

Оптимизация схемы микропрограммного автомата Мили на FPGA

Баркалов А.А., Красичков А.А., Бойков В.А.

Abstract

Barkalov A.A., Krasichkov A.A., Boykov V.A.. Optimization of the Mealy FSM on FPGA.

The method of Mealy FSM optimization with encoding of the fields of compatible microoperations is proposed. Method is based on verticalization of initial flow-chart of algorithm that provides the total compatibility all microoperations of control algorithm. The transformation of serial hot-one code of microoperation into parallel code of microinstruction is proposed to improve the performance of a digital system with such control unit. Key words: REGISTER, FPGA, FLOWCHART, MICROOPERATION, LUT.

1. Введение

В настоящее время появилась возможность реализации цифровой системы, используя одну микросхему типа «система-на-кристалле» (SoC, system-on-chip) [1]. Как правило, произвольная логика в SoC реализуется на FPGA (field programmable gate array), которые состоят из элементов табличного типа (LUT, look-up table) [2] с ограниченным числом входов. Такое ограничение приводит к необходимости использования функциональной декомпозиции [3] систем реализуемых булевых функций, что снижает быстродействие цифровой системы. Важной частью любой цифровой системы является устройство управления, которое может быть реализовано в виде микропрограммного автомата (МПА) Мили [4]. Стремление к уменьшению числа LUT-элементов в схеме МПА при сохранении заданного быстродействия системы приводит к реализации МПА в виде многоуровневой структуры [5]. Одним из таких методов является кодирование полей совместимых микроопераций [6], при котором система микроопераций реализуется на дешифраторах. В настоящей работе предлагаются две структуры схем МПА Мили, основанных на вертикализации [6] исходного алгоритма управления и позволяющих уменьшить число дешифраторов до одного.

2. Постановка задачи

Пусть алгоритм управления цифровой системы представлен в виде граф-схемы алгоритма (ГСА) Г [4], вершины которой содержат микрооперации (МО) из множества

$Y = \{y_1, \dots, y_N\}$ и логические условия (ЛУ) из множества $X = \{x_1, \dots, x_L\}$. Микрооперации $y_i, y_j \in Y$ называются совместимыми микрооперациями (СМО), если они всегда находятся в разных вершинах ГСА Γ . Пусть $\Pi_Y = \{Y_1, \dots, Y_I\}$ - разбиение множества Y на классы СМО. Закодируем микрооперации $y_n^i \in Y_i$ двоичными кодами $K(y_n^i)$ разрядности $G_i = \lceil \log_2(|Y_i| + 1) \rceil$ и используем элементы множества $Z = \{z_1, \dots, z_G\}$ для такого кодирования, где $G = G_1 + G_2 + \dots + G_I$. В этом случае МПА Мили реализуется в виде PD-автомата [5] (рис. 1), в котором схема D состоит из I дешифраторов и i-й дешифратор реализует микрооперации $y_n^i \in Y^i$ ($i = 1, \dots, I$).

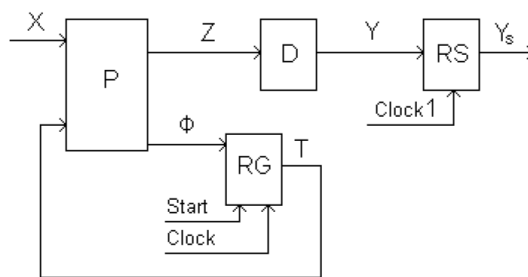


Рисунок 1. МПА Мили в виде PD-автомата.

Здесь регистр RG хранит коды $K(a_m)$ состояний МПА Мили $a_m \in A = \{a_1, \dots, a_M\}$ разрядности $R = \lceil \log_2 M \rceil$, регистр RS необходим для организации устойчивой работы цифровой системы [7]. Схема P реализует систему функций возбуждения триггеров RG

$$\Phi = \Phi(T, X) \quad (1)$$

и систему функций

$$Z = Z(T, X), \quad (2)$$

использующих для формирования микроопераций Y

$$Y = Y(Z). \quad (3)$$

Здесь $T = \{T_1, \dots, T_R\}$ - множество внутренних переменных, используемых для кодирования состояний $a_m \in A$. По сигналу Start в RG записывается код начального состояния $a_1 \in A$, по сигналу Clock происходит смена кодов состояний в RG, по сигналу Clock1 микрооперации Y записываются в RS и стабильные МО Y_s поступают в различные блоки цифровой системы.

