

# УСКОРЯВАНЕ НА УСЛОВНИТЕ ПРЕХОДИ В УПРАВЛЯВАЩИ АВТОМАТИ С ПРОГРАМНО ЗАКРЕПЕНА ЛОГИКА

Тянев Д. С.

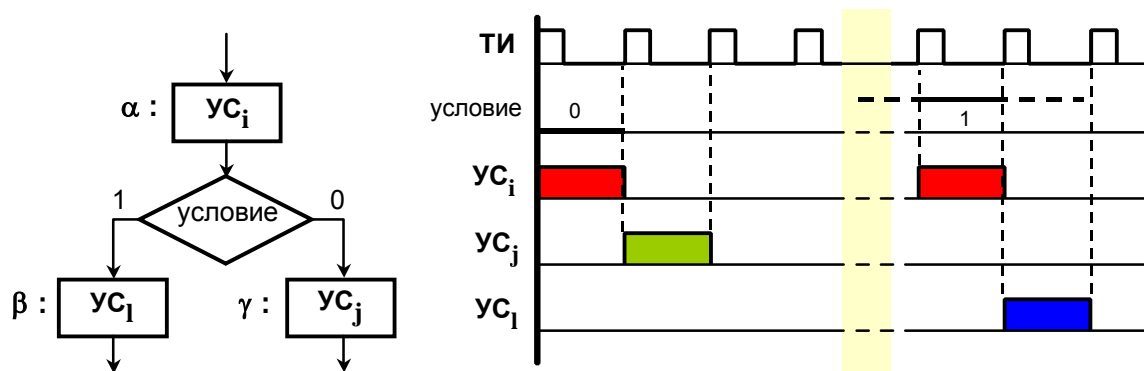
*Предлага се метод за ускоряване изпълнението на алгоритми в микропрограми управляващи автомати с програмно закрепена логика. Ускоряването се извършва най-вече в участъци с комплексни условия за преход. Техническото решение се постига чрез включване на допълнителен външен източник на адрес за преход, в който изборът на съответния адрес се постига чрез подходящо функционално преобразуване. Разгледани са възможностите за реализация на това преобразуване и възможностите за разпределение на адресното пространство.*

## 1. Въведение в проблема

В съвременни условия средствата за построяване на управляващи автомати с програмно закрепена логика (УАПЗЛ) са твърде разнообразни [3-10]. Благодарение на технологичното си изпълнение те представляват достъпно и универсално средство за решаване на задачите при синтез на различни управляващи автомати в цифровите изчислителни устройства. Въпреки това, че имат фиксирана логическа структура, тези средства позволяват реализация на допълнителни подходи, с помощта на които може да се оптимизира програмната реализация на някои алгоритмични преходи така, че изпълнението на микропрограмата да е еквивалентно на това във варианта на автомат с апаратно закрепена логика.

Управлението на логическите структури на ниво микрооперации се постига чрез управляващи сигнали, чиято логическа последователност се съдържа в алгоритъма на управление. Следването на управляващите сигнали, според този алгоритъм при всеки алгоритмичен преход, е желателно да става без загуба на тактове. За съжаление в микропрограмните УАПЗЛ това не винаги може да се постигне, тъй като изчислението на стойността на логическото условие при условни алгоритмични преходи е програмно. Това е проблем както за структурите с принудителна адресация, така и за структурите с естествена адресация. Проблемът на снижени темп на управляващата последователност се изяснява по-надолу.

Ако условието на алгоритмичния преход се формира в управляваната част, неговата стойност е известна и възможностите за преход се изразяват в блок-схемите както е показано на фигура 1.

