

УДК 004.051 + 53.097 + 537.39

**И.А. Самсонова, А.Н. Мирошкин**

Донецкий национальный технический университет, г. Донецк  
кафедра компьютерной инженерии

## **ИССЛЕДОВАНИЕ ВЛИЯНИЯ МЕТОДИКИ НАЗНАЧЕНИЯ АДРЕСОВ МИКРОКОМАНД НА ХАРАКТЕРИСТИКИ СХЕМ УСТРОЙСТВ УПРАВЛЕНИЯ**

### *Аннотация*

*Самсонова И.А., Мирошкин А.Н. Исследование влияния методики назначения адресов микрокоманд на характеристики схем устройств управления. В статье рассматривается методика назначения адресов микрокоманд, использование которой приводит к снижению энергопотребления. Предлагаемая методика заключается в назначении адресов таким образом, чтобы уменьшить количество термов в системе функций возбуждения памяти блока формирования адресов микрокоманд, что при тех же временных характеристиках получаемой схемы управляющего устройства приводит к снижению потребляемой мощности.*

***Ключевые слова:** мощность, энергопотребление, ПЛИС, устройства управления, алгоритм, метод назначения адресов, триггер.*

**Постановка проблемы.** Из-за стремительного развития цифровых технологий в наше время требования к их производительности и функциональности быстро возрастают. При решении некоторых вопросов разработчики встали перед проблемой использования более эффективных методов распределения энергии. Сложность современных приложений приводит к необходимости снижения энергопотребления систем путём применения оптимальных решений в процессе их проектирования.

На всем протяжении своего существования активные полупроводниковые приборы практически не изменились. Вероятно, единственным их значительным изменением было уменьшение минимальных размеров элементов в соответствии с законом Мура. Но со временем дальнейшее масштабирование и другие способы повышения рабочих характеристик кремниевых транзисторов могут привести к достижению физических пределов создаваемых структур. С уменьшением размеров транзисторов растёт и ток

утечки отключенного транзистора. Таким образом, транзистор практически никогда не бывает полностью отключен. Увеличение тока приводит к повышению температуры прибора, а с повышением температуры растет и ток. Круг замкнулся, и в режиме покоя возможен тепловой пробой транзистора[1]. Кроме этих проблем существует масса других, так же усложняющих дальнейшую работу схемы.

**Цель статьи** — исследование влияния методики назначения адресов микрокоманд на характеристики схем устройств управления.

**Постановка задачи исследования.** Снижения энергопотребления возможно не только аппаратным методом, но и программным. Большинство крупных компаний по разработке программного обеспечения давно внедряют наработки по решению данного вопроса в свои продукты. Это как специальные режимы работы, так и определенные надстройки и параметры. Снижением энергопитания занимаются также и независимые разработчики, создавая алгоритмы, при реализации которых будет затрачиваться меньше мощности. Один из программных способов рассматривается в данной работе.

### **Анализ современного элементного базиса программируемых логических интегральных схем (ПЛИС)**

Основной особенностью ПЛИС архитектуры FPGA, которая широко используется для реализации управляющих устройств, является наличие трех типов элементов, конфигурация которых может изменяться разработчиком при проектировании конкретного устройства. Этими элементами являются:

- блоки ввода/вывода (БВВ);
- конфигурируемые логические блоки (КЛБ);
- межсоединения.

Все FPGA имеют практически одинаковую базовую структуру. Исходя из отличительных особенностей различных серий микросхем, можно выделить типы:

- D-триггерная архитектура (XC3000 - XC4000);
- архитектура с использованием КЛМ и ИВВ (XC5000);
- однокристалльная архитектура (Virtex, Spartan).

Различаются они главным образом максимальным числом и сложностью КЛБ и БВВ.

Базовая архитектура FPGA микросхем фирмы Xilinx имеют одинаковую архитектуру и различаются лишь параметрами, зависящими от напряжения

питания (потребляемой мощностью, быстродействием, уровнями входных и выходных сигналов).



