

## ОПТИМИЗАЦИЯ СХЕМ УСТРОЙСТВ УПРАВЛЕНИЯ НА ПРОГРАММИРУЕМЫХ БИС

Баркалов А.А., Ковалев С.А., Красичков А.А., Зеленева И.Я.

Донецкий национальный технический университет

В настоящее время для синтеза логических схем устройств управления (УУ) широко применяются программируемые логические устройства типа ПЛМ, ПМЛ, ППЗУ [1]. При этом традиционно актуальной является задача минимизации аппаратных затрат в логической схеме УУ. Один из путей решения этой задачи – учет особенностей интерпретируемой граф-схемы алгоритма (ГСА). В настоящей работе рассматриваются методы оптимизации числа ПЛУ в схемах микропрограммных автоматов (МПА) Мили и Мура, реализуемых по линейным ГСА, в которых доля операторных вершин составляет не менее 75% от общего числа вершин.

Исходной информацией для синтеза МПА является граф-схема алгоритма (ГСА). В операторных вершинах ГСА записываются наборы микроопераций  $Y_i \subseteq Y$ , где  $Y = \{y_1, \dots, y_N\}$  – множество микроопераций. В условных вершинах ГСА записываются элементы множества логических условий  $X = \{x_1, \dots, x_L\}$ . После отметки ГСА элементами множества состояний  $A = \{a_1, \dots, a_M\}$  строится прямая структурная таблица (ПСТ) со столбцами [1]:  $a_m$  – исходное состояние;  $K(a_m)$  – код состояния  $a_m \in A$  разрядности  $R = \lceil \log_2 M \rceil$ , для кодирования состояний используются внутренние переменные, образующие множество  $T = \{T_1, \dots, T_R\}$ ;  $a_s$ ,  $K(a_s)$  – состояние перехода и его код соответственно;  $X_h$  – конъюнкция некоторых элементов (или их отрицаний) множества  $X$ , определяющая переход  $\langle a_m, a_s \rangle$ ;  $Y_h$  – набор микроопераций, формируемый на переходе  $\langle a_m, a_s \rangle$  (автомат Мили) или в состоянии  $a_m$  (автомат Мура);  $\Phi_h$  – набор функций возбуждения памяти из множества  $\Phi = \{\phi_1, \dots, \phi_R\}$ , принимающих единичное значение для переключения памяти из  $K(a_m)$  в  $K(a_s)$ ;  $h = 1, \dots, H$  – номер перехода.

Логическая схема МПА строится по системе функций  $\Phi = \Phi(T, X)$  и  $Y = Y(T, X)$ , формируемой по ПСТ в виде:

$$\varphi_r = \bigvee_{h=1}^H C_{rh} F_h \quad (r=1, \dots, R),$$

$$y_n = \bigvee_{h=1}^H C_{nh} F_h \quad (n=1, \dots, N),$$
(1)

где  $C_{rh}$  ( $C_{nh}$ ) – булева переменная, равная единице, если и только если в  $h$ -й строке ПСТ записана функция  $\varphi_r$  ( $y_n$ ); переменная  $F_h$  называется термом ПСТ и определяется как

$$F_h = A_m^h X_h \quad (h=1, \dots, H),$$
(2)

где  $A_m^h$  – конъюнкция внутренних переменных, соответствующая коду состояния  $a_m \in A$  из  $h$ -й строки ПСТ.

Ограничимся в дальнейшем такими представлениями класса ПЛУ как программируемые логические матрицы (ПЛМ) и программируемые постоянные запоминающие устройства (ППЗУ). Отметим, что для минимизации множества  $\Phi$  память автомата строится как регистр RG с информационными входами D-типа [3].

Рассмотрим более подробно методы оптимизации схемы автомата Мили.

При выполнении условий

$$\begin{aligned} L + R &\leq S, \\ N &\leq q, \\ N + R &\leq t, \end{aligned}$$
(3)

где  $s$ ,  $t$ ,  $q$  – число входов, выходов и промежуточных шин (термов) ПЛМ соответственно, автомат тривиально реализуется на одной ПЛМ. Однако для задач практической сложности условие (3) выполняется крайне редко, поэтому схема МПА реализуется как сеть ПЛУ.