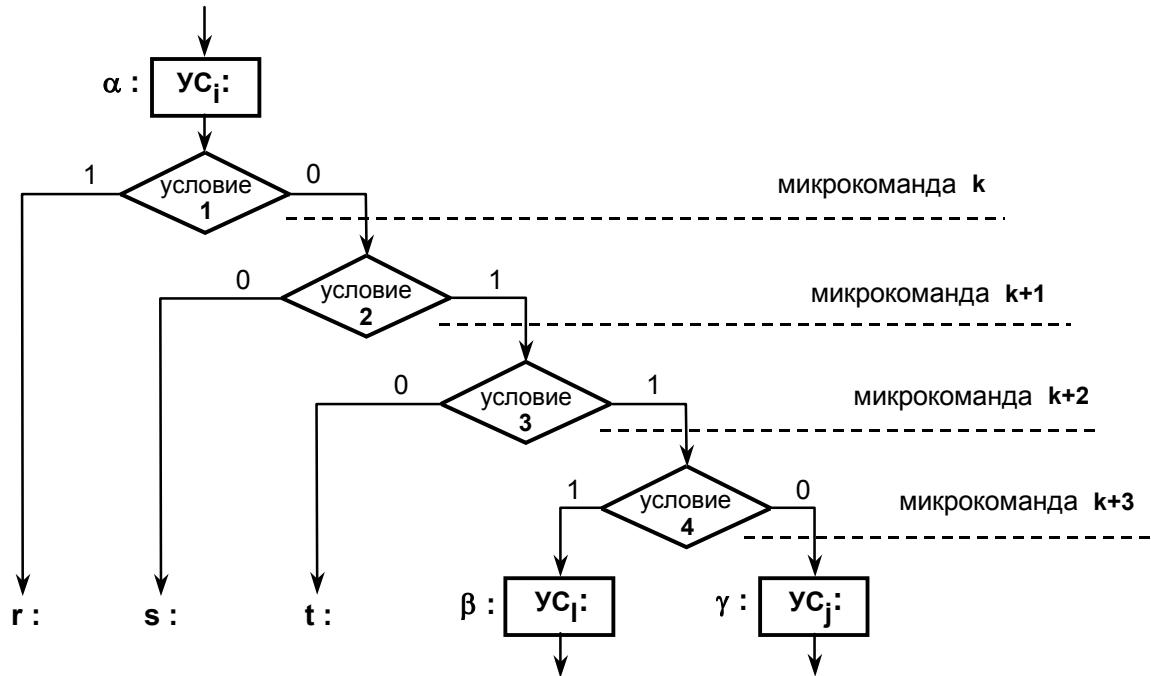


Фиг. 1

Както може да се види от времедиagramата, след управляващия сигнал  $UC_i$ , независимо от стойността на логическото условие, издаването на следващия управляващ сигнал ( $UC_j$  или  $UC_1$ ) се осъществява без загуба на такт. Това е така, защото в структурата на микрокомандата на УАПЗЛ с принудителна адресация се съдържат два адреса за преход

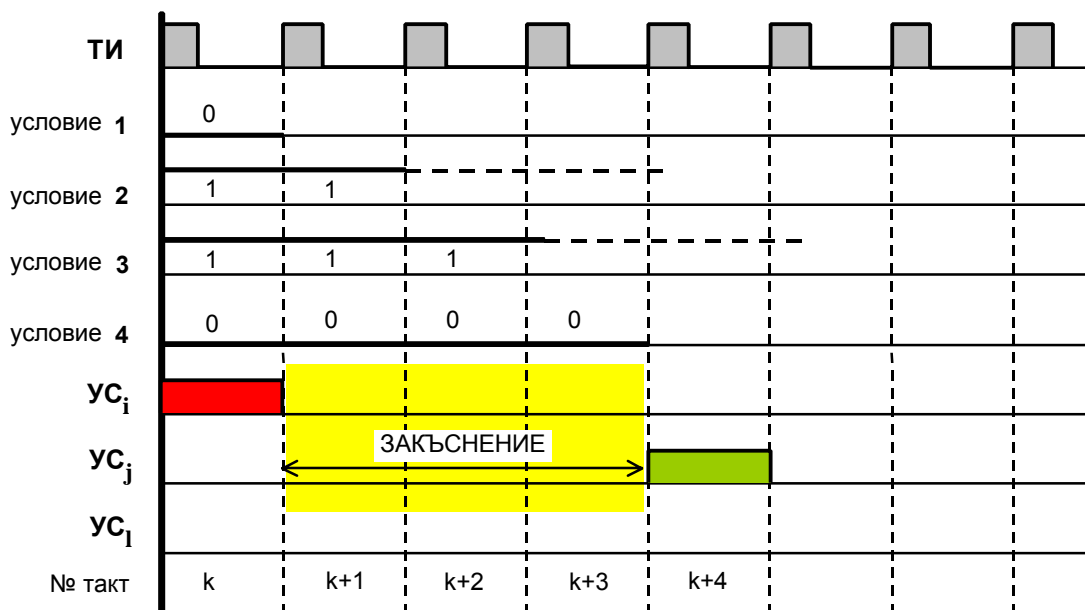
( $\beta$  и  $\gamma$ ). В структурата на микрокомандата на УАПЗЛ с естествена адресация може да бъде посочен само един адрес, обикновено в прехода "истина" (т.е. адресът  $\beta$  - виж фигура 1), а адресът в прехода "лъжа" се реализира чрез микропрограмния брояч, по закона  $\gamma = \alpha + 1$ .

