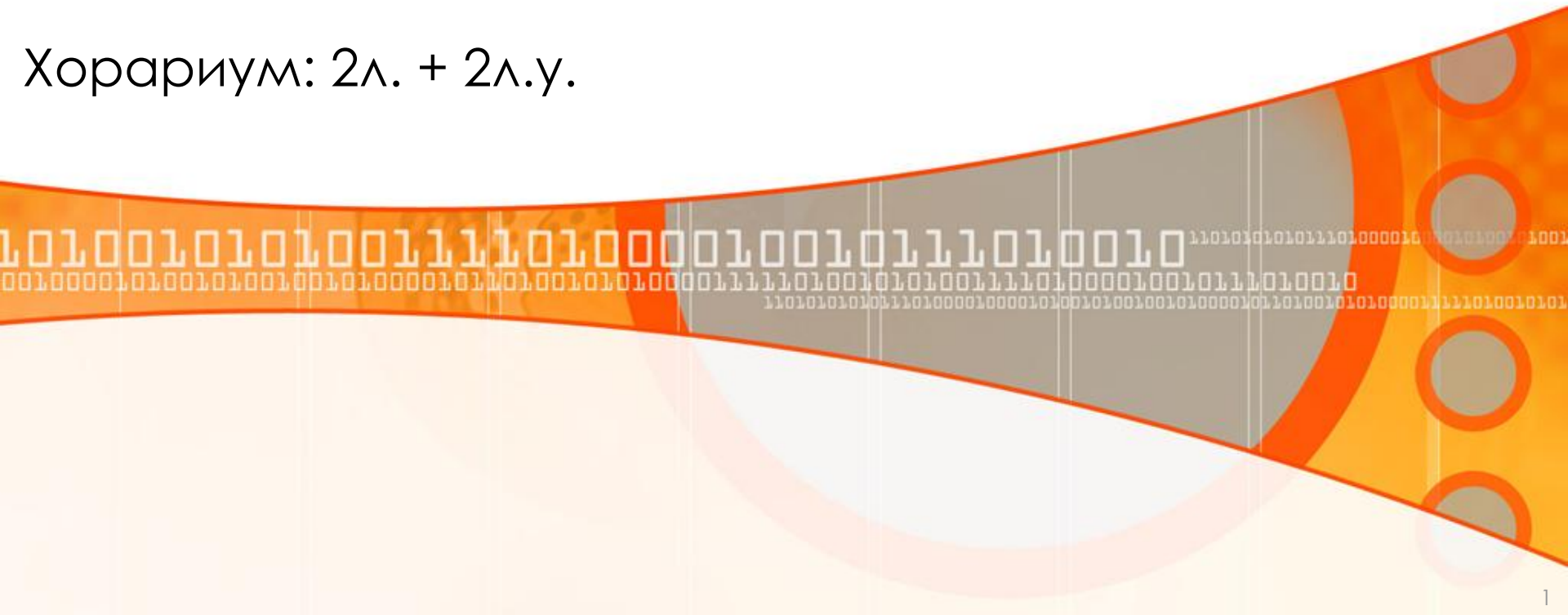
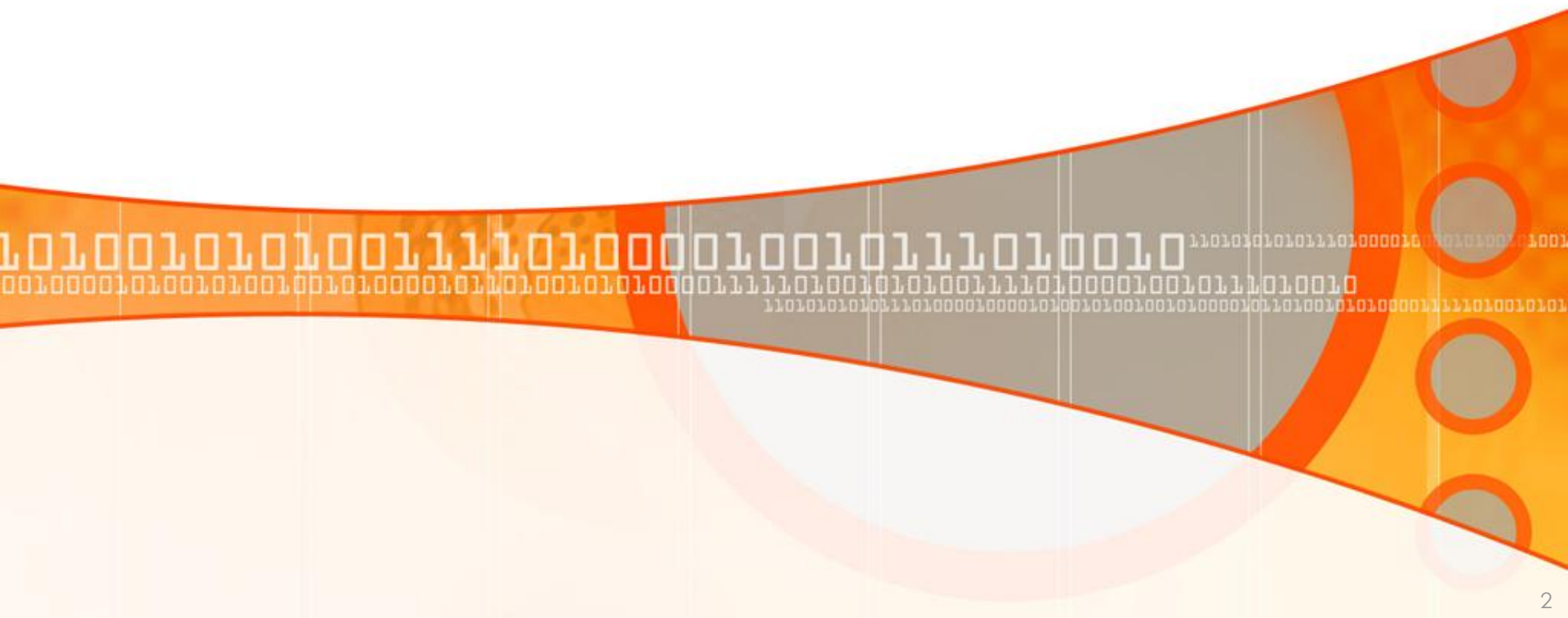


Организация на компютъра

Хорариум: 2л. + 2л.у.



Лекция 15: Организация на управлението



Съдържание

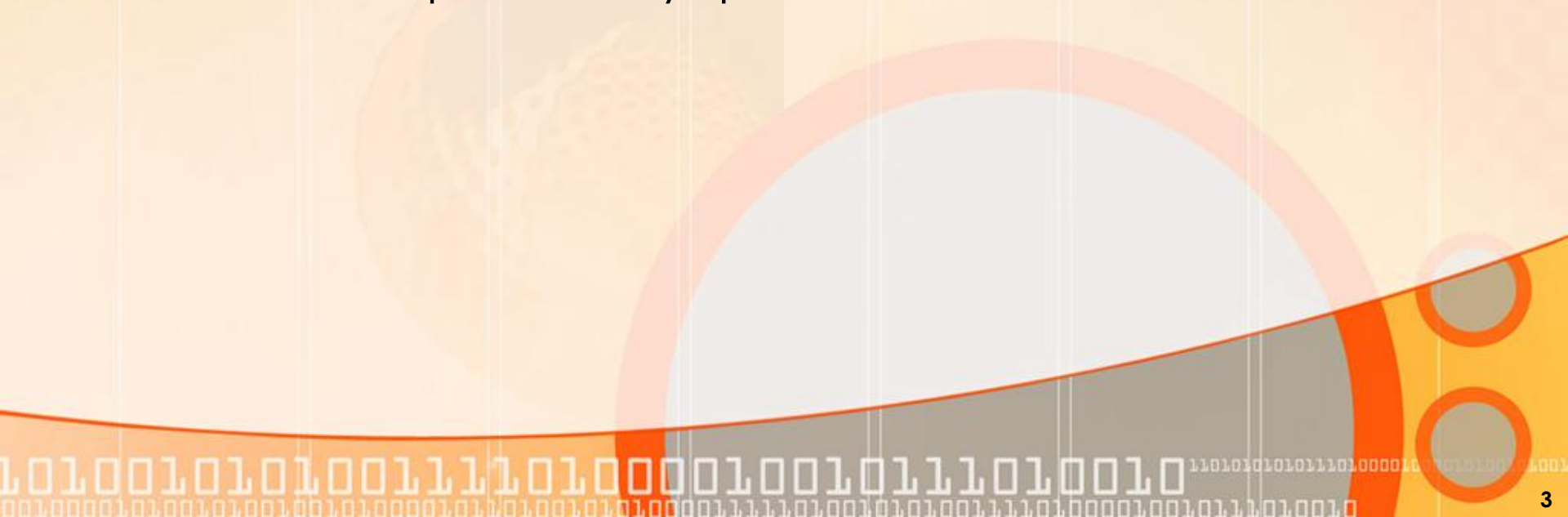
Командният цикъл като основен алгоритъм за функциониране на управляващото устройство

Синхронен и асинхронен метод на управление

Управляващи устройства с програмно закрепена логика

Логическа структура на УА с принудителна и с естествена адресация

Методи за кодиране на управляващите сигнали



Организация на управлението

Общи положения

Изчислителният процес е **структурен** и неговите елементи на най-ниско ниво са **микрооперациите**.

Микрооперациите се изпълняват от **логическите възли** на отделните устройства.

Микрооперациите, които са функции на логическите възли, се активират чрез **управляващите сигнали**.

Управляващите сигнали от своя страна са резултат от изпълнението на дадена **микропрограма**, която е вложена в **управляващия автомат** на съответното **операционно устройство**.



Организация на управлението

Общи положения

Микропрограмата, която представя функционирането на операционното устройство във времето, се превръща по време на своето изпълнение в **последователност от управляващи сигнали.**

Управляващите сигнали обикновено се появяват в началото на всеки текущ такт и изчезват с началото на следващия такт.

Управляващите сигнали (УС) представляват правоъгълни импулси, определяни като логически сигнали. Те се интерпретират като логически, защото като микрозаповеди имат само две значения:

УС = 0 - няма заповед (не се изпълнява микрооперация)

УС = 1 - заповед за микрооперация



Организация на управлението

Общи положения

Съвкупността от едновременно подадените в рамките на текущия такт управляващи сигнали се нарича **микрокоманда**, а последователността от микрокоманди – **микропрограма**.

Последователността от управляващи сигнали или точно от микрокоманди, се реализира от специален клас логически устройства, които се наричат крайни автомати или още автомати с памет, или просто **управляващи автомати**.

С течение на времето, при функциониране, управляващият автомат се превключва последователно от едно вътрешно състояние към друго.

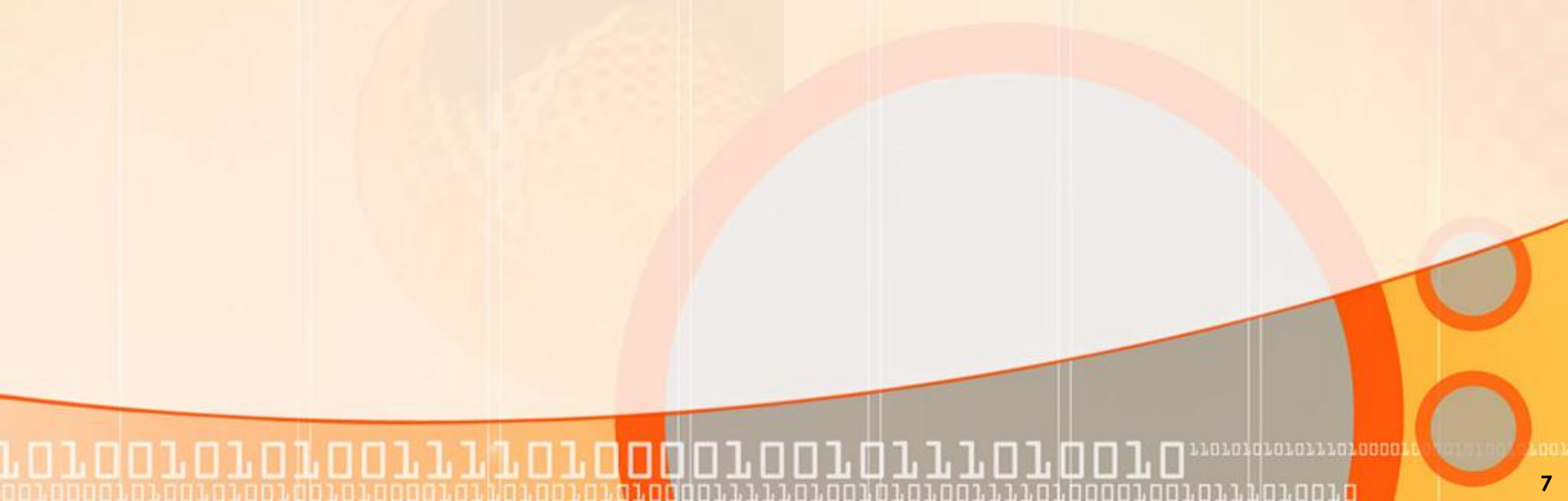


Организация на управлението

Общи положения

Всеки управляващ автомат има едно вътрешно състояние, в което престоява обикновено най-дълго време и то се нарича **изходно (начално)** състояние $S^{(0)}$.

След изпълнение на своята функция, управляващият автомат се оказва **автоматично** отново в изходното си състояние.

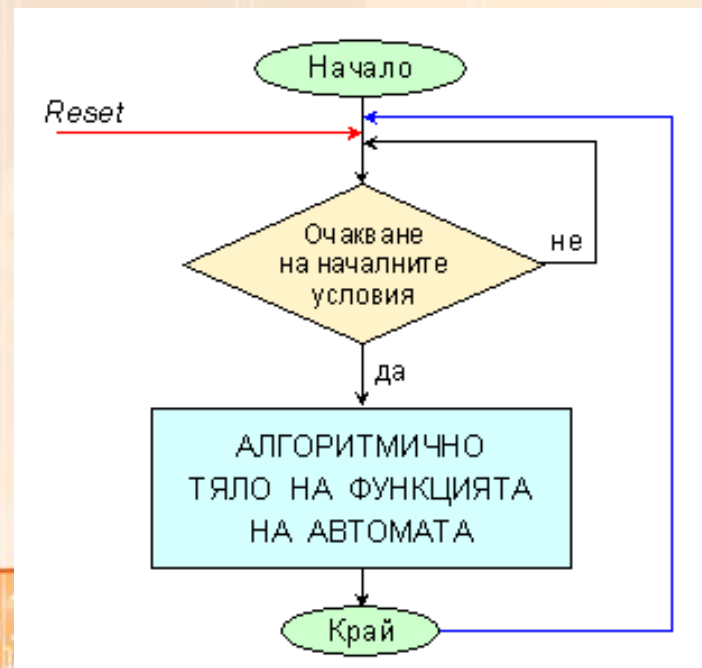


Организация на управлението

Общи положения

Удържането на автомата неограничено дълго време в изходно състояние се осигурява от зацикления условен преход, както е показано на фигурата.

От това състояние той може да бъде изваден само чрез външен сигнал, с цел да реализира своята функция за *пореден път*. Така изходното (началното) състояние може да се определи като състоянието, в което се очаква изпълнението на определени начални условия - сигнал "Пуск".



Организация на управлението

Синхронен и асинхронен метод за управление

Причината за изчезване на даден управляващ сигнал, т.е. за превключване на управляващото устройство от едно в друго състояние може да бъде само една - завършване на преходния процес на превключването в текущо управлявания възел и установяване на всичките му изходи в истинните логически стойности!

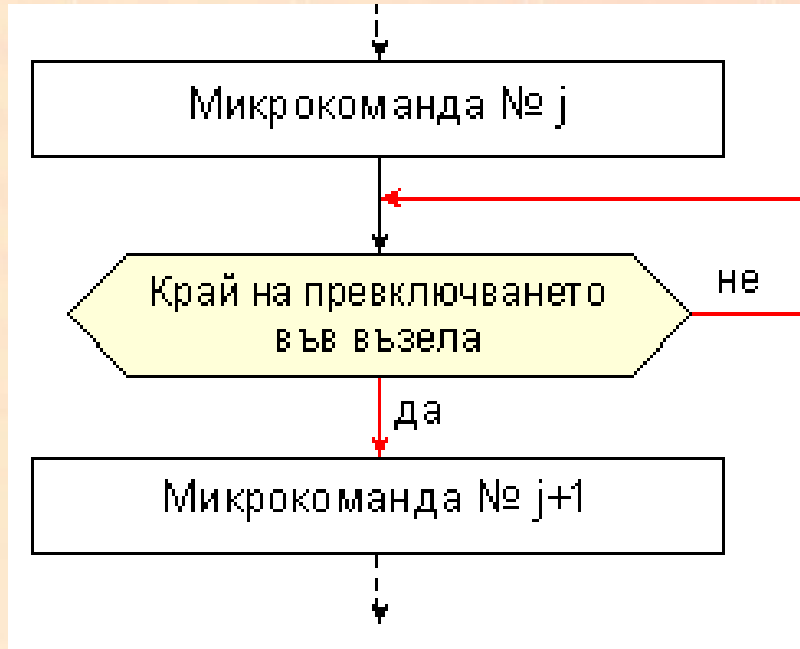
Когато за този факт управляваният възел съобщава чрез оповестяващ сигнал сам, се говори, че осъществяваното от автомата управление е **асинхронно**.

Когато за този факт се съди по априорни оценки на бързодействието, постигани чрез допълнителен външен сигнал (обикновено периодичен), се говори, че управлението е **синхронно**. Управлението е синхронно именно с този външен периодичен сигнал.

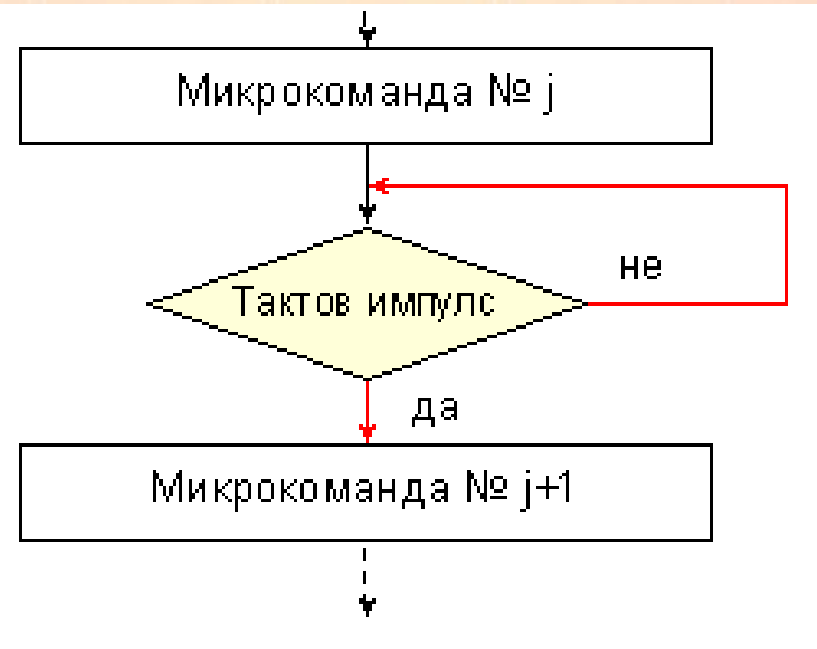


Организация на управлението

Асинхронен и синхронен метод за управление



асинхронно



синхронно

управление

Може да се използва и синхронно-асинхронен метод за управление.

Организация на управлението

Синхронен и асинхронен метод за управление

Основният и масово прилаган метод за управление е *синхронният*.

За реализиране на този метод в структурата на автомата се въвежда тактов генератор, за който се говори, че е *двигателят* на всяка цифрова система.

Тактовите импулси, които се наричат още *синхронизиращи*, фиксират моментите на превключване в логическите схеми, които съответно се наричат *синхронни*.



Организация на управлението

Синхронен и асинхронен метод за управление

Крайният автомат, който интерпретира микропрограмата на операционното устройство, се нарича **микропрограмен автомат**.

Синтезираните по съответната методика управляващи автомати се определят като **автомати с апаратно закрепена логика**.

Логиката на изчисление на управляващите сигнали, които се генерират от схемата на управляващия автомат, е фиксирана твърдо в логическите схеми и не може да бъде променена. С други думи препрограмирането, т.е. промяната на алгоритъма на функциониране или добавянето на нови функции в структурата на автомата, е свързано с неговото физическо разрушаване.

Организация на управлението

Управляващи автомати с програмно закрепена логика

Управляващите автомати могат да бъдат синтезирани въз основа на **принципа за програмно управление**.

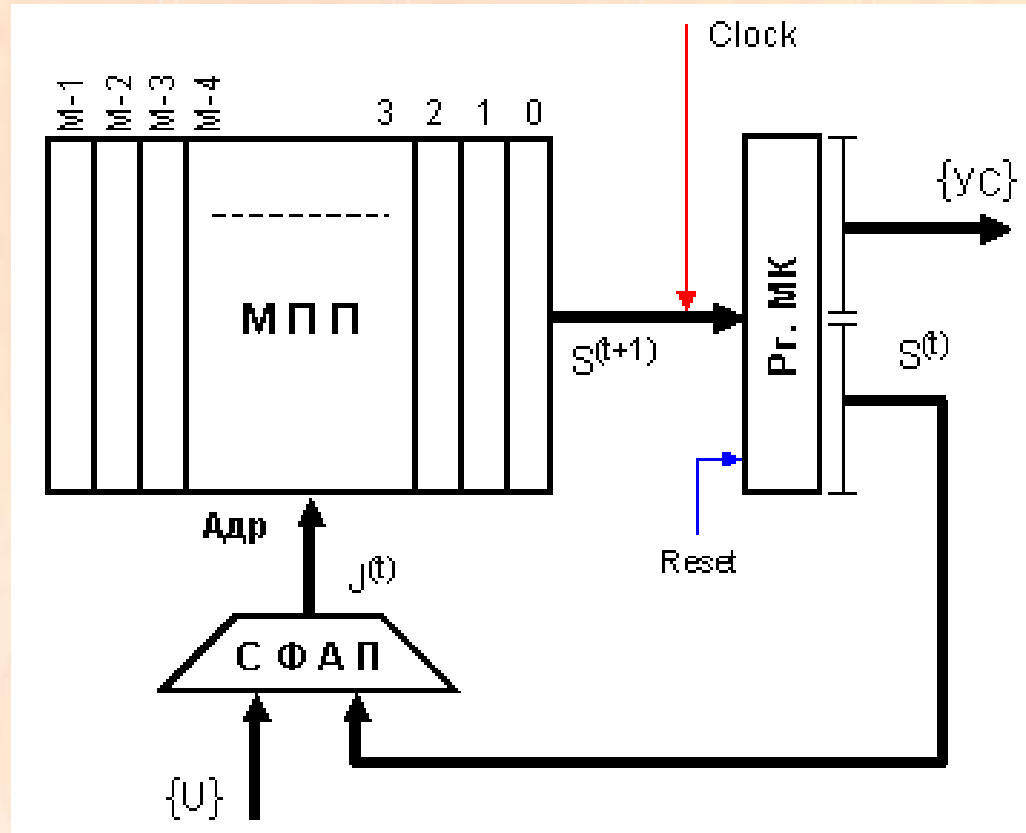
Според този принцип, при синтеза се изхожда от **алгоритмичната същност** на управляващия автомат. Алгоритмичната същност се изразява чрез блок-схемата на микропрограмата, в която еднозначно е изразено управлението.

В този смисъл и според посочения принцип, цялата последователност от микрокоманди може да бъде запаметена в паметта на автомата, където **алгоритмичните преходи се превръщат в адресни преходи**. С други думи, **логиката на алгоритмичните преходи се превръща в логика на адресните преходи**.



Организация на управлението

Управляващи автомати с програмно закрепена логика



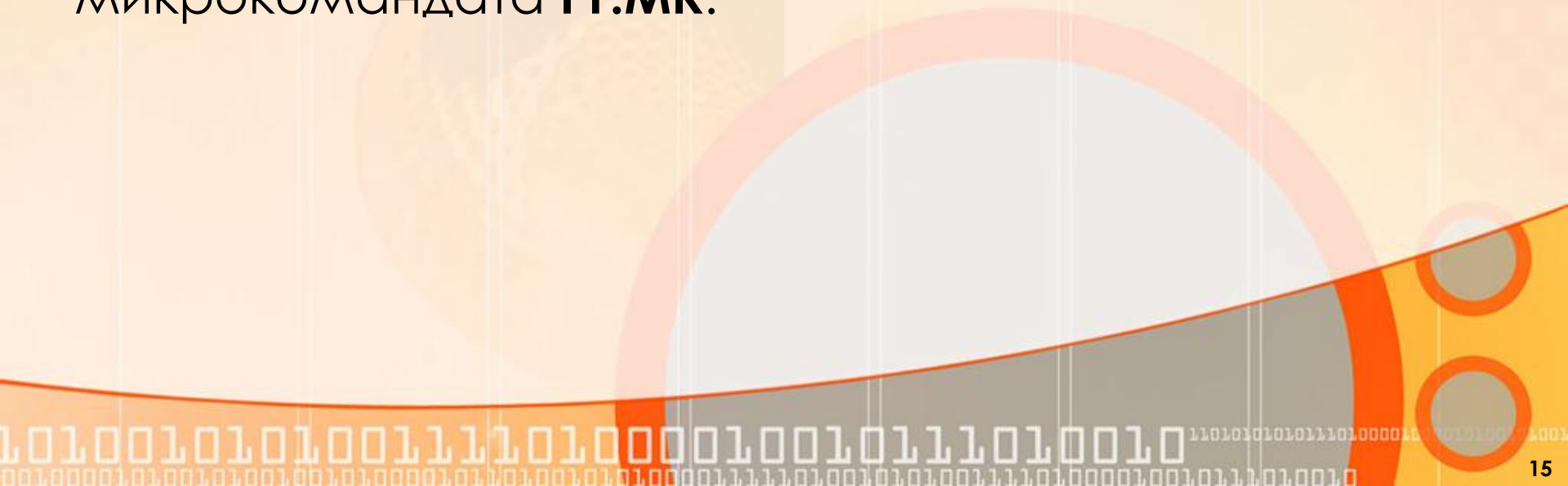
Обща структура на управляващ автомат с програмно закрепена логика

Организация на управлението

Управляващи автомати с програмно закрепена логика

Микропрограмната памет (**МПП**) има поне толкова клетки, колкото са микрокомандите в блок-схемата на микропрограмата.

Текущо изпълняваните микрооперации в операционното устройство се инициират от управляващите сигнали {УС} на текущата микрокоманда, която в продължение на текущия такт се съхранява в регистъра на микрокомандата **Рг.МК**.



Организация на управлението

Управляващи автомати с програмно закрепена логика

Алгоритмичният преход към следващата микрокоманда се реализира чрез адресния механизъм, заложен в схемата за формиране на адреса за преход **СФАП**.

Формираният от тази схема адрес **Адр** отваря съответната клетка в микропрограмната памет, откъдето се извлича следващата микрокоманда.

Схемата **СФАП** формира адреса за преход според методите за адресиране.



Организация на управлението

Управляващи автомати с програмно закрепена логика

Тъй като в паметта на управляващия автомат се съдържа цялата микропрограма, той се нарича *микропрограмен управляващ автомат* със *съхранена микропрограма (МПУА)*, а най-често се нарича просто *микропрограмен*.

Обикновено тези автомати са **синхронни**. Това означава, че появата (записът) на микрокомандите в **Рг.МК** е синхронна с тактовите импулси.

Всяка микрокоманда престоява в този регистър поне един период на тактовата последователност, което определя и времето (продължителността) за активност на управляващите сигнали.



Организация на управлението

Управляващи автомати с програмно закрепена логика

Основна задача на микрокомандата е да подаде управляващите сигнали към операционното устройство. Ето защо една нейна част е предназначена за изобразяване на множеството на управляващите сигнали $\{УС\}$. Тази част се нарича **операционна**.

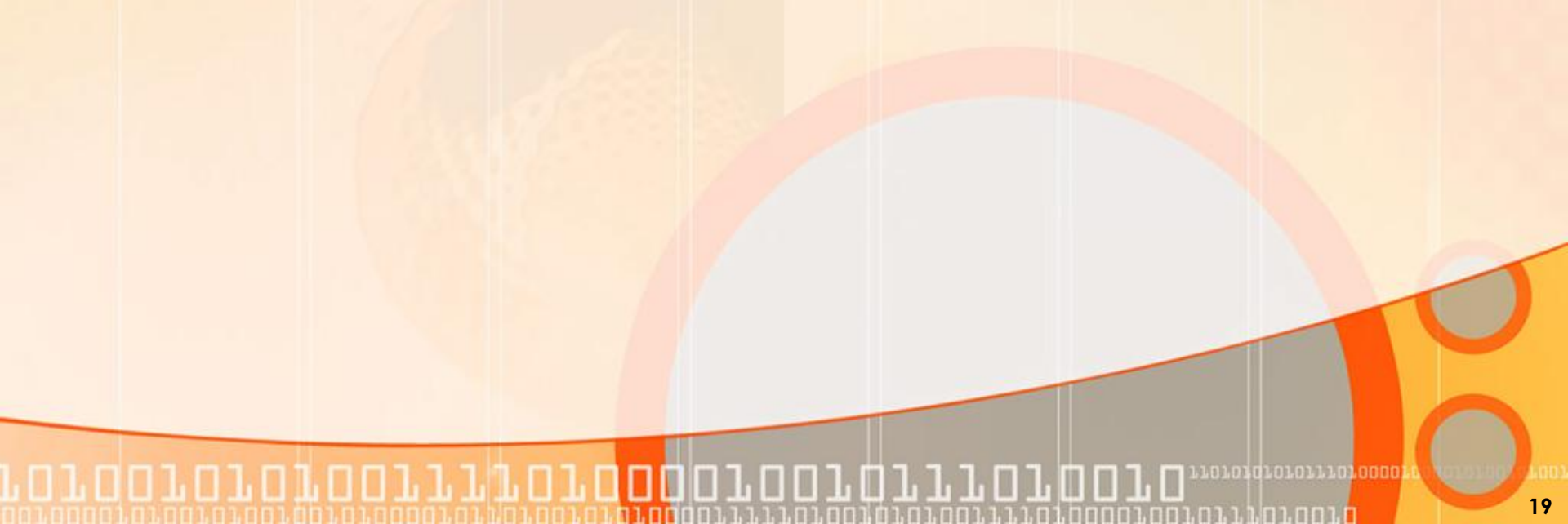
Освен тази задача, по силата на принципа за програмно управление, микрокомандата има за задача още да **управлява адресния преход** към следващата (според алгоритъма) микрокоманда. Ето защо в нейната структура има още една, втора част, наречена **адресна** част. Така структурата на микрокомандата съответствува на структурата на машинната команда, което е следствие от принципа на програмно управление.

Организация на управлението

Управляващи автомати с програмно закрепена логика

За управляващия автомат по-важна е задачата за определяне на **адреса за преход** към следващата микрокоманда.

От гледна точка на микропрограмния софтуер, в този смисъл микрокомандите могат да се интерпретират единствено и само като **команди за управление на прехода – безусловни или условни.**



Организация на управлението

Управляващи автомати с програмно закрепена логика

Тъй като функциите на управляващия автомат най-често са постоянни и непроменящи се, то **съдържанието на паметта на автомата е постоянно**, откъдето следва, че тя може да се реализира като постоянна – от тип ROM.

Съдържанието на паметта на автомата може винаги да бъде сменено, без да се променя нещо в структурата му, което означава, че **апаратната реализация не зависи от алгоритмичният "пълнеж" на паметта**.

Смяната на съдържащата се в паметта микропрограма с друга означава смяна на функцията, нещо което е невъзможно за автоматите с твърдо закрепена логика. Ето защо **МПУА** се наричат още **препрограмируеми автомати**.

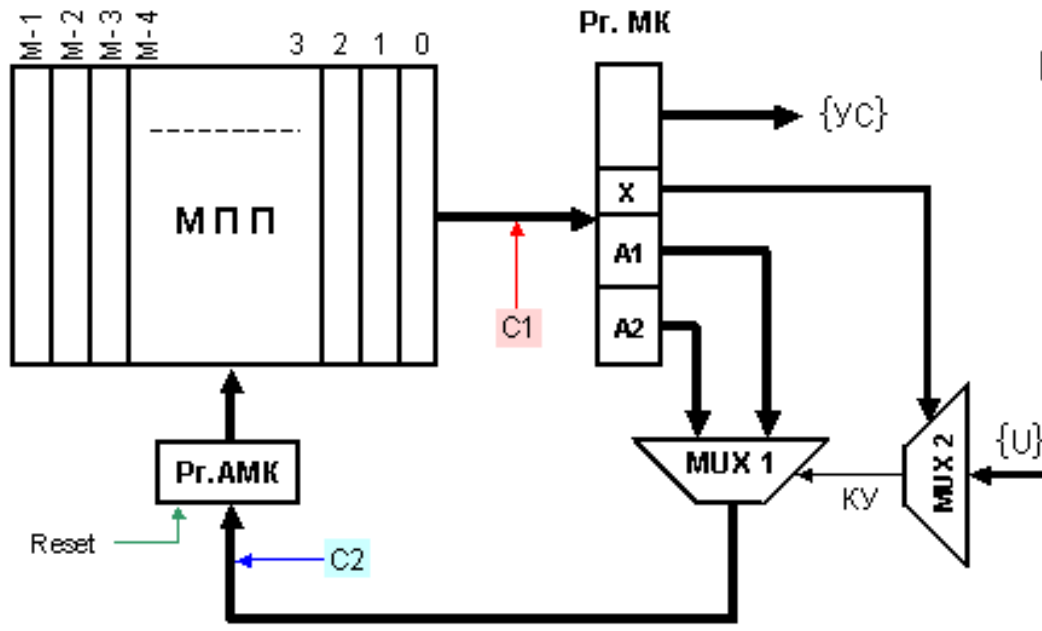
Организация на управлението

Логическа структура на УА с принудителна адресация

Структура на микрокомандата



Примерна логическа структура



$KY = \begin{cases} 0, & \text{условието не е изпълнено, т.е. "false";} \\ 1, & \text{условието е изпълнено, т.е. "true".} \end{cases}$

КУ управлява MUX1 по правилото:

if (KY) then GO TO A2
else GO TO A1
end if

Организация на управлението

Логическа структура на УА с принудителна адресация

Двуадресната структура на микрокомандата позволява изключително **ефективно използване на адресното пространство**. При принудителния метод на адресиране следващата микрокоманда може да се намира в произволна клетка, а това позволява **максимално уплътняване на паметта**.

Наличието на два адреса в структурата на микрокомандата не винаги е необходимо. В случаите, когато тя реализира безусловен алгоритмичен преход, например, съдържанието на полето A2 е без значение. В този смисъл микрокомандата неефективно използва клетката, в която се намира, тъй като полезната информация заема само част от нея. В това се състои **недостатъкът** на описаната структура.

Организация на управлението

Логическа структура на УА с принудителна адресация

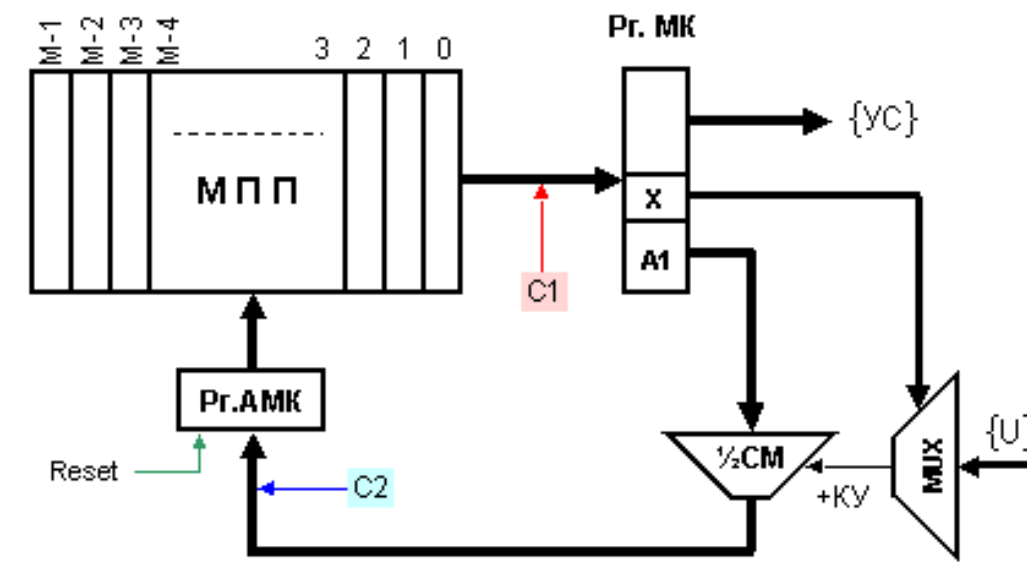
Едноадресна структура на микрокомандата

- Вторият адрес не присъства в микрокомандата, а, когато е необходим, се изчислява като функция на първия: $A_2 = f(A_1)$.
- Една от най-лесните за апаратна реализация функционални зависимости е : $A_{пр} = A_1 + KU$,
т.е. адресът за преход $A_{пр}$ е равен на A_1 при безусловен преход, тъй като тогава $KU=0$, или е равен на сумата (A_1+1) , тъй като тогава $KU=1$.

Организация на управлението

Логическа структура на УА с принудителна адресация

Примерна логическа структура



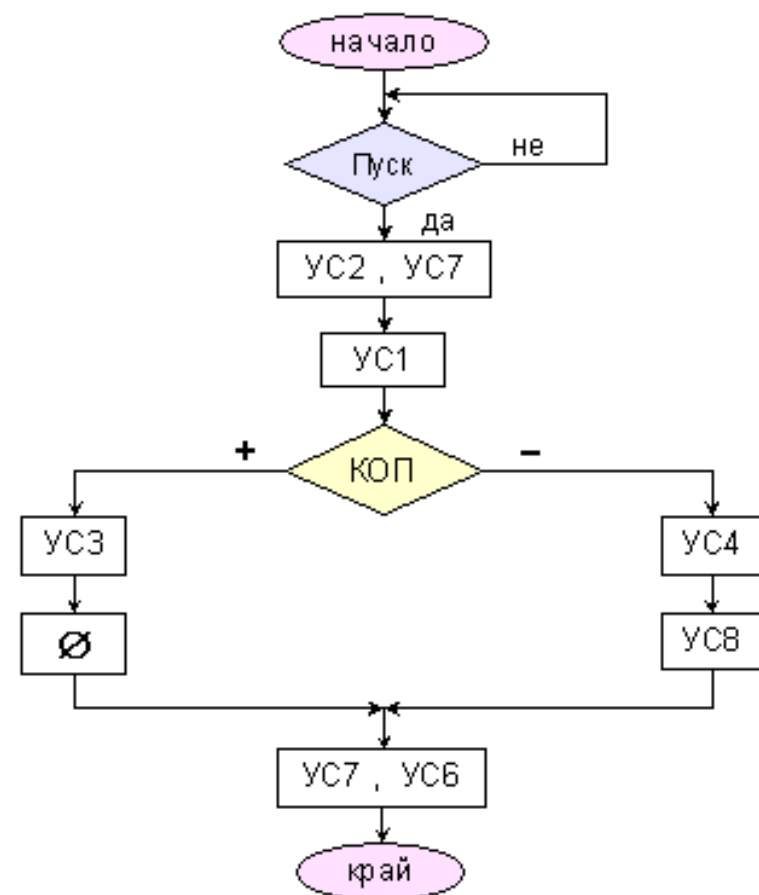
Когато микрокомандата е за условен преход, чрез полето X в мултиплексора на оповестяващите сигнали **MUX**, се избира за проверка съответният оповестяващ сигнал. Ако последният не е активен, то $KY=0$ и тогава адресът за преход е $A_{пр}=A1$. Ако сигнаът е активен, то $KY=1$ и $A_{пр}=(A1+1)$.

- Тази конструктивна реализация означава, че когато при програмиране се употребява микрокомандата за условен преход, в адресното пространство е необходима **свободна двойка съседни клетки**. Ако в адресното пространство не могат да се намерят два съседни свободни адреса, условният преход не може да бъде реализиран.

Организация на управлението

Логическа структура на УА с принудителна адресация

Блок-схема на управляващ алгоритъм на микропрограмата за събиране и изваждане



Формат на микрокомандата

УС		X		A1		A2	
8	1	2	1	3	1	3	1

Структура на мултиплексорите: **MUX1** - (2x3) към (1x3); **MUX2** - 4 към 1.

Входовете на **MUX2** и съответните им комбинации:

X=00 : “Пуск”; X=01 : КОП; X=10 и X=11 : Нула.

Организация на управлението

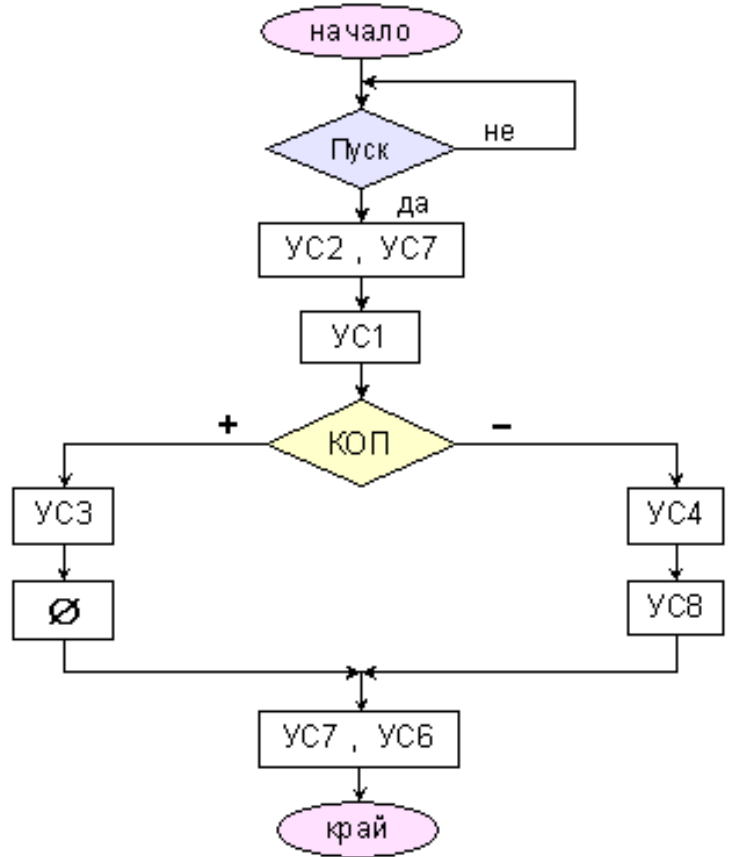
Логическа структура на УА с принудителна адресация

Формат на микрокомандата

УС		X	A1		A2	
8	1	2 1	3	1	3	1

X=00 : "Пуск" ; X=01 : КОП ; X=10 и X=11 : Нула.

Код (текст) на микропрограмата:



		Съдържание на МПП				Коментар на микрокомандата
		Управляващи сигнали	X	A1	A2	
№	адрес	8 7 6 5 4 3 2 1	2 1	3 2 1	3 2 1	
0	000	0 0 0 0 0 0 0 0	0 0	0 0 0	0 0 1	Ако "Пуск"=0, то преход по A1 – очакване ; Ако "Пуск"=1, то преход по A2 – старт .
1	001	0 1 0 0 0 0 1 0	1 1	0 1 0	---	Безусловен преход по A1, сигнали УС2, УС7.
2	010	0 0 0 0 0 0 0 1	0 1	0 1 1	1 1 0	Ако КОП=0, то преход по A1, сигнал УС1 ; Ако КОП=1, то преход по A2, сигнал УС1 .
3	011	0 0 0 0 0 1 0 0	1 1	1 0 0	---	Безусловен преход по A1, сигнал УС3, (+).
4	100	0 0 0 0 0 0 0 0	1 1	1 0 1	---	Безусловен преход по A1, празен такт.
5	101	0 1 1 0 0 0 0 0	1 1	0 0 0	---	Безусловен преход по A1, сигнали УС6, УС7.
6	110	0 0 0 0 1 0 0 0	1 1	1 1 1	---	Безусловен преход по A1, сигнал УС4. (-).
7	111	1 0 0 0 0 0 0 0	1 1	1 0 1	---	Безусловен преход по A1, сигнал УС8

Организация на управлението

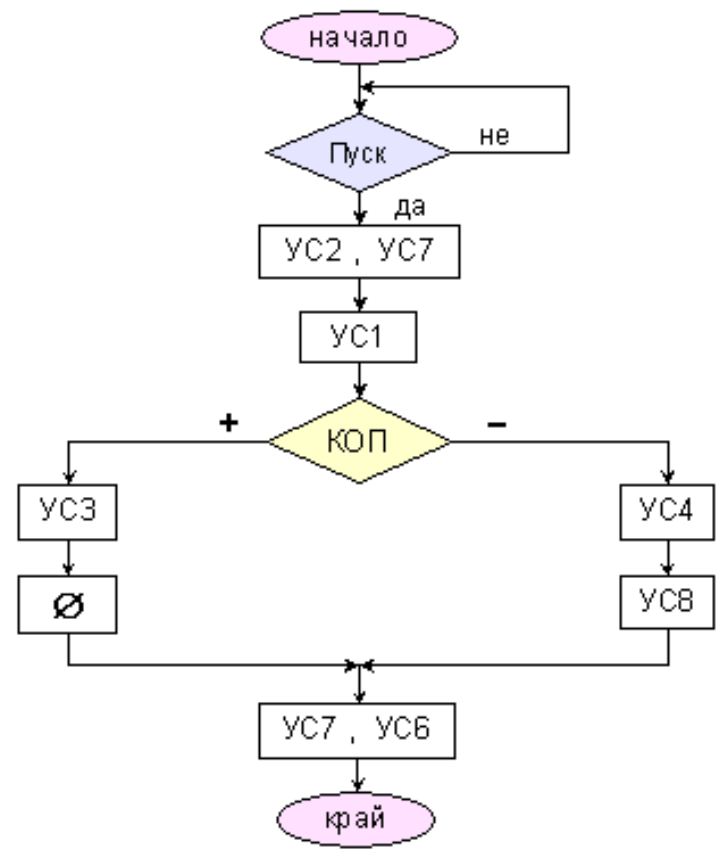
Логическа структура на УА с принудителна адресация

Формат на микрокомандата

УС		X		A1	
8	1	2	1	3	1

Код (текст) на микропрограмата:

		Съдържание на МПП			Коментар на микрокомандата
		Управляващи сигнали	X	A1	
№	адрес	8 7 6 5 4 3 2 1	2 1	3 2 1	
0	000	0 0 0 0 0 0 0 0	0 0	0 0 0	Ако "Пуск"=0, то преход по A1 – очакване ; Ако "Пуск"=1, то преход по A1+1 – старт .
1	001	0 1 0 0 0 0 1 0	1 1	0 1 0	Безусловен преход по A1, сигнали УС2, УС7.
2	010	0 0 0 0 0 0 0 1	0 1	0 1 1	Ако КОП=0, то преход по A1, сигнал УС1 ; Ако КОП=1, то преход по A1+1, сигнал УС1 .
3	011	0 0 0 0 0 1 0 0	1 1	1 1 0	Безусловен преход по A1, сигнал УС3, (+).
4	100	0 0 0 0 1 0 0 0	1 1	1 0 1	Безусловен преход по A1, сигнал УС4. (-).
5	101	1 0 0 0 0 0 0 0	1 1	1 1 1	Безусловен преход по A1, сигнал УС8
6	110	0 0 0 0 0 0 0 0	1 1	1 1 1	Безусловен преход по A1, празен такт.
7	111	0 1 1 0 0 0 0 0	1 1	0 0 0	Безусловен преход по A1, сигнали УС6, УС7.



Организация на управлението

Логическа структура на УА с естествена адресация

- Управляващият автомат трябва да работи с микрокоманди, които имат **различна структура** – такива, които ще подават само управляващи сигнали и такива, които ще реализират различните преходи.

За различаване на двата вида микрокоманди в структурата им се включва отделно поле за код на вида – **микрокоманден код (МКК)**.

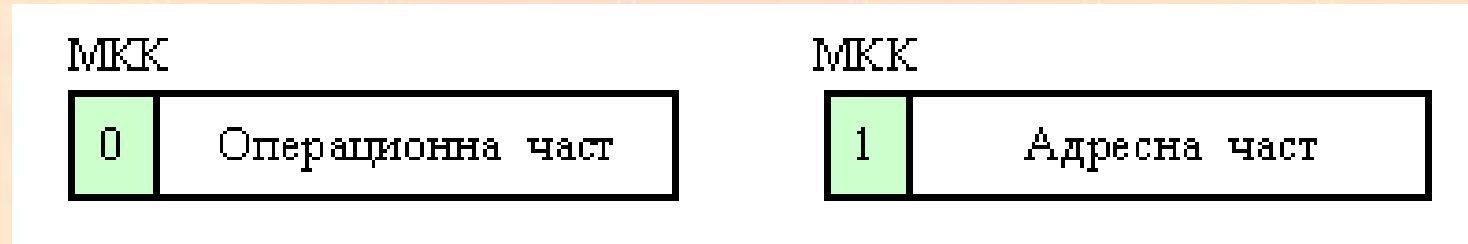
Кодирането е следното:

- ✓ $МКК=1$ – изпълнима микрокоманда (за безусловен преход);
- ✓ $МКК=0$ – микрокоманда за управление на прехода (за условен преход).

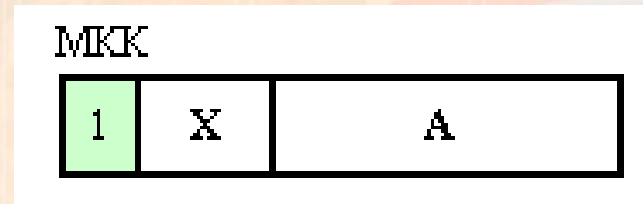
Организация на управлението

Логическа структура на УА с естествена адресация

- Останалата част от структурата на микрокомандата се интерпретира според кода ѝ, както следва:



- Адресната част на микрокомандата за управление на прехода при метода на естественото адресиране има структурата:



където с X е означен кодът за избор на условието за преход, с A е означен адресът за преход.

Организация на управлението

Логическа структура на УА с естествена адресация

- Адрес за преход при "изпълнима" микрокоманда:

$$\text{Рг.АМК} := (\text{Рг.АМК}) + 1$$

Реализацията на този закон изисква брояч.



Организация на управлението

Логическа структура на УА с естествена адресация

- Ако извлечената микрокоманда е от вида "за преход", то адресът за преход се формира според стойността на кода на проверяваното условие КУ и записаната в полето А на микрокомандата стойност.

Възможни са следните случаи:

1) Когато е употребен код за безусловен преход (БП), $X=БП$, то винаги $KУ=1$ и адресният регистър получава съдържанието:

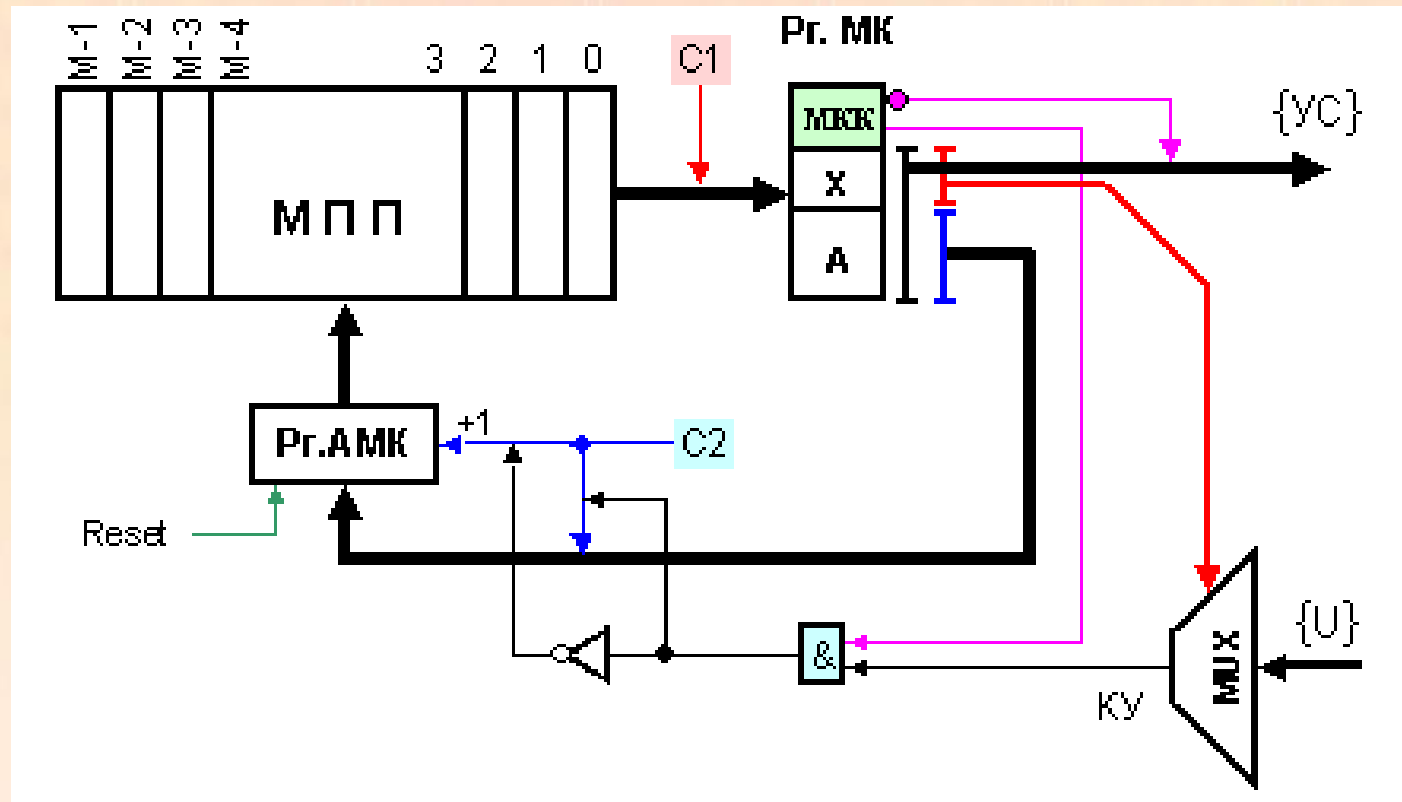
$$Pг.АМК := A$$

2) Когато е употребен код за условен преход (УП), $X=УП$, то адресът за преход е функция от стойността на кода на условието КУ, както следва:

$$\text{ако } КУ = \begin{cases} 0, & \text{то } Pг.АМК := (Pг.АМК) + 1; \\ 1, & \text{то } Pг.АМК := A. \end{cases}$$

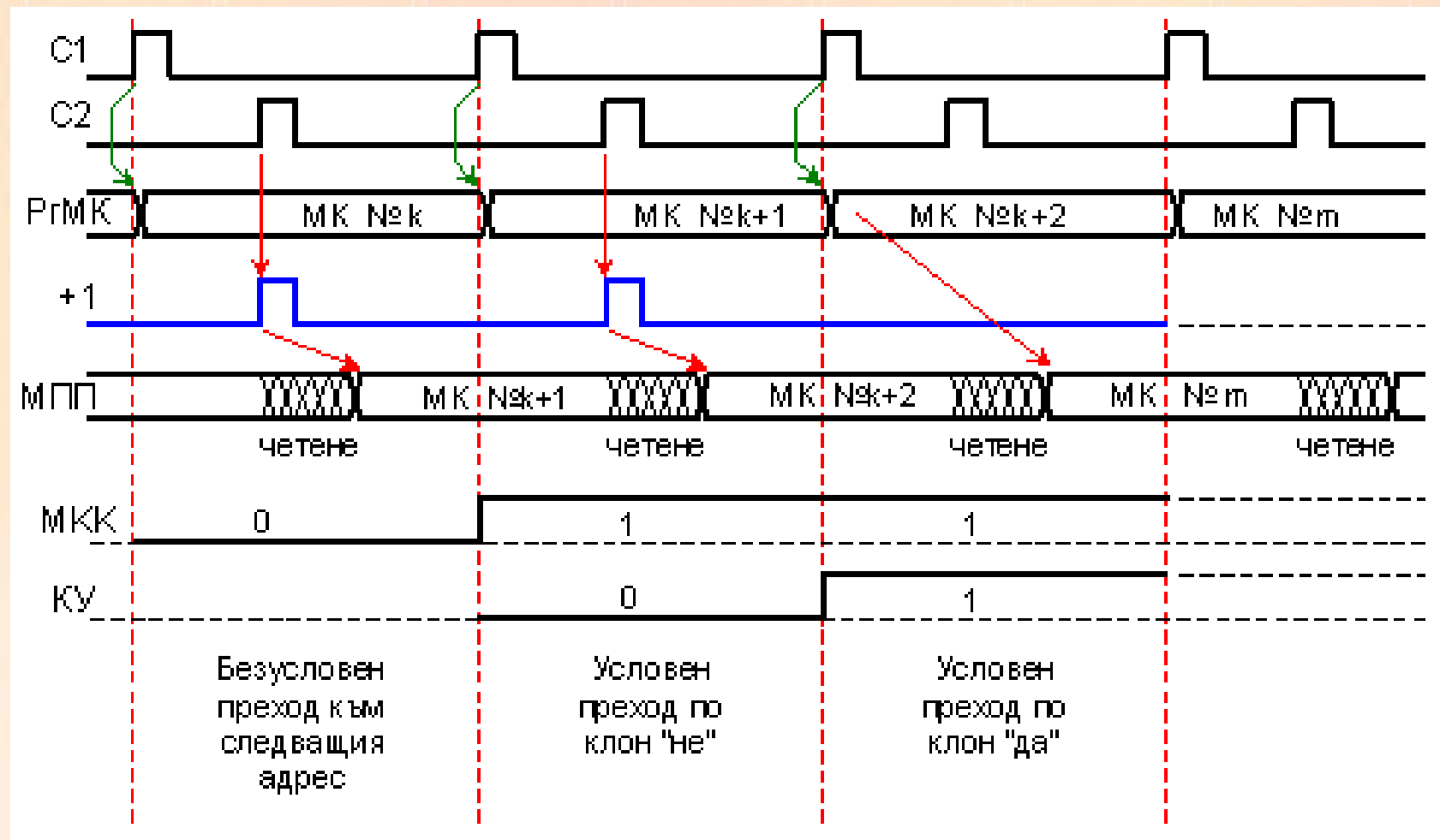
Организация на управлението

Логическа структура на УА с естествена адресация



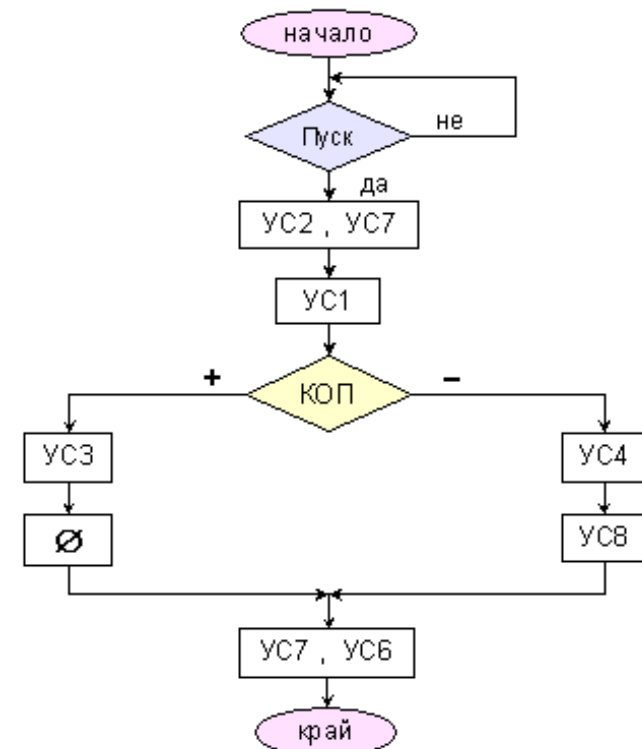
Организация на управлението

Превключвания в МПУА с естествена адресация



Организация на управлението

Микропрограма в МПУА с естествена адресация



		Съдържание на МПП			Коментар на микрокомандата
		МКК	Х	А	
№	адрес	9	8 7 6 5	4 3 2 1	
0	0000	1	0 0 0 0	0 0 1 0	Ако "Пуск"=0, то преход към следващия поред адрес ; Ако "Пуск"=1, то преход към А=2 – старт .
1	0001	1	0 0 1 1	0 0 0 0	Безусловен преход към адрес А=0 .
2	0010	0	0 1 0 0	0 0 1 0	БП към следващия поред адрес, сигнали УС2, УС7.
3	0011	0	0 0 0 0	0 0 0 1	БП към следващия поред адрес, сигнал УС1 .
4	0100	1	0 0 0 1	1 0 0 1	Ако "КОП"=0, то преход към следващия поред адрес ; Ако "КОП"=1, то преход към адрес А=9 .
5	0101	0	0 0 0 0	0 1 0 0	БП към следващия поред адрес, сигнал УС3 .
6	0110	0	0 0 0 0	0 0 0 0	БП към следващия поред адрес, празен такт .
7	0111	0	0 1 1 0	0 0 0 0	БП към следващия поред адрес, сигнали УС6, УС7.
8	1000	1	0 0 1 1	0 0 0 0	БП към адрес А=0 .
9	1001	0	0 0 0 0	1 0 0 0	БП към следващия поред адрес, сигнал УС4 .
10	1010	0	1 0 0 0	0 0 0 0	БП към следващия поред адрес, сигнал УС8 .
11	1011	1	0 0 1 1	0 1 1 1	БП към адрес А=7 .

Организация на управлението

Оптимизация на операционната част на микрокомандата

- **Хоризонтално кодиране** - двоична комбинация от стойностите на управляващите сигнали в операционната част на микрокомандата (за всеки управляващ сигнал в операционната част се отделя един разряд)

✓ **Предимства:**

Позволява изобразяването на повече от един активен сигнал, т.е. позволява паралелно изпълнение на много микрооперации, стига те да са възможни в структурата на операционното устройство.

По този начин хоризонталното кодиране позволява да се реализира в пълна степен бързодействието, което конструкторът на операционното устройство е постигнал в алгоритъма на неговото функциониране.

Организация на управлението

Оптимизация на операционната част на микрокомандата

- **Хоризонтално кодиране** - двоична комбинация от стойностите на управляващите сигнали в операционната част на микрокомандата

✓ **Недостатъци:**

Операционната част на микрокомандата е твърде голяма по дължина.

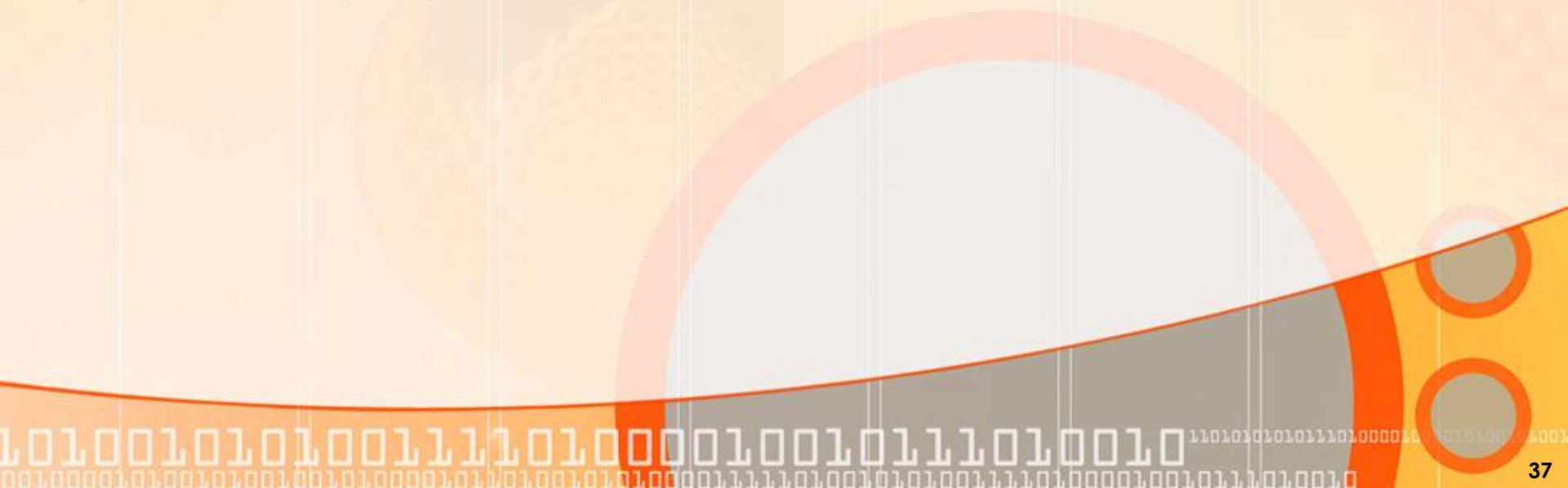
Голяма част разрядите в операционната част съдържат *неактивни* управляващи сигнали.

Организация на управлението

Оптимизация на операционната част на микрокомандата

Ако бързодействието на управляващия автомат не е от съществено значение за неговите функции, то като първостепенен критерий при синтеза на формата на микрокомандата може да остане **минимизирането** на нейната операционна част.

Това означава **минимизиране на общите апаратни разходи** при реализация на автомата.



Организация на управлението

Оптимизация на операционната част на микрокомандата

Методът, при който в операционната част на микрокомандата не се изобразява множеството управляващи сигнали, а неговият двоичен код, се нарича **вертикално кодиране**.

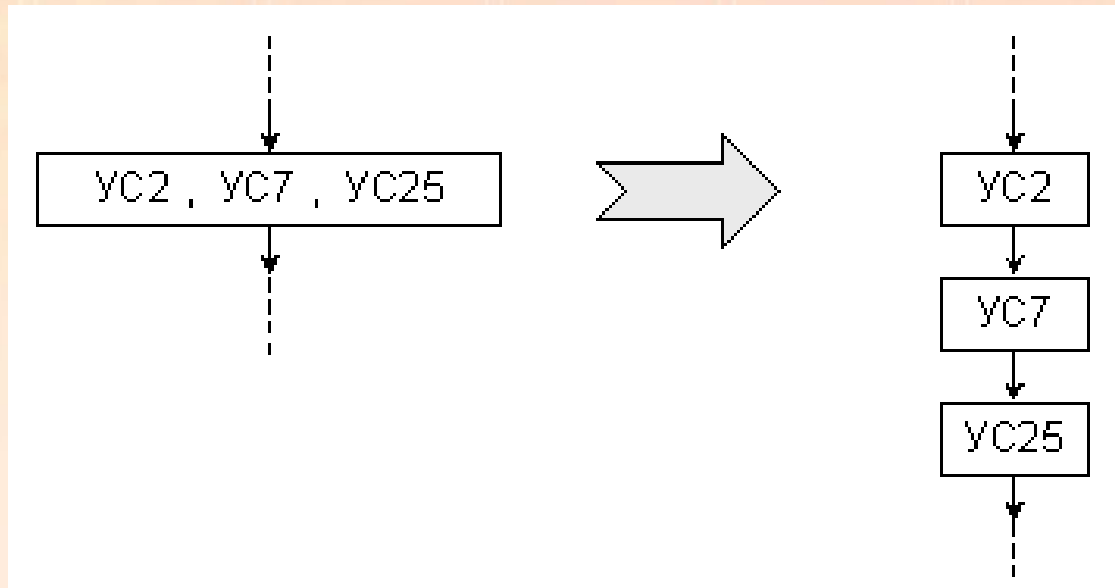
Всяка микрокоманда може да заповядва само една микрооперация, т.е. да подава само един управляващ сигнал, тъй като в нейната операционна част може да бъде записана само една кодова комбинация. Това означава, че всяка комбинация от едновременно възможни за изпълнение микрооперации, ще се разпадне на няколко последователни, които ще се подредят вертикално в блок-схемата на алгоритъма. С други думи този метод изисква **декомпозиция на операционните блокове**.

Организация на управлението

Оптимизация на операционната част на
микрокомандата

Вертикално кодиране

Вертикална декомпозиция на микрокомандата



Организация на управлението

Оптимизация на операционната част на микрокомандата

Вертикално кодиране

Предимства:

- Свободните кодови комбинации могат да се използват за реализиране на микрокоманди без управляващи сигнали, т.е. на празен такт (каквото в посочения алгоритъм е необходим).
- Съставянето на микропрограмата в този случай на кодиране **не зависи от структурата на управляващия автомат**, което означава, че то може да бъде приложено както в структури с принудително адресиране, така и в структури с естествено адресиране, а така също и в смесени.

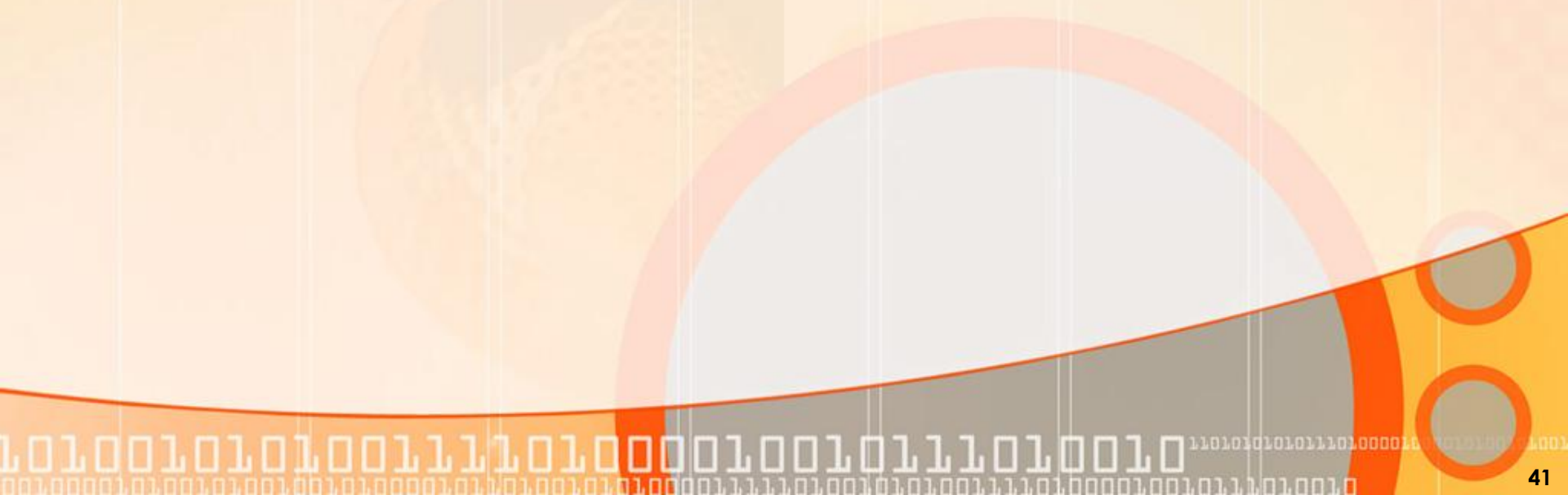
Организация на управлението

Оптимизация на операционната част на
микрокомандата

Вертикално кодиране

Недостатъци:

- ✓ Снизено бързодействие;
- ✓ Изисква допълнителен логически възел - дешифратор



Организация на управлението

Оптимизация на операционната част на
микрокомандата

*Смесени методи за кодиране на управляващите
сигнали:*

- вертикално-хоризонтален
- хоризонтално-вертикален.

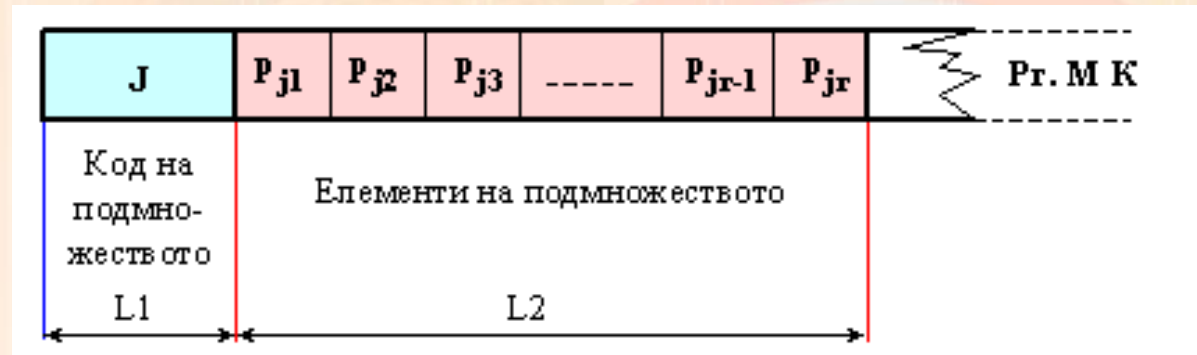


Организация на управлението

Оптимизация на операционната част на микрокомандата

- При вертикално-горизонталния метод за кодиране принадлежността към съответното подмножество j се кодира вертикално, а поредният номер на елемента r , се кодира хоризонтално.

В полето ОЧ на микрокомандата се определят две подполета - едно за изобразяване на номера на подмножеството и второ за изобразяване на елементите на подмножествата.

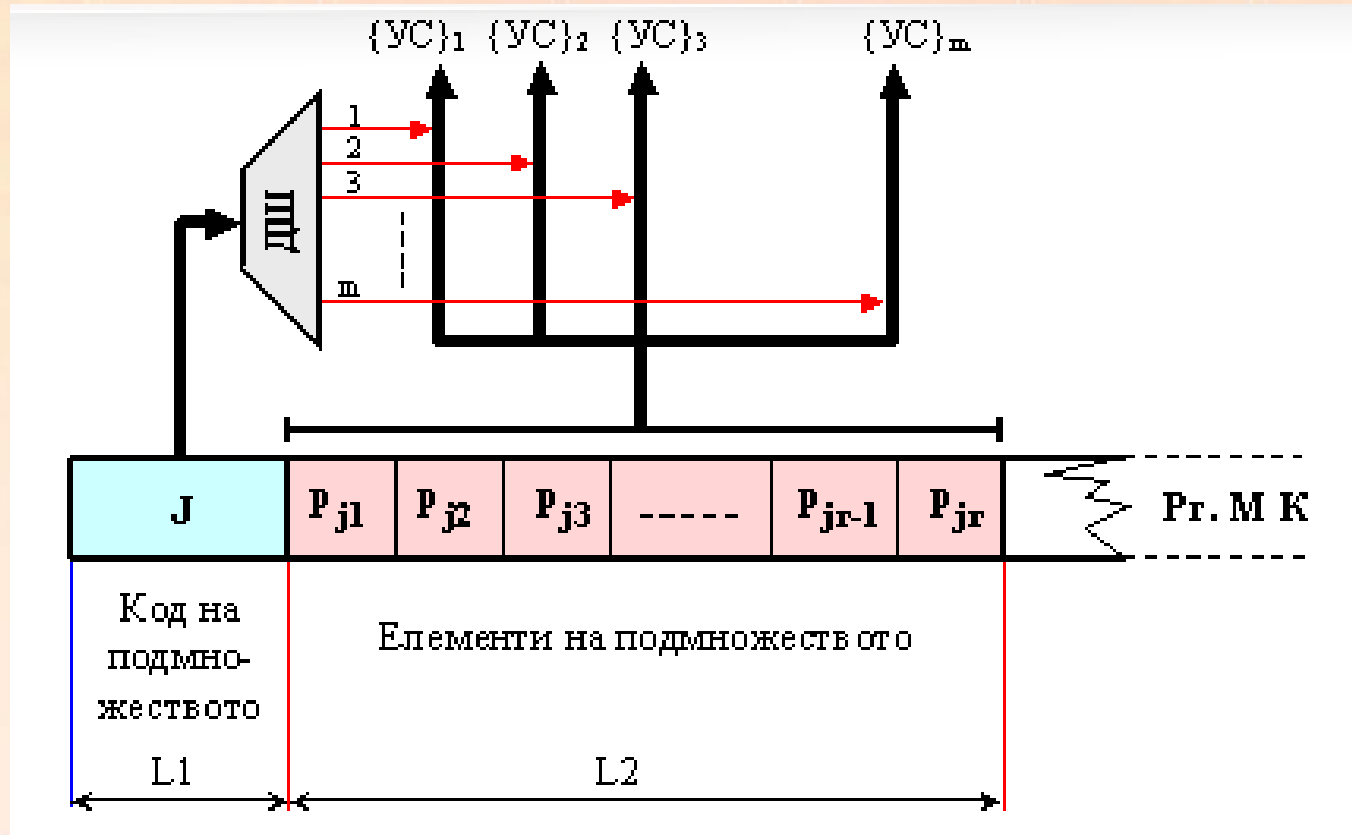


Структура на операционната част на микрокоманда при вертикално-горизонтално кодиране

Организация на управлението

Оптимизация на операционната част на микрокомандата

Формиране на управляващите сигнали при вертикално-горизонтално кодиране



Организация на управлението

Оптимизация на операционната част на микрокомандата

При хоризонтално-вертикалния метод за кодиране, принадлежността към съответното подмножество j се кодира хоризонтално, а поредният номер на елемента r , се кодира вертикално.

Това означава, че в полето на операционната част на микрокомандата се определят толкова подполета, колкото са подмножествата, тъй като при хоризонталното им кодиране те трябва да бъдат изобразени едновременно.

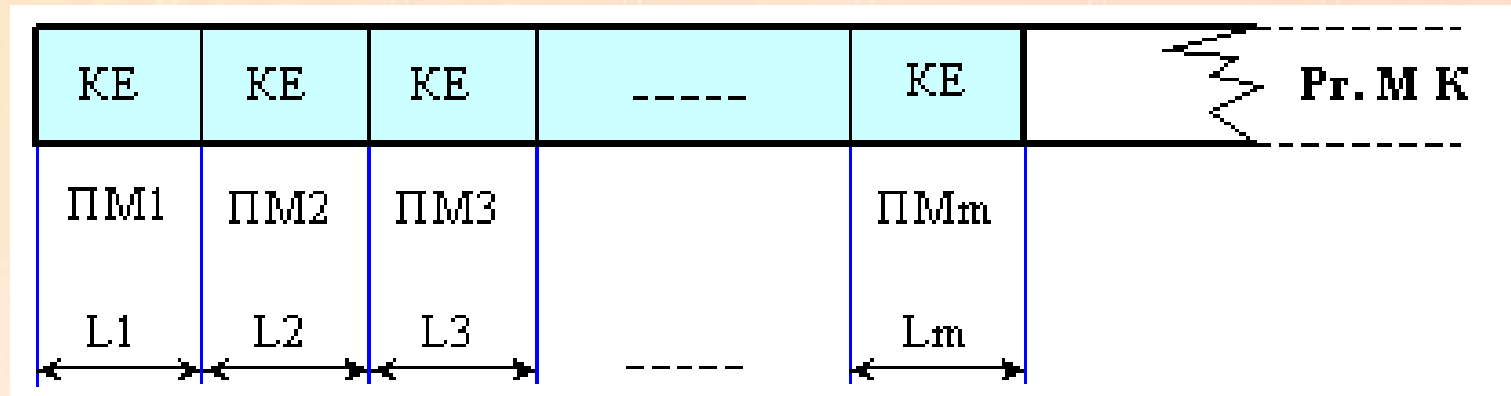
Във всяко подполе за изобразяване на подмножество се съдържа код на активния в момента елемент, представляващ управляващ сигнал.



Организация на управлението

Оптимизация на операционната част на микрокомандата

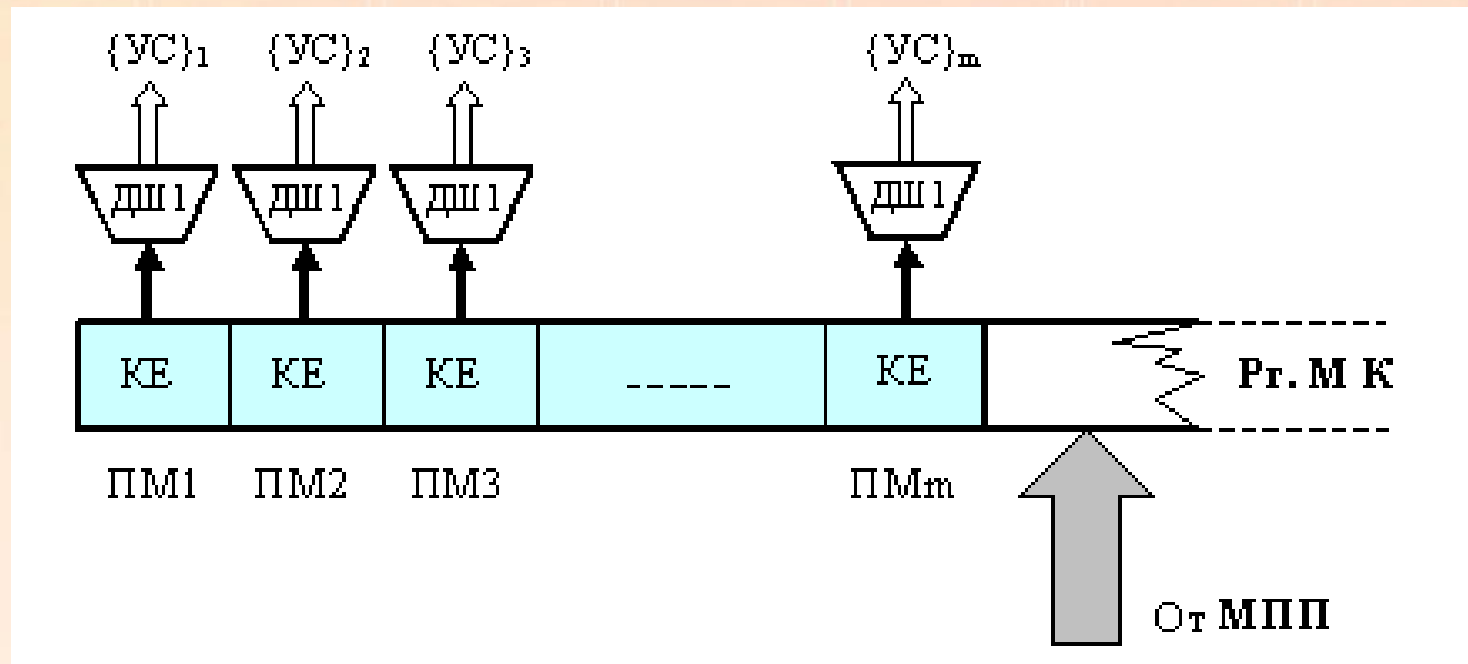
Структура на операционната част на микрокоманда при хоризонтално-вертикално кодиране



Организация на управлението

Оптимизация на операционната част на микрокомандата

Дешифриране на управляващите сигнали при
хоризонтално-вертикално кодиране



Литература

[1]. <http://tyanev.com/> - On-line книги – ОРГАНИЗАЦИЯ НА КОМПЮТЪРА – книга [1]

Глава 7. ОРГАНИЗАЦИЯ НА УПРАВЛЕНИЕТО

- [7.1. Общи положения](#)
- [7.2. Логическа структура на управляващи автомати с програмно закрепена логика](#)
 - [7.2.1. Микропрограмен автомат с принудителна адресация](#)
 - [7.2.2. Микропрограмен автомат с естествена адресация](#)
- [7.3. Оптимизация на операционната част на микрокомандата](#)
- [7.4. Оптимизация на адресната част на микрокомандата](#)

[2]. Димитър Тянев, ОРГАНИЗАЦИЯ НА КОМПЮТЪРА, том първи (ISBN 978-954-20-0412-7), Варна 2008г.

