

КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

A. A. Barkalov, A. A. Krasichkov,
A. V. Matvienko

THE REALIZATION OF THE MOORE AUTOMATION ON FPGA AS A PETRI NETWORK

The method of Moore FSM as a Petri network on FPGA is offered. The method is based on the flow-chart transformation to a Petri network with its realization as principal circuit. The features of the Moore FSM circuit realization on FPGA are considered and the method of its synthesis on the base of flow-chart is offered. An example of proposed method application is given.

Предложен метод синтеза автоматов Мура в виде сети Петри на микросхемах FPGA. Метод основан на преобразовании исходной граф-схемы алгоритма к сети Петри с дальнейшей ее реализацией в виде принципиальной схемы. Рассмотрены особенности реализации схемы автомата на микросхемах FPGA. Предложена методика ее синтеза по граф-схеме алгоритма. Приведен пример применения предложенного метода.

© А.А. Баркалов, А.А. Красичков,
А.В. Матвиенко, 2006

УДК 681.324

А.А. БАРКАЛОВ, А.А. КРАСИЧКОВ,
А.В. МАТВИЕНКО

РЕАЛИЗАЦИЯ АВТОМАТА МУРА НА FPGA В ВИДЕ СЕТИ ПЕТРИ

В настоящее время для реализации схем управляющих автоматов (УА) применяются программируемые логические интегральные схемы (ПЛИС) с архитектурой FPGA [1]. Ввиду особенностей функционального базиса FPGA, предпочтение отдается УА с жесткой логикой [2]. При этом актуальными являются задачи минимизации логической схемы и повышение быстродействия УА.

На рис. 1 показана базовая структура УА Мура, в которую входят регистр состояния (Pr), комбинационные схемы для формирования функций возбуждения (P_1) и выходных сигналов (P_2).

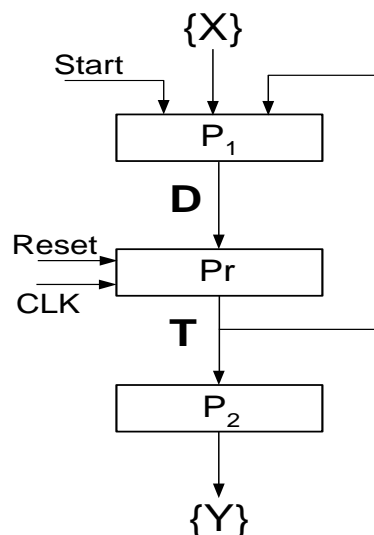


РИС. 1. Структурная схема канонического автомата Мура

Традиционно синтез УА выполняется по указанной базовой структуре, а оптимизация схемы достигается за счет ряда методов, таких как оптимальное кодирование состояний, выделение подуровней в схемах P_1 и P_2 , трансформация кодов объектов [3,4].

Общая структура FPGA включает множество идентичных конфигурируемых логических блоков (КЛБ) с R входами, которые могут реализовать произвольную булеву функцию R переменных. В зависимости от модели микросхемы FPGA $R = 2,5$ [2]. При реализации традиционных структур УА число переменных значительно больше, что приводит к невыгодно громоздкой схеме на вентиляльном уровне и снижению быстродействия.

К выходу каждого КЛБ подключен DC-триггер, что позволяет наиболее оптимально реализовывать на FPGA схемы с древовидной структурой и большим числом состояний, в частности сети Петри [5].

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) $\{t \mid \langle S', t, S'' \rangle \in F\} = T$.

Условия в п. 5 свидетельствуют о том, что для каждого перехода $t \in T$ существует единственный элемент $\langle S', t, S'' \rangle \in F$, задающий для него входное множество мест S' и выходное множество S'' .

Модель сети Петри является принципиально асинхронной, служит для отображения и анализа причинно-следственных связей в системе. Для привязки к определенным моментам времени тех или иных переходов используются события [6]. Сеть Петри имеет наглядную форму представления в виде графа, который включает четыре базовых элемента: позиции (места), переходы, дуги и метки. На рис. 2 показан пример графа, задающего простую сеть Петри с тремя местами, тремя метками и двумя переходами.