Если критерием оптимальности схемы МПА является максимальное быстродействие, то при выполнении условия

$$L_{\max} + R < S,$$
(4)

где  $L_{\max}$  – максимальное число букв в термах (2), автомат реализуется в виде одноуровневой схемы, называемой в дальнейшем Р-автоматом. Р-автоматы, однако, отличаются значительной избыточностью реали-

зации [3]. Кроме того, если (4) не выполняется, то необходимо преобразовывать исходную ГСА.

Оптимизация схемы МПА возможна за счет увеличения числа уровней (структурная редукция), использования различного базиса для различных уровней (гетерогенная реализация) и оптимального кодирования состояний и промежуточных переменных (алгоритмические методы). Эти методы, как правило, используются совместно [4].

Методы структурной редукции включают в себя:

1. Замену логических условий  $X$  переменными  $P = \{p_1, \dots, p_G\}$ , где  $G \ll L$ . Это порождает МР-автоматы, М-подсхема которых реализуется на мультиплексорах МХ. Преобразование исходной ГСА позволяет менять число переменных в множестве  $P$ , что порождает  $M_1P$ -,  $M_2P$ -, ...,  $M_GP$ -автоматы [5]. Оптимизация М-подсхемы возможна за счет оптимального кодирования состояний ( $M_gCP$ -автоматы) и преобразования кодов состояний в коды логических условий ( $M_gLP$ -автоматы) [5].
2. Кодирование наборов микроопераций  $Y_t \subseteq Y$  двоичными кодами разрядности  $Q = \lceil \log_2 T_0 \rceil$ , где  $T_0$  – число наборов. Это порождает РУ-автоматы, Y-подсхема которых реализуется на ППЗУ.
3. Кодирование полей совместимых микроопераций, что порождает РD-автоматы. Вертикализация исходной ГСА [6] позволяет менять число классов от 1 до  $I$ , где параметр  $I$  определяется характеристиками ГСА. Такое преобразование порождает  $PD_1$ -,  $PD_2$ -, ...,  $PD_I$ -автоматы.
4. Кодирование строк ПСТ двоичными кодами  $K(F_h)$  разрядности  $Q_0 = \lceil \log_2 H \rceil$ . Это порождает РF-автоматы, в которых F-подсхема реализуется на ППЗУ.

Для минимизации числа дополнительных переменных автомат Мили может быть реализован как РR-автомат [7], в котором термы (2) реализуются в виде

$$F_h = A_m^h A_s^h \quad (h=1, \dots, H). \quad (5)$$

Естественно, состояния  $a_m$  и  $a_s$  должны кодироваться различными переменными, что достигается за счет введения в схему МПА дополнительного регистра  $RG_1$  (Рис.1).

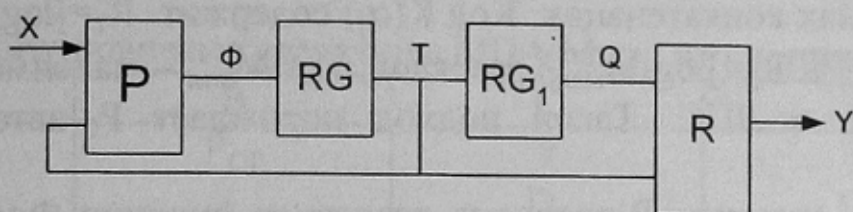


Рисунок 1 - Структурная схема PR-автомата Мили

Если в ПСТ имеется несколько переходов  $\langle a_m, a_s \rangle$  с различными выходными наборами, то корректность PR-автомата нарушается. В этом случае необходимо “расширить” исходные состояния так, чтобы PR-автомат был эквивалентен исходному автомату Мили. Это приводит к увеличению числа состояний и строк ПСТ, однако эти параметры можно уменьшить за счет использования методов, аналогичных методам оптимизации автомата Мура [8]:

1. Оптимальное кодирование состояний ( $P_E R$ -автоматы), при котором коды “расщепленных” состояний входят в один обобщенный интервал пространства кодирования.
2. Преобразование кодов состояний в коды классов состояний ( $P_K R$ -автоматы).
3. Преобразование исходной ГСА ( $P_r R$ -автоматы).

