

УДК 004.3

А.А. Баркалов, Р.М. Бабаков, А.А. Егошина
Университет Зеленогурский (Польша),
Государственный университет информатики и искусственного интеллекта
a.barkalov@ie.uz.zgora.pl, cpld@mail.ru

Микропрограммное устройство управления с относительной адресацией

Предложен новый метод адресации микрокоманд в микропрограммном устройстве управления, подобный относительной адресации в универсальных микропроцессорах. Разработаны форматы микрокоманд для различных методов относительной адресации, а также структурная модель микропрограммного устройства управления с относительной адресацией.

Ключевые слова: микропрограммное устройство управления, адресация микрокоманд, структурная модель, управляющая память, аппаратные затраты.

Общая постановка проблемы

Важной составной частью цифровых систем является устройство управления (УУ), координирующее работу всех блоков системы [1-3]. Одним из способов реализации УУ является микропрограммное устройство управления (МУУ) [4], характеризующееся высокой универсальностью схемы по сравнению с автоматом с «жесткой» логикой.

В МУУ могут использоваться различные методы адресации микрокоманд, определяющие такие характеристики устройства, как быстродействие и аппаратные затраты [4, 5]. Высокие требования к современным вычислительным системам делают актуальной задачу уменьшения затрат аппаратуры в схеме МУУ, решение которой может быть достигнуто путем разработки специальных методов оптимизации, учитывающих особенности данного класса устройств управления.

Постановка задач и целей исследования

В универсальных микропроцессорах (МП) одним из множества возможных методов адресации является относительная адресация [6, 7]. Примером тому является МП Intel 8086, в котором для хранения адресного смещения выполняемой микрокоманды используется специальный регистр IP (Instruction Pointer – указатель команды). Изменение содержимого регистра IP осуществляется как автоматически при переходе к следующей по порядку команде, так и с помощью команд условных и безусловных переходов.

В МП 8086 каждая команда перехода использует в качестве аргумента численное смещение, измеряемое в байтах и являющееся целочисленной знаковой величиной. При выполнении команды перехода указанное в команде смещение арифметически суммируется с

содержимым регистра IP, в результате чего значение IP может увеличиваться (переход выполнен к команде с большим адресом) или уменьшаться (переход выполнен к команде с меньшим адресом).

Особенностью МП 8086 является поддержка т.н. «дальних» и «ближних» переходов. При использовании «дальнего» перехода смещение является двухбайтовой величиной и может лежать в диапазоне $[-32768; +32767]$ байт. При использовании «ближнего» перехода смещение лежит в диапазоне $[-128; +127]$ байт и является однобайтовой величиной (занимает вдвое меньший объем памяти по сравнению с «дальним» смещением). Таким образом, при использовании «ближних» переходов имеет место экономия памяти, однако дальность переходов невелика. Для перехода к командам, находящимся вне диапазона, доступного для «ближнего» перехода, могут быть использованы несколько последовательных «коротких» переходов, либо один «дальний». Компиляторы ассемблерных программ и программ на языках высокого уровня имеют возможность оптимизировать объем машинного кода, используя по возможности команды «ближних» переходов.

В настоящей работе предлагается использовать принцип относительной адресации при организации микропрограммных переходов в микропрограммном устройстве управления. Целью данного исследования является разработка новой структурной модели МУУ с относительной адресацией микрокоманд, схема которой обладает меньшими затратами аппаратуры по сравнению с ранее известной структурной моделью МУУ.

Организация МУУ с относительной адресацией

На рис. 1 приведена структурная модель микропрограммного устройства управления [4].

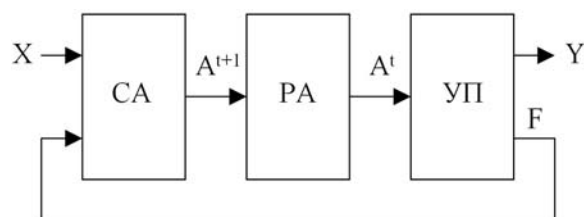


Рисунок 1 – Структурна модель мікропрограмного пристрою управління

Здесь схема адресации СА формирует адрес следующей микрокоманды A^{t+1} на основании множества логических условий X и множества адресных полей F, поступающих из управляющей памяти УП. УП обычно реализуется в базе ПЗУ или ППЗУ и служит для хранения множества микрокоманд, образующих микропрограмму. Регистр адреса РА хранит адрес текущей микрокоманды.

