

ВЕРОЯТНОСТНЫЙ ПОДХОД ПРИ ПОСТРОЕНИИ ИНИЦИАЛИЗИРУЮЩЕЙ ПОСЛЕДОВАТЕЛЬНОСТИ ДЛЯ ПОСЛЕДОВАТЕЛЬНОСТНЫХ СХЕМ

Чебанов П.А.

Луганский национальный педагогический университет им. Т. Шевченко, г. Луганск

Abstract

Tchebanov P.A. A probability approach for initialization sequences computation for synchronous sequential circuits. This paper presents a new approach to the automated generation of an initialization sequence for synchronous sequential circuits. We propose a new algorithm providing a sequence that aims at initializing the highest number of flip flops with the lowest number of vectors.

Общая постановка проблемы. Для синхронных последовательностных схем существует важная проблема инициализации схемы, т.е. перехода схемы из неизвестного состояния в известное, например, при включении. Зачастую это производится путем подачи сигнала сброса ко всем элементам схемы. В других случаях, разработчик схемы предоставляет инициализирующую последовательность, которая способна инициализировать все триггеры известными значениями. Тем не менее, существуют случаи, когда необходимо иметь инструмент, способный определить, возможно ли построить инициализирующую последовательность для произвольной схемы и, в случае необходимости, сгенерировать её. В частности, к таким случаям относятся алгоритмы построения тестов для цифровых схем.

Анализ исследований. Ранее было предложено несколько способов построения инициализирующих последовательностей. Пиксли (Pixley) и другие разработали точный метод, основанный на двоичных диаграммах решения (Binary Decision Diagrams) и технике символьного манипулирования (symbolic manipulation techniques), который способен решить такую проблему для схем малой и большой размерности [1]. Для схем большой размерности этот метод неприменим из-за большого расхода памяти. Структурная декомпозиция описанная в работе Веба (Wehbeh) и Сааба (Saab) позволяет облегчить задачу определения возможности построения инициализирующей последовательности [2]. Недостатком этого метода является то, что для некоторых схем он не может построить необходимую последовательность. Подход, объединяющий двоичные диаграммы решений и жадные алгоритмы, описан в работе Кейма, Беккера и Стеннера [3]. Он эффективен во многих случаях, но иногда он требует очень много процессорного времени.

Решение задач. Для обозначения физических сигналов, распространяющихся по схеме, при логическом моделировании используется *алфавит моделирования*. В процессе моделирования входные, выходные и переменные состояний могут принимать только значения из алфавита моделирования. Поскольку дискретное устройство работает с информацией, представленной в двоичном виде, только несколько логических значений сигналов необходимо для представления физических сигналов.

Минимальным алфавитом для моделирования работы ДУ является двоичный алфавит $B_2 = \{0,1\}$, в котором, как правило, 0 соответствует низкому уровню сигнала, а 1 – высокому.

Для представления неопределённых начальных состояний и сигналов, чьё значение невозможно определить из-за состязаний сигналов в схеме или генерации, используется неизвестное значение u . При этом получаем трёхзначный алфавит $E_3 = \{0,1,u\}$. Также при моделировании используется понятие последовательной глубины схемы, которое определяется как количество триггеров, через которые должен пройти сигнал от входа схемы до самого дальнего вентиля. Исходя из этого определения, можно определить ранг триггера как его номер в последовательности достижения самого дальнего вентиля.

Для построения инициализирующей последовательности входных сигналов предлагается использовать вероятностный подход при моделировании логической схемы. Для этого используется алфавит $P = \{u, H_p, L_p\}$, определяющий вероятности инициализации той или иной линии. В этом алфавите символ H_p используется при вероятности инициализации линии равной 1, символ u используется при вероятности инициализации линии равной 0, и, соответственно, символ L_p используется при вероятности инициализации линии больше нуля и меньшей единицы. Эти символы можно понимать следующим образом, символ H_p показывает, что значение на этой линии будет определенным вне зависимости от сигналов на входах схемы, символ u — значение на этой линии будет неопределенным вне зависимости от сигналов на входах схемы и символ L_p — значение на этой линии будет определенным в зависимости от сигналов на входах схемы. Определим для такого алфавита стандартные функции булевой алгебры, используемые при описании цифровых схем на логическом уровне. Для этого зададим их следующими таблицами истинности.

