

УДК 004.312.02

Е.Н. Коробкова

Белгородский государственный технологический университет имени В.Г. Шухова, Россия

ПРИЛОЖЕНИЕ МЕТОДА ДЕКОМПОЗИЦИИ К СИНТЕЗУ ФОРМИРОВАТЕЛЕЙ ДВУХФАЗНОЙ ПЕРИОДИЧЕСКОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ИМПУЛЬСОВ ТИПА МЕАНДР

Предлагается метод синтеза формирователей двухфазной периодической последовательности импульсов типа меандр с перестраиваемыми временными параметрами, основанный на представлении проектируемого автомата в виде функционально-блочной композиции нескольких взаимосвязанных отдельных блоков, каждый из которых определяет один из временных параметров формируемой импульсной последовательности. Начальное состояние входящих в композицию блоков трактуется как состояние нуля. Любое другое состояние каждого из входящих в композицию блоков трактуется как состояние единицы, что позволяет трактовать входящие в композицию блоки как простейшие элементы памяти, имеющие два состояния.

Ключевые слова: цифровые автоматы, декомпозиция, логические функции, синтез, минимизация.

Введение

Постановка проблемы. В связи с бурным развитием программируемых интегральных схем и внедрением их в практику проектирования цифровых устройств [1] вновь возрос интерес к логическому синтезу, причём, не только на уровне автоматизированного проектирования, но и на более низком - «ручного синтеза». Работа посвящена синтезу формирователей двухфазной периодической последовательности импульсов типа меандр с перестраиваемыми временными параметрами.

Анализ исследований и публикации, посвящённых проблеме проектирования блоков и узлов, многие из которых вошли в библиотеку типовых, посвящено большое число работ, простое перечисление которых представляет собой далеко не тривиальную задачу [2]. Известные методы проектирования позволяют по заданному алгоритму построить любой конечный автомат. Построение конечного автомата с жёсткой логикой, не вызывает никаких проблем для случая фиксированных параметров выходных сигналов. При проектировании автомата с перестраиваемыми (программируемыми) параметрами, появляются проблемы, связанные даже не с синтезом схемы автомата как такового, а с нахождением оптимального варианта схемы с минимальным числом межэлементных связей и минимальной сложностью представления функций, определяющих настройку автомата на заданные параметры.

В работах [3, 4] предложен метод синтеза цифровых автоматов, основанный на представлении логических функций в обобщённой форме, позволивший упростить процедуру анализа и синтеза цифровых автоматов.

Цель статьи – продолжить исследования, направленные на разработку нетрадиционных методов

анализа и синтеза цифровых автоматов с перестраиваемыми параметрами, и их приложение к синтезу автоматов, в частности формирователей двухфазной периодической последовательности импульсов типа меандр с перестраиваемыми параметрами.

Метод решения

В [5] предложен метод, основанный на представлении проектируемого автомата в виде функционально-блочной композиции нескольких взаимосвязанных отдельных блоков, каждый из которых, не зависимо от числа разрядов позволяет трактовать входящие в композицию блоки как простейшие элементы памяти, имеющие два состояния: состояние нуля (если значение во всех разрядах блока равно нулю) и состояние единицы (если хотя бы в одном разряде блока значение равно единице), а проектируемый автомат в целом – как устройство, состоящее из двух (или более) таких элементов. Число блоков равно числу временных параметров формируемой выходной последовательности. Каждый блок представляет собой конечный цифровой циклический автомат.

В качестве типовых блоков могут быть использованы: двоичные, двоично-десятичные суммирующие, вычитающие или реверсивные счётчики, пересчётные схемы с любым коэффициентом пересчёта и с любым кодированием состояний; циклические автоматы, выполненные на сдвигающих регистрах (счётчики с унитарным кодированием, счётчики Джонсона, линейные счётчики), типовые синхронные или асинхронные триггеры.

При воздействии тактового импульса один или несколько блоков могут оставаться в прежнем состоянии (режим хранения), могут перейти в следующее состояние в соответствии с алгоритмом функционирования данного блока (режим последовательного перехода), могут изменить направление

последовательного перехода (реверс), могут перейти в любое состояние из его множества, задаваемое внешним управляющим словом (режим параллельной загрузки), могут перейти в начальное (нулевое) состояние (режим установки нуля). Для обеспечения возможности этих переходов каждый из блоков должен иметь соответствующие входы, формирование активного сигнала (равного 0 или 1 в зависимости от выбранной схемы) на одном из которых обеспечивает выбранный режим.

