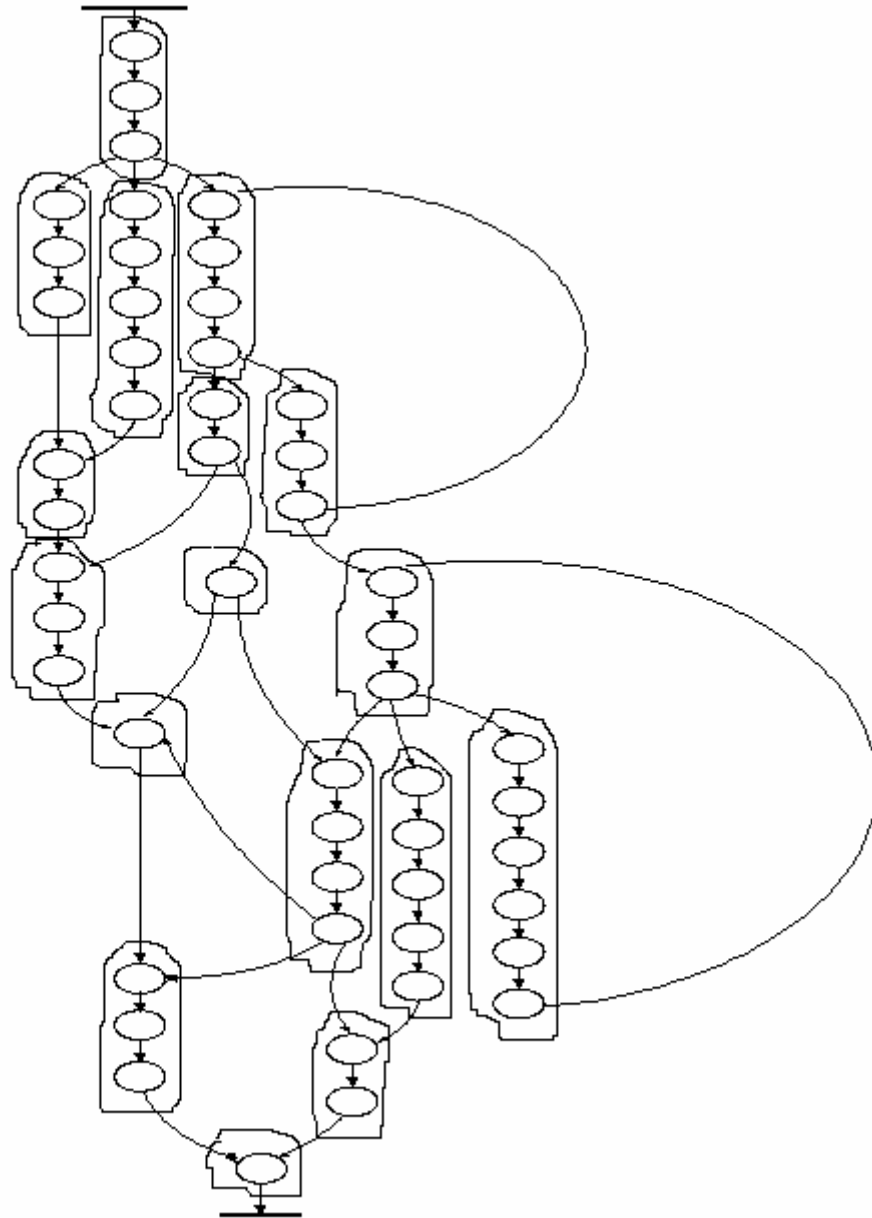


Рисунок 2 – Граф переходів, поділений на ЛПС⁴

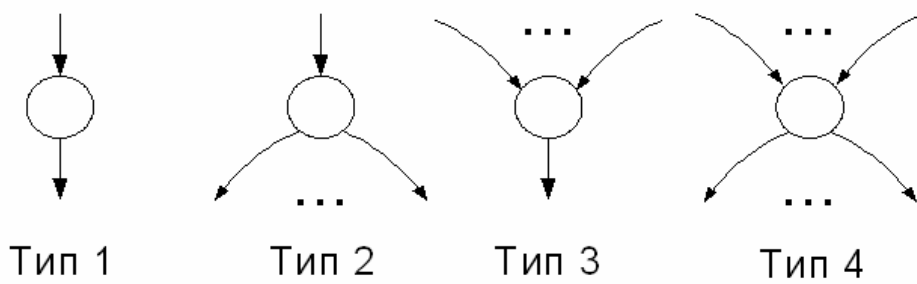


Рисунок 3 – Типи вершин графу переходів

Нехай на деякому етапі розбивки ГСА формується ЛПС_i та розглядається вершина a_j , до якої здійснюється перехід з вершини $a_{j-1} \in$ ЛПС_i, при чому вершина a_{j-1} типу 1 або 3:

