

Y_2 являются

$$Y_1 = \bar{x}y_2 + y_1 \cdot (\overline{\bar{x}y_2}),$$

$$Y_2 = x\bar{y}_1 + y_2 \cdot (\overline{x\bar{y}_1}).$$

На рис. 4.52 показана реализация этих уравнений без введенных элементов задержки. Заметим, что мы должны использовать логический элемент ИЛИ для реализации \bar{Z}_1 и логический элемент НЕ-И для реализации \bar{Z}_2 в отличие от общей

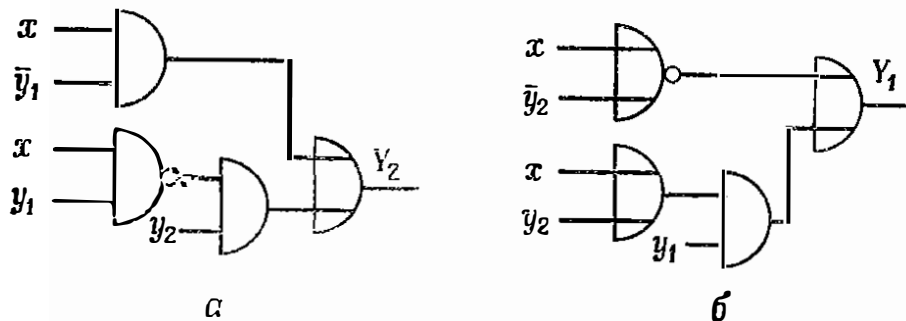


Рис. 4.52. Реализация цепей без элементов задержки, использующая специальные схемы.

формы, показанной на рис. 4.50. Это упрощение стало возможным потому, что как Z_1 , так и Z_2 содержит только один терм и нет необходимости в явном дополнении входных переменных.

4.9. ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ С НЕОГРАНИЧЕННЫМИ ЗАДЕРЖКАМИ

Различные модели асинхронных цепей были предложены Мюллером и Бартки [17]. Предполагается, что в ветвях цепи отсутствуют паразитные задержки, а границы значений паразитных задержек, связанных с логическими элементами, неизвестны, хотя и считается, что они положительны и конечны. Эти допущения оказываются полезными при исследовании поведения цепей, в которых задержки логических элементов могут изменяться в неограниченно широком диапазоне. Цепи, поведение которых не зависит от значений паразитных задержек, называются *независимыми от скорости* цепями.

При допущении об отсутствии ограничений на время задержки, вызываемой логическими элементами, невозможно определить требуемый при основном режиме функционирования интервал между последующими изменениями входных переменных. Следовательно, необходимо, чтобы сама цепь генерировала сигнал для указания своей готовности к получению следующего входного сигнала. Было предложено несколько способов проектирования цепей, которые генерируют такие *сигналы готовности*

(завершения), для того чтобы функционировать в условиях указанной неопределенности задержки, вносимой логическими элементами [2, 8, 16]. Ниже в общих чертах будет изложен один из этих методов [2].

Цели с сигналами завершения

На рис. 4.53 показана общая модель цепей с сигналами завершения. Подсхема, названная логической цепью, может реализовать комбинационную или последовательностную функцию. Детектор завершения генерирует сигнал завершения, когда выходы логической цепи достигли устойчивого состояния. Входы и выходы кодируются таким образом, чтобы устойчивые состояния могли быть определены детектором завершения. Получая

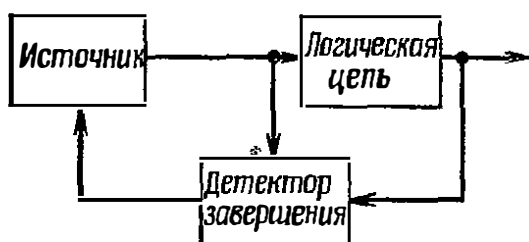


Рис. 4.53. Цепь с сигналами завершения

сигнал завершения, источник генерирует на входе следующий код, поступающий в логическую цепь. Необходимость связи между входом логической цепи и детектором завершения будет объяснена ниже.

