

УДК 0.043

**А.Н. Галдин, А.Н. Мирошкин**  
*Донецкий национальный технический университет*

## **ИССЛЕДОВАНИЕ ПРОБЛЕМЫ СНИЖЕНИЯ АППАРАТУРНЫХ ЗАТРАТ ПРИ РЕАЛИЗАЦИИ МИКРОПРОГРАММНЫХ УСТРОЙСТВ УПРАВЛЕНИЯ В FPGA-БАЗИСЕ**

### **Аннотация**

*Галдин А.Н., Мирошкин А.Н. Исследование проблемы снижения аппаратурных затрат при реализации микропрограммных устройств управления в fpga-базисе. Рассмотрены проблема снижения аппаратурных затрат, возникающая при реализации микропрограммных устройств управления, широко используемые методы формирования операционной части микрокоманд, а также модификация метода кодирования полей совместимых микроопераций. Проведена сравнительная характеристика методов, освещены их преимущества, недостатки, предложены условия их оптимального использования, приведен пример, показывающий преимущество использования модификации метода кодирования полей совместимых операций при реализации микропрограммного устройства управления в базе программируемой пользователем вентильной матрицы.*

**Ключевые слова:** *снижение аппаратурных затрат, микропрограммные устройства управления, микрокоманда, модификация метода кодирования полей совместимых микроопераций, программируемая пользователем вентильная матрица.*

**Введение.** Процесс проектирования микропрограммных устройств управления (МПУУ) [2, 3] сопровождается рядом проблем, одной из которых является снижение аппаратурных затрат. Данная проблема остается актуальной особенно при использовании базиса программируемых логических интегральных схем (ПЛИС) [1], которые на данный момент распространены, в частности, широко используется одна из архитектурных разновидностей ПЛИС, а именно программируемая пользователем вентильная матрица (ППВМ) или FPGA. Современные микросхемы FPGA дают преимущество в конфигурации топологий связей в процессе использования за счет наличия в своей архитектуре блоков встроенной памяти (БВП), которые позволяют микросхеме перепрограммироваться и менять топологию в любой момент процесса использования. Недостатком таких микросхем является относительно небольшой объем БВП, что делает актуальной задачу сокращения разрядности слова микропрограммы.

Задачей данного исследования является рассмотрение широко используемых методов снижения аппаратурных затрат при реализации МПУУ.

Целью работы является снижение аппаратурных затрат при проектировании микропрограммного устройства управления с использованием модифицированного метода кодирования полей совместимых операций, позволяющего уменьшить аппаратурные затраты, необходимые для реализации МПУУ в FPGA-базисе.

Известно немалое количество методов оптимизации, которые позволяют уменьшить аппаратурные затраты на разных участках схемы МПУУ. К примеру, метод сегментации микрокоманд предусматривает разделение микропрограммы на сегменты, что приводит к экономии памяти до 20 % [4]. Некоторые методы основываются на преобразованиях с целью снижения аппаратурных затрат еще на уровне формирования алгоритма. К примеру, в методе минимизации емкости управляющей памяти при вертикализации граф-схемы алгоритма (ГСА) предлагается свести исходную ГСА к вертикальному виду, т.е. к такой ГСА, в которой каждая вершина будет содержать всего одну микрооперацию. Исследования, проведенные авторами, показали, что данный метод позволяет в большинстве случаев позволяет уменьшить емкость постоянного запоминающего устройства (ПЗУ) при незначительном увеличении времени выполнения алгоритма. Экономии аппаратурных затрат также можно добиться используя методы кодирования операционной части микрокоманды (МК).

**Организация операционной части микрокоманды.** Кодирование микрокоманд (МК) может быть выполнено согласно одной из трех стратегий [4]:

- горизонтальное (унитарное) кодирование;
- вертикальное (максимальное) кодирование;
- горизонтально-вертикальное кодирование(кодирование полей совместимых операций).