Таблица 1. Таблица истинности логического AND

AND	u	L_p	H_p
u	u	L_p	L_p
L_p	L_p	L_p	L_p
H_p	L_p	L_p	H_p

Таблица 2. Таблица истинности логического OR

OR	u	L_p	H_p
u	u	L_p	L_p
L_p	L_p	L_p	L_p
H_p	L_p	L_p	H_p

Таблица 3. Таблица истинности логического XOR

XOR	u	L_p	H_p
u	u	L_p	L_p
L_p	L_p	L_p	L_p
H_p	L_p	L_p	H_p

Таблица 4. Таблица истинности логического NOT

NOT	u	L_p	H_p
A	u	L_p	H_p
\bar{A}	u	L_p	H_p

Все таблицы истинности элементарно получаются из таблиц истинности логических элементов на трехзначном алфавите E_3 . Например, при моделировании логического умножения, если на оба входа подать значения с вероятностью инициализации равной 1, т.е. на входах вентиля будут значения либо 0, либо 1, вентиль на выходе даст определенное значение по своему табличному описанию в алфавите E_3 . Если же один из входов будет инициализироваться с вероятностью 0, т.е. на нем будет значение u , то вероятность инициализации выхода вентиля будет меньше 1, но больше нуля. Это легко объяснить тем фактом, что если инициализированный вход примет значение 0, то на выходе однозначно будет 0, а если инициализированный вход примет значение 1, то значение на выходе будет равно u . Соответственно, при вероятности инициализации обоих входов большей нуля, вероятность инициализации выхода будет также больше нуля, но меньше 1.

Используя алфавит P и булевские функции, мы можем моделировать работу любой цифровой схемы. Моделирование схемы на таком алфавите имеет целью разбиение последовательностной схемы на ряд подсхем, причем количество этих подсхем будет равно

последовательностной глубине схемы. Разбиение на подсхемы производится следующим образом:

1. Пусть n - это последовательностная глубина схем.
2. Подаем последовательность входных сигналов длиной 1 состоящую из элементов H_p на внешние входы схемы (при моделировании в алфавите P всей схемы, мы также заменяем входы триггеров на псевдовыходы схемы, а выходы триггеров – на псевдовыходы схемы). После моделирования мы получим схему, в которой будет видно все линии и триггера, т.к. на этих линиях значения будут отличны от u , для которых можно построить инициализирующую последовательность.
3. Удаляем из схемы элементы, имеющие на выходах значение u , и получаем таким образом первую подсхему. Если в результате удаления вентиля получается линия с неизвестным значением, то заменяем эту линию на сигнал $constU$.
4. Подаем последовательность входных сигналов длиной 1 состоящую из элементов H_p на внешние входы полной схемы и псевдовыходы первого ранга.
5. Удаляем из схемы элементы, имеющие на выходах значение u , и получаем таким образом вторую подсхему.
6. Повторяем шаги 4,5 $n-2$ раз, каждый раз увеличивая размер последовательности входных сигналов на количество триггеров соответствующего ранга.
7. В результате получаем n подсхем, необходимых для моделирования.

Рассмотрим приведенный алгоритм на примере схемы, представленной на рисунке 1.