В случаи когато условието за преход е комплексно, както е показано на фигура 2, алгоритмичните преходи се забавят, в следствие на последователната програмна проверка на условията за преход.



Фиг. 2

Без да разглеждаме всички възможни преходи се вижда, че преходът от адрес  $\alpha$  (микрокоманда  $k$ , която издава управляващ сигнал  $UC_i$ ) към адрес  $\gamma$  или  $\beta$ , не може да се осъществи в следващия такт, тъй като той зависи от още три непроверени условия (2, 3 и 4). Тези условия изискват проверка чрез три последователни микрокоманди в резултат на което, издаването на управляващия сигнал  $UC_j$  (или  $UC_i$ ) се забавя на три такта, както е показано на времедиagramата от фигура 3.



Фиг. 3

От показаното на фигура 3 може да се направи следния извод: ако логическите стойности на условия 2, 3 и 4 се формират още в предишния (k-1)-ви такт, то те ще трябва да се съхраняват във времето достатъчно дълго време, за което ще трябва да се вземат подходящи решения.

Проблемът на закъснелия управляващ сигнал съществува както за структурите с принудителна адресация, така и за структурите с естествена адресация. Теоретично проблемът има решение и то е изложено в [1]. Решението се състои от една страна в усложняване на адресната част на микрокомандата, а от друга страна в подходящо сегментиране на микропрограмата и адресното пространство. За съжаление в съвременни условия не съществува възможност да се изменя структурата на елементната база на схемата за формиране на адреса на следващата команда, тъй като тя най-често се предлага във вид на една интегрална схема. Този й завършен вид предопределя и структурата на микрокомандата. Известните и масово прилагани структури на фирми като Texas Instruments, Intel, Advanced Micro Devices, Signetics и други, не съдържат възможности за решаване на проблема. Единствено в серия K587 на руската електронна промишленост има възможност за реализация на многоалтернативните преходи [2].

