

УДК 681.3

СИНТЕЗ АВТОМАТА МУРА С ПАРАЛЛЕЛЬНЫМИ СОСТОЯНИЯМИ НА FPGA

*Стародубцева А.Ю., Красичков А.А.,
Донецкий національний технічний університет*

Согласно принципу микропрограммного управления цифровая система представляет собой композицию операционного автомата (ОА) и управляющего автомата (УА). В настоящее время цифровые устройства, в частности, управляющие автоматы (УА), строятся на ПЛИС с архитектурой FPGA [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. Кроме того вычисления или операции ОА могут производиться параллельно, однако 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.

В работе [2] 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.

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 [3]. В работе [2] 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.

Сеть Петри является двудольным ориентированным графом (пример простой сети Петри приведен на рис. 1), состоящим 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.

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 работает 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 работает 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

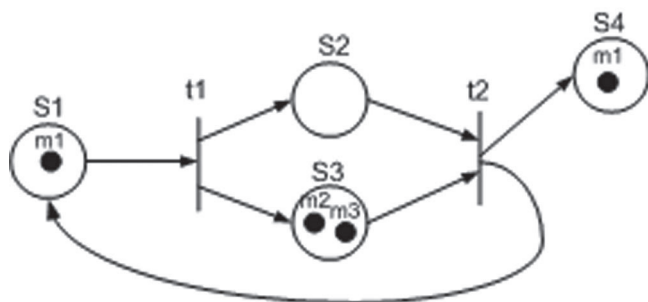


Рисунок 1 - Пример простой сети Петри

S4 появится хотя бы по одному маркеру. Исходное размещение меток по местам является начальным условием и может быть произвольным.

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.

Рассмотрим реализацию УА Мура, ГСА которого приведена на рис. 2, в виде сети Петри. В данном алгоритме присутствуют 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 заданной ГСА можно построить сеть Петри с десятью состояниями a_0 - a_9 и пятью условиями – Start, x_1, x_2, x_3, x_4 . Такая сеть Петри приведена на рис. 3. При этом не указаны управляющие сигналы, чтобы не загромождать схему. Их формулы соответствуют формулам управляющих сигналов при традиционном синтезе автомата Мура.

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

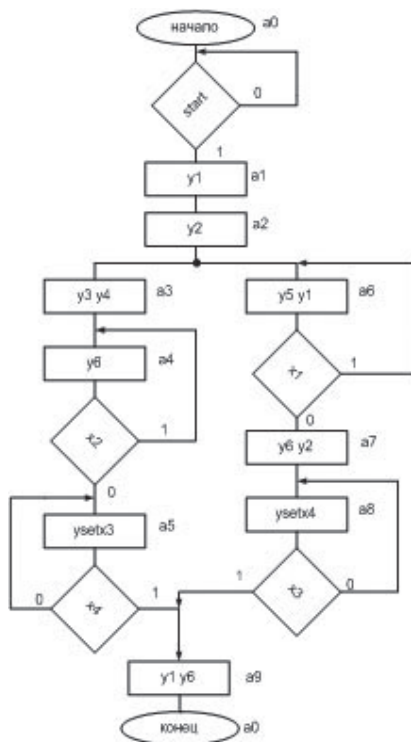


Рисунок 2 - Исходная граф-схема алгоритма

безусловный переход в позицию a2 через t2. Из a2 происходит переход через t3 одновременно в позиции a3 и a6 - далее происходит параллельная работа алгоритма. Из a1 происходит безусловный переход в позицию a2 через t2. Из a2 происходит переход через t3 одновременно в позиции a3 и a6 - далее происходит параллельная работа алгоритма. Переход t8 сработает только при завершении выполнения обеих параллельных ветвей. По завершении работы алгоритма сработает система перейдет в исходное состояние.

Для реализации схемы данного автомата на FPGA, необходимо заменить компоненты сети на рисунке 3 их функциональными аналогами. Если в качестве позиции (состояния автомата) использовать D-триггер, то маркером будет служить

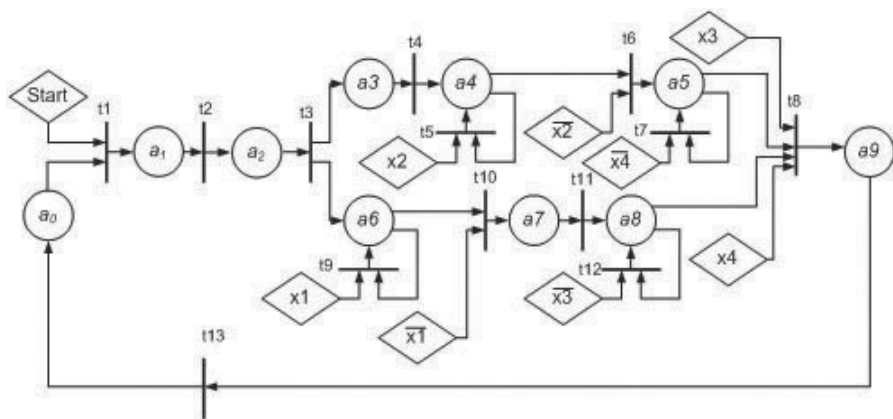


Рисунок 3 - Сеть Петри для исходной ГСА

сигнал логической единицы, записанной в триггер текущего состояния. Позиции, которые соответствуют условным вершинам 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 подается 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 рис. 4 показана функциональная схема УА, полученная из сети Петри для рассматриваемой в примере ГСА.

Схема работает следующим образом. По сигналу $Reset='1'$ 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 $Start='1'$, 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

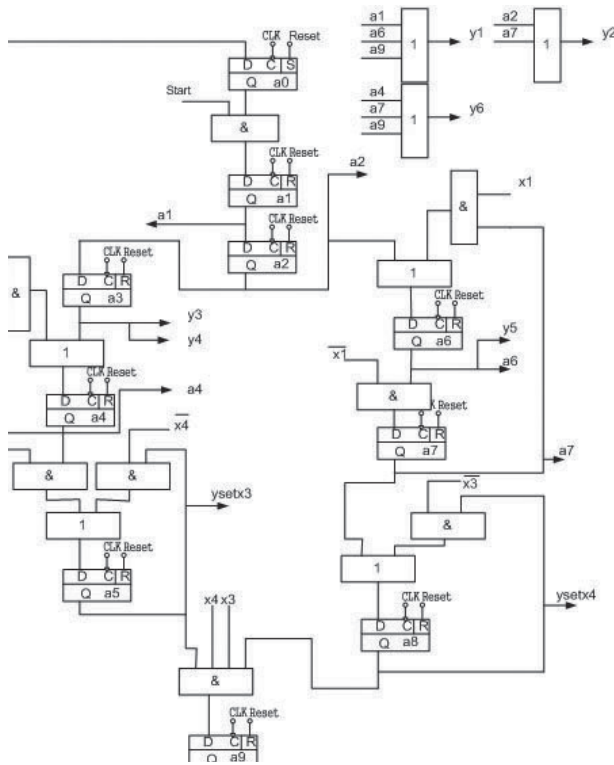


Рисунок 4 - Функциональная схема УА Мура по ГСА

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 a2 в параллельные ветви алгоритма, то есть одновременно в состояния a3 и a6, триггер a2 сбрасывается в '0', а триггеры a3 и a6 устанавливаются в '1'. Далее обе эти ветви работают независимо друг от друга, аналогично ранее рассмотренной логике. Переход в состояние a9 осуществится в том случае, если переход будет одновременно из состояний a5 и a8. Отработав состояние a9 система возвращается в исходное состояние a0.

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

Преимуществом такой реализации УА на FPGA являются:

- присутствие параллельно-выполняющихся блоков алгоритма;
- идентичность блоков и регулярность структуры управляющего автомата;
- минимальная комбинационная часть, распределенная по всей схеме FPGA;
- высокое быстродействие автомата.

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.

Литература

- [1] Грушвицкий Р. И., Мурсаев А. Х., Угрюмов Е. П. Проектирование систем на микросхемах программируемой логики. – СПб.: БХВ-Петербург, 2002. – 608 с.
- [2] 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.
- [3] Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984 – 368с.
- [4] Котов В.Е. Сети Петри. – М.: Наука, 1984 – 263 с.