Проектирование заданного конечного автомата, как и в случае классических методов, начинается с его словесного описания с последующим составлением таблицы переходов. При классических методах проектирования полное число переменных определяется суммой всех разрядов автомата в целом и числа настроечных переменных, следовательно, число наборов таблицы (с учётом неиспользуемых) равно 2^{k+v} , где k – число элементов памяти (триггеров, разрядов) проектируемого автомата в целом, v – число настроечных переменных. В предлагаемом методе полное число наборов таблицы (равное 2^n , n – число блоков) зависит только лишь от числа разрядов используемых блоков и от числа настроечных переменных независимо от числа разрядов используемых блоков и от числа настроечных переменных, что существенно уменьшает общее число наборов таблицы переходов, по сравнению с классическим её представлением.

Граф переходов автомата представляют в виде замкнутых колец, равных числу входящих в композицию блоков, с общей вершиной, соответствующей начальному (нулевому) состоянию каждого блока.

При таком представлении каждое последующее состояние S^{r+1} отдельного блока в $(r+1)$ -м такте зависит не от конкретного его состояния S^r в рассматриваемом такте, а от состояния всех блоков. Состояние блока характеризуется значением сигнала F_i на его выходе (в частности, на выходе переноса в случае использования вычитающего счётчика), а состояние автомата в целом в $(r+1)$ -м такте определяется значениями сигналов на выходах всех блоков в r -м такте.

После выбора необходимого числа и типа блоков, определяемых в соответствии с описательным алгоритмом (приведенными эпюрами, графом), составляется таблица переходов и совмещённая с ней таблицы режимов и функций возбуждения (значений управляющих сигналов, обеспечивающих переход из рассматриваемого состояния в следующее) в соответствии с приведенным графом. Таблица переходов содержит $2n$ столбцов (n – число блоков). В первых n столбцах перечислены все комбинации значений сигналов (переменных) на выходах блоков (даже если некоторые комбинации в процессе функционирования автомата не используются) в r -м такте и n столбцов, в которых приведены значений сигналов (переменных) на выходах блоков в $(r+1)$ -м такте.

Таблица режимов настройки содержит n столбцов, в каждой строке которых комментируется режим настройки блока на данную комбинацию значений сигналов в r -м такте. В соответствии таблицей режимов настройки заполняется таблица функций возбуждения (управления), которая содержит число столбцов, равное числу требуемых управляющих сигналов формируемых на входах из множества $L, D_0 - D_{n-1}, R, U$: L – вход разрешения параллельной загрузки; $(D_0 - D_{n-1})$ – входы подачи загружаемых данных; R – вход разрешения режима счёта (последовательного перехода); R – вход установки в нулевое состояние; U – вход управления направлением перехода (реверса). Число используемых (необходимых) управляющих входов в каждом из блоков зависит от конкретного алгоритма функционирования автомата в целом и способа разбиения его на блоки.

Таблица заполняется в соответствии с приведенным комментарием реакции блоков (режимов настройки) на каждую из комбинаций, при этом следует иметь в виду, что каждому управляющему входу присвоен приоритет, поэтому, если в некоторой строке таблицы для какого-то блока проставлено активное значение управляющего сигнала (значение функции возбуждения) на входе, имеющего более высокий приоритет, то для сигналов на остальных входах этого блока, имеющих меньший приоритет, можно проставлять значение 0 или 1 (знак избыточности). Полученная таким образом таблица позволяет найти необходимые функции возбуждения (управления), определяющие сигналы, подаваемые на управляющие входы каждого блока. При нахождении функций возбуждения неиспользуемые и избыточные комбинации доопределяются значением 0 или 1, исходя из требования минимизации структуры автомата или устранения риска.

Выходной сигнал чаще всего снимается с выхода одного из блоков, или является некоторой функцией сигналов с выходов нескольких блоков. Значение выходного сигнала равно нулю, если во всех разрядах блока значения равны нулю. Если хотя бы в одном разряде блока значение равно единице, значение выходного сигнала равно единице.