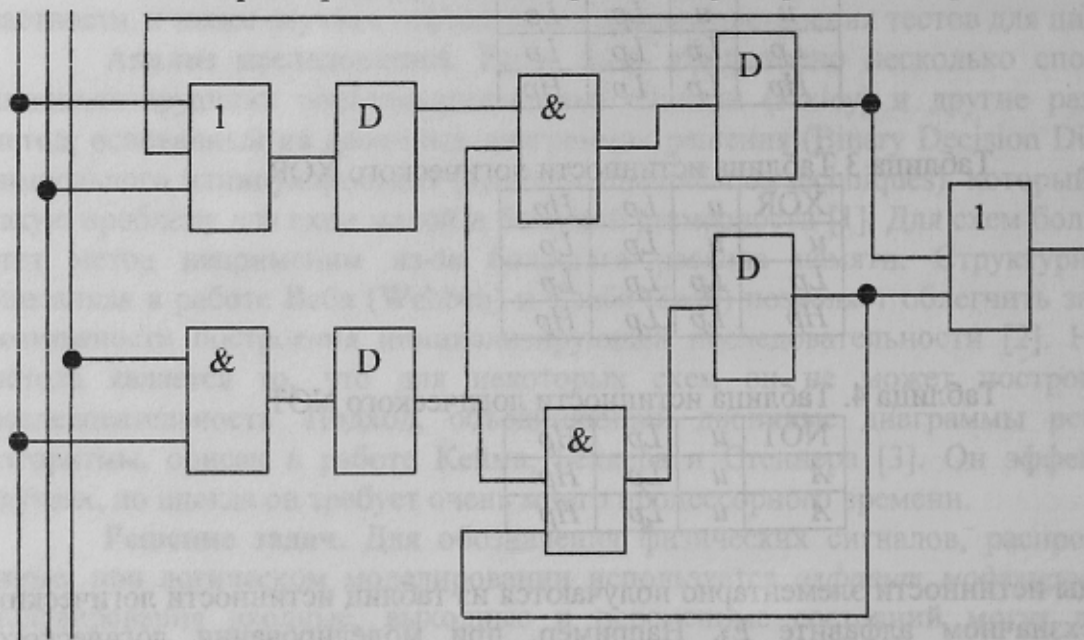


Рисунок 1 - Исходная схема

После выполнения 1 шага алгоритма получаем следующие значения сигналов на линиях (см. рис. 2).

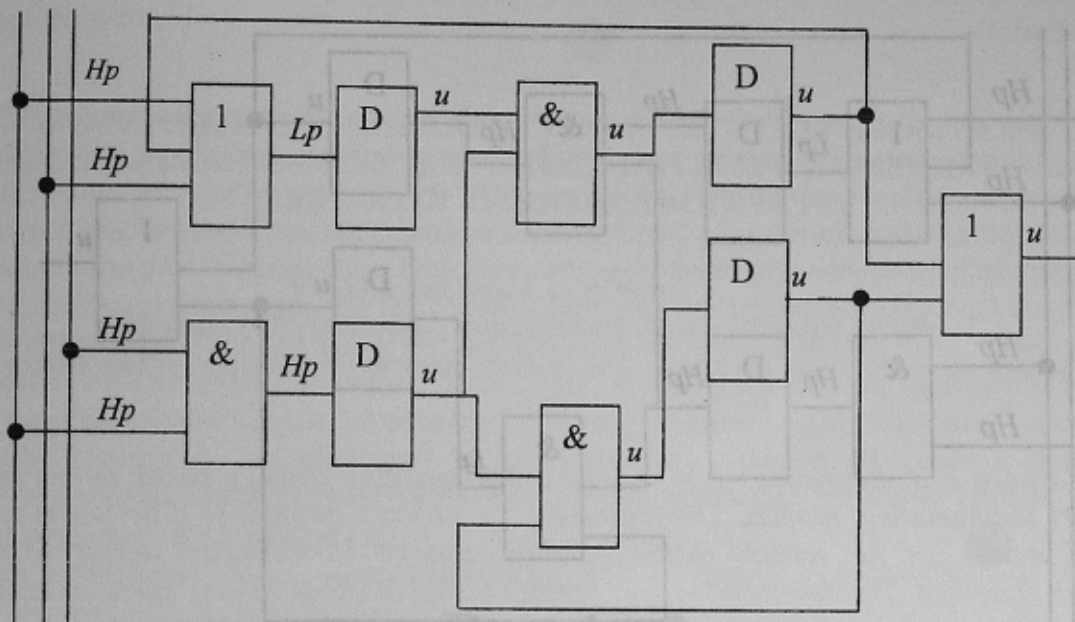


Рисунок 2 - Значения линий

После удаления неопределенных элементов и замены неопределенных линий, получаем первую подсхему (см. рис. 3).