Использование различных методов адресации – принудительной (ПА), естественной (ЕА) и комбинированной (КА) приводит к различным форматам микрокоманд [4, 5], которые представлены на рис. 2.

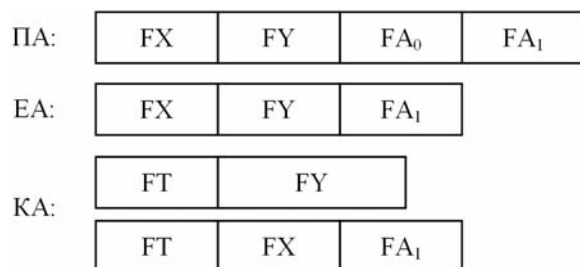


Рисунок 2 – Форматы микрокоманд для различных методов адресации

При этом адрес перехода A^{t+1} определяется для различных режимов адресации следующим образом:

$$ПА : A^{t+1} = \begin{cases} FA_0, & \text{если } x = 0; \\ FA_1, & \text{если } x = 1. \end{cases}$$

$$ЕА : A^{t+1} = \begin{cases} A^t + 1, & \text{если } x = 0; \\ FA_1, & \text{если } x = 1. \end{cases}$$

$$КА : A^{t+1} = \begin{cases} A^t + 1, & \text{если } FT = 0; \\ A^t + 1, & \text{если } FT = 1 \text{ и } x = 0; \\ FA_1, & \text{если } FT = 1 \text{ и } x = 1. \end{cases}$$

Поскольку в МУУ вся информация, относящаяся к реализуемому алгоритму управления, хранится в УП, то можно утверждать, что стоимостные характеристики схемы МУУ во многом определяются характеристиками схемы УП. Снижение емкости

УП, традиционно реализуемой в базе ПЗУ или ППЗУ, возможно за счет уменьшения разрядности полей в используемом формате микрокоманды. С этой целью выполним следующее.

Модифицируем рассмотренные форматы микрокоманд так, чтобы в адресных полях вместо полного адреса A перехода содержалось знаковое смещение S относительно текущего адреса. Величина смещения измеряется в количестве микрокоманд. При $S > 0$ выполняется переход к более высоким адресам, при $S < 0$ – к более низким. При этом уменьшение информационной емкости схемы УП возможно при условии

$$R_S < R_A, \quad (1)$$

где R_A и R_S – разрядности адреса и смещения соответственно. Назовем данный способ обозначения точки перехода *относительной адресацией микрокоманд* (ОА).

Применяя принцип относительной адресации к трем известным методам адресации, получим три метода относительной адресации: принудительная ОА (ПОА), естественная ОА (ЕОА) и комбинированная ОА (КОА). Форматы микрокоманд для данных типов ОА представлены на рис. 3.

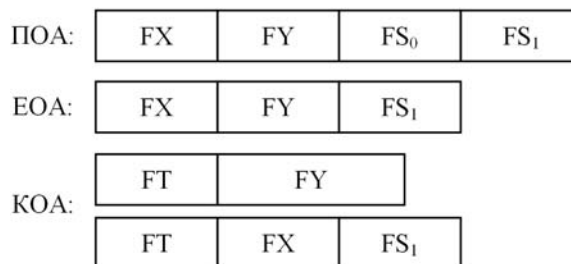


Рисунок 3 – Форматы микрокоманд для различных методов относительной адресации

Здесь поле FS_0 содержит смещение перехода при невыполнении ЛУ, FS_1 – смещение перехода при выполнении ЛУ.