Одним из кодов, которые могут быть использованы в цепях, генерирующих сигналы завершения, является *автосинхронный код*. В этом коде каждая двоичная входная или выходная переменная представляется двумя переменными, которые должны указывать друг другу о завершении. А именно $x_1 = 0$ представляется как $x_{11} = 0, x_{10} = 1$, а $x_1 = 1$ представляется как $x_{11} = 1, x_{10} = 0$. Переходы между двумя кодовыми словами всегда совершаются через *промежуточное нулевое состояние*, в котором все переменные состояния обращаются в 0. Например, две переменные x_1, x_2 , которые изменяются от 01 до 10, должны быть представлены четырьмя переменными $x_{11}, x_{10}, x_{21}, x_{20}$, подвергающимися переходам $0110 \rightarrow 0000 \rightarrow 1001$.

Логическая цепь должна быть спроектирована таким образом, чтобы она выдавала выходное кодовое слово только тогда, когда получается входное кодовое слово. Выход из промежуточного нулевого состояния может произойти одновременно либо по мере изменения переменных состояний. Переход в промежуточное нулевое состояние определяется по наличию всех нулей на входах и выходах. Цикл работы цепи должен быть следующим: источник посылает кодовое слово в цепь, которая генерирует соответствующее выходное кодовое слово. Проверая этот выход, детектор завершения генерирует сигнал завершения, который заставляет источник генерировать

нулевое состояние. Цепь реагирует на вход в промежуточное нулевое состояние, вырабатывая сигнал выхода этого состояния. После этого детектор завершения генерирует отличный от других сигнал завершения, по получении которого источник выдает следующее кодовое слово.

Пример 4.25. Допустим, что реализуемая функция должна быть $y = x_1 \oplus x_2$. Кодирова вход и выход с помощью автосин-

x_{11}	x_{10}	x_{21}	x_{20}	y_1	y_0
0	1	0	1	0	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	0	1

Рис. 4.54. Таблица истинности для функции $y = x_1 \oplus x_2$, закодированной в автосинхронном коде.

хронного кода, получаем частичную таблицу истинности, представленную на рис. 4.54. Так как x_{i1} и x_{i0} не могут быть оба одновременно равны 1 при $i = 1, 2$, то можно обрабатывать такие входные комбинации без особых забот. Другие комбинации могут быть пройдены в процессе переходов, и как y_1 , так

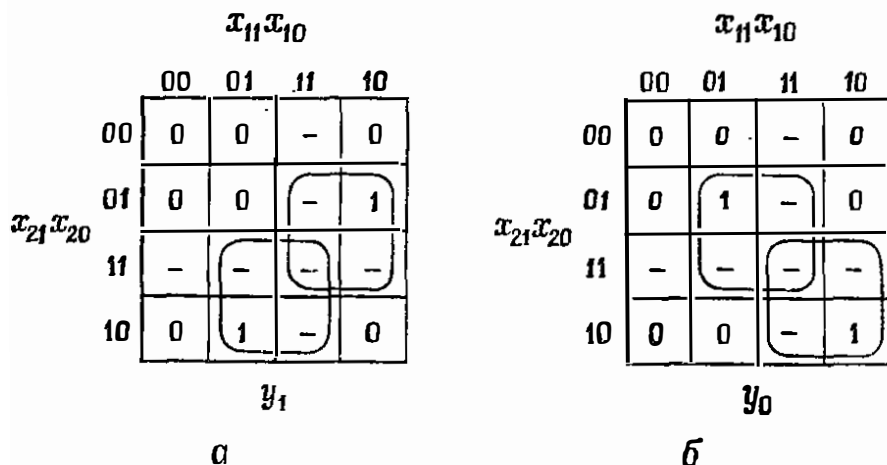


Рис. 4.55. Карты Карнау для автосинхронных функций.