## 2. Решение на проблема

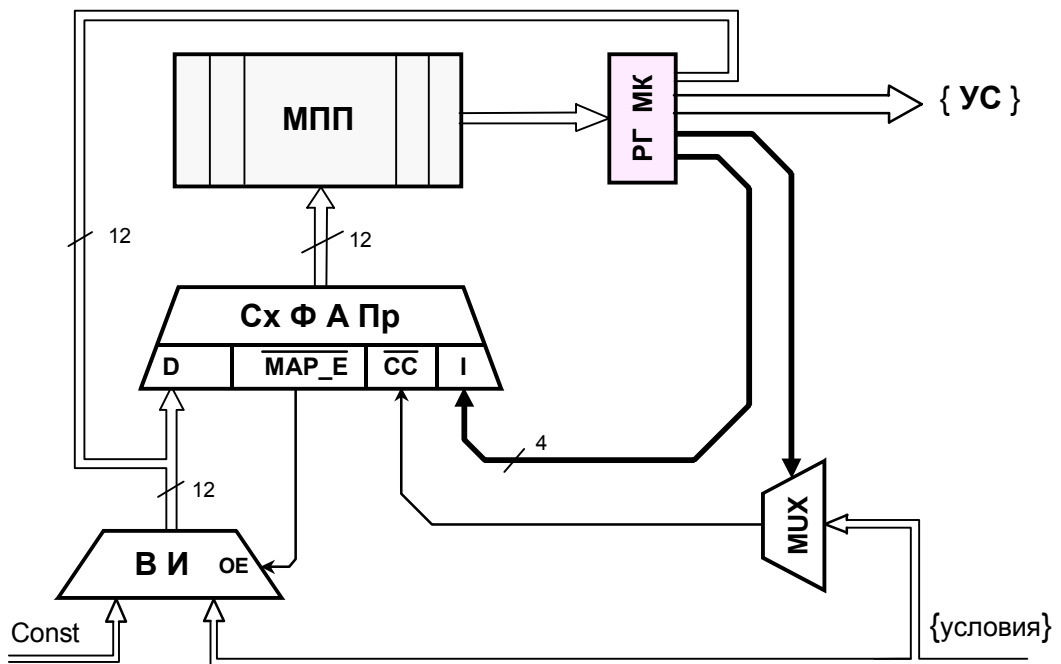
Тук ще бъде разгледана една възможност за решаване на изяснения проблем в структури от вида Am2910 (на фирмата AMD) или нейния руски еквивалент KP1804BY4. Както е известно [10, 5] тези структури са проектирани в съответствие с метода за естествена адресация на микрокомандите. Адресната част на микрокомандите в тази структура съдържа само един от адресите за преход. При условни преходи алтернативният адрес се извлича най-често от микропрограмния брояч или от друг вътрешен за схемата източник, например стек. При това положение многоалтернативните преходи от вида показан на фигура 2 се реализират последователно в микропрограмата, което води до изпълнение, съответстващо на фигура 3.

Изпълнението на един конкретен преход от многото възможни (виж фиг. 2) за един такт не е възможно без допълнителна апаратура. В допълнителната апаратура следва да се формира един адрес за преход, който е функция на няколко логически условия. Функционалното преобразование може да бъде частично (върху част от адреса) или пълно (върху целия адрес). Функционалното преобразование следва да се реализира за един такт, което изисква комбинационна схема. Функционалното преобразование е детерминирано и еднозначно, тъй като реалният алгоритмичен преход (при всеки отделен проход на програмата) е единствен.

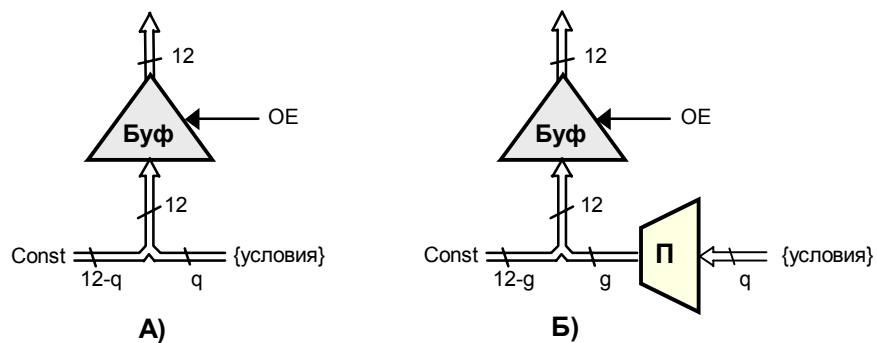
Осъществяването на адресен преход, който е различен от посочените в началото на този пункт, в цитираните структури може да се осъществи чрез специална микрокоманда за безусловен преход "JMAP", имаща микрокоманден код МКК=0010. Характерно за тази микрокоманда е това, че тя генерира разрешаващия сигнал  $\sim$ MAP\_ENABLE, който разрешава изходите на външен източник на адрес за преход. И така микрокоманда JMAP замества всички микрокоманди от фигура 3 (k-та, k+1-ва, k+2-ра, k+3-та), като разрешава адресът за преход да излезе от външен източник, както е показано на фигура 4. Така задачата се свежда до синтез на този източник.