Исходя из этого, любая граф-схема алгоритма (ГСА) для УА является частным случаем сети Петри. При этом существует только одна метка, соответствующая текущему состоянию автомата. Множество мест S соответствует множеству состояний автомата, множество переходов t – множеству условных вершин, множество событий – множеству наборов управляющих сигналов.

Таким образом, произвольный УА может быть представлен в виде сети Петри. В том случае, если события возникают при наличии метки в определенном месте, сеть отображает поведение УА Мура. Если события формируются во время перемещений меток между местами, в зависимости от переходов, то сеть отображает поведение автомата Мили.

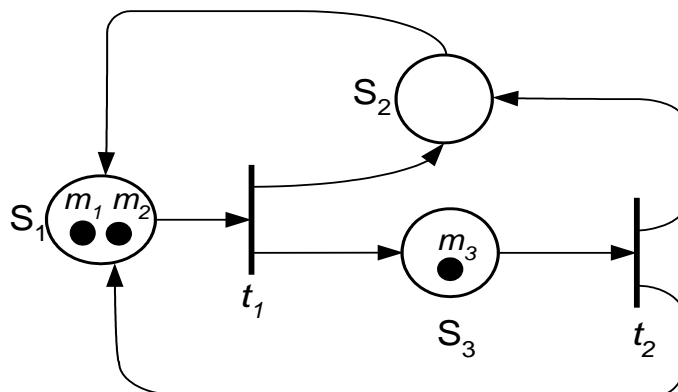


РИС. 2. Пример простой сети Петри

Рассмотрим реализацию УА Мура в виде сети Петри, заданного ГСАГ на рис. 3.

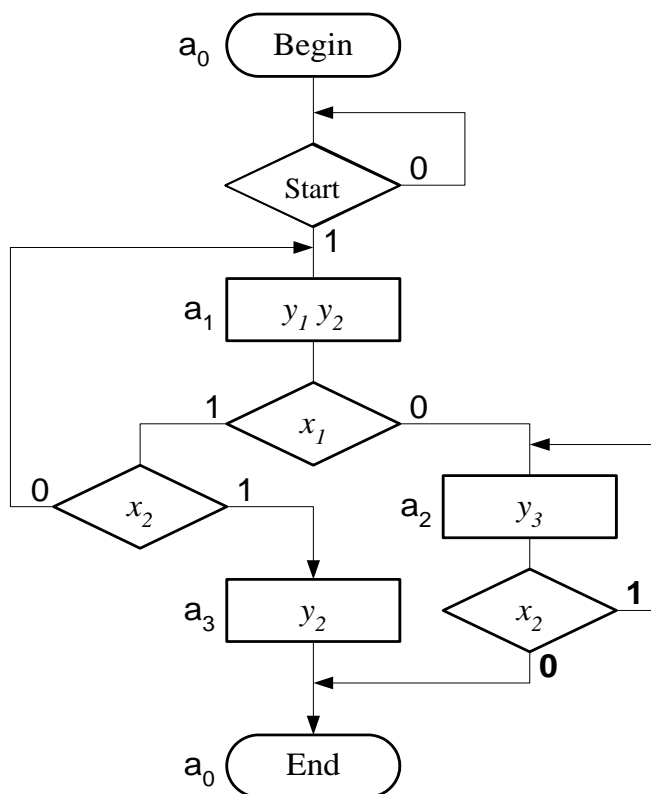


РИС. 3. Исходная граф-схема алгоритма Г

Данная ГСА соответствует сети Петри с четырьмя состояниями – a_0 - a_3 , четырьмя переходами с тремя условиями – $Start, x_1, x_2$, и тремя событиями – y_1 - y_3 , (рис. 4).