При этом адрес перехода A^{t+1} будет формироваться следующим образом:

$$ПОА : A^{t+1} = \begin{cases} A^t + FS_0, & \text{если } x = 0; \\ A^t + FS_1, & \text{если } x = 1. \end{cases}$$

$$ЕОА : A^{t+1} = \begin{cases} A^t + 1, & \text{если } x = 0; \\ A^t + FS_1, & \text{если } x = 1. \end{cases}$$

$$КОА : A^{t+1} = \begin{cases} A^t + 1, & \text{если } FT = 0; \\ A^t + 1, & \text{если } FT = 1 \text{ и } x = 0; \\ A^t + FS_1, & \text{если } FT = 1 \text{ и } x = 1. \end{cases}$$

Таким образом, при использовании ОА в МУУ может иметь место уменьшение емкости

ПЗУ схемы УП за счет сокращения разрядности адресных полей в форматах микрокоманд.

Использование ОА в МУУ требует модификации структуры устройства. Между УП и регистром адреса ставится сумматор смещения, выполняющий суммирование содержимого регистра адреса (адреса текущей микрокоманды) и смещения, поступающего из УП. Например, структурная модель МУУ для случая ПОА приведена на рис. 4.

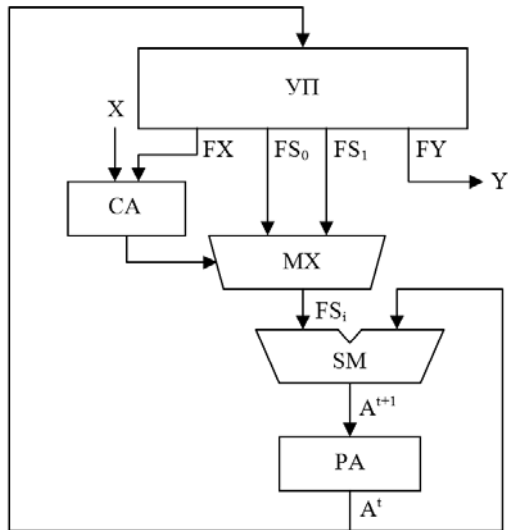


Рисунок 4 – Структурная модель МУУ с принудительной операционной адресацией

Здесь схема адресации СА на основании содержимого поля FX и значений ЛУ X управляет мультиплексором MX и пропускает на вход сумматора SM либо поле FS_0 , либо поле FS_1 .

Диапазон смещения S зависит от разрядности R_S . Являясь числом со знаком, S может принимать максимально возможное положительное значение, равное $(2^{(R_S-1)} - 1)$, и максимально возможное отрицательное значение, равное $-2^{(R_S-1)}$. Так, при $R_S=6$ имеем 5 значащих разрядов (один – знаковый), что позволяет использовать смещение в диапазоне $[+31, -32]$.

Если при заданном значении R_S невозможно выполнить переход из МК с адресом A_i в МК с адресом A_j , в ГСА могут быть добавлены дополнительные «пустые» МК, которые будут выступать промежуточными при переходе из A_i в A_j . При этом следует помнить, что рост числа дополнительных МК приводит к увеличению времени выполнения алгоритма.

Пример синтеза МУУ с относительной адресацией

Рассмотрим пример построение МУУ с принудительной операционной адресацией на примере ГСА G (рис. 5). Данная ГСА

характеризуется следующими основными параметрами: число микрокоманд $M=18$ микрокоманд a_1-a_{18} и количество анализируемых ЛУ $L=4$ (три ЛУ x_1-x_3 и безусловный переход (БП), для которого проверяемое ЛУ считается всегда истинным). Поскольку множество формируемых микроопераций не оказывает принципиального влияния на процесс синтеза устройства, в рассматриваемом примере содержимое микрокоманд не детализировано, и с этой точки зрения ГСА G носит абстрактный характер.

Поскольку адресация микрокоманд в случае традиционного МУУ может быть произвольной, будем считать адрес микрокоманды равным его индексу, уменьшенному на единицу. При построении содержимого ПЗУ будем придерживаться формата микрокоманды согласно рис. 2 для принудительной адресации.