и y_0 присваивается значение 0, как показано на картах Карнау, приведенных на рис. 4.55. Таким образом, цепь будет выдавать на выходе кодовое слово только после того, как входы завершат свой переход к кодовому слову. Две выходные функции могут быть реализованы в следующей двухуровневой форме суммы произведения:

$$y_1 = x_{11}x_{20} + x_{10}x_{21},$$

$$y_0 = x_{11}x_{21} + x_{10}x_{20}.$$

Реализация этих функций и детектора завершения показана на рис. 4.56. Детектор завершения имеет два выхода, помеченных S и C. Значение $S = 1$ указывает, что было установлено

нулевое состояние, что в свою очередь послужило причиной выдачи источником следующего кодового слова. Аналогичным образом, значение $C = 1$ указывает, что было определено кодовое слово, что заставляет источник генерировать сигнал перехода в нулевое состояние. Заметим, что детектор, сигнализирующий о переходе в нулевое состояние, присоединяется к выходам логической цепи, так же как к ее входам, что обсуждалось ранее. Если детектор присоединяется только к выходам

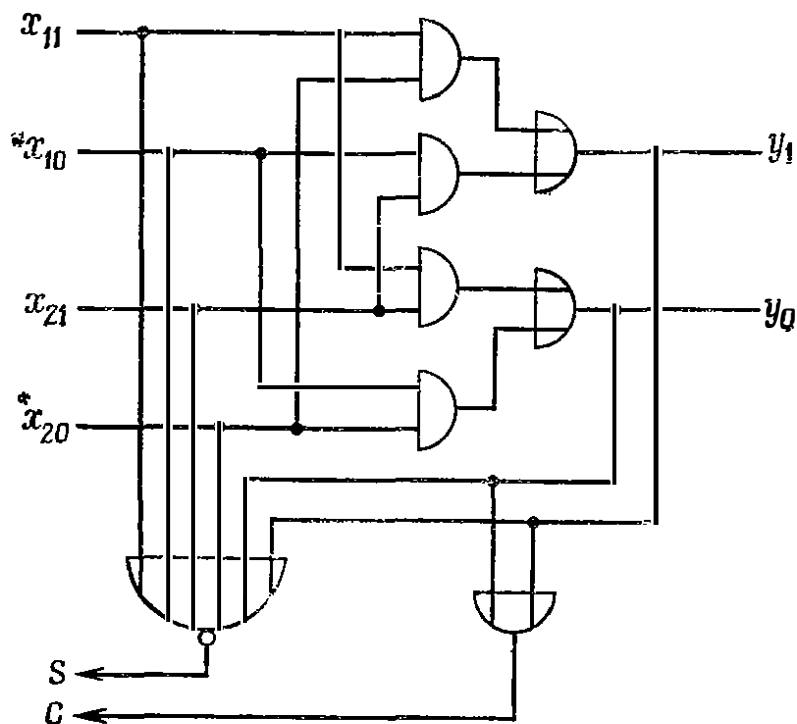


Рис. 4.56. Реализация автосинхронных функций и детектора завершения.

цепи, то может быть выработан преждевременный сигнал завершения, который, в свою очередь может послужить причиной того, что источник изменит свое состояние до того, как цепь завершит свою реакцию на предшествующий переход.

Если с помощью этого метода реализуется функция с многими выходами, то детектор выходного кодового слова необходимо модифицировать, связывая каждую пару выходов с логическим элементом ИЛИ и соединяя все выходы элементов ИЛИ со схемой И. Вместо автосинхронного кода для кодирования входов и выходов могут быть использованы коды m/n (m из n), где m переменных из полного числа n переменных являются 1 для любого кодового слова.

Этот метод может быть также модифицирован для реализации последовательных цепей при допущениях об отсутствии определенных границ для величины времени запаздывания логических элементов. Как и в случае комбинационных цепей, предполагается, что требуются входы и выходы, отличающиеся для кодовых слов (автосинхронных или m/n) и для промежу-

точных нулевых состояний. Если последовательная цепь идентифицируется с помощью таблицы переходов, то мы дополняем эту таблицу введением столбца промежуточного нулевого состояния (как и в случае реализации цепей с многократным изменением входов, который обсуждался ранее). Столбец промежуточного нулевого состояния содержит только