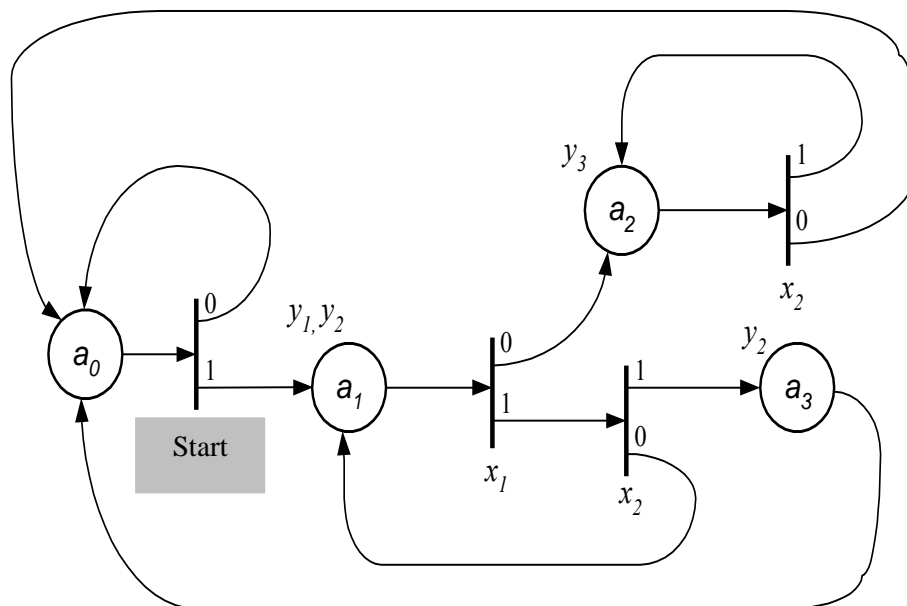


РИС. 4. Сеть Петри для ГСАГ

Сеть функционирует следующим образом. В исходном состоянии метка находится в месте a_0 . При этом не формируется никаких событий. Данное место является ждущей вершиной. Переход метки в место a_1 осуществляется через первый переход только после выполнения условия $Start=1$. Пока метка находится в месте a_1 , действительны события y_1 и y_2 . Из a_1 , благодаря двум переходам (по x_1 и x_2), возможны три варианта перемещений – в a_1 , a_2 или a_3 . Месту a_2 соответствует событие y_3 , а месту a_3 – событие y_2 . Данная сеть не имеет тупиковых состояний, поскольку соответствует замкнутому алгоритму УА.

Для реализации схемы данного автомата на FPGA необходимо заменить компоненты сети (рис. 4) их функциональными аналогами. Если в качестве места (состояния) использовать ДС-триггер, то меткой будет служить сигнал логической единицы, записанной в триггер текущего состояния. Множество входов в каждое состояние объединяется элементом “ИЛИ” и подается на соответствующий триггер. Функцию перехода выполняет обычный демультиплексор на два выхода, управляемый одним условием. События (управляющие сигналы) формируются как логическое “ИЛИ” выходов соответствующих им состояний (триггеров). Дугам, соединяющим места и переходы, соответствуют обычные электрические соединения. На рис. 5 показана функциональная схема УА, полученная из сети Петри для ГСАГ.