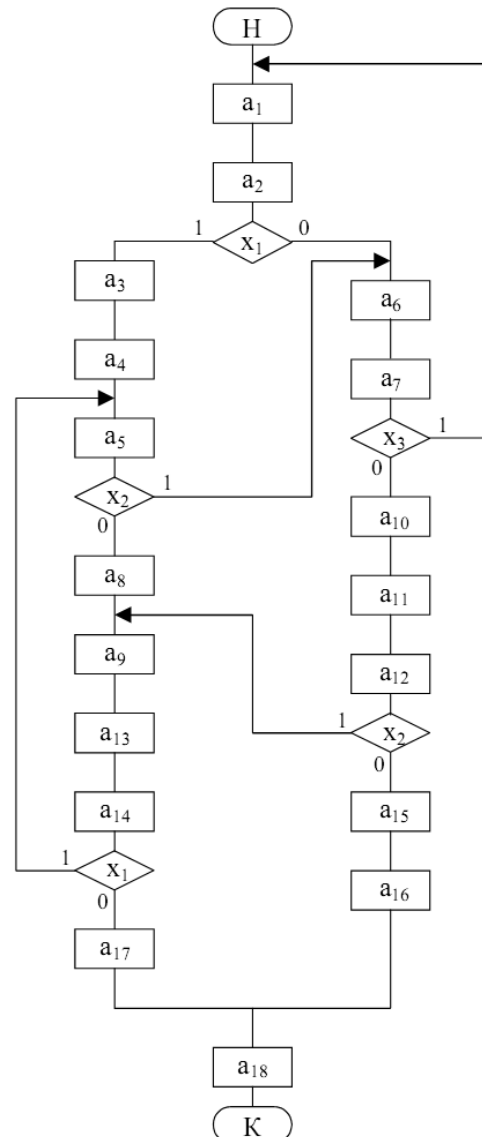


Рисунок 5 – Граф-схема алгоритма G

Также закодируем логические условия следующим образом: $K(x_1)=1$, $K(x_2)=2$, $K(x_3)=3$, $K(\text{БП})=0$. Число двоичных разрядов для кодирования ЛУ $R_L = \lceil \log_2 L \rceil = 2$.

Сформируем содержимое ПЗУ управляющей памяти для традиционной реализации МУУ согласно методике, изложенной в [4] (табл. 1). В таблице символ «*» обозначает произвольное содержимое данного поля.

Таблица 1. Содержимое ПЗУ управляющей памяти для традиционной реализации МУУ

a_i	Адрес	FX	FA_0	FA_1
a_1	0	0	*	1
a_2	1	1	5	2
a_3	2	0	*	3
a_4	3	0	*	4
a_5	4	2	7	5
a_6	5	0	*	6
a_7	6	3	10	1
a_8	7	0	*	8
a_9	8	0	*	12
a_{10}	9	0	*	10
a_{11}	10	0	*	11
a_{12}	11	2	14	8
a_{13}	12	0	*	13
a_{14}	13	1	16	4
a_{15}	14	0	*	15
a_{16}	15	0	*	17
a_{17}	16	0	*	17
a_{18}	17	*	*	*

Для адресации $M=18$ микрокоманд требуются $R_A = \lceil \log_2 M \rceil = 5$ двоичных разрядов адреса, что определяет число строк схемы УП равным $2^5 = 32$. Хотя содержимое полей FX, FA_0 и FA_1 приводится в табл. 1 в десятичной системе счисления, при прошивке ПЗУ следует использовать эквивалентные двоичные значения минимально достаточной разрядности.

В табл. 1 приведено содержимое лишь тех строк, адреса которых соответствуют микрокомандам ГСА G. Содержимое остальных строк УП для рассматриваемого примера может быть произвольным. Поскольку предполагается, что переход из a_{18} выполняется в конечную вершину и приводит к останову устройства, содержимое полей для МК a_{18} также может быть произвольным. Подчеркнем также, что в силу абстрактного характера ГСА содержимое поля FУ в таблице не показано, поскольку может быть определено лишь после детализации содержимого микрокоманд.

Итак, разрядности адресных полей микрокоманды $R(FA_0)=R(FA_1)=5$. Тогда общая емкость ПЗУ управляющей памяти, соответствующая адресным полям, в случае принудительной адресации составит $V_{ПА} = 32 * (5+5) = 320$ бит.

Выполним для ГСА G синтез МУУ с принудительной относительной адресацией.