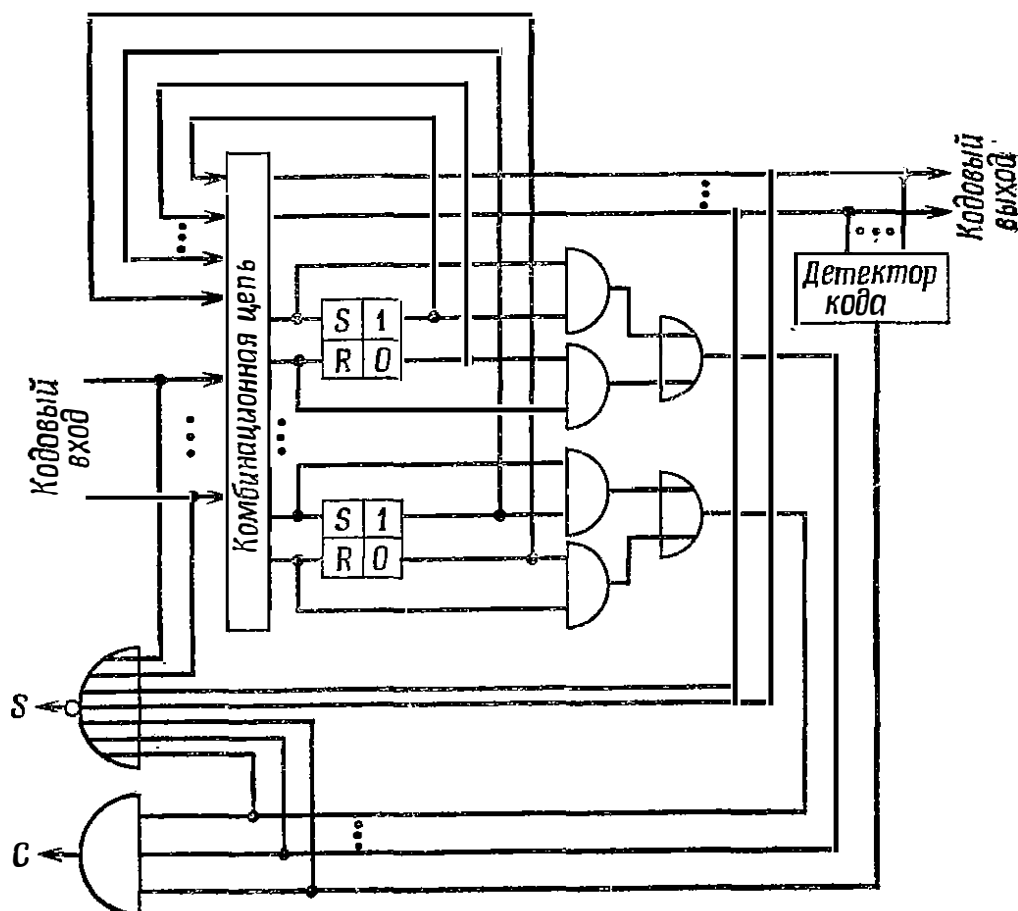


Рис 4.57. Реализация последовательной цепи, использующая сигнал завершения.

устойчивые состояния, но теперь определение выходов в столбце промежуточного нулевого состояния производится таким образом, чтобы выход из него не зависел от состояний. Все переходы состояний происходят тогда, когда состояния входов изменяются от положений, соответствующих промежуточному нулевому состоянию, к положению, соответствующему надлежащему кодовому слову.

На рис. 4.57 показан этот метод реализации последовательной функции. Для каждой переменной состояния используется триггер. Когда в цепи имеет место переход в промежуточное нулевое состояние, все триггеры функций возбуждения находятся в нулевом состоянии. Если входом является кодовое слово, все триггеры цепи, реализующей функцию возбуждения, находятся в состоянии 0 независимо от того, находились ли они уже в желаемых состояниях или нет. В этой схеме каждый триггер должен быть в таком состоянии, в котором он возбуж-

дается, когда цепь завершает реакцию на переход от промежуточного нулевого состояния к кодовому слову. В дополнение к этому выходы цепи должны быть кодовыми словами. Реакция на переход от кодового слова к промежуточному нулевому состоянию завершается, когда входы, выходы и состояние триггеров цепи возбуждения все являются нулями.