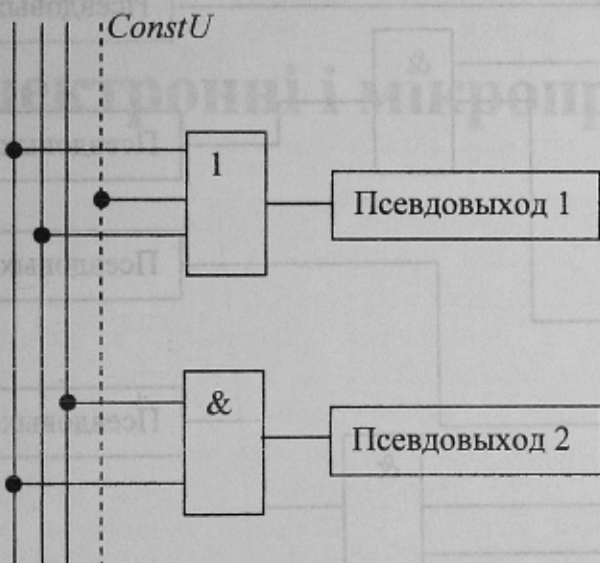


Рисунок 3 - Подсхема №1

Далее, моделируем полную схему на алфавите P , задавая значениями H_p псевдовыходы первого ранга. В результате получаем следующие значения линий схемы (см. рис. 4). После выполнения шага 5 мы получим вторую подсхему, изображенную на рисунке 5.

Перебирая тем или иным способом последовательности входных сигналов длины 1 и моделируя первую полученную схему на двузначном алфавите B_2 , мы можем получить все инициализирующие последовательности для триггеров 1 ранга. После получения таких последовательностей мы можем приступить к моделированию второй подсхемы, используя подготовленные на предыдущем этапе последовательности для инициализации триггеров 2 ранга и т.д. В конечном итоге мы получим инициализирующие последовательности для всех триггеров схемы.