С помощью специальной методики, не рассматриваемой в данной работе, адресуем микрокоманды так, как показано в табл. 2.

Таблица 2 – Адресация микрокоманд МУУ с принудительной операционной адресацией

a_i	$A(a_i)$	a_i	$A(a_i)$
a_1	4	a_{10}	29
a_2	0	a_{11}	14
a_3	3	a_{12}	10
a_4	31	a_{13}	7
a_5	2	a_{14}	6
a_6	30	a_{15}	13
a_7	1	a_{16}	12
a_8	5	a_{17}	9
a_9	8	a_{18}	11

Адреса в табл. 2 подобраны так, что переходы в заданной ГСА могут быть выполнены с использованием смещений в диапазоне $[-4; +3]$. Например, переход из $A(a_{12})=10$ осуществляется либо в $A(a_9)=8$ (смещение $FS_1 = -2$), либо в $A(a_{15})=13$ (смещение $FS_0 = +3$). При этом сложение текущего адреса со смещением выполняется по модулю 32: например, переход из $A(a_3)=3$ в $A(a_4)=31$ выполняется путем прибавления смещения, равного -4 .

Главным достоинством выбранного диапазона смещений является то, что все его числа могут быть представлены трехразрядными двоичными числами со знаком в дополнительном коде: числу -4 соответствует код 100_2 , числу $+3$ – код 011_2 . Используя вместо полей FA_0 и FA_1 поля FS_0 и FS_1 (см. рис. 3) разрядности $R_S=3$, мы для 32 строк ПЗУ получим суммарную емкость полей FS_0 и FS_1 равной $V_{ПОА}=32 * R_S * 2 = 192$ бита.

В рассматриваемой ГСА G некоторые переходы не могут быть выполнены с помощью выбранного диапазона смещений. Например, при переходе из $A(a_{10})=29$ в $A(a_{11})=14$ смещение составляет 15 адресов. Такой переход может быть осуществлен в несколько шагов через дополнительные «пустые» микрокоманды. Поскольку для выбранного диапазона смещений максимальное отрицательное смещение равно -4 , то переход длиной 15 адресов может быть осуществлен минимум за 4 шага. Таким образом, в ГСА должны быть введены 3 дополнительных МК с «пустым» полем FУ.

Добавим в ГСА G между МК a_{10} и a_{11} дополнительные МК $a_{19}-a_{21}$ с адресами $A(a_{19})=25$, $A(a_{20})=21$, $A(a_{21})=17$. В результате получим преобразованную ГСА G, приведенную на рис. 6. На данном рисунке дополнительные МК показаны штриховкой. Кроме того, около каждой микрокоманды показан ее адрес согласно табл. 2, а около каждой ветви перехода – величина используемого смещения.