Кодирование состояний для этого типа реализации должно быть ОВП-кодированием, один из вариантов которого был предложен Лю. Этот вид кодирования обладает тем свойством, что все неустойчивые состояния приводят к одному и тому же устойчивому состоянию в любом столбце и само устойчивое состояние выделяется из всех других состояний в этом столбце за счет подмножества переменных состояния. В результате триггерные функции возбуждения могут быть реализованы в виде суммы произведений таким образом, что не более чем одна импликанта функции изменяется к 1 в течение любого перехода, предотвращая тем самым преждевременную генерацию сигналов завершения. Если две или более импликанты функции переходят в 1 в течение перехода, то вырабатывается сигнал завершения после того, как первая из них перейдет в 1, и прежде чем цепь завершит реакцию на входной переход. Это может привести к тому, что источник изменит свои выходы прежде, чем цепь придет в устойчивое состояние, что в свою очередь приведет к неправильному ее функционированию.

При проектировании сложных схем удобно строить их путем соединения модулей, реализующих простейшие функции. Такие соединения модулей допустимы между модулями, генерирующими сигналы завершения. Если выходы модуля A соединяются с модулем B , то сигналы завершения, генерируемые модулем B , возвращаются по цепи обратной связи к модулю A . Модули должны быть спроектированы таким образом, чтобы выходы модуля A не изменялись до тех пор, пока модуль B не завершит реакцию на предшествующий выход A . Это может быть достигнуто путем соединения элементов памяти (таких, как триггеры) с выходами. Обсуждавшийся выше основной подход к проектированию комбинационных и последовательностных цепей может быть модифицирован для того, чтобы обеспечить запоминание на выходе в пределах каждого модуля. Более сложные схемы могут быть реализованы за счет соединений спроектированных таким способом модулей как для комбинационных, так и для последовательностных цепей.

БИБЛИОГРАФИЯ И КОММЕНТАРИИ

Модель асинхронных последовательностных автоматов и цепей и использование таблиц переходов предложены Хаффманом [10], который также получил универсальное $2S_0 - 1$ коди-

рование состояний [11]. Условия основного режима функционирования были формально определены Мак-Класки [15]. Проблема сокращения таблицы состояний для синхронных автоматов рассматривалась Паулем и Ангером [18], а для асинхронных автоматов Ангером [23]. Универсальное кодирование с помощью множества разделяемых строк для 8 и 12 состояний было предложено Саусером [19]. Кашеф и Макхи [12] разработали относительный класс кодирования, называемый кодами с дополнительной проверкой на четность. Универсальное ОВП-кодирование состояний предложено Хэммингом [9]. Лю [13] и Трэйси [21] разработали процедуры для получения однозначного ОВП-кодирования, а Фридман, Грэхэм и Ульман [6] указали на границы универсальных ОВП-кодирований, а также рассмотрели многозначные ОВП-кодирования. Влияние риска сбоя, их предотвращение и определение были рассмотрены Ангером [22] и Айшельбергером [4]. Девятизначная алгебра для определения динамических рисков сбоя была разработана Фантоци [5]. Аналогичная, но менее формализованная процедура была предложена Бреуром и Гаррисоном [3].

Реализация цепей с одновременным изменением входных цепей была выполнена Фридманом и Меноном [7], а учет неограниченных изменений входных сигналов был выполнен Ангером [24]. Армстронг, Фридман и Менон [2] рассматривали модель, в которой задержки линии были меньше, чем задержки логических элементов, а также модель линии с неограниченной задержкой. Модель задержки, не имеющей ограничений, была разработана Мюллером и Бартки [17]. Методы конструирования были разработаны Миллером [16] и Хэммелом [8]. Использование автосинхронных кодов при проектировании асинхронных цепей было впервые предложено Симсом и Греем [20].

ЛИТЕРАТУРА

1. Armstrong D. B., Friedman A. D., Menon P. R., Realization of Asynchronous Sequential Circuits Without Inserted Delay Elements, *IEEE Trans. Computer*, vol. C-17, pp. 129—134, February 1968.
2. Armstrong D. B., Friedman A. D., Menon P. R., Design of Asynchronous Circuits Assuming Unbounded Gate Delays, *IEEE Trans. Computers*, vol. C-18, pp. 1110—1120, December 1969.
3. Breuer M. A., Harrison R. L., Procedures for Eliminating Static and Dynamic Hazards in Test Generation, *IEEE Trans. Computers*, vol. C-23, pp. 1069—1078, October 1974.
4. Eichelberger E. B., Hazard Detection in Combinational and Sequential Switching Circuits, Proc. Fifth Annual Symposium on Switching Circuit Theory and Logical Design, pp. 111—120, 1964, Also IBM J. of Res. & Dev., pp. 90—99, 1965.
5. Fantauzzi G., An Algebraic Model for the Analysis of Circuits, *IEEE Trans. Computers*, vol. C-23, pp. 576—581, June 1974.