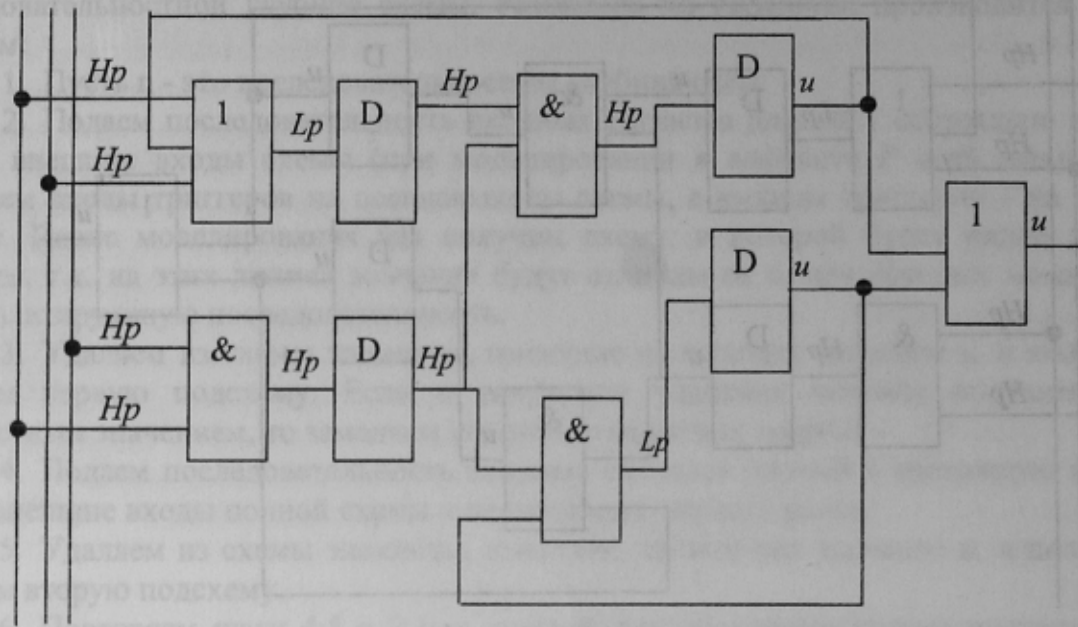


Рисунок 4 - Значения линий схемы после шага 4

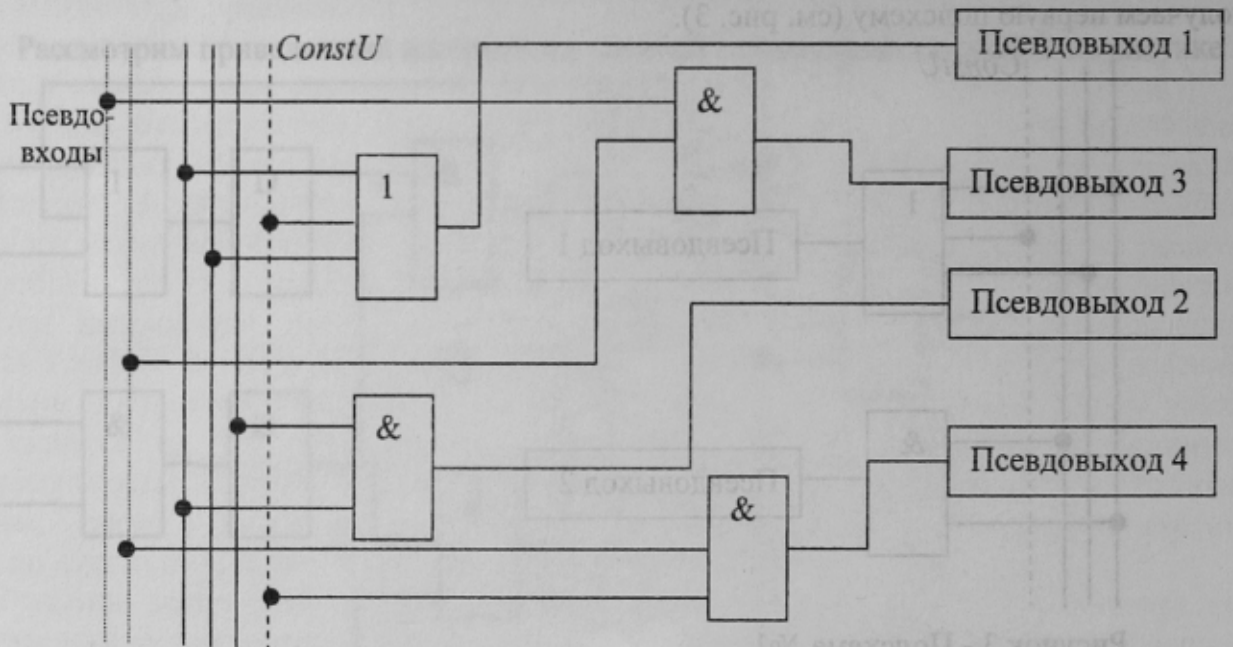


Рисунок 5 - Подсхема №2

Выводы. Предложен на основе многозначных моделей новый структурный алгоритм построения инициализирующих последовательностей для последовательностных схем, которые позволяют перевести схему из неопределенного состояния в некоторое начальное.

Литература

1. C. Pixley, S. Jeong, G. Hatchel, "Exact Calculation of Synchronization Sequences based on Binary Decision Diagrams," IEEE Trans. on CAD, Vol. 13, pp. 1024-1034, 1994.
2. J.A. Wehbeh, D.G. Saab, "Initialization of Sequential Circuits and its Application to ATPG," Proc. IEEE VLSI Test Symp., 1996, pp. 246-251.
3. M. Keim, B. Becker, B. Stenner, "On the (Non-)Resettability of Synchronous Sequential Circuits," Proc. IEEE VLSI Test Symp., 1996, pp. 240-245.
4. Скобцов Ю.А., Скобцов В.Ю. Логическое моделирование и тестирование цифровых устройств.- Донецк, 2005.