Проектируемое устройство обеспечивает формирование двухфазной периодической последовательности импульсов типа меандр с перестраиваемой длительностью при воздействии тактовых импульсов, подаваемых на его вход с выхода внешнего тактового генератора.

Длительность импульсов, кратная периоду тактовых импульсов, определяется значением двоичного слова $V = b_{n-1}b_{n-2} \times \dots \times b_1b_0$, подаваемого на входы настройки. При этом, следует отметить, что спроектированное устройство будет работать надлежащим образом в том случае, если его конструкция гарантирует, что при включении источника питания все блоки его

войдут в начальное (нулевое) состояние. Для этого необходимо в каждом блоке предусмотреть наличие входа установки, на котором при включении питания аналоговой схемой генерируется активное значение сигнала R, подаваемого на входы асинхронной установки 0 всех блоков, а затем происходит самоблокировка этого состояния, т.е. устройство будет оставаться в нулевом состоянии пока на его вход запуска (Start) не поступит активное значение сигнала.

Подача активного значения сигнала на вход Start снимает блокировку нулевого состояния, подготавливая все блоки автомата к переходу в состояние готовности к формированию заданной импульсной последовательности, начинающегося с поступлением первого тактового импульса после снятия блокировки. Кроме того, необходимо также предусмотреть наличие входа остановки (Stop).

Подача активного значения сигнала на вход Stop обеспечивает подготовку всех блоков к переходу в начальное состояние, но собственно переход происходит только по окончании формирования на выходе второй фазы, предотвращая тем самым искажение выходной последовательности независимо от момента поступления сигнала Stop. Функции блокировки и неискажённого формирования заданной последовательности возлагаются на четвёртый блок (старт-стопное устройство), формирующий активный сигнал R на входах асинхронной установки 0 всех блоков по окончании формирования последнего импульса заданной импульсной последовательности.

Временные диаграммы, иллюстрирующие алгоритм функционирования формирователя, для варианта настройки $t_{и} = 5T$ приведены на рис. 1.

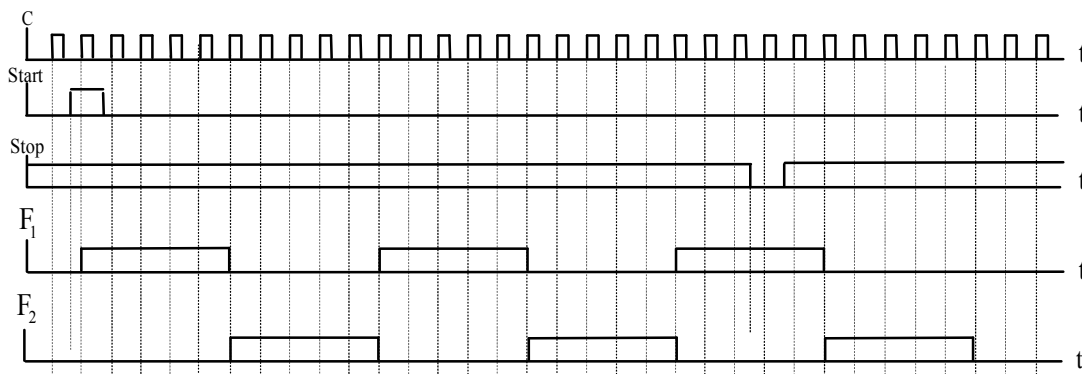


Рис. 1. Временные диаграммы, иллюстрирующие алгоритм функционирования.

Анализируя приведенные эпюры, можно заключить, что проектируемый автомат характеризуется тремя параметрами: длительность импульсов, временной интервал формирования импульсов первой фазы, временной интервал формирования импульсов второй фазы, следовательно, функции, определяющие эти параметры, можно возложить на три блока: блок, определяющий длительность импульсов и два блока, определяющие временные интервалы формирования первой и второй фазы. Оптимальным вариантом первого блока будет типовой вычитающий (реверсивный) двоичный счётчик со входом разрешения синхронной параллельной загрузки C, входами подачи загружаемых данных $D_0 - D_{n-1}$, входом разрешения счёта и входом асинхронной установки нуля. Число разрядов счётчиков определяет диапазон перестройки длительности импульсов. Функции, определяющие формирование импульсов первой и второй фазы можно возложить на два типовых синхронных D – триггера со входом асинхронной установки нуля, на выходе одного из которых формируются импульсы первой фазы, а на выходе второго – второй.