Основным достоинством PD-автомата по сравнению одноуровневой схемой P-автомата, где

$$\Phi = \Phi(T, X), \quad (4)$$

$$Y = Y(T, X), \quad (5)$$

является уменьшение числа функций, зависящих от переменных Т и Х. Как правило, для реализации системы (5) требуется гораздо больше LUT-элементов, чем для реализации системы (3). Кроме того, дешифраторы являются регулярными библиотечными элементами, что упрощает процесс проектирования МПА. Недостатком PD-автомата по сравнению с Р-автоматом является потенциальная возможность увеличения времени такта за счет введения дополнительного уровня.

В настоящей работе предлагается ряд структур схемы PD-автомата, основанные на вертикализации [6] исходной ГСА Г .

3. Основная идея метода оптимизации

Граф-схема алгоритма Г называется вертикальной ГСА (ВГСА), если каждая ее операторная вершина содержит не более одной микрооперации. [6] В этом случае все микрооперации $y_n \in Y$ являются попарно совместимыми и для их кодирования достаточно

$$Q = \lceil \log_2 N \rceil \quad (6)$$

элементов множества Z.

В общем случае произвольная ГСА Г не является вертикальной, однако полная совместимость МО может быть достигнута за счет вертикализации ГСА Г и формирования вертикализированной ГСА $v(\Gamma)$. [6] Процесс вертикализации заключается в расщеплении операторной вершины $b_k \in B$ на n_k операторных вершин, где $n_k = |Y(b_k)|$. Здесь B – множество операторных вершин ГСА Г, $Y(b_k) \subseteq Y$ – набор микроопераций, содержащихся в вершине $b_k \in B$, $|B| = K$.

На рис.2а показан фрагмент ГСА Г₁ и два способа отметки состояний вертикализированной ГСА $v(\Gamma_1)$ (рис. 2b, 2c).

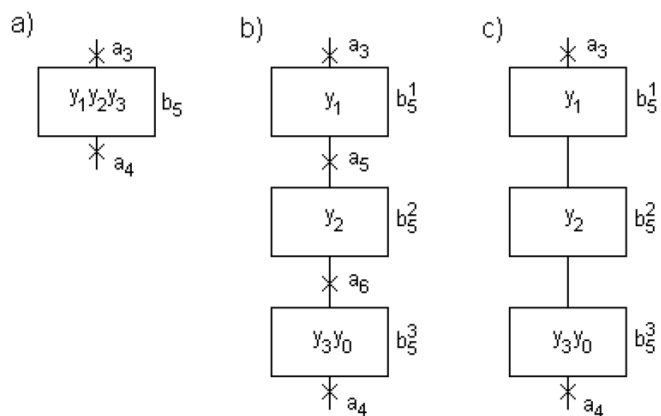


Рисунок 2. Фрагмент ГСА и два способа отметки состояний вертикализированной ГСА.

В обоих случаях вершине b_5 соответствует последовательность вершин $\beta_5 = \langle b_5^1, b_5^2, b_5^3 \rangle$, но в одном случае (метод U_1) вводятся новые состояния МПА Мили

(рис.2b), а во втором случае (метод U_2) сохраняется исходная отметка (рис. 2c). Назначение сигнала y_0 рассмотрим несколько позже.

Автомат Мили, синтезированный по ВГСА $v(\Gamma)$, требует больше времени для выполнения алгоритма управления, чем МПА, синтезированный по ГСА Γ . Кроме того, последовательное выполнение микроопераций $y_n \in Y(b_k)$ не всегда возможно из-за возможной зависимости МО по данным [6]. Первый недостаток можно уменьшить, а второй – устранить, для чего в работе предлагается преобразование последовательности β_k в параллельный код $Y(b_k)$ и запуск операционного автомата системы только после формирования всех микроопераций $y_n \in Y(b_k)$ ($k=1, \dots, K$). Признаком окончания формирования микроопераций $y_n \in Y(b_k)$ служит сигнал $y_0=1$. Параллельный код последовательно загружается в регистр RS, для чего используется возможность локальной синхронизации триггеров, входящих в состав SoC [1].