Рис. 1 — Блок ввода/вывода FPGA

На рисунке 1 показана общая структура БВВ базовых FPGA. Каждый БВВ включает программируемый входной канал и программируемый выходной буфер. Как видно из рисунка, в качестве триггера используется D-триггер. Буферизованный входной сигнал поступает на вход триггера и на один из входов программируемого мультиплексора. Выходной сигнал триггера поступает на другой вход мультиплексора. Наличие программируемого мультиплексора дает возможность пользователю выбрать либо прямой ввод сигнала, либо ввод с запоминанием на триггере.

При D-триггерной архитектуре общая структура кристалла FPGA серии состоит из БВВ, КЛБ, трехстабильных буферов с доступом к горизонтальным длинным линиям, глобального тактового буфера. Особенностью данных FPGA является то, что триггером в КЛБ могут быть только D-триггер, запоминающий состояние данных на входе по переднему фронту тактового сигнала.

При структуре FPGA с использованием КЛМ и БВВ вместо матрицы КЛБ на кристалле FPGA имеется матрица конфигурируемых логических модулей (КЛМ). Каждый КЛМ состоит из КЛБ и локальной трассировочной матрицы (ЛТМ), позволяющей конфигурировать связи внутри КЛМ.

Семейство FPGA Virtex позволяет реализовывать высокопроизводительные, цифровые устройства на одном кристалле. Запоминающими элементами в каждой секции КЛБ Virtex могут конфигурироваться как динамические триггеры (чувствительные к фронту сигнала) D-типа, так и триггеры-защелки, чувствительные к уровню сигнала. D-вход триггера может управляться либо от функционального генератора в рамках той же секции КЛБ, либо непосредственно от входов данной секции КЛБ, минуя функциональные генераторы[2].

Кроме FPGA также распространены ПЛИС архитектуры CPLD. Каждая микросхема представляет собой подсистему, состоящую из множества функциональных блоков (ФБ) и БВВ, соединенных переключательной матрицей. Функциональный блок состоит из независимых макроячеек (МЯ). Любая МЯ в микросхеме может выполнять как логическую комбинаторную, так и регистровую функции. Запоминающий элемент в МЯ может быть сконфигурирован или как D-триггер, или как тактируемый триггер-защелка, или же он может отсутствовать. В последнем случае сигнал логической функции пропускается напрямую для использования в других МЯ[3].

Из сказанного выше можно сделать вывод, что по большей части в схемах ПЛИС используются именно триггеры D-типа, что играет большую роль в создании методов кодирования.

### **Исследование различных методик назначения адресов микрокоманд**

Существуют различные варианты назначения адресов микрокоманд для микропрограммных устройств управления. Каждый способ приводит к различным выражениям функций возбуждения памяти и функций выходов. Можно сделать вывод, что сложность комбинационной схемы автомата во многом зависит от выбранного способа назначения адресов.

Задача назначения адресов является одной из основных задач канонического метода структурного синтеза автоматов. Назначение заключается в установлении взаимно-однозначного соответствия между множеством  $A = \{a_1, \dots, a_M\}$  состояний автомата и множеством R-компонентных векторов  $\{K_1, \dots, K_M\}$ ,  $K_m = (e_{m1}, \dots, e_{mR})$ , где  $e_{m\gamma}$  - состояние  $\gamma$ -го элемента памяти,  $\gamma = 1, \dots, R$ . Если  $e_{m\gamma} \in \{0, 1\}$ , т.е. алфавит состояний элементов памяти двоичный,  $R \geq \lceil \log_2 M \rceil$ .

Переход автомата из одного состояния в другое осуществляется за счет изменения состояний элементов памяти. Если автомат переходит из состояния  $a_m$  с кодом 0101 в состояние  $a_s$  с кодом 1001, то это означает, что триггер  $T_1$  переходит из состояния 0 в состояние 1, триггер  $T_2$  - из состояния 1 в состояние 0, а состояния триггеров  $T_3$  и  $T_4$  не изменяются.