В случае горизонтального кодирования, каждой микрооперации (МО) ставится в соответствие отдельный бит в операционной части МК. В связи с этим, данный метод кодирования называют также прямым управлением или неупакованной организацией МК [50]. В этом случае, разрядность поля операционной части МК определяется количеством уникальных (отличных друг от друга) МО в составе исходного алгоритма. Из этого следует, что схема формирования МО в случае унитарного кодирования МК состоит лишь из регистра микрокоманд (РМК), разрядностью, определенной разнообразностью микроопераций. Преимуществом этой схемы является отсутствие дополнительных уровней, что приводит к снижению аппаратурных затрат, связанных с формированием МК. Недостатком же является прямо пропорциональная зависимость между количеством уникальных микроопераций и разрядностью РМК. Таким образом, несмотря на простоту реализации, данный метод кодирования приводит к значительным аппаратурным затратам в случае реализации алгоритмов с большим количеством уникальных МО.

В случае вертикального кодирования, поле операционной части МК представляется кодом определенной последовательности МО. Разрядность поля зависит от количества МК, по закону:

$$N = \lceil \log_2 T \rceil, \quad (1)$$

где  $T$  – количество микрокоманд, а  $N$  – разрядность операционной части МК.

Таким образом, по отношению к предыдущему методу, незначительное увеличение количества МО не всегда приводит к увеличению разрядности операционной части МК. Этот факт является преимуществом данного подхода. Однако, недостатком максимального кодирования является добавление двух уровней в структурную схему, а именно уровней дешифрации полученного из поля МК кода. С увеличением количества МК растет разрядность не только РМК, но и дешифратора, а также второй и третий уровни в схеме вносят дополнительную задержку в работе.

Своеобразным компромиссом двух вышеприведенных методов является метод кодирования полей совместимых микроопераций. Совместимыми называются такие МО, которые никогда не встречаются вместе в одной МК. Принцип данного метода состоит в том, что все множество МО разбивается на подмножества. Члены подмножества кодируются максимально, а сами подмножества между собой – унитарно. Таким образом, в операционной части МК для каждого подмножества выделяется свое поле разрядностью, определяемой кодированием каждого подмножества, а разрядность всей операционной части поля определяется как сумма разрядностей полей всех подмножеств. Выходы каждого поля операционной части МК подключаются к отдельному дешифратору, разрядность которого равна разрядности соответствующего кода МО. В результате, структурная схема имеет только один дополнительный уровень, что приводит к лучшему быстродействию, нежели при максимальном кодировании, и приводит к меньшим аппаратурным затратам относительно РМК, нежели при унитарном кодировании.

Однако, метод имеет свои ограничения [5]:

- каждая МО может принадлежать только одному подмножеству
- МО, которые входят в одну и ту же МК должны быть в разных подмножествах.

В связи с приведенными ограничениями, преимущества данного метода, указанные выше, имеют относительный характер. Ограничения могут привести к некоторому количеству подмножеств, которым будет принадлежать всего одна МО, что увеличит разрядность операционной части МК.

**Идея модифицированного метода кодирования полей совместимых операций.** В статье [6] рассматривается подход, позволяющий преодолеть эти недостатки. Основная идея предлагаемого метода заключается в преодолении ограничениями приведенного метода. Для дальнейшего изложения материала введем несколько понятий.

Если существуют такие подмножества МО  $P_i$  и  $P_j$ , для которых справедливо следующее выражение:

$$P_i \cap P_j = y_k, \quad (2)$$

где  $y_k$  – определенная МО, то такая МО называется *зависимой*, причем она может являться пересечением любого количества подмножеств. Множество, которое формирует такую МО, определяется в зависимости от наличия других МО, формируемых в данный момент.

Если существует такая МО  $y_k$ , которая принадлежит только одному подмножеству  $P_i$ , то она называется *независимой*, поэтому источник ее формирования не зависит от присутствия других МО в той же вершине граф-схемы алгоритма.

Если существуют такие МО  $y_i, y_j$ , которые встречаются в МК только вместе, то они называются *связанными*, причем таких МО может быть любое количество.