Този начин за формиране на адреса за преход обаче има съществения недостатък да не е в състояние да гарантира достатъчна дължина на отделните адресни подпространства. Външният източник (ВИ), управляван от сигнала  $\sim$ MAP\_ENABLE, трябва да има буферирани изходи.

При обикновените преходи тези изходи ще бъдат изключени чрез третото логическо състояние на буферите. При преход чрез микрокоманда JMAP ще се изключва адресната шина, свързваща входа D на адресния секвенсор (АС) с регистъра на микрокомандара РгМК и ще се подключва външният източник ВИ, който ще подаде формирания в него адрес за преход, функция на входните условия. Така следващата микрокоманда ще бъде първата от избрания чрез условията клон на алгоритъма. Разположението на програмните участъци, реализиращи отделните клонове на алгоритъма в адресното пространство, зависи от това, колко сложна е реализацията на външния източник ВИ. Най-евтината реализация е тази при която ВИ представлява само набор от 12 буфера (виж фигура 5а). Старшата част на адреса (базовия адрес на програмните разклонения) конкатенира с младшата му част, формирана от стойностите на условията, подлежащи на проверка. Така адресът за преход се формира непосредствено от условията.



Фиг. 4



Фиг. 5

Това от своя страна изисква специално преподреждане на условията спрямо адресните разряди - задача, която не винаги е решима. Нещо повече - такова решение е алгоритмично зависимо. Изход от неразрешима ситуация може да се постигне чрез схемно функционално преобразуване на комбинацията от стойностите на условията в друга, достатъчно отдалечена в адресното пространство. Например в случая, представен на фигура 3, са възможни следните комбинации:

Усл. 1	Усл. 2	Усл. 3	Усл. 4	Изход
1	x	x	x	по адрес r
0	0	x	x	по адрес s
0	1	0	x	по адрес t
0	1	1	0	по адрес γ
0	1	1	1	по адрес β

Ако последните две комбинации са в младшата част на адреса:

$$A3A2A1A0 = \text{Усл.1} | \text{Усл.2} | \text{Усл.3} | \text{Усл.4}$$

то за програмиране на алгоритмичния клон с начален адрес γ няма нито една свободна клетка, тъй като следващият адрес е начало на друг клон.

На фигура 5б е показано как чрез включването на един преобразовател П q-разрядната комбинация от условия може да се преобразува в g-разрядна така, че да се осигури необходимия обем за всеки програмен клон.

### **Литература:**

- [1]. Майоров С.А., Новиков Г.И., *Принципы организации цифровых машин*, Ленинград, "Машиностроение", 1974.
- [2]. *Микропроцессорные БИС и микро-ЭВМ. Построение и применение*, под ред. Васенкова А.А., Москва, "Советское радио", 1980.
- [3]. Воробьев В.С., *Эффективность использования БИС и микропрограммирование*, Вопросы радиоэлектроники - серия ВТ, 1975, выпуск 3.
- [4]. Гринкевич В.А., Файзулаев Б.Н., *Пути построения элементной базы ЭВМ на основе БИС*, Вопросы радиоэлектроники - серия ВТ, 1977, выпуск 5.
- [5]. Воробьев Н.М., Васенков А.А. и др., *Микропроцессорные наборы БИС и серия ЭВМ "Электроника НЦ"* - Электронная промышленность, 1978, № 5.
- [6]. Березенко А.И., Корягин Л.Н., Назарян А.Р., *Микропроцессорные комплекты повышенного быстродействия*, Москва, "Радио и связь", 1981.
- [7]. Grasselli A., Montanari U., *On the minimization of READ-ONLY memories in microprogrammed digital computers*, IEEE Trans. Comput., 1970, vol. 19, №11.
- [8]. Edwards M., Dagless E., *LSI microprogrammable microprocessors*, Microprocessors, 1977, №7.
- [9]. Calvan N., *Structure and applications of field programmable logic arrays*, Microelectronics and Reliability, 1976, vol.15, p. 285-295.
- [10]. Baron M., *Control your next pipeline design with a microprogram sequencer*, END, 1980, vol. 25, №6, p. 157-162.