Старт-стопное устройство также может быть выполнено на типовом синхронном D – триггере, одно состояние которого обеспечивает блокировку

нулевого состояния блоков, а второе – снятие блокировки, разрешение режима формирования выходной последовательности.

Для более детального понимания алгоритма функционирования формирователя (после включения источника питания и после подачи стартового импульса) с заданными параметрами представим его в виде графа (рис. 2).

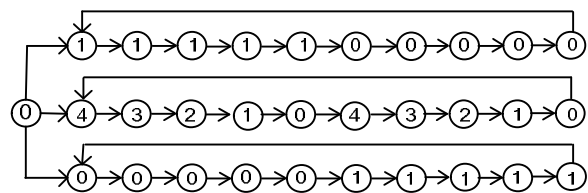


Рис. 2. Граф

Граф состоит из трёх колец с общей вершиной, соответствующей начальному (нулевому) состоянию всех (трёх) блоков: верхнее кольцо – граф переходов первого D-триггера, формирующего импульсы первой фазы; среднее кольцо – граф переходов счётчика, определяющего длительность формируемых импульсов; нижнее кольцо – граф переходов второго D-триггера, формирующего импульсы второй фазы. При изображении графа вершины его

представлены в виде кружков, внутри которых записывается состояния (содержимое) блоков.

После включения источника питания в течение некоторого промежутка времени, определяемого быстродействием элементной базы, происходит формирование активного сигнала на асинхронных входах установки нуля всех блоков, обеспечивая установку и самоблокировку нулевого состояния счётчика, первого и второго триггера, формируя нулевое значение сигналов на их выходах, т.е. на прямых выходах триггеров и на инверсном выходе переполнения (переноса) вычитающего счетчика.

Поскольку режим асинхронной установки счётчика и триггеров в нулевое состояние имеет приоритет по отношению ко всем остальным режимам, то до тех пор, пока на их входах асинхронной установки нуля будет сохраняться активный уровень, никаких изменений в состояниях счётчика и триггеров при поступлении тактовых импульсов не будет, т.е. значение сигналов на их выходах будет оставаться равным нулю.

Значения этих сигналов (выходных переменных -000) обеспечивают настройку счётчика на режим синхронной параллельной загрузки, второго триггера – на режим перехода в единичное состояние, первого триггера – на режим хранения нуля (перехода в нулевое состояние).

При поступлении импульса запуска происходит разблокировка режима асинхронной установки нуля и тогда при поступлении первого (после окончания переходного процесса, связанного с запуском) тактирующего импульса (С) по его фронту происходит параллельная загрузка счётчика значениями настроечных переменных, подаваемых на входы $D_0 - D_{n-1}$, второй триггер переходит в единичное состояние, нулевое состояние первого триггера остаётся неизменным. В результате этого перехода происходит настройка счётчика на режим вычитания, режимы первого и второго триггеров останутся неизменными. С приходом каждого последующего тактового импульса содержимое счётчика будет уменьшаться, состояние первого и второго триггеров оставаться неизменным до тех пор пока содержимое счётчика не станет равным 0. В результате этого перехода происходит настройка счётчика на режим синхронной параллельной загрузки, второго триггера – на режим перехода в нулевое состояние, первого триггера – на режим перехода в единичное состояние, и тогда при поступлении следующего тактового импульса происходит параллельная загрузка счётчика, второй триггер переходит в нулевое состояние, первый триггер переходит в единичное состояние. В результате этого перехода снова происходит настройка счётчика на режим вычитания, режимы первого и второго триггеров останутся неизменными. С приходом каждого последующего тактового импульса содержимое счёт-

чика будет уменьшаться, состояние первого и второго триггеров оставаться неизменным до тех пор пока содержимое счётчика не станет равным 0.

В результате этого перехода снова происходит настройка счётчика на режим синхронной параллельной загрузки, второго триггера – на режим перехода в единичное состояние, первого триггера – на режим перехода в нулевое состояние. При поступлении следующих импульсов рассмотренные выше процессы повторяются.