При функционировании автомата могут появиться так называемые состязания. Элементы памяти имеют различные, хотя и достаточно близкие, времена срабатывания, а также различны задержки сигналов возбуждения, поступающих на входные каналы элементарных автоматов по логическим цепям неодинаковой длины. Если при переходе автомата из одного состояния в другое должны изменить свои состояния сразу несколько запоминающих элементов, то между ними начинаются состязания. Поэтому в процессе перехода из состояния  $a_m$  в состояние  $a_s$  под действием входного сигнала  $z_f$  автомат может оказаться в некотором промежуточном состоянии  $a_k$  или  $a_l$  в зависимости от того, какой элемент памяти выиграет состязания. Если затем при том же входном сигнале автомат из  $a_k$  и  $a_l$  перейдет в состояние  $a_s$ , то такие состязания являются допустимыми, или некритическими. Если же в этом автомате есть переход, например, из  $a_k$  в  $a_j \neq a_s$  под действием того же сигнала  $z_f$ , то автомат может перейти в  $a_j$ , а не в  $a_s$  и правильность его работы тем самым будет нарушена. Такие состязания называются критическими состязаниями или гонками. При кодировании состояний гонки должны быть устранены. Кодирование с устранением гонок называется противогоночным.

Одни из способов ликвидации гонок состоит в тактировании входных сигналов автомата импульсами определенной длительности. Предполагается, что кроме входных каналов  $x_1, \dots, x_L$  имеется еще один канал  $p$  от генератора синхроимпульсов, по которому поступает сигнал  $p = 1$  в момент прихода

импульса и  $p = 0$  при его отсутствии. В связи с этим входным сигналом на переходе  $(a_m, a_s)$  будет не  $z_f$ , а  $pz_f$ . Тогда, если длительность импульса  $t_p$  меньше самого короткого пути прохождения тактированного сигнала обратной связи по комбинационной схеме, то к моменту перехода в промежуточное состояние  $a_k$  сигнал  $p$  равен нулю и, следовательно,  $pz_f = 0$ , что исключает гонки.

Другой способ ликвидации гонок заключается во введении двойной памяти. В этом случае каждый элемент памяти дублируется, причем перепись из нижнего элемента памяти в верхний происходит в момент отсутствия тактирующего импульса ( $p = 0$ ). Сигналы обратной связи для получения функции возбуждения и функций выходов автомата снимаются с верхнего ряда триггеров. Таким образом, состояния могут возникнуть только между нижними триггерами, сигналы обратной связи не смогут измениться до тех пор, пока  $p$  не станет равным нулю. Но тогда входной сигнал  $pz_f$ , также равен нулю, а потому гонок быть не может.

Пусть  $(\alpha, \beta)$  и  $(\gamma, \delta)$  - две пары двоичных кодов длины  $R$ . Эти пары будут называться развязанными, если при некотором  $i$  ( $1 \leq i \leq R$ )  $i$ -ый разряд кода принимает значение 1 на паре  $(\alpha, \beta)$  и противоположное значение на паре  $(\gamma, \delta)$ . Если это не соблюдается, пары называются связанными. Идея противогоночного кодирования состоит в следующем: последовательно просматривая все пары переходов, имеющие хотя бы один общий входной сигнал, осуществляющий эти переходы, следует присваивать разрядам кодов такие значения, чтобы эти пары кодов состояний были развязаны. В результате развязывания пар переходов длина кода оказывается не минимальной, т.к. при введении нового разряда могут быть развязаны пары, которые были развязаны раньше. Следовательно, алгоритм противогоночного кодирования предусматривает минимизацию длины получаемых кодов состояний. Суть заключается в следующем: исключается один из разрядов кодов, в результате чего некоторые пары могут оказаться связанными. Применяют алгоритм развязывания и исключают еще один разряд, а затем снова применяют алгоритм развязывания. И так до тех пор, пока длина кода не перестанет уменьшаться. Если в результате работы алгоритма некоторые значения разрядов будут неопределенны, то их определяют произвольно.

Существует частный способ кодирования - соседнее кодирование. Условие отсутствия гонок при соседнем кодировании всегда выполняется. Суть этого кодирования состоит в том, что два состояния связанные дугой графа кодируются наборами, отличающимися лишь в одном элементе памяти. Существует несколько алгоритмов, но они не всегда поддаются формализации.