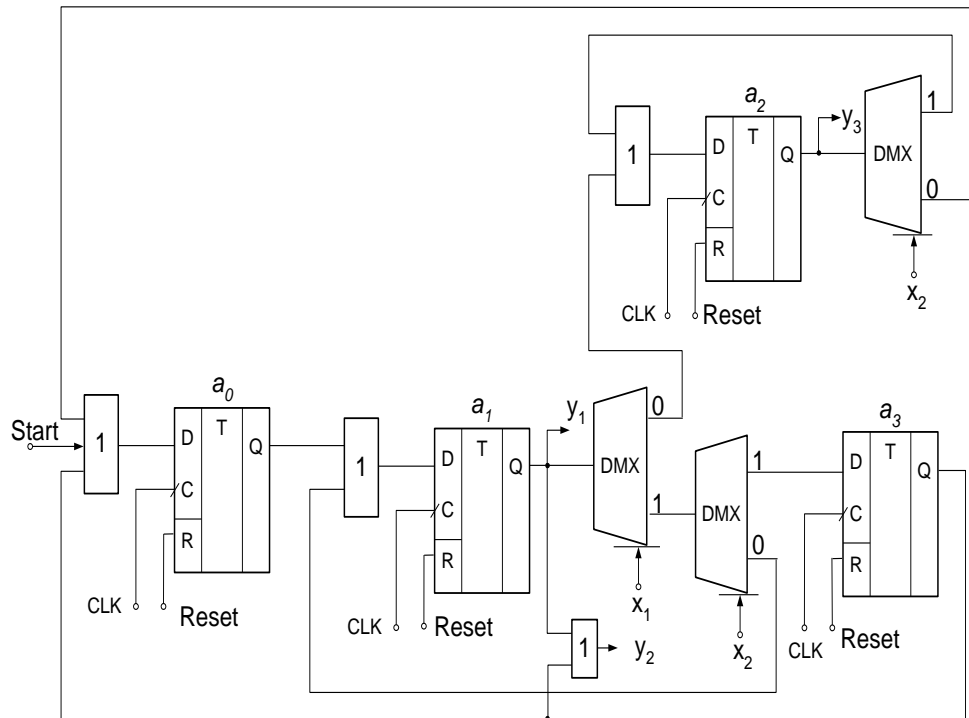


РИС. 5. Функциональная схема УА Мура по ГСАГ

Схема работает следующим образом. По сигналу $Reset = "1"$ все триггеры сбрасываются в "0", что соответствует начальному состоянию автомата a_0 , поскольку при этом не вырабатывается никаких управляющих сигналов. При появлении сигнала $Start$, в триггер a_0 по переднему фронту синхросигнала CLK записывается логическая "1", что разрешает дальнейшее функционирование схемы УА по заданному алгоритму. Такая реализация позволяет не использовать демультиплексор, который соответствует на рис. 4 переходу с условием $Start$. В каждом такте CLK в один из триггеров записывается "1", что соответствует переходу из одного состояния в другое. 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.

Далее полученная схема описывается на языке $VHDL$ для последующего автоматического синтеза и окончательной имплементации на $FPGA$ [2].

Преимущества такой реализации УА на $FPGA$:

- идентичность блоков, регулярность структуры автомата;
- минимальная комбинационная часть, распределенная по всей схеме;

- небольшое число входов комбинационных схем, не зависящее от числа состояний автомата;
- высокое быстродействие автомата;
- отсутствие синтеза.

Удобство проектирования и исследования таких УА заключается в том, что можно автоматизировать генерацию конечного VHDL-кода по исходной ГСА. Синтез, оптимизация и оценка параметров полученной реализации выполняется САПР. При имплементации данной схемы УА на FPGA серии Spartan-II использовано 5 четырехходовых КЛБ и 4 триггера (в составе КЛБ). Максимальная задержка сигналов в схеме составила 7,3 нс. При реализации данной логической схемы УА на других семействах FPGA возможны другие показатели быстродействия и затрат.

1. *Кнышев Д.А., Кузелин М.О.* ПЛИС фирмы Xilinx: структура и применение. – М.: ДОДЭКА, 2000. – 270 с.
2. *Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П.* Проектирование систем на микросхемах программируемой логики. – СПб.: БХВ-Петербург, 2002. – 608 с.
3. *Баркалов А.А., Палагин А.В.* Синтез микропрограммных устройств управления. – Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины, 1997 – 135 с.
4. *Баркалов А.А.* Синтез устройств управления на программируемых логических устройствах. – Донецк: ДонНТУ, 2002 – 262 с.
5. *Питерсон Дж.* Теория сетей Петри и моделирование систем. – М.: Мир, 1984 – 368 с.
6. *Котов В.Е.* Сети Петри. – М.: Наука, 1984 – 263 с.

Получено 28.02.2006