Если число операторных вершин исходной ГСА превышает 75% общего числа вершин (линейная ГСА), то оптимизация схемы возможна за счет использования счетчика СТ вместо регистра RG [8]. При этом возникает задача нахождения линейных последовательностей состояний (ЛПС), аналогичных линейным операторным цепям [6]. В пределах каждой ЛПС  $\alpha_j$  ( $j=1, \dots, J$ ) состояния  $a_m$  и  $a_s$ , соответствующие соседним компонентам ЛПС, кодируются в естественном порядке

$$K(a_m) = K(a_m) + 1. \tag{6}$$

Назовем одноуровневые схемы автоматов Мили на счетчике  $P_{CT}$ -автоматами. Для оптимизации характеристик  $P_{CT}$ -автоматов в настоящей работе предлагается использовать идею работы [8] и представить код состояния в виде конкатенации кода ЛПС  $K(\alpha_j)$  и кода компоненты  $K(a_m')$ , соответствующей состоянию  $a_m$  в ЛПС  $\alpha_j$ :

$$K(a_m) = K(\alpha_j) * K(a_m'), \tag{7}$$

где \* – знак конкатенации. Код  $K(\alpha_j)$  содержит  $R_1 = \lceil \log_2 J \rceil$  разрядов, код  $K(a_m')$  –  $R_2 = \lceil \log_2 M_{\max} \rceil$  разрядов, где  $M_{\max}$  – максимальное число элементов в ЛПС. Такой подход порождает  $P_C$ -автоматы Мили (рис.2).

В  $P_C$ -автомате  $P$ -подсхема реализует функции  $\Phi = \Phi(T, X)$ , формирующие в регистре  $RG$  код  $K(\alpha_j)$ , и  $\psi = \psi(T, X)$ , формирующие в счетчике  $CT$  коды входов ЛПС,  $P'$ -подсхема формирует функции  $Y = Y(T, \tau, X)$ . Схема является одноуровневой, здесь  $P$ - и  $P'$ -подсхемы выделены, чтобы показать различие их функций.

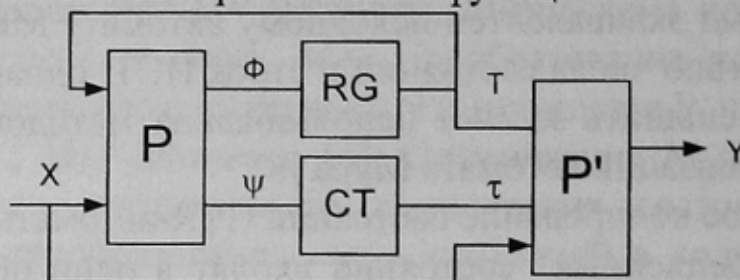


Рисунок 2 - Структурная схема  $P_C$ -автомата Мили

Такой подход целесообразен, если

$$R_1 + R_2 = R, \quad (8)$$

в противном случае аппаратные затраты в  $P_C$ -автомате могут превысить соответствующие параметры  $P_{CT}$ -автомата.

Предложенный метод позволяет:

1. Повысить быстродействие по сравнению с  $P_{CT}$ -автоматом за счет уменьшения разрядности счетчика.
2. Уменьшить число ПЛМ в  $P$ -подсхеме за счет уменьшения числа сигналов обратной связи по сравнению с  $P_{CT}$ -автоматом (с  $R$  до  $R_1$ ).

Применение данной методики к  $PR$ -автоматам порождает  $P_C R$ -,  $P_{CE} R$ -,  $P_{CK} R$ -,  $P_{CF} R$ -автоматы, где второй индекс в обозначении  $P$ -подсхемы указывает на метод оптимизаций  $PR$ -автомата.

Дальнейшая оптимизация логической схемы МПА Мили на ПЛУ и счетчиках возможна за счет увеличения числа уровней в схеме. Все многообразие структур МПА приведено в табл.1.

Генерация структур  $S_i$  по табл.1 аналогична формированию слов вида  $A * B * C * D$ , где  $A$  – приставка,  $B$  – корень,  $C$  – суффикс,  $D$  – окончание. При этом корень должен присутствовать всегда, а остальные части слова могут быть нулевыми.