Соседнее кодирование однако не всегда возможно. Требование к графу автомата, допускающего соседнее кодирование таково:

- 1) в графе не должно быть циклов с нечетными числом вершин;
- 2) два соседних состояния не должны иметь более двух состояний, лежащих между ними.

Рассмотрим алгоритм, который используется при кодировании состояний и позволяет упростить функции возбуждения:

- 1) каждому состоянию  $a_m$  ставим в соответствие число  $N_m$  равное числу переходов в это состояние, или равное числу ветвей, входящих в это состояние на графе;
- 2) числа  $N_1, N_2, \dots, N_M$  сортируются по убыванию;
- 3) состояние  $a_i$  с наибольшим  $N_i$  кодируется кодом 0000...0;
- 4) следующие I-состояний (I - число элементов памяти) кодируется кодами с одной "1": (00..01), (00..10), ..., (10...00);
- 5) следующие I-состояний из оставшихся кодируются кодами содержащие две "1".

В результате получаем такое кодирование, при котором чем больше переходов в некоторое состояние, тем меньше единиц содержит код этого состояния. Аналогичные соображения могут быть использованы и при кодировании выходных сигналов для минимизации функции выходов[4].

### **Модификация существующих методик назначения адресов микрокоманд с учетом особенностей функционирования композиционного микропрограммного устройства управления**

Выше был сделан вывод, что D-триггеры являются основными типами триггеров, которые используются в схемах ПЛИС. Из принципа работы D-триггера следует, что каждый раз, когда на вход поступает единичное значение, триггер срабатывает и принимает значение единицы независимо от предыдущего состояния на единицу, не зависимо от того, в каком состоянии он был до этого. Соответственно, на каждое такое переключение затрачивается энергия. Логично было бы считать, что чем меньше количество таких переключений, тем меньше потребляемая при работе схемы мощность.

При данном подходе адреса микрокоманд необходимо задать так, чтобы суммарное количество единиц двоичных кодов адресов было минимально. Тогда количество переключений триггеров будет соответствовать полученному минимальному значению, а соответственно уменьшится и потребляемая мощность. Расчет количества переключений зависит не только от фактического количества единиц всех двоичных кодов адресов, но и количества ссылок на определенную микрокоманду. Если на одну и тут же вершину ссылается сразу несколько микрокоманд, то количество переключений увеличивается во столько раз, сколько ссылок на нее было сделано.

Предположим, что граф-схема состоит из 7 вершин, причем на седьмую ссылается сразу три микрокоманды, а на шестую — две. Пример схемы приведен на рисунке 2. При последовательном назначении адресов суммарное количество единиц будет равно:

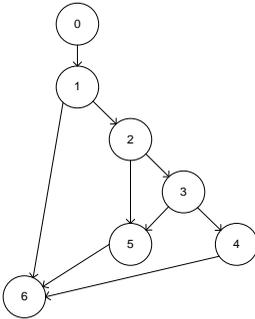


Рис. 2 — Пример граф-схемы

$S = (0+0+0)+(0+0+1)+(0+1+0)+(0+1+0)+$   
 $+(0+1+1)+(1+0+0)+2*(1+0+1)+3*(1+1+0) = 14$ , что не является оптимальным.

Для данного метода была разработана программа, которая анализирует все возможные варианты назначения адресов и подсчитывает общее количество единиц для каждой. Минимальным значением для данной схемы является 10. Рассмотрим оба варианта с последовательным и минимальным назначением адресов.

Одной из минимальных последовательностей для рассматриваемой схемы является [0 1 3 5 6 2 4]. При последовательной адресации микрокоманда, соответствующая седьмой вершине, имеет адрес 110.

Количество термов системы функций возбуждения памяти, порождаемых адресом данной МК, определяется как количество единиц в адресе, умноженное на количество переходов на данную микрокоманду, определяемое по граф-схеме алгоритма управления. Т.е.,  $R = 2 * 3 = 6$ .