4. Полученные структуры МПА Мили

На рис. 3 показана структурная схема PD₁-автомата Мили, основанная на методе U_1 . В этом случае схема P реализует системы (1), (2) и функцию $y_0 = y_0(T, X)$, для чего требуется

$$t_1 = R_1 + Q + 1 \quad (7)$$

выходов. Здесь $R_1 = \lceil \log_2 M_1 \rceil$, где M_1 число состояний PD₁-автомата

$$M_1 = M + \sum_{k=1}^K (n_k - 1). \quad (8)$$

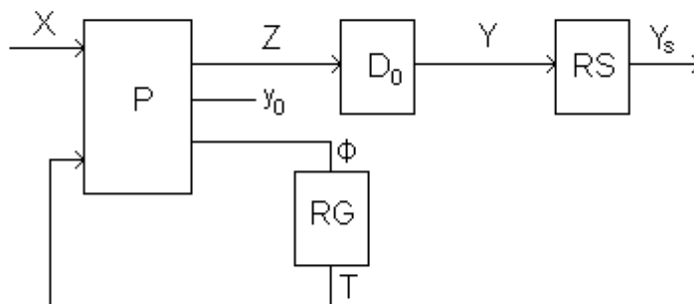


Рисунок 3. Структурная схема PD₁-автомата Мили.

Дешифратор D_0 имеет Q входов и N выходов. Схема синхронизации RG, RS и операционного автомата здесь не показана. Отметим, что в общем случае $M_1 > M$ и $R_1 \geq R$.

На рис. 4 показана структурная схема PD₂-автомата Мили, основанная на методе U_2 .

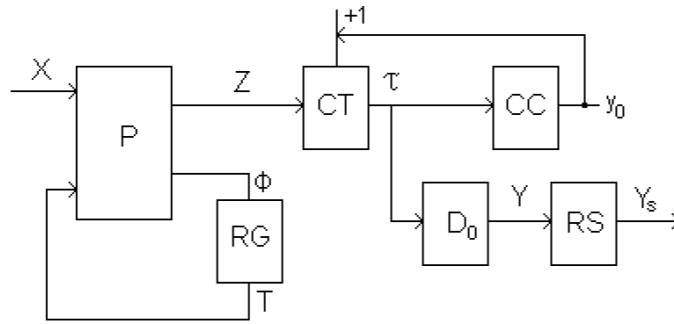


Рисунок 4. Структурная схема PD₂-автомата Мили.

В этом случае схема P реализует системы (1), (2), комбинационная схема CC (Combinational circuit) формирует сигнал

$$y_0 = y_0(\tau), \quad (9)$$

где τ – выходы счетчика CT (Counter), формирующего коды микроопераций $y_n \in Y(b_k)$ ($k=1, \dots, K$). Микрооперации формируются дешифратором D₀ в виде

$$Y = Y(\tau), \quad (10)$$

а функции Z используются как функции возбуждения CT.

Теперь схема P имеет

$$t_2 = R + Q_1 \quad (11)$$

выходов, где $Q_1 \geq Q$ - число переменных, необходимое для уникального кодирования микроопераций $y_n^k \in Y(b_k)$ ($k=1, \dots, K$).

Параметр Q_1 может быть найден следующим образом. Из принципа организации PD₂-автомата следует, что внутри каждой последовательности β_k коды микроопераций должны возрастать на единицу. Например, если $Q_1 = 4$, то для последовательности β_5 при коде $K(y_1^5) = 0100$ код $K(y_2^5) = 0101$ и код $K(y_3^5) = 0110$. Таким образом, кодированию подлежат микрооперации из множества Y_0 , объединяющего микрооперации уникальных микрокоманд (наборов микроопераций). Это следует из того, что при равенстве $Y(b_i) = Y(b_j)$ одинаковые микрооперации из множеств $Y(b_i)$ и $Y(b_j)$ могут иметь одинаковые коды. Пусть ГСА Г имеет J уникальных наборов микроопераций, тогда множество Y_0 ВГСА $v(\Gamma)$ содержит

$$N_0 = \sum_{j=1}^J n_j \quad (12)$$

элементов. Следовательно, $Q_1 = \lceil \log_2 N_0 \rceil$.

Очевидно, что схема управления регистром RS в PD₂-автомате сложнее, чем в PD₁-автомате, так как выходы дешифратора, соответствующие микрооперации $y_n \in Y$ с разными верхними индексами должны быть объединены по схеме «ИЛИ» или с

использованием трехстабильных буферов. Отметим, что рост числа кодирующих переменных Z в PD_2 -автомате компенсируется уменьшением числа входов схемы P по сравнению с PD_1 -автоматом Мили. Кроме того, принцип кодирования микроопераций в PD_2 -автомате позволяет использовать методы оптимизации схемы CC , аналогичные методам оптимизации композиционных микропрограммных устройств управления [5].