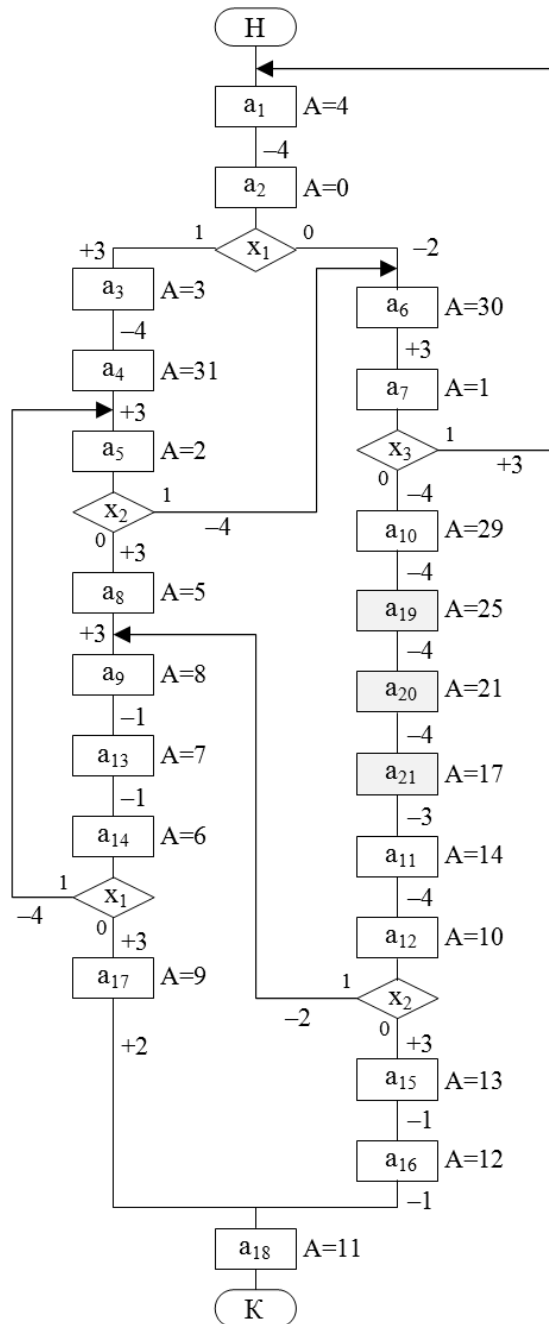


Рисунок 6 – Преобразованная ГСА G

По преобразованной ГСА G составим содержимое ПЗУ управляющей памяти (табл. 3). Как и в случае табл. 1, здесь указаны десятичные значения информационных полей, которые перед процессом прошивки должны быть преобразованы в двоичные значения, представленные в дополнительном коде.

Отметим, что поскольку ГСА G начинается с микрокоманды a_1 , то в структуре на рис. 4 регистр адреса в начале работы должен инициализироваться значением $A(a_1)=4_{10}=00100_2$. Данный момент должен быть реализован

схмотехнически при проектировании схемы регистра адреса.

Таблица 3. Содержимое ПЗУ управляющей памяти для МУУ с ПОА

a_i	Адрес	FX	FS ₀	FS ₁
a_2	0	1	-2	+3
a_7	1	3	+3	-4
a_5	2	2	+3	-4
a_3	3	0	*	-4
a_1	4	0	*	-4
a_8	5	0	*	+3
a_{14}	6	1	+3	-4
a_{13}	7	0	*	-1
a_9	8	0	*	-1
a_{17}	9	0	*	+2
a_{12}	10	2	-2	+3
a_{18}	11	*	*	*
a_{16}	12	0	*	-1
a_{15}	13	0	*	-1
a_{11}	14	0	*	-4
	15	*	*	*
	16	*	*	*
a_{21}	17	0	*	-3
	18	*	*	*
	19	*	*	*
	20	*	*	*
a_{20}	21	0	*	-4
	22	*	*	*
	23	*	*	*
	24	*	*	*
a_{19}	25	0	*	-4
	26	*	*	*
	27	*	*	*
	28	*	*	*
a_{10}	29	0	*	-4
a_6	30	0	*	+3
a_4	31	0	*	+3

Оценим для рассмотренного примера эффективность использования относительной адресации с точки зрения уменьшения информационной емкости ПЗУ управляющей памяти.

Абсолютный выигрыш в емкости ПЗУ схемы УП составляет $V = V_{\text{ПОА}} - V_{\text{ПА}} = 128$ бит, что составляет 40% от исходной емкости полей FA_0 и FA_1 .

Пусть известна разрядность $R(FY)$ поля FY . Тогда отношение информационной емкости строки ПЗУ управляющей памяти МУУ с ПА к эквивалентному параметру МУУ с ПОА определяется выражением (2).

$$E = \frac{R(FX) + R(FY) + R(FA_0) + R(FA_1)}{R(FX) + R(FY) + R(FS_0) + R(FS_1)} \quad (2)$$

Поскольку величина $R(FY)$ не зависит от структуры ГСА, рассмотрим изменение величины E при изменении $R(FY)$ в диапазоне от 0 до 50 с шагом 5, используя для остальных параметров

выражения (2) значения из рассмотренного примера. Полученные результаты приведены в табл. 4 и показаны графически на рис. 7.