Согласно предлагаемому алгоритму, по подмножествам распределяются в первую очередь связанные МО, что обеспечивает первичное уменьшение разрядности МК. Затем распределяются все остальные МО по классическому методу. В последнюю очередь распределяются зависимые МО. В случае необходимости, копии зависимых МО включаются в несколько подмножеств. В виду того, что изначально размещаются только связанные МО и зависимые МО могут располагаться в нескольких подмножествах, групп с одной МО не возникает, что позволяет преодолеть ограничения классического метода.

Данная модификация может быть эффективна применительно к ПЛИС ППВМ. Рассмотрим его эффективность на примере устройства серии Stratix фирмы Altera. Микросхема содержит блоки встроенной памяти, которые отличаются гибкостью управления. К примеру, один из видов блоков памяти имеет общий объем в 576 бит и может хранить разное количество слов в зависимости от их разрядности. Блок памяти имеет 7 режимов:  $512 \times 1$ ,  $256 \times 2$ ,  $128 \times 4$ ,  $64 \times 8$ ,  $64 \times 9$ ,  $32 \times 16$ ,  $32 \times 18$ , где первое значение определяет количество слов памяти, а второе – разрядность одного слова.

Пусть, имеется исходная ГСА, содержащая 50 различных друг от друга МО. Фрагмент ГСА без условных вершин, которые не влияют на формирование операционной части МК, приведен на рис. 1.

Допустим, путем использования классического метода кодирования полей совместимых операций получена микропрограмма (МП), состоящая из 40 МК, каждая из которых имеет разрядность 11 бит. Тогда, запись МП в память TriMatrix устройства Stratix [7] может быть выполнена двумя способами, приведенными на рис. 2. Полученная разрядность МК и их количество не позволяют разместить всю МП в одном блоке памяти. На рис. 2а блоки памяти используются в режиме  $64 \times 9$ , одна часть микрокоманды располагается в блок памяти. На рис. 2б блоки памяти используются в режиме  $32 \times 16$ , МП делится на части и размещается в двух блоках памяти.

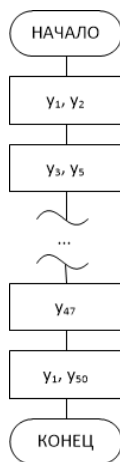


Рисунок 1 – Пример ГСА

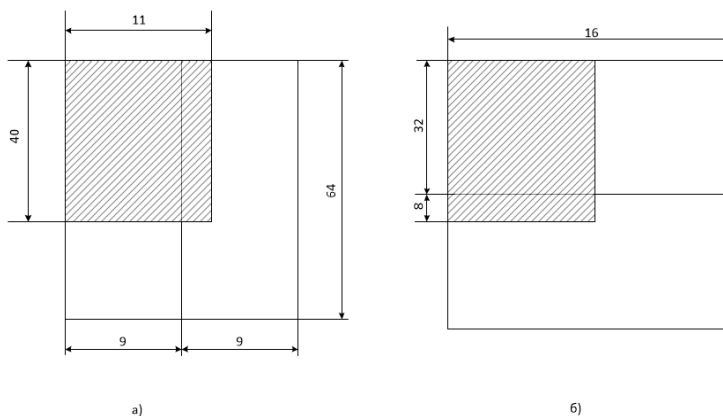


Рисунок 2 – Принципы размещения микропрограммы в блоках памяти устройства при использовании классического метода кодирования полей совместимых операций.

Из рис. 2 видно, что блоки встроенной памяти используются не полностью. Для них можно подсчитать усредненное значение коэффициента использования по формуле:

$$h = \frac{N_{mp} * T_{mc}}{T}, \quad (3)$$

где  $N_{mp}$  – количество МК в МП,  $T_{mc}$  – разрядность одной МК,  $T$  – количество ячеек в блоке памяти, максимально доступное в данном режиме.

Таким образом, чем больше значение параметра  $h$ , тем рациональнее используются ресурсы памяти микросхемы. Следует заметить, что о рациональности идет речь только исходя из аппаратурных затрат. Блок памяти может быть заполнен на 100% несколькими способами, один из которых можно выбрать исходя из соображений о других параметрах, к примеру, бо воздействии.