При минимальной последовательности, для той же седьмой вершины, теперь с адресом 100, количество переключений будет  $R = 1*3 = 3$ , что, соответственно, в два раза меньше, а следовательно и число формируемых термов и энергия, затрачиваемая на работу конкретного триггера, в два раза меньше.

Из всего выше сказанного можно сделать вывод, что данный подход возможен в качестве одного из вариантов способов программного уменьшения энергопотребления в ПЛИС.

## Вывод

Проблема снижения энергопотребления достаточно актуальна на данный момент. Ее решение возможно как аппаратными, так и программными способами, и второй все становится все более распространенным, так как аппаратные решения проблемы связаны с определенными физическими ограничениями элементной базы. Программных способов множество, и один из них заключается в различных методах назначения адресов. Предложенный в данной работе метод может являться одним из таких, которые приведут к снижению энергии, потребляемой микросхемой, что и будет проверено в дальнейшем в магистерской работе.

### Список литературы

1. Robert Oshana. Embedded applications using integrated DSP and microprocessorcores/ Интернет-ресурс. Режим доступа: [www/URL: www.dsp-ПЛИС.com/articles/id/?2546](http://www.URL: www.dsp-ПЛИС.com/articles/id/?2546).
2. Зинченко Ю.Е. «Принцип построения ПЛИС» / Интернет-ресурс. Режим доступа: [www/URL: ftp://ftp.cs.dgtu.donetsk.ua/incoming/FPGA/for%20КC04%Е0%Е1/%СВ%Е5%ЕА%F6%Е8%Е8/%СВ%Е5%ЕА%2005%20%20%СF%F0%Е8%ЕD%F6%Е8%ЕF%FВ%20%ЕF%ЕЕ%F1%F2%F0%ЕЕ%Е5%ЕD%Е8%F%20%СF%СВ%С8%D1%20%F2%Е0%20FPGA.doc](http://www/URL: ftp://ftp.cs.dgtu.donetsk.ua/incoming/FPGA/for%20КC04%Е0%Е1/%СВ%Е5%ЕА%F6%Е8%Е8/%СВ%Е5%ЕА%2005%20%20%СF%F0%Е8%ЕD%F6%Е8%ЕF%FВ%20%ЕF%ЕЕ%F1%F2%F0%ЕЕ%Е5%ЕD%Е8%F%20%СF%СВ%С8%D1%20%F2%Е0%20FPGA.doc)
3. Зинченко Ю.Е. «CPLD ведущих мировых производителей ПЛИС»/ Интернет-ресурс. Режим доступа: [www/URL: ftp://ftp.cs.dgtu.donetsk.ua/incoming/FPGA/for%20КC04%Е0%Е1/%СВ%Е5%ЕА%F6%Е8%Е8/%СВ%Е5%ЕА%2004%20%20CPLD%20%F4%Е8%F0%ЕC%20ALTERA%20%Е8%20XILI NX.doc](http://www/URL: ftp://ftp.cs.dgtu.donetsk.ua/incoming/FPGA/for%20КC04%Е0%Е1/%СВ%Е5%ЕА%F6%Е8%Е8/%СВ%Е5%ЕА%2004%20%20CPLD%20%F4%Е8%F0%ЕC%20ALTERA%20%Е8%20XILI NX.doc)
4. Бобров М., Андриющенко Ю., Лаптева Н. «Теория автоматов» / Интернет-ресурс. Режим доступа: [www/URL: http://ofap.ulstu.ru/vt/Theory\\_of\\_automats/content.htm](http://www/URL: http://ofap.ulstu.ru/vt/Theory_of_automats/content.htm)
5. Мирошкин А.Н., Баркалов.А.А. «Синтез и исследование композиционных микропрограммных устройств управления с базовой структурой» / Интернет-ресурс. Режим доступа: [www.uran.donetsk.ua/~masters/2007/fvti/miroshkin/diss/index.htm+%&cd=1&hl=uk&ct=clnk&gl=ua&client=firefox-a](http://www/URL: www.uran.donetsk.ua/~masters/2007/fvti/miroshkin/diss/index.htm+%&cd=1&hl=uk&ct=clnk&gl=ua&client=firefox-a)