Таблица 4 – Зависимость $E(R(FY))$

R(FY)	E
0	1,50
5	1,30
10	1,22
15	1,17
20	1,14
25	1,12
30	1,10
35	1,09
40	1,08
45	1,07
50	1,06

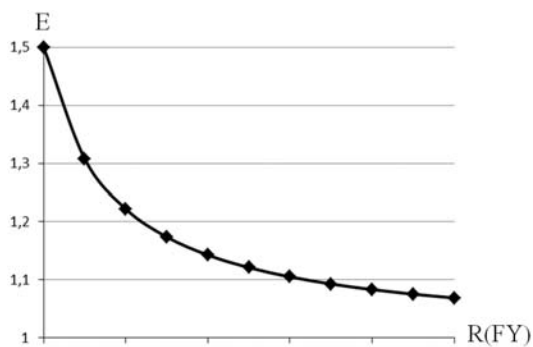


Рисунок 7 – График зависимости $E(R(FY))$

Анализ графика показывает, что данная зависимость экспоненциально убывает с ростом $R(FY)$ и при $R(FY) \rightarrow \infty$ стремится к единице

Анализ полученных результатов

В настоящей работе предложен новый способ адресации микрокоманд в

микропрограммном устройстве управления, основанный на использовании адресного смещения. По результатам проведенных авторами исследований могут быть сделаны следующие выводы:

1. Применение относительной адресации в МУУ позволяет уменьшить аппаратные затраты в схеме устройства за счет снижения информационной емкости УП.

2. В структурную модель МУУ вводится сумматор, выполняющий знаковое сложение адреса текущей микрокоманды со смещением, поступающим из управляющей памяти. За счет этого быстродействие схемы может незначительно снижаться.

4. Добавление дополнительных микрокоманд в общем случае увеличивает среднее число микрокоманд, выполняющихся за один проход микропрограммы. Результатом является увеличение времени выполнения микропрограммы, точное значение которого можно получить, зная вероятности выполнения логических условий, анализируемых в ГСА [8].

5. Применение относительной адресации не снижает универсальность схемы устройства. Использование стандартных функциональных элементов позволяет легко реализовать схему устройства с использованием стандартных библиотечных элементов САПР ПЛИС [9, 10].

6. Процесс адресации микрокоманд в случае относительной адресации, при которой достигается аппаратный выигрыш, не является очевидным из структуры ГСА. Для адресации микрокоманд должны использоваться специальные методы, основывающиеся на выбранных критериях оптимизации МУУ. Разработка таких методов планируется авторами в дальнейших исследованиях.

Литература

1. Глушков В.М. Синтез цифровых автоматов / В.М. Глушков. – М.: Физматгиз, 1962. – 476 с.
2. Майоров С.А. Принципы организации цифровых машин / С.А. Майоров, Г.И. Новиков. – Л.: Машиностроение, 1974. – 432 с.
3. Baranov S. Logic and system design of digital systems. – Tallinn: TUT Press, 2008. – 266 pp.
4. Баркалов А. А. Синтез микропрограммных устройств управления / А.А. Баркалов, А.В. Палагин. – К.: ИК НАН Украины, 1997. – 136 с.
5. Баркалов А.А. Синтез устройств управления на программируемы логических устройствах / А.А. Баркалов. – Донецк: ДНТУ, 2002. – 262 с.
6. Корнеев В.В. Современные микропроцессоры / В.В. Корнеев, А.В. Киселев. – М.: НОЛИДЖ, 1998. – 240 с.
7. Таненбаум Э. Архитектура компьютера / Э. Таненбаум. – СПб.: Питер, 2002. – 704 с.
8. Баркалов А.А. Определение вероятности кэш-попаданий в композиционных микропрограммных устройствах управления по граф-схеме алгоритма // Автоматика и вычислительная техника / А.А. Баркалов, С.А. Ковалев, Р.М. Бабаков. – 2001. – № 4. – С. 44-52.
9. Грушвицкий Р.И. Проектирование систем на микросхемах программируемой логики / Р.И. Грушвицкий, А.Х. Мурсаев, Е.П. Угрюмов. – СПб.: БХВ-Петербург, 2002. – 608 с.
10. Соловьев В.В. Логическое проектирование цифровых систем на основе программируемых логических интегральных схем / В.В. Соловьев, А. Климович. – М.: Горячая линия – Телеком, 2008. – 376 с.

Надійшла до редакції 18.02.2011