Таблица 1

Многоуровневые структуры МПА Мили на счетчике

A	B	C	D
M <sub>1</sub>	P <sub>CT</sub>	Y	Y
M <sub>1</sub> C		D <sub>1</sub>	D <sub>1</sub>
M <sub>1</sub> L	P <sub>C</sub>	.	.
.		.	.
.		D <sub>1</sub>	.
.		H	.
M <sub>G</sub>	P <sub>C</sub>	R	D <sub>1</sub>
M <sub>G</sub> C	P <sub>CE</sub>		
M <sub>G</sub> L	P <sub>CK</sub>		
	P <sub>CT</sub>		

Как видно из табл.1, существует две одноуровневые структуры вида S=A, 6G двухуровневых структур вида S=A\*B, 2(I+1)+4 двухуровневых структур вида S=B\*C (C≠H), 6G(I+1)+12G трехуровневых структур вида S=A\*B\*C (C≠H), 4(I+1) трехуровневых структур вида S=B\*C\*D (C=R), 18G(I+1) четырехуровневых структур вида S=A\*B\*C\*D (C=R или C=H). Таким образом, табл.1 задает структуры МПА Мили, число уровней в которых варьируется от одного до четырех. Общее число структур определяется формулой

$$N = 14 + 8I + 42G + 24GI. \tag{9}$$

Так, при G = I = 6, что соответствует алгоритмам управления средней сложности, выражение (9) определяет n = 1178 различных структур.

Для оптимизации МПА Мура применимы все ранее рассмотренные методы, включая методы оптимизации PR-автоматов:

1. Оптимальное кодирование состояний, при котором коды одного класса псевдоэквивалентных состояний входят в один обобщенный интервал R-мерного булева пространства. Это позволяет добиться наиболее экономичного решения за счет исключения дополнительных переменных и аппаратных блоков. Однако такое решение не всегда возможно из-за отсутствия необходимого числа безразличных состояний автомата.

2. Преобразование кодов состояний, при котором коды псевдоэквивалентных состояний одного класса преобразовываются в код класса псевдоэквивалентных состояний. Такой подход позволяет максимально сократить число строк прямой структурной таблицы, однако связан с введением дополнительной аппаратуры.
3. Преобразование исходной граф-схемы алгоритма, при котором каждому классу псевдоэквивалентных состояний соответствует дополнительная операционная вершина. Этот подход связан с увеличением числа тактов работы устройства управления и обеспечивает средний уровень аппаратных затрат по сравнению с двумя предыдущими методами.

Аналогичные расчеты показывают, что при  $G=I=6$  существует  $N_2=1064$  различных варианта реализации схемы автомата Мура.

Таким образом, для автомата средней сложности существует  $N_1 + N_2 = 2242$  различных вариантов реализации.

Проведенные авторами исследования позволили сделать следующие выводы:

1. При выполнении условия (8) для линейных ГСА представление кода состояния автоматов Мили и Мура в виде конкатенации (7) позволяет уменьшить аппаратные затраты по сравнению с автоматами на счетчиках на 18-22%.
2. Наличие значительного числа структур логической схемы автоматов Мура и Мили на ПЛУ и счетчике – формула (9) – вызывает проблему априорного выбора структуры, порождающей схему с минимальными аппаратными затратами и заданным быстродействием. В настоящее время аналитических методов решения этой задачи не существует. Для эффективного ее решения современные САПР должны включать экспертные системы выбора оптимальной структуры.

#### Список источников

1. Баранов С.И. Синтез микропрограммных автоматов. – Л.: Энергия, 1979. – 232 с.
2. Соловьев В.В. Проектирование функциональных узлов цифровых систем на программируемых логических устройствах. – Минск: Бестпринт, 1996 – 252с.
3. Баранов С.И., Складов В.А. Цифровые устройства на программируемых БИС с матричной структурой. – М.: Радио и связь, 1986 – 272с.
4. Баркалов А.А. Разработка формализованных методов структурного синтеза композиционных автоматов. – Диссертационная работа доктора технических наук: 05.13.08 / Донецк: ДонГТУ, 1994. – 201с.