Рассчитаем параметр  $h$  для случаев, изображенных на рис. 2:

$$h_1 = \frac{40 \cdot 11}{512 \cdot 2} = 0,43, \quad (4)$$

$$h_2 = \frac{40 \cdot 11}{576 \cdot 2} = 0,38 \quad (5)$$

Исходя из (4) и (5), можно сделать вывод о нерациональном использовании аппаратных ресурсов, т.к. в обоих случаях не заполнено даже половины всего пространства памяти, но следует заметить, что реализация первого случая является более оптимальной для классического метода.

Применим модификацию метода кодирования полей совместимых микроопераций, которая позволяет преодолеть ограничения классического метода. Пусть, ГСА, приведенная на рис. 1, построена так, что с помощью предложенной модификации, получена МП, состоящая из такого же количества МК, однако разрядность одной МК уменьшилась до 9 бит за счет отсутствия подмножеств с одной МО, которые не образуются при использовании данного метода. Исходя из этого, можно использовать один блок ПАМЯТИ в режиме 64x9. Принцип размещения изображен на рис. 3.

Рассчитаем коэффициент рациональности для данного случая:

$$h_3 = \frac{40 \cdot 9}{576} = 0,625 \quad (6)$$

Сравним коэффициенты (4), (5) и (6). Можно сделать вывод об эффективности предложенного метода касательно аппаратурных затрат. Использование данной модификации привело к сокращению количества используемых блоков памяти вдвое.

**Выводы.** Рассмотрены широко используемые методы кодирования операционной части микрокоманд, модификация метода совместимых микроопераций, которая имеет ряд преимуществ в снижении аппаратурных затрат. Метод эффективен только в граничных случаях, т.е. в случаях, когда незначительное уменьшения разрядности может привести к существенному снижению аппаратурных затрат. Целью дальнейшего исследования является формализация алгоритма данного метода, а также получения точных условий его рационального использования.

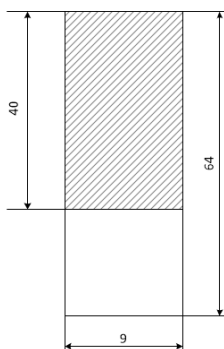


Рисунок 3 – Принцип размещения микропрограммы в блоке памяти устройства при использовании модификации метода кодирования полей совместимых операций.

### Список литературы

1. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.
2. Baranov S. Logic Synthesis for Control Automata – Boston: Kluwer Academic Publishers, 1994 – 312 pp.
3. Баркалов А.А., Бабаков Р.М., Ахмад Бадер. Исследование аппаратных характеристик автомата Мили с кодированием фрагмента микроопераций по VHDL-моделям // Искусственный интеллект. – 2007, №1, – С. 117–122.
4. Баркалов А.А., Палагин А.В. Синтез микропрограммных устройств управления. – Киев, 1997. – 135 с.
5. Модифікація алгоритму Шварца для оптимізації операційної частини композиційних мікропрограмних пристроїв керування з урахуванням обмежень базису FPGA. Зеленцова І.Я. Мірошкін О.М., Казачанський А.В. // Наукові праці Донецького національного технічного університету. Серія «Обчислювальна техніка та автоматизація». Випуск 147 (16). / Редкол. Башков Є.О. (голова) та ін. – Донецьк: ДонНТУ, 2009 г. – С. 105–115.
6. Модификация кодирования полей совместимых микроопераций. Баркалов А.А., Визор Я.Е., Мирошкин А.Н. // Международный научный журнал «Управляющие системы и машины». Випуск 6(224). / Редкол. Чарчиян С.П., Савенко Н.И. – Киев: Международный научно-учебный центр информационных технологий и систем НАН и МОН Украины, 2009 г. – С. 50–55
7. Stratix IV FPGA: High Density, High Performance and Low Power. Интернет-ресурс. Режим доступа:  
<http://www.altera.com/devices/fpga/stratix-fpgas/stratix-iv/stxiv-index.jsp>