1. Якщо вершина a_j типу 1, то включаємо її до ЛПС_i та переходимо до розглядання наступної вершини (до якої здійснюється перехід з a_j).
2. Якщо вершина a_j типу 2, то включаємо її до ЛПС_i, завершуємо ЛПС_i та починаємо ЛПС_{i+1} з однієї з вершин, до яких здійснюється перехід з a_j , а інші заносимо до списку перших у ЛПС вершин.
3. Якщо вершина a_j типу 3, то завершуємо ЛПС_i та починаємо ЛПС_{i+1} з a_j .
4. Якщо вершина типу 4, то завершуємо ЛПС_i, формуємо ЛПС_{i+1} з a_j та починаємо ЛПС_{i+2} з однієї з вершин, до яких здійснюється перехід з a_j , а інші заносимо до списку перших у ЛПС вершин.
5. Якщо перехід з a_j здійснюється у кінець графу переходів і ще залишилися нерозподілені вершини, то починаємо формування наступної ЛПС з вершини зі списку перших.

Тепер розглянемо розбивку графу переходів на ЛПС². Ці ЛПС відрізняються від вищерозглянутих тим, що вони можуть мати декілька виходів.

Перед початком розбивання графу переходів на ЛПС² пропонується зробити розбивку на ЛПС⁴ та замінити граф переходів графом ЛПС⁴, де вершини – це ЛПС⁴, ребра – переходи між ними, про кожну вершину такого графу зберігається інформація про кількість станів у цій ЛПС.

Приклад роботи алгоритму

На рисунку 4 наведено граф ЛПС⁴. Цей граф повністю зберігає якості графу переходів та значно скорочує його розміри. При такій заміні зберігаються також типи вершин.

При розбиванні на ЛПС² вершини типів 3 та 4 можуть бути в ЛПС лише першими. Для безлічі вершин, перехід до яких здійснюється з вершини типу 2 або 4, одна буде приєднана до попередньої, а усі інші стануть початковими для наступних ЛПС.

Оскільки вершини типів 2 та 4 можуть бути не останніми в ЛПС², то виникає варіантність у виборі вершини, яку приєднати до попередньої, якщо вона типу 2 або 4. Потрібно визначити умови, за якими можна буде здійснювати той чи інший вибір.

Розміри схеми автомату, та його характеристики залежать від розрядності двох величин:

- кількість ЛПС,
- довжина найбільшої ЛПС,

при чому перший параметр важливіший за другий.

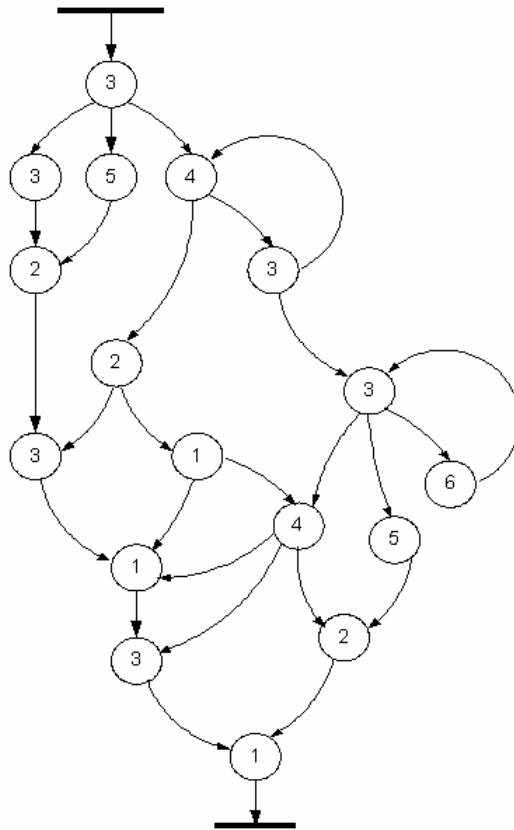


Рисунок 4 – Граф ЛПС⁴

Таким чином, необхідно мінімізувати кількість ЛПС, та при незмінній їх кількості мінімізувати довжину найбільшої ЛПС. Але якщо виконувати розбивання за всіма правилами, та навмисно не розбивати ЛПС там де це не потрібно, кількість ЛПС буде величиною постійною, яка не залежить від варіанту розбивання. Таким чином мінімізувати потрібно лише довжину ЛПС. Це можна зробити якщо обираючи яку вершину приєднати до ЛПС обирати ту, що дасть меншу довжину ЛПС.

На рис. 5 наведено граф ЛПС⁴, розбитий на ЛПС².