Результаты этих рассуждений представляем в виде таблицы переходов (первые шесть столбцов), совмещённой с ней таблицы режимов настройки (следующие три столбца) и таблицы функций возбуждения (последние четыре столбца). В первых трёх столбцах представлены все наборы выходных переменных размещённых в порядке возрастания номеров наборов, определяемых значениями сигналов на выходах каждого блока Q_1, \bar{P}_4, Q_2 (Q_1 – сигнал на выходе первого триггера; \bar{P}_4 – сигнал на выходе переноса счётчика; Q_2 – сигнал на выходе второго триггера).

В каждой строке следующих трёх столбцов представлены состояния каждого блока после перехода. При заполнении таблицы, режимы и соответствующие им значения функций возбуждения для не имеющих места состояний (комбинаций наборов), отмечаем знаком #.

Для нахождения минимальных форм аналитического представления функций возбуждения и последующей реализации их с использованием типовых элементов и соединений воспользуемся методом, основанным на представлении их в картах Карно с соседним кодированием, при этом, значения функций возбуждения (управления) для не имеющих места состояний (комбинаций наборов) и на наборах, на которых значение функций безразлично, также будем отмечать знаком # с последующим доопределением их значением нуля или единицы, исходя из условия получения минимального представления функций в выбранном логическом базисе (рис. 3).

Карта значений функции D_1 первого триггера приведена на рис. 4, второго – на рис. 5.

Доопределяя функцию D_1 на втором и пятом наборах значением 0, а на седьмом наборе значением 1, записываем минимальную форму функции D_1 первого триггера:

$$D_1 = \bar{P}_4 + Q_1.$$

Доопределяя функцию D_2 на втором и пятом наборах значением 1, а на седьмом наборе значением 0, записываем минимальную форму функции D_2 второго триггера:

$$D_2 = \bar{P}_4 + Q_1.$$

Q ₁	\bar{P}_4	Q ₂	Q ₁	\bar{P}_4	Q ₂	Тр 1	СТ	Тр 2	D ₁	\bar{L}	\bar{P}	D ₂
0	0	0	1	1	0	3-1	Загр	3-0	1	0	#	0
0	0	1	1	1	0	3-1	Загр	3-0	1	0	#	0
0	1	0	#	#	#	#	#	#	#	#	#	#
0	1	1	0	0	1	3-0	–	3-1	0	1	0	1
1	0	0	0	1	1	3-0	Загр	3-1	0	0	#	1
1	0	1	#	#	#	#	#	#	#	#	#	#
1	1	0	1	0	0	3-1	–	3-0	1	1	0	0
1	1	1	#	#	#	#	#	#	#	#	#	#

При заполнении таблицы режимов приняты сокращения: «Заг» – режим загрузки; «–» – режим вычитания; «3-0» - запись 0; «3-1» - запись 1; «#» – нет перехода или значение безразлично.

Рис. 3. Выбранный логический базис

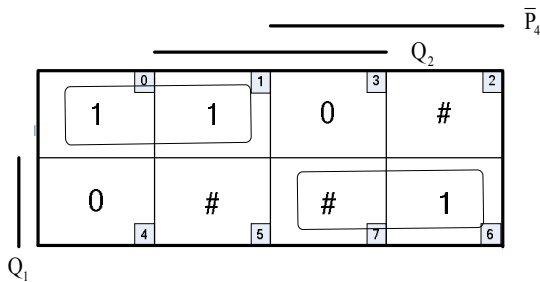


Рис. 4. Карта значений функции D₁ 1-го триггера

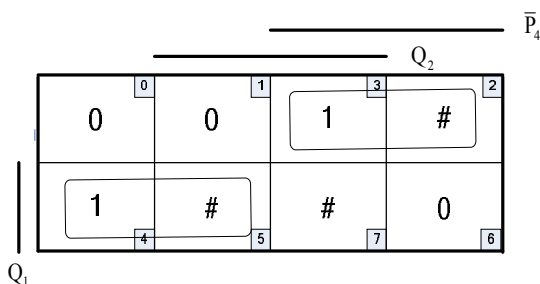


Рис. 5. Карта значений функции D₁ 2-го триггера

Карта значений функции \bar{L} счётчика приведена на рис. 6.

