

УДК 004.3

А.А. Баркалов (д.т.н, профессор),
А.С. Лаврик (ассистент), **А.Ю. Стародубцева** (студент)
Донецкий национальный технический университет
A.Barkalov@iie.uz.zgora.pl
lavrik@cs.dgtu.donetsk.ua

СИНТЕЗ ПАРАЛЛЕЛЬНОГО АВТОМАТА МУРА НА FPGA С УНИТАРНЫМ КОДИРОВАНИЕМ СОСТОЯНИЙ

В статье предложен метод синтеза для автомата Мура с унитарным кодированием состояний. Метод направлен на увеличение быстродействия устройства за счёт представления управляющего автомата в виде сети Петри, что допускает наличие параллельно выполняющихся блоков алгоритма. Приведены метод синтеза, а также пример применения.

Устройство управления, сеть Петри, алгоритм, FPGA

Общая постановка проблемы

Согласно принципу микропрограммного управления цифровая система представляет собой композицию операционного автомата (ОА) и управляющего автомата (УА). Многие вычисления или операции ОА могут производиться параллельно, однако тут встает проблема реализации управления такими операционными автоматами. В настоящее время цифровые устройства, в частности, управляющие автоматы (УА), строятся на ПЛИС с архитектурой FPGA [1]. Функциональный базис FPGA представляет собой множество идентичных конфигурируемых логических блоков (КЛБ) с R входами, реализующих произвольную булеву функцию R аргументов. В зависимости от модели микросхемы FPGA $R = \overline{2,5}$ [1]. Ввиду особенностей функционального базиса FPGA, предпочтение отдается УА с жесткой логикой. Однако при этом реализованные в известных пакетах автоматизированного проектирования (фирм Altera, Xilinx и других) методы синтеза основаны на традиционных подходах, которые не позволяют эффективно использовать архитектурные возможности новой элементной базы. Помимо этого традиционный подход к синтезу УА не позволяет реализовать алгоритмы параллельного управления путем синтеза одного УА. А иерархическое построение нескольких управляющих автоматов значительно снижает быстродействие таких алгоритмов.

Пути оптимизации схем управляющих автоматов, предложенные в работах [2] – [4] (а именно: метод синтеза УА Мили на FPGA, основанный на функциональной декомпозиции [2], методы декомпозиции, позволяющие сократить число КЛБ реализуемых функций [3], методы синтеза комбинационных схем УА в виде многоуровневых структур [4]), приводят к тому, что оптимизированный автомат Мура работает либо с одинаковым быстродействием по сравнению с традиционной реализацией, либо же оптимизация приводит к уменьшению быстродействия автомата.

В работе [5] предложено применение унитарного кодирования состояний УА Мура для вырождения схем дешифратора и уменьшения числа аргументов СФФВ. А также предложено представление схемы УА Мура в виде сети Петри, что позволило исключить традиционный синтез автомата по таблице переходов. В результате повышено быстродействие синтезируемого автомата и исключен традиционный синтез, что позволит реализацию параллельного управляющего автомата с жесткой логикой на FPGA.

Постановка задач и целей исследований

На рисунке 1 приведена базовая структура УА Мура, в которую входят регистр состояния (Pг), комбинационные схемы для формирования функций возбуждения (СФФВ), микроопераций (СФМО), а также дешифратор состояний (DC).

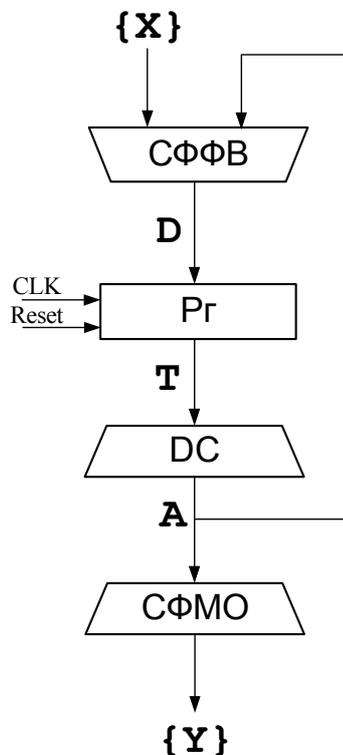


Рисунок 1 – Базовая структура УА Мура

В FPGA в состав каждого КЛБ входит D-триггер, что позволяет наиболее оптимально реализовывать на FPGA схемы с древовидной структурой и большим числом состояний, в частности, сети Петри [6]. В работе [5] предлагается применить унитарное кодирование состояний УА Мура, что приведет к вырождению схемы дешифратора и уменьшит число аргументов СФФВ. Также предлагается представление схемы УА Мура в виде сети Петри, что позволит исключить традиционный синтез автомата по таблице переходов.

Простой сетью Петри называется набор $N=(S,T,M,P,F)$, где:

1. $S = \{s_1, \dots, s_N\}$ – множество мест;
2. $T = \{t_1, \dots, t_N\}$ – множество переходов таких, что $S \cap T = \emptyset$;
3. $M = \{m_1, \dots, m_N\}$ – множество меток;
4. $P = \{p_1, \dots, p_N\}$ – множество событий;
5. $F \subseteq \mu S \times T \times \mu S$ – отношение инцидентности такое, что
 - а) $\forall (S'_1, t_1, S''_1), (S'_2, t_2, S''_2) \in F : (S'_1, t_1, S''_1) \neq (S'_2, t_2, S''_2) \Rightarrow t_1 \neq t_2$;
 - б) $\{t \mid \langle S', t, S'' \rangle \in F\} = T$.

Сеть Петри представляет собой двудольный ориентированный граф, состоящий из вершин двух типов – позиций и переходов, соединенных между собой дугами, вершины одного типа не могут быть соединены непосредственно. В позициях размещены метки (маркеры), способные перемещаться по сети. Событием в сети Петри называется срабатывание перехода, при котором метки из входных позиций перемещаются в выходные позиции. Переход срабатывает, если имеется как минимум 1 метка в каждом из его исходных позиций.

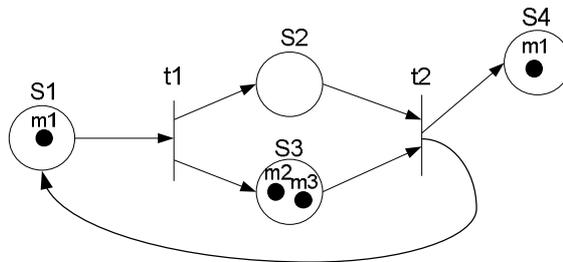


Рисунок 2 – Пример простой сети Петри

Функционирование данной сети заключается в перемещении меток m_1 - m_4 по дугам между местами S_1 - S_4 и формировании событий. Переход t_1 сработает только в том случае, если в позиции S_1 находится хотя бы один маркер, при этом событии в позициях S_2 и S_3 появится хотя бы по одному маркеру. Переход t_2 сработает только в том случае, если и в позиции S_2 и в позиции S_3 находится хотя бы по одному маркеру, при этом событии в позициях S_1 и S_4 появится хотя бы по одному маркеру.

Исходное размещение меток по местам является начальным условием и может быть произвольным.

Исходя из этого, произвольная граф-схема алгоритма (ГСА) для УА является частным случаем сети Петри с одной меткой, которая соответствует текущему состоянию автомата, при выполнении последовательной части алгоритма и с несколькими метками, чье количество соответствует количеству параллельных ветвей алгоритма, при выполнении параллельной части алгоритма управления. Множество переходов t при этом соответствует множеству условных вершин, множество событий – множеству наборов управляющих сигналов, а множество мест S при этом соответствует множеству состояний автомата и множеству флагов операционного автомата.

Таким образом, параллельный УА может быть представлен в виде сети Петри.

Решение задач и результаты исследований.

Рассмотрим реализацию УА Мура в виде сети Петри, заданного ГСА на рисунке 3.

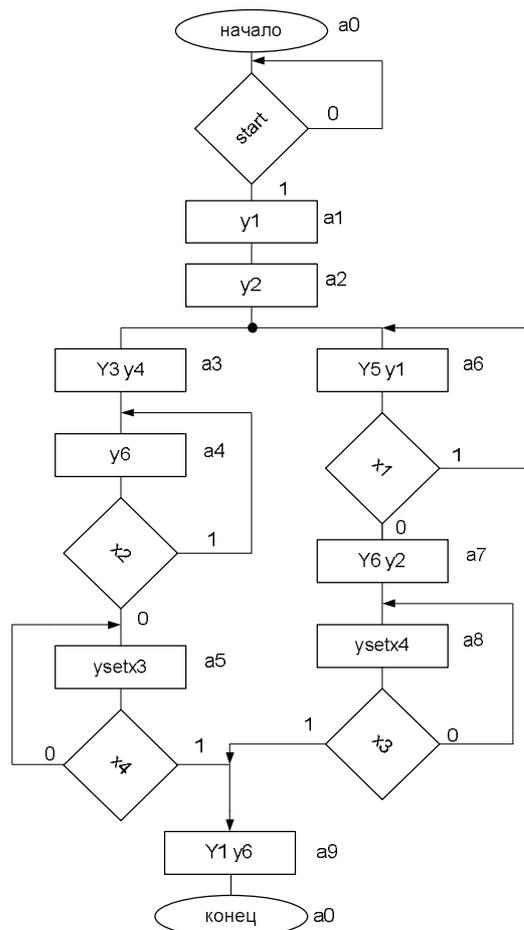


Рисунок 3 – Исходная граф-схема алгоритма

Данная ГСА имеет параллельно выполняющиеся ветви с операторными вершинами a_3, a_4, a_5 , принадлежащими одной ветви и вершинами a_6, a_7, a_8 – другой. Вершины a_5 и a_8 введены в данный алгоритм для синхронизации, то есть последующая последовательная часть не будет выполняться, пока не отработают обе параллельные ветви. Существуют алгоритмы, где такая синхронизация не требуется - это учитывается при разработке алгоритма и на синтез управляющего автомата не влияет.

По данной ГСА можно построить сеть Петри с десятью состояниями – a_0 - a_9 , с пятью условиями – $Start, x_1, x_2, x_3, x_4$. Такая сеть Петри приведена на рисунке 4. При этом не указаны управляющие сигналы, чтобы не загромождать схему. Их формирование происходит как и при традиционном синтезе автомата Мура.

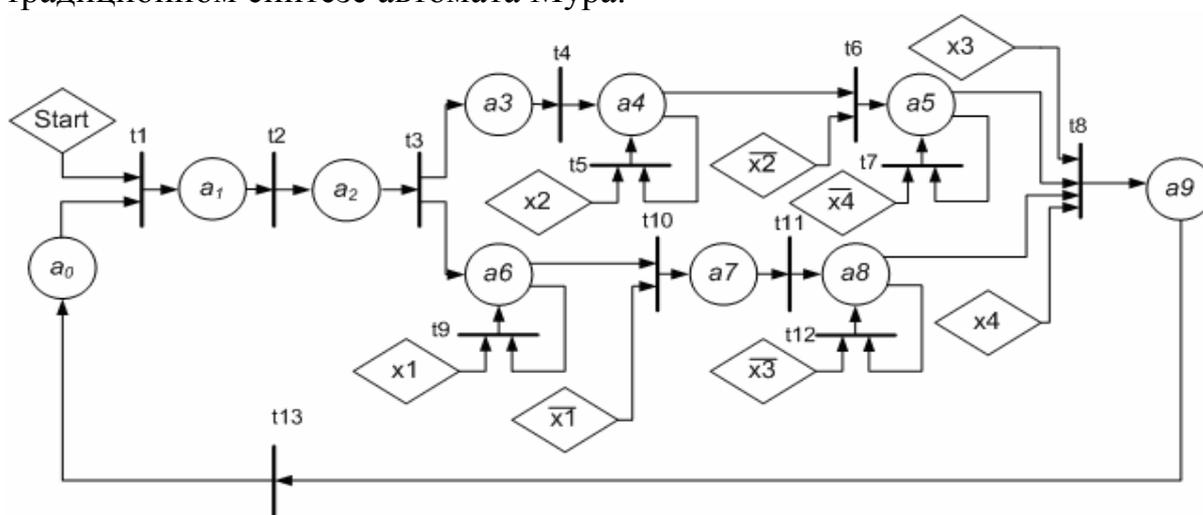


Рисунок 4 – Сеть Петри для исходной ГСА

Сеть функционирует следующим образом. В исходном состоянии метка находится в позиции a_0 . При этом не формируется никаких событий. Данная позиция является ждущей вершиной. Переход маркера в позицию a_1 осуществляется через переход t_1 только при выполнении условия $Start = 1$. Из a_1 происходит безусловный переход в позицию a_2 через t_2 . Из a_2 происходит переход через t_3 одновременно в позиции a_3 и a_6 - далее происходит параллельная работа алгоритма. Переход t_8 сработает только при завершении выполнения обеих параллельных ветвей. По завершении работы алгоритма сработает переход t_{13} и система перейдет в исходное состояние.

Для реализации схемы данного автомата на FPGA, необходимо заменить компоненты сети на рисунке 4 их функциональными аналогами. Если в качестве позиции (состояния автомата) использовать D-триггер, то маркером будет служить сигнал логической единицы, записанной в триггер текущего состояния. Позиции, которые соответствуют условным

вершинам, не заменяются на триггеры, а просто подаются в качестве входных сигналов. Множество входов в каждое состояние объединяется элементом «ИЛИ» и подается на соответствующий триггер. Переходам соответствует элемент «И», на вход которого подаётся выход триггера, соответствующий позиции, из которой осуществляется переход, а также флаги условий. Если переход безусловный, то ему соответствует обычное электрическое соединение. Множество входов в позицию объединяется элементами «ИЛИ». Дугам, соединяющим места и переходы, соответствуют обычные электрические соединения. На рисунке 5 показана функциональная схема УА, полученная из сети Петри для рассматриваемой в примере ГСА.

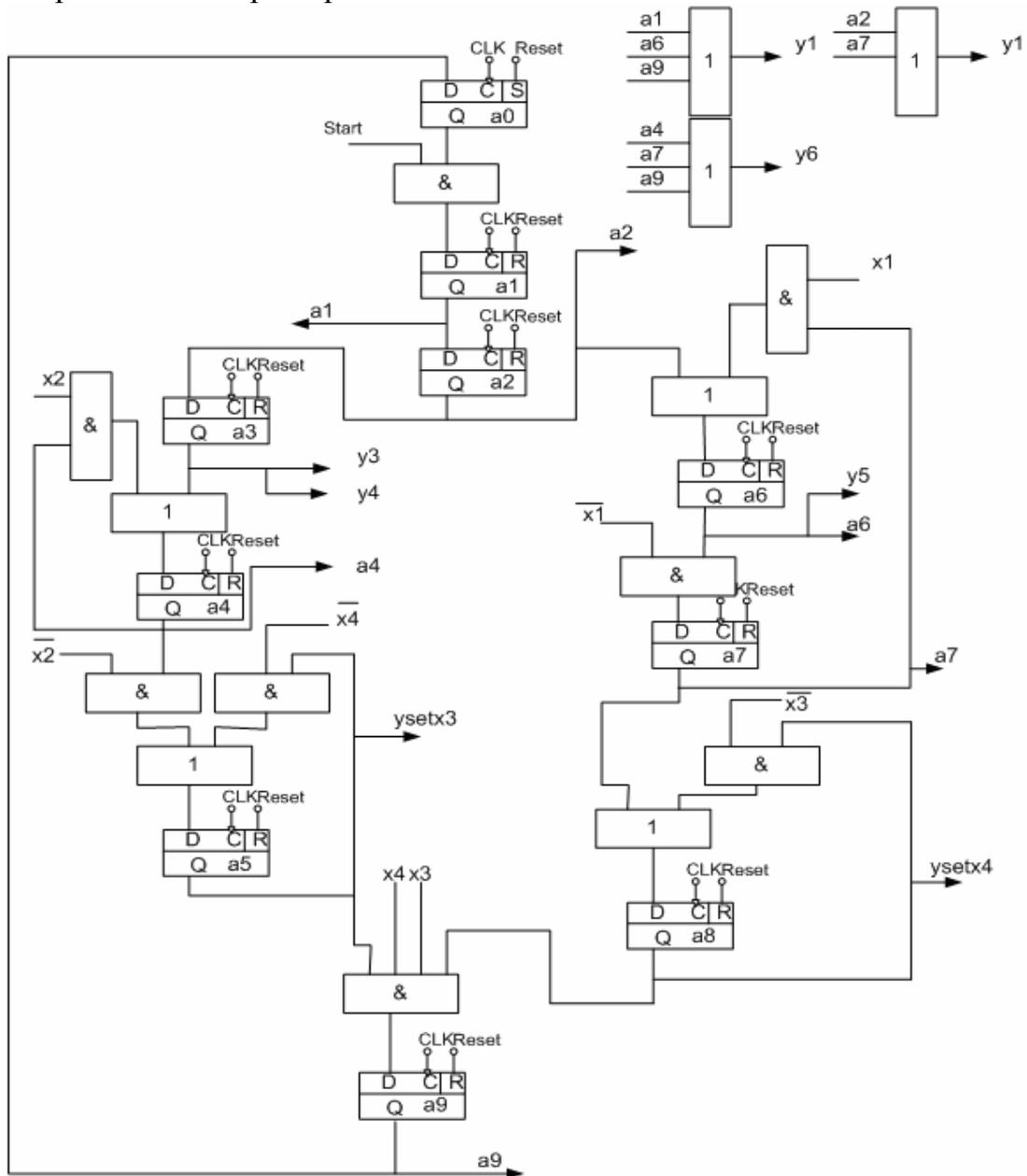


Рисунок 5 – Функциональная схема УА Мура по ГСА

Схема работает следующим образом. По сигналу $\text{Reset}=1$ все триггеры, кроме триггера a_0 сбрасываются в '0', триггер a_0 при этом устанавливается в '1', что соответствует состоянию автомата a_0 . При появлении сигнала $\text{Start}=1$, в триггер a_1 по переднему фронту синхросигнала CLK записывается логическая '1', что соответствует переходу автомата в первое состояние. В последовательной части алгоритма (вершины a_1 - a_2) на каждом такте CLK в один из триггеров записывается '1', что соответствует переходу из предыдущего состояния в следующее. При этом в триггер, из которого происходит запись '1', записывается '0'. Для обеспечения корректной работы схемы сигнал Start должен иметь длительность не более одного такта и вырабатываться один раз при запуске УА. При переходе из состояния a_2 в параллельные ветви алгоритма, то есть одновременно в состояния a_3 и a_6 , триггер a_2 сбрасывается в '0', а триггеры a_3 и a_6 устанавливаются в '1'. Далее обе эти ветви работают независимо друг от друга, аналогично ранее рассмотренной логике. При прохождении состояний a_5 и a_8 управляющими сигналами устанавливаются флаги x_3 и x_4 соответственно, которые сигнализируют о том, что работа ветви завершена. Пока эти флаги не установлены – та ветвь, которая закончила работу раньше, будет находиться в своем последнем состоянии и ожидать завершения другой ветви. Переход в состояние a_9 осуществится в том случае, если переход будет одновременно из состояний a_5 и a_8 . Отработав состояние a_9 система возвращается в исходное состояние a_0 .

Далее полученная схема описывается на языке VHDL для последующего автоматического синтеза и окончательной имплементации на FPGA.

Преимуществом такой реализации УА на FPGA являются:

1. присутствие параллельно выполняющихся блоков алгоритма;
2. идентичность блоков и регулярность структуры автомата;
3. минимальная комбинационная часть, распределенная по всей схеме FPGA;
4. высокое быстродействие автомата.

Выводы

Формализация синтеза и анализа таких схем УА позволила автоматизировать генерацию конечного VHDL-кода по исходной ГСА, заданной текстовым файлом. Синтез, окончательная оптимизация размещения схемы по КЛБ и оценка параметров полученной реализации выполняется уже непосредственно в САПР.

Список литературы

1. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. – СПб.: БХВ-Петербург, 2002. – 608 с.
2. Красичков А.А. Синтез микропрограммных автоматов на FPGA. Наукові праці ДонНТУ. Серія: “Інформатика, кібернетика та обчислювальна техніка”. Випуск 64. – Донецьк: Вид-во ДонНТУ, 2003. – 280 с.
3. Баркалов А. А., Красичков А.А. Основные подходы к декомпозиции булевых функций при реализации на FPGA. Наукові праці ДонНТУ. Серія: “Інформатика, кібернетика та обчислювальна техніка”. Випуск 70. – Донецьк: Вид-во ДонНТУ, 2003. – с.243–250.
4. Баркалов А. А. Синтез устройств управления на программируемых логических устройствах. – Донецьк: ДонНТУ, 2002 – 262 с.
5. Баркалов А.А., Красичков А.А., Халед Баракат. Синтез автомата Мура на FPGA с унитарным кодированием состояний. Наукові праці Донецького національного технічного університету. Серія “Обчислювальна техніка та автоматизація”. Випуск №106 – Донецьк: ДонНТУ, 2005 – С.157-162.
6. Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984 – 368с.
7. Котов В. Е. Сети Петри. – М.: Наука, 1984 – 263 с.

Надійшла до редколегії 26.09.2009р Рецензент: д.т.н., проф. Скобцов Ю.О.

О.О. Баркалов, О.С. Лаврік, Г.Ю. Стародубцева
Донецький національний технічний університет

Синтез параллельного автомата Мура на FPGA з унітарним кодуванням станів. У статті запропонований метод синтезу для автомату Мура з унітарним кодуванням станів. Метод спрямований на покращення швидкодії пристрою за рахунок представлення керуючого автомату у вигляді мережі Петрі, що припускає наявність блоків алгоритма що виконуються паралельно. Наведено метод синтезу а також приклад застосування.

Пристрій керування, мережа Петрі, алгоритм, FPGA

A.A. Barkalov, A.S. Lavrik, A.Y. Starodubtseva
Donetsk National Technical University

Synthesis of the Moore paralleling FSM with the unitary states coding on FPGA. Synthesis method for Moore machine with hot-one state coding is proposed in the article. A method is directed on increasing the fast-acting of device due to presentation of control unit as a Petri network, that assumes the presence of the parallel executed blocks of algorithm. Method of synthesis and example of application are given.

Control unit, Petri network, algorithm, FPGA