Нехай на деякому етапі розбивки ГСА формується ЛПС_{*i*} та розглядається вершина a_j , до якої здійснюється перехід з вершини $a_{j-1} \in \text{ЛПС}_i$, при чому вершина a_{j-1} типу 1 або 3:

1. Якщо вершина a_j типу 1, то включаємо її до ЛПС_{*i*} та переходимо до розглядання наступної вершини (до якої здійснюється перехід з a_j).
2. Якщо вершина a_j типу 2, то включаємо її до ЛПС_{*i*}, умовно завершуємо ЛПС_{*i*} та формуємо список умовно перших вершин з індексом i з вершин, до яких здійснюється перехід з a_j . З однієї з цих вершин починаємо формування ЛПС. Коли ЛПС будуть сформовані з вершин усього списку, найкоротшу приєднаємо до ЛПС_{*i*}.
3. Якщо вершина a_j типу 3, то завершуємо ЛПС_{*i*} та починаємо ЛПС_{*i+1*} з a_j .

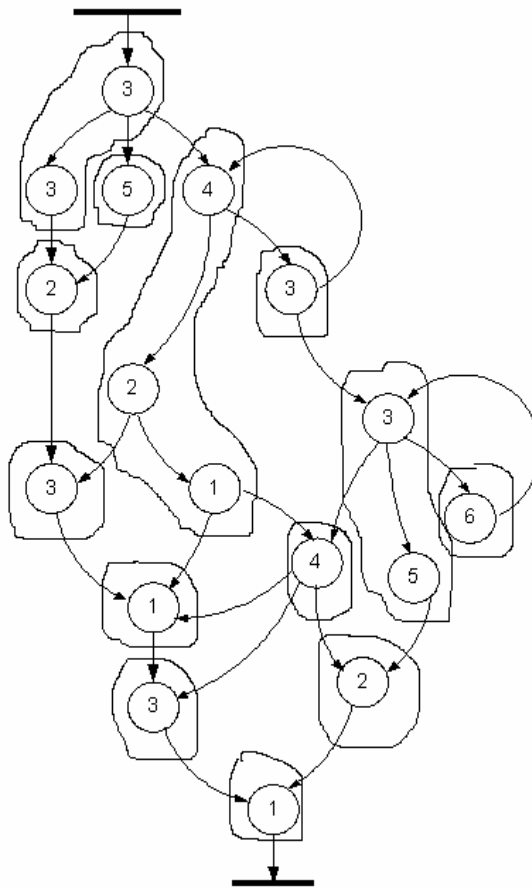


Рисунок 5 – Граф ЛПС⁴

4. Якщо вершина типу 4, то завершуємо ЛПС_i, починаємо ЛПС_{i+1} з a_j , формуємо список умовно перших вершин з індексом $i+1$ з вершин, до яких здійснюється перехід з a_j . З однієї з цих вершин починаємо формування ЛПС. Коли ЛПС будуть сформовані з вершин усього списку, найкоротшу приєднаємо до ЛПС_{i+1}.

5. Якщо перехід з a_j здійснюється у кінець графу переходів і ще залишились нерозподілені вершини, то починаємо формування наступної ЛПС з вершини зі списку перших або умовно перших.

Перейдемо до розбивання графу переходів на ЛПС³. Ці ЛПС можуть мати декілька входів та лише один вихід.

При розбиванні на ЛПС³ вершини типів 2 та 4 можуть бути в ЛПС лише останніми. Для безлічі вершин, перехід з яких здійснюється до вершини типу 3 або 4, вона буде приєднана до однієї з попередніх.

Висновки

При дослідженні особливостей цього розбивання було виявлено таку закономірність: якщо транспонувати граф, тобто інвертувати усі його

зв'язки, для розбивання графу на ЛПС³ стає застосовним алгоритм розбивання на ЛПС², лише з тією різницею, що граф може мати декілька входів. Потрібно лише замінити перший блок алгоритму формуванням списку перших вершин. Таким чином алгоритм розбивання на ЛПС² стає універсальним, що значно спрощує подальшу розробку системи автоматичного синтезу автоматів.

Література

1. Грушвицкий Р.И. Проектирование систем на микросхемах программируемой логики / Р.И. Грушвицкий, А.Х. Мурсаев, Е.П. Угрюмов – СПб., БХВ-Петербург, 2002. – 608 с.
2. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах / А.А. Баркалов. – Донецк: ДонНТУ, 2002. – 262с.
3. Красичков А.А. Методы синтеза управляющих автоматов на конфигурируемых логических блоках / А.А. Красичков – Диссертация канд. тех. наук: 05.13.13. – Донецк: ДонНТУ, 2004. – 137с.



Баркалов Александр Александрович.

Доктор технических наук, профессор. Работает профессором на кафедре “Электронные вычислительные машины” Донецкого национального технического университета.

Научные интересы: цифровые устройства управления, системы-на-кристалле.



Красичков Алексей Александрович.

Кандидат технических наук, доцент. Работает доцентом на кафедре “Электронные вычислительные машины” Донецкого национального технического университета.

Научные интересы: цифровые устройства управления.



Выприцкая Полина Александровна.

В 2008 году заканчивает магистратуру по специальности “Системное программирование” в Донецком национальном техническом университете.

Научные интересы: программирование, моделирование