

МЕТОДЫ ОПТИМИЗАЦИИ ЛОГИЧЕСКОЙ СХЕМЫ АВТОМАТА С ДВУНАПРАВЛЕННЫМИ ПЕРЕХОДАМИ

Баркалов А.А., Зеленева И.Я., Саломатин В.А.

Кафедра ЭВМ ДГТУ

barkalov@dstu.donetsk.ua

Abstract

Barkalov A.A., Zeleneva I.J., Salomatina V.A. The optimization methods of the logic circuits for automaton with bi-directional transitions. The methods oriented on the reducing of parameters of multiplexor in the two-level circuit of control unit with bi-directional transitions are proposed. The methods are based on the optimal encoding of the states and transformation of the logical condition's codes to reducing the amount of control inputs of the multiplexor.

Key words: control unit, bi-directional transition, multiplexor, state encoding.

Введение

В состав любой цифровой системы входит устройство управления [1], которое на практике часто реализуется в виде автомата с «жесткой» логикой [2]. В настоящее время для реализации логических схем автоматов широко используются программируемые логические устройства (ПЛУ). Для минимизации числа ПЛУ в схеме автомата часто используется метод замены логических условий, порождающих двухуровневую структуру, называемую МР-автомат [3]. В настоящей работе предлагаются методы оптимизации схемы замены логических условий, основанные на специальном кодировании состояний или логических условий. При этом для определенности рассматривается автомат Мили, хотя предлагаемые методы применимы и для автомата Мура.

Кодирование логических условий

Пусть задан автомат Мили, состояния которого образуют множество $A = \{a_1, \dots, a_m\}$, логические условия – множество $X = \{x_1, \dots, x_l\}$, микрооперации – множество $Y = \{y_1, \dots, y_n\}$. Для кодирования состояний используются внутренние переменные $T_r \in T = \{T_1, \dots, T_R\}$, где $R = \lceil \log_2 M \rceil$. Пусть переходы из любого состояния $a_m \in A$ зависят не более чем от одного логического условия. При этом из каждого состояния возможно не более двух переходов. Назовем такие устройства автоматами с двунаправленными переходами, и в дальнейшем будем говорить только о них. Назовем подобные автоматы, в которых используется метод замены логических условий, M_0P -автоматами. В M_0P -автоматах элементы множества X заменяются новой переменной p , определяемой, как

$$p = \bigvee_{m=1}^M A_m x_{ml}, \quad (1)$$

где A_m – конъюнкция внутренних переменных, соответствующая коду $k(a_m) \in A$, x_{ml} – логическое условие, определяющее переходы из состояния $a_m \in A$. Как правило, автомат Мили задается прямой структурной таблицей [3] со строками $\langle a_m, k(a_m), a_s, k(a_s), X_h, Y_h, \Phi_h, h \rangle$, где a_m – исходное состояние; a_s – состояние перехода; X_h – логическое условие (или его отрицание), определяющее переход из a_m в a_s ; $Y_h \subseteq Y$ – выходной сигнал на

h -ом переходе, Φ_h – набор функций возбуждения, принимающих единичное значение для переключения памяти автомата из $k(a_m)$ в $k(a_s)$, $\Phi_h \subseteq \Phi = \{\varphi_1, \dots, \varphi_R\}$; $h=1..N$ – номер перехода.

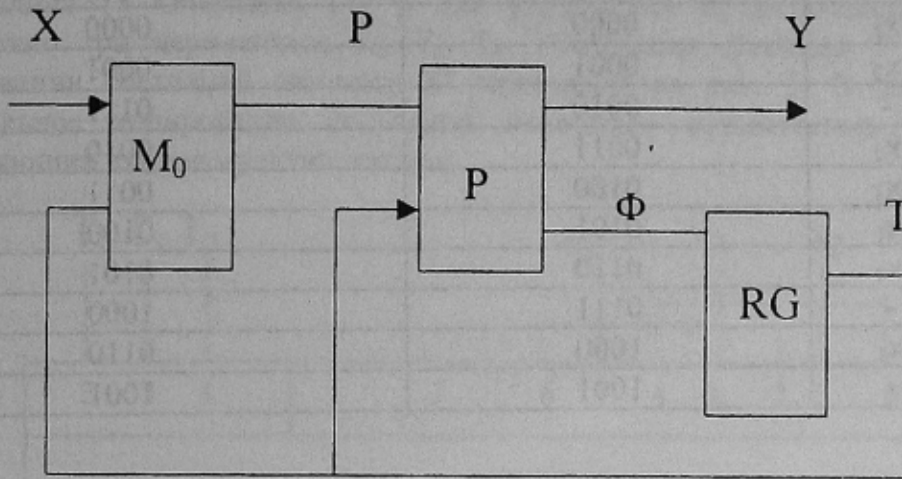


Рис. 1 Структурная схема M_0 -автомата

Для синтеза M_0P -автомата необходимо в каждой строке таблицы заменить переменную X_h переменной p согласно (1). Структурная схема M_0P -автомата (рис. 1) включает M_0 -подсхему, реализующую (1), и P -подсхему, реализующую системы Y и Φ в виде:

$$y_n = \bigvee_{h=1}^N C_{nh} A_m P_h \quad (n=1..N);$$

$$\varphi_r = \bigvee_{h=1}^N C_{rh} A_m P_h \quad (r=1..R).$$

В системе (2) $C_{nh}(C_{rh})$ – булева переменная, равная единице, если и только если в h -ой строке таблицы записана функция $y_n(\varphi_r)$, P_h – значение переменной p (прямое или инверсное) из h -ой строки таблицы.

Очевидно, функция (1) наиболее естественно реализуется на мультиплексоре, на управляющие входы которого подаются внутренние переменные $T_r \in T$, а на информационные входы – переменные $x_i \in X$. Система (2) зависит от $R+1$ переменных и определена более, чем на 50% всевозможных наборов, что делает естественной ее реализацию на ПЗУ.

Рассмотрим пример применения метода замены логических условий для автомата Мили S1, у которого $x(a_1)=x(a_6)=x_1$, $x(a_2)=x(a_4)=x_2$, $x(a_5)=x(a_9)=x_3$, $x(a_7)=x_4$, $x(a_3)=x(a_8)=x(a_{10})=0$. Здесь $x(a_m)$ – множество логических условий, определяющих переходы из состояния $a_m \in A$. Итак, для автомата S1 $M=10$, $L=4$, $R=4$, $T=\{T_1, \dots, T_4\}$. Закодируем состояния $a_m \in A$ так, чтобы код $k(a_m)$ соответствовал двоичному эквиваленту числа $m-1$, то есть $k(a_1)=0000$, $k(a_2)=0001, \dots, k(a_{10})=1001$. Назовем такое кодирование состояний тривиальным кодированием.

В таблице 1 приведена информация для синтеза M_0 -подсхемы. Здесь на пересечении строки a_m и столбца P записаны логические условия x_i в соответствии с (1), в столбце "Управление 1" приведены коды состояний автомата.

Таблица 1

Замена логических условий автомата S1

a _m	P	Управление 1	Управление 2
		T ₁ T ₂ T ₃ T ₄	T ₁ T ₂ T ₃ T ₄
a ₁	x ₁	0000	0000
a ₂	x ₂	0001	0001
a ₃	-	0010	0111
a ₄	x ₂	0011	0010
a ₅	x ₃	0100	0011
a ₆	x ₁	0101	0100
a ₇	x ₄	0110	0101
a ₈	-	0111	1000
a ₉	x ₃	1000	0110
a ₁₀	-	1001	1001

Реализация M₀-подсхемы приведена на рис. 2. В данной реализации функциональные возможности мультиплексора используются не оптимально, так как задействовано 7 информационных входов из 16.

В настоящей работе для уменьшения требований к параметрам мультиплексора и, как следствие, для уменьшения стоимости схемы автомата, предлагается метод оптимального кодирования состояний.

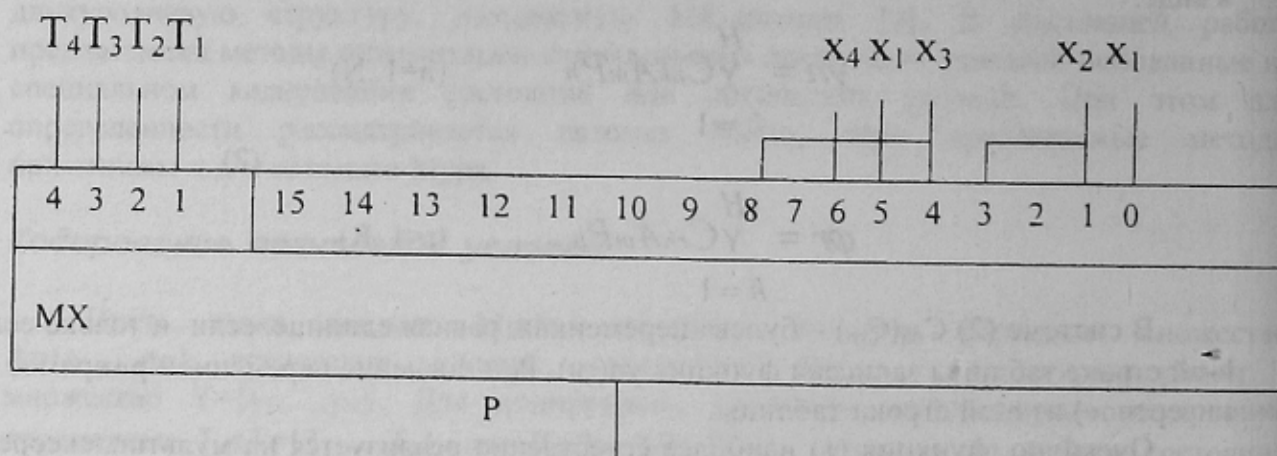


Рис. 2 Реализация схемы замены логических условий при тривиальном кодировании состояний

Оптимальное кодирование состояний

Сформируем множество состояний $V \subseteq A$, включающее состояния $a_m \in A$, у которых $x(a_m) \neq 0$, и начальное состояние a_1 вне зависимости от $x(a_1) \neq 0$. Пусть множество V включает M_1 элементов и $R_1 = \lceil \log_2 M_1 \rceil$. Если выполняется условие

$$R_1 < R \tag{3}$$

то для оптимизации M₀-подсхемы достаточно закодировать состояния следующим образом. Состояниям $a_m \in V$ присваиваются первые коды от нулевого (для a₁) до двоичного эквивалента числа M₁-1. Состояния $a_m \notin V$ кодируются произвольным образом оставшимися M-M₁ кодами. Назовем такое кодирование оптимальным. В результате оптимального кодирования состояний первые R-R₁ переменных в кодах

состояний $a_m \in B$ будут неизменными. Таким образом, для управления мультиплексором можно использовать внутренние переменные T_i, \dots, T_r , где $i=R-R_1+1$.

Для автомата S1 множество $B=\{a_1, a_2, a_4, a_5, a_6, a_7, a_9\}$, $M_1=7$, $R_1=3$ и условие (3) выполняется. Закодируем состояния автомата S1, как показано в столбце "Управление 2" (таблица 1). Очевидно, $i=2$ и для управления мультиплексором M_0 -подсхемы достаточно три переменных T_2, T_3, T_4 . Реализация функции P при оптимальном кодировании состояний автомата S1 приведена на рис. 3. В данной реализации оптимальное кодирование состояний позволило существенно уменьшить число управляющих входов мультиплексора.

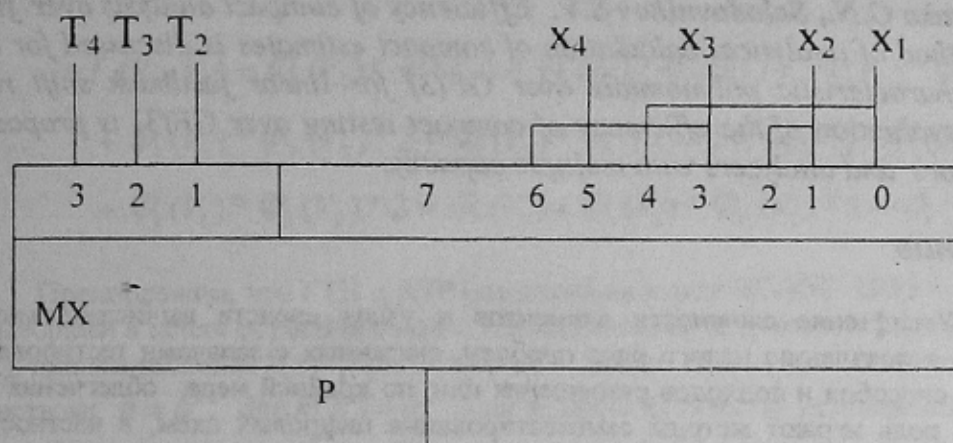


Рис. 3 Реализация схемы замены логических условий при оптимальном кодировании состояний.

Заключение

Для оптимизации схемы замены логических условий автомата с двунаправленными переходами возможно применение следующих методов:

1. Оптимальное кодирование состояний, применимое если выполняется условие (3) и нет других требований к кодированию состояний.
2. Преобразование кодов логических условий, применимое для реализации системы (2), если это ведет к уменьшению аппаратных затрат в схеме устройства управления.

Как видно из реализации M_0 -подсхемы на рис. 2., при тривиальном кодировании состояний функциональные возможности мультиплексора используются только на 44%. Реализация функции P при оптимальном кодировании состояний автомата S1, приведенная на рис. 3., позволила уменьшить число управляющих входов мультиплексора на 25%, при этом информационные входы используются на 88%.

Литература

1. Баркалов А.А. Палагин А.В. Синтез микропрограммных устройств управления. – Киев: ИК НАН Украины, 1997 – 135 с.
2. Скляр В.А. Синтез автоматов на матричных БИС. – Минск: Наука и техника, 1984. – 287 с.
3. Соловьев В.В. Проектирование функциональных узлов цифровых систем на программируемых логических устройствах. – Минск: Бестпринт, 1996. – 252 с.