6. Friedman A. D., Graham R. L., Ulman J. D., Universal Single Transition Time Asynchronous State Assignments, *IEEE Trans. Computers*, vol. C-18, pp. 541—547, June 1969.
7. Friedman A. D., Menon P. R., Synthesis of Asynchronous Sequential Circuits with Multiple-Input Changes, *IEEE Trans. Computers*, vol. C-17, pp. 559—566, June 1968.
8. Hummel D., Ideas on Asynchronous Feedback Networks, Proc. Fifth Annual Symposium on Switching Circuit Theory and Logical Design, pp. 4—11, 1964.
9. Hamming R. W., Error Detecting and Correcting Codes, *BSTJ*, vol. 29, pp. 147—160, April 1950.
10. Huffman D. A., The Synthesis of Sequential Switching Circuits, *J. Franklin Inst.*, vol. 257, pp. 161—190, March 1954, pp. 275—303, April 1954.
11. Huffman D. A., A Study of Memory Requirements of Sequential Switching Circuits, Technical Report № 293, Research Laboratory of Electronics, MIT, Cambridge, Mass., 1955.
12. Kashef R. S., McGhee R. B., Augmented Parity Check Codes for Encoding of Asynchronous Sequential Machines, *IEEE Trans. Computers*, vol. C-22, pp. 891—896, October 1973.
13. Liu C. N., A State Variable Assignment Procedure for Asynchronous Sequential Switching Circuits, *J. ACM*, vol. 10, pp. 209—215, April 1963.
14. Magó G., Realization Methods for Asynchronous Sequential Circuits, *IEEE Trans. Computers*, vol. C-20, pp. 290—297, March 1971.
15. McCluskey E. J., Fundamental Mode and Pulse Mode Sequential Circuits, Proc. IFIP Congress, Munich, Germany, North Amsterdam Publishing Company, Amsterdam, pp. 725—730, 1962.
16. Miller R. E., *Switching Theory*, vol. 2, Wiley, New York, 1965.
17. Muller D. E., Bartky W. S., A Theory of Asynchronous Circuits, Proc., of an International Symp. on the Theory of Switching, vol. 29, Annals of the Computation Laboratory of Harvard Univ., Harvard Univ. Press, pp. 204—243, 1959.
18. Paull M. C., Unger S. H., Minimizing the Number of States in Incompletely Specified Sequential Switching Functions, *IRE Trans. Elec. Computers*, vol. EC-8, pp. 356—367, September 1959.
19. Saucier G., Encodings for Asynchronous Sequential Networks, *IEEE Trans. Elec. Computers*, vol. EC-16, pp. 365—369, June 1967.
20. Sims J. C., Gray H. J., Design Criteria for Autosynchronous Circuits, *Proc. Eastern Joint Comp. Conf.*, pp. 94—99, December 1958.
21. Tracey J. H., Internal State Assignments for Asynchronous Sequential Machines, *IEEE Trans. Elec. Computers*, vol. EC-15, pp. 551—560, August 1966.
22. Unger S. H., Hazards and Delays in Asynchronous Sequential Switching Circuits, *IRE Trans. Circuit Theory*, vol. CT-6, pp. 12—25, March 1959.
23. Unger S. H., *Asynchronous Sequential Circuits*, Wiley Interscience, New York, 1969.
24. Unger S. H., Asynchronous Sequential Switching Circuits with Unrestricted Input Changes, *IEEE Trans. Computers*, vol. C-20, pp. 1437—1444, December 1971.

УПРАЖНЕНИЯ

4.1. Рассмотрите таблицу переходов, представленную на рис. 4.58. Содержимое $N(2,01)$ обозначено как A . Охарактеризуйте таблицу как таблицу с однократным изменением выхода, с многократным изменением выхода, нормального режима, основного режима и т. д., если а) $A = 1$, б) $A = 2$, в) $A = 3$, г) $A = 4$.