5. Заключение

Предлагаемые в работе методы организации схем автоматов Мили позволяют уменьшить число LUT-элементов схемы по сравнению с PD -автоматом, основанным на кодировании полей совместимых микроопераций. Предлагаемые методы основаны на вертикализации исходной граф-схемы алгоритма, что позволяет использовать только один дешифратор для реализации системы микроопераций. Проведенные авторами исследования показали, что применение PD_1 -автомата эффективно, если $R_1 = R$, то есть при совпадении числа переменных обратной связи автоматов. В противном случае более эффективно применение модели PD_2 -автомата Мили. Отметим, что конкретный выигрыш зависит от характеристик самой ГСА, определяющих параметры t_1 и t_2 . Как показали исследования авторов, лучшая в конкретных условиях модель автомата Мили (PD_1 или PD_2) позволяет на 24-32% уменьшить число LUT-элементов по сравнению с PD -автоматом.

Уменьшение аппаратных затрат в данном случае сопровождается увеличением времени выполнения алгоритма управления по сравнению с PD -автоматом. Предложенный в работе метод преобразования последовательности микроопераций в параллельный код, соответствующий микрооперациям для конкретной операторной вершины ГСА, позволяет уменьшить влияние вертикализации на производительность цифровой системы.

Литература

1. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики.– Петербург: БХВ – 2002. – 636 с.
2. [Salc98] Salcic Z. VHDL and FPGAs in digital systems design, prototyping and customization. –Kluwer Academic Publishers, 1998, 576 p.
3. [Sas99] Sasao T. Switching theory for logic synthesis.–Kluwer Academic Publishers, 1999, 355p.
4. [Bar94] S.Baranov, Logic Synthesis for Control Automata, Kluwer Academic Publishers, 1994, 405p.
5. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах.– Донецк: ДонНТУ, 2002. – 262 с.

6. Баркалов А.А., Палагин А.В. Синтез микропрограммных устройств управления – Киев: ИК НАН Украины, 1997.- 136 с.
7. Баркалов А.А. Синтез операционных устройств. – Донецк: ДонНТУ, 2003. – 306 с.

Abstract

Barkalov A.A., Krasichkov A.A., Boykov V.A.. Optimization of the Mealy FSM on FPGA.

The method of Mealy FSM optimization with encoding of the fields of compatible microoperations is proposed. Method is based on verticalization of initial flow-chart of algorithm that provides the total compatibility all microoperations of control algorithm. The transformation of serial hot-one code of microoperation into parallel code of microinstruction is proposed to improve the performance of a digital system with such control unit. Key words: REGISTER, FPGA, FLOWCHART, MICROOPERATION, LUT.

Аннотация

Баркалов А.А., Красичков А.А., Бойков В.А. Оптимизация схемы микропрограммного автомата Мили на FPGA. Предложен метод оптимизации автомата Мили кодированием полей совместимых микроопераций. Метод основан на вертикализации исходной граф-схемы алгоритма, которая обеспечивает полную совместимость всех микроопераций управляющего алгоритма. Для улучшения быстродействия цифровой системы на базе такого автомата, предложено преобразование унитарного кодирования микроопераций в параллельный код микроинструкций. Ключевые слова: РЕГИСТР, FPGA, ГРАФ-СХЕМА, МИКРООПЕРАЦИЯ, КЛБ.

Анотація

Баркалов О.О., Красічков А.А., Бойков В.А. Оптимізація схеми мікропрограмного автомата Мілі на FPGA. Запропоновано метод оптимізації автомата Мілі кодуванням полей сумісних мікрооперацій. Метод засновано на вертикалізації вихідної граф-схеми алгоритму, котра забезпечить повну сумісність усіх мікрооперацій керуючого алгоритму. Для поліпшення швидкодії цифрової системи на базі такого автомату, запропоновано перетворення унітарного кодування мікрооперацій до паралельного коду мікроінструкцій. Ключові слова: РЕГІСТР, FPGA, ГРАФ-СХЕМА, МИКРООПЕРАЦИЯ, КЛБ.