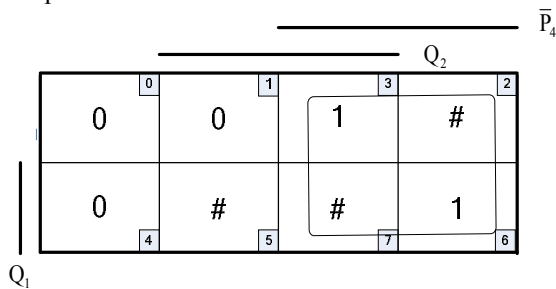


Рис. 6. Карта значений функции \bar{L} счётчика

Доопределяя функцию на втором и седьмом пятом наборах значением 1, а на пятом наборе значением 0, записываем минимальную форму функции

$$\bar{L} = \bar{P}_4.$$

Непосредственно из таблицы видно, что значение функции \bar{P} равно 0, на третьем и шестом наборах, на втором, пятом и седьмом не определено, на нулевом, первом и четвёртом безразлично, откуда следует, что её можно доопределить на этих наборах значение 0 и тогда $\bar{P} = 0$.

Как уже было отмечено выше, схема, построенная в соответствии с полученными выражениями, требует наличия цепи обнаружения включения источника питания, обеспечивающей переход в начальное (нулевое) состояние, а также наличия входа запуска и обеспечение возврата в начальное состояние по окончании формирования заданной серии. Эти функции выполняет старт-стопное устройство, предложенное в [6].

Схема формирователя, построенная на типовых блоках (реверсивный счётчик и два синхронных D-триггера со входами асинхронной установки нуля) в соответствии с полученными выше представлениями функций управления D₁, \bar{L} , \bar{P} , D₂ и приведенным в [6] описанием старт-стопного устройства, также выполненного на типовом синхронном D-триггере со входом асинхронной установки нуля, показана на рис. 7.

Компьютерное моделирование схемы в САПР Quartus – 2 фирмы Altera подтвердило достоверность полученных результатов проведенного синтеза.

Заключение

Схема формирователя двухфазной периодической последовательности импульсов с перестраиваемыми параметрами получена в результате проведенного синтеза, основанного на представлении проектируемого автомата в виде функционально-блочной композиции трёх основных взаимосвязанных блоков, начальное состояние которых принимается как состояние логического 0, а любое другое состояние каждого из входящих в композицию блоков принимается как состояние логической единицы, трактуя входящие в композицию блоки как простейшие элементы памяти, имеющие два состояния: состояние нуля и состояние единицы, а проектируемый формирователь в целом – как устройство, состоящее из трёх элементов, что обеспечило тем самым независимость числа наборов от числа разрядов используемых блоков, т.е. от диапазона перестройки параметров формируемой серии. Это позволило, в отличие от известных подходов, сократить общее число наборов таблицы состояний, что существенно упростило проце-

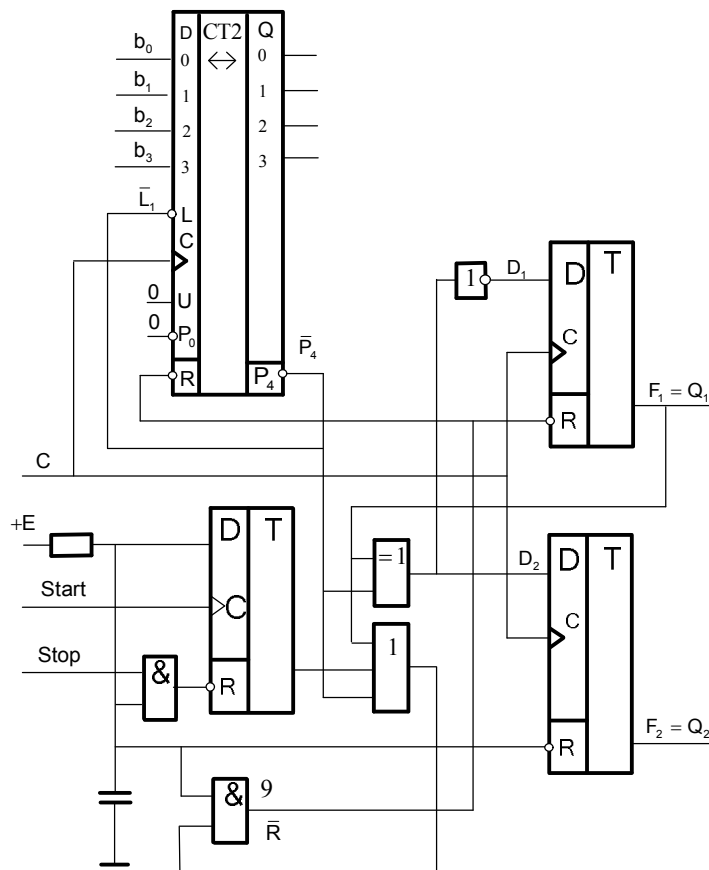


Рис. 7. Схема формувача

дурі синтезу порівняно з відомими класическими методами. Отримана при цьому схема, виконана на типових елементах, по складності вигідно відрізняється від відомих, крім того, наведено стаття переслідує ще одну мету – на прикладі проектування розглянутого конкретного формувача проілюструвати можливості і достоїнства запропонованого апарату синтезу і дати навики синтезу інших схем цифрових автоматів.

Список літератури

1. Уэйкерли Дж. Ф. Проектирование цифровых устройств: пер. с англ. / Дж. Уэйкерли. – М.: Постмаркет, 2002. – 544 с.
2. Соловьев В.В. Проектирование цифровых систем на основе программируемых логических интегральных схем / В.В. Соловьев.- М.: Горячая линия-Телеком, 2001.-636 с.
3. Обобщенные логические функции и системы на программируемой логике. Моногр. / Н.Г. Коробков, Е.Н. Короб-

кова, В.Г. Рубанов, В.С. Харченко; под общ. ред. Харченко В.С. – К.: МОН України, Х.: Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», 2008. – 351 с.

4. Рубанов В.Г. Логическое проектирование цифровых устройств, основанное на представлении функций в обобщенной форме. Моногр. / В.Г. Рубанов, Е.Н. Коробкова. – Белгород: Изд-во БГТУ, 2009. – 240 с.

5. Rubanov V. G. Application of Decomposition Method to Cyclic Finite State Machine Synthesis with Reconfigurable Time Parameters of Output Signals / V.G. Rubanov, E.N. Korobkova, E.P. Dobrinskiy // World Appl.Sci. J. – 2013. – 25 (1). – P. 69-77.

6. Коробкова Е.Н. Формувач тактових імпульсів. Патент України на корисну модель №53542 / Н.Г. Коробков, Е.Н. Коробкова, В.С. Харченко. – Бюллетень №19, 2010.

Поступила в редакцію 7.02.2014

Рецензент: д-р техн. наук, проф. О.Е. Федорович, Національний аерокосмічний університет ім. Н.Е. Жуковського «ХАИ», Харків/

ДОПОВНЕННЯ МЕТОДУ ДЕКОМПОЗИЦІЇ ДО СИНТЕЗУ ФОРМУВАЧІВ ДВОФАЗНОЇ ПЕРІОДИЧНОЇ ПОСЛІДОВНОСТІ ІМПУЛЬСІВ ТИПУ МЕАНДР

О.М. Коробкова

Пропонується метод синтезу формувачів двофазної періодичної послідовності імпульсів типу меандр з перебудовуваними часовими параметрами, заснований на представленні проектного автомата у вигляді функціонально-блокової композиції декількох взаємопов'язаних окремих блоків, кожен з яких визначає один з часових параметрів формованої імпульсної послідовності. Початковий стан входить в композицію блоків трактується як стан нуля. Будь-яке інше стан кожного з вхідних в композицію блоків трактується як стан одиниці, що дозволяє трактувати входить в композицію блоки як найпростіші елементи пам'яті, що мають два стани.

Ключові слова: цифрові автомати, декомпозиція, логічні функції, синтез, мінімізація.

**ANNEX TO THE SYNTHESIS DECOMPOSITION METHOD CONDITIONERS TWO-PHASE
A PERIODIC SEQUENCE OF PULSES TYPE MEANDER**

E.N. Korobkova

We propose a method of synthesis of the two-phase generators periodic pulse sequence type less Andrei tunable time parameters based on representing the projected machine as functional block composition several interrelated separate units, each of which defines one of the temporal parameters of the pulse sequence generated. The initial state of the blocks included in the composition is treated as a zero state. Any other condition of each of the blocks in the composition is treated as a state of unity that allows us to treat incoming blocks in the composition as a simple memory elements having two states.

Keywords: digital machines , decomposition, logic functions , synthesis, minimization.