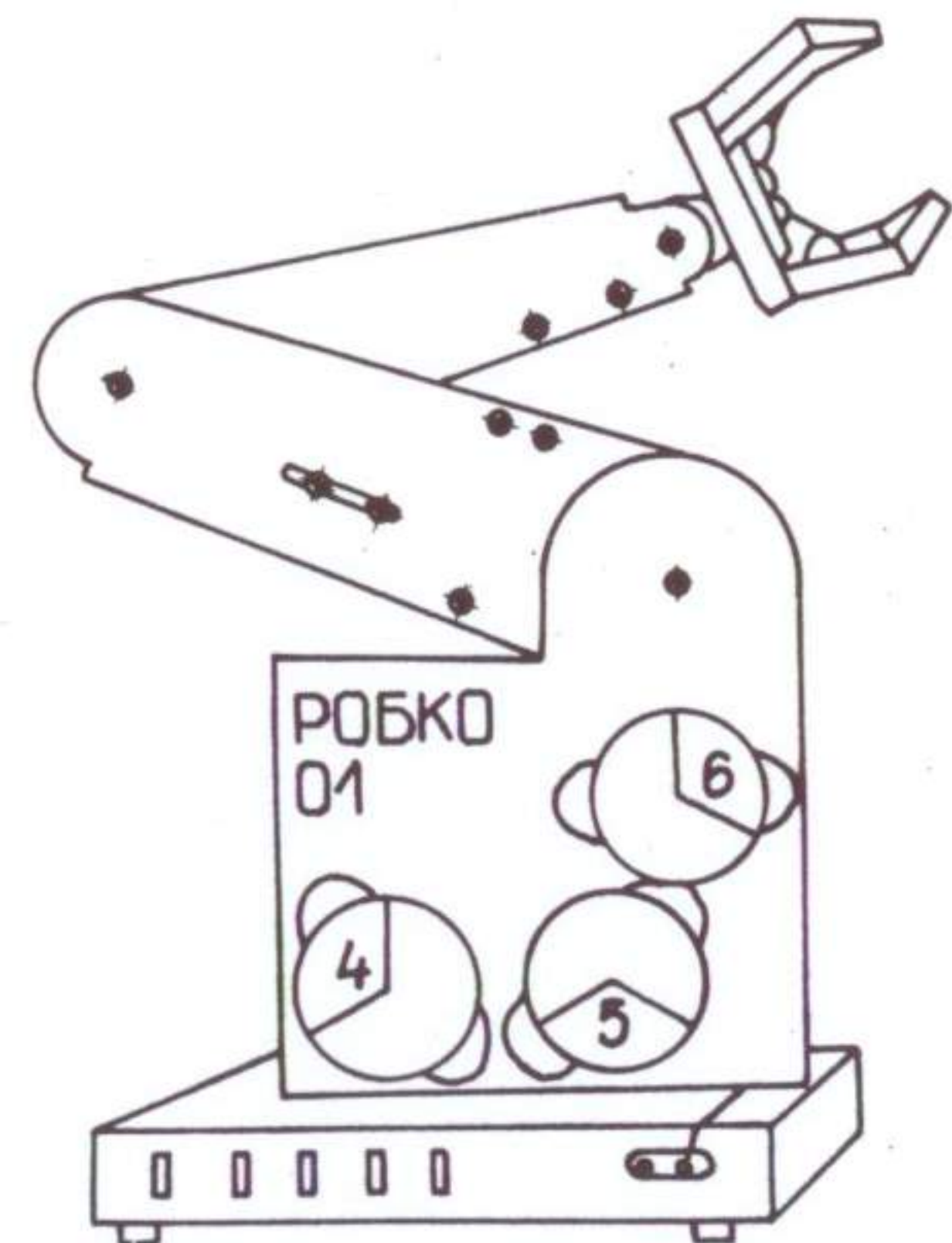


Иван Абрамов

МИНИСТЕРСТВО НА МАШИНОСТРОЕНЕТО  
ДСО "ПРИБОРОСТРОЕНЕ И АВТОМАТИЗАЦИЯ"  
ЗАВОД ЗА МЕДИЦИНСКА ТЕХНИКА  
ИТКР-БАН

МИНИ РОБОТ  
РОБКО-01

РУКОВОДСТВО ЗА ЕКСПЛОАТАЦИЯ



1. МЕХАНІЧНА ЧАСТ



## 1. МЕХАНИЧНА ЧАСТ

В тази глава са описани основните елементи на механичната част на миниробота РОБКО - 01.

РОБКО - 01 е антропоиден тип робот с шест степени на свобода, осъществени чрез ротационни кинематични връзки от пети клас.

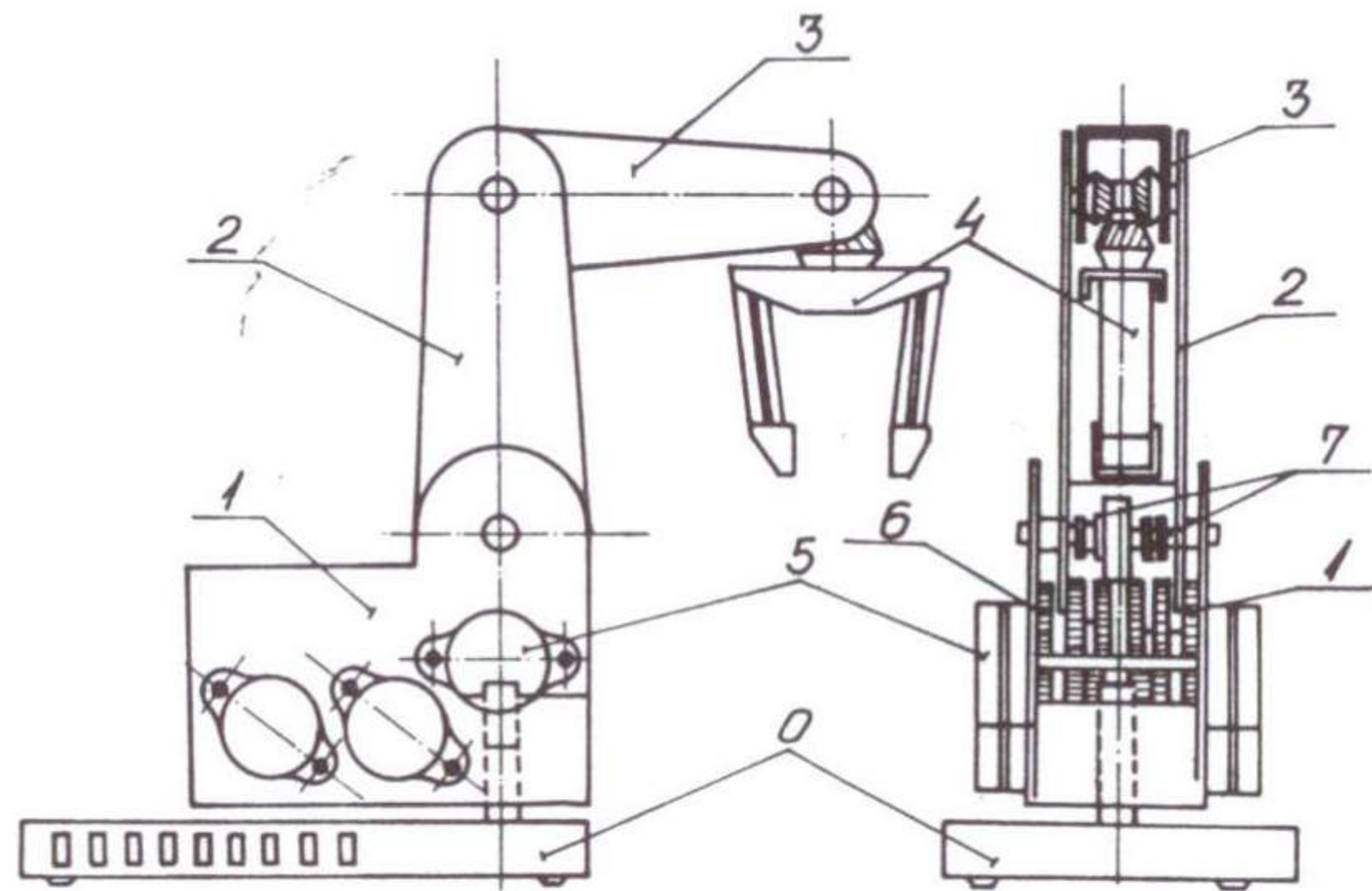
Всички двигатели на робота са монтирани върху първото звено. Движението към звената се предава с помощта на система от въжета и ролки от осите на зъбни редуктори, съединени с двигателите. Зъбните колела и ролките са изработени от пластмаса, което олекотява конструкцията.

Външният вид на робота е показан на фиг. 1.1. С позиция 0 е означена базата (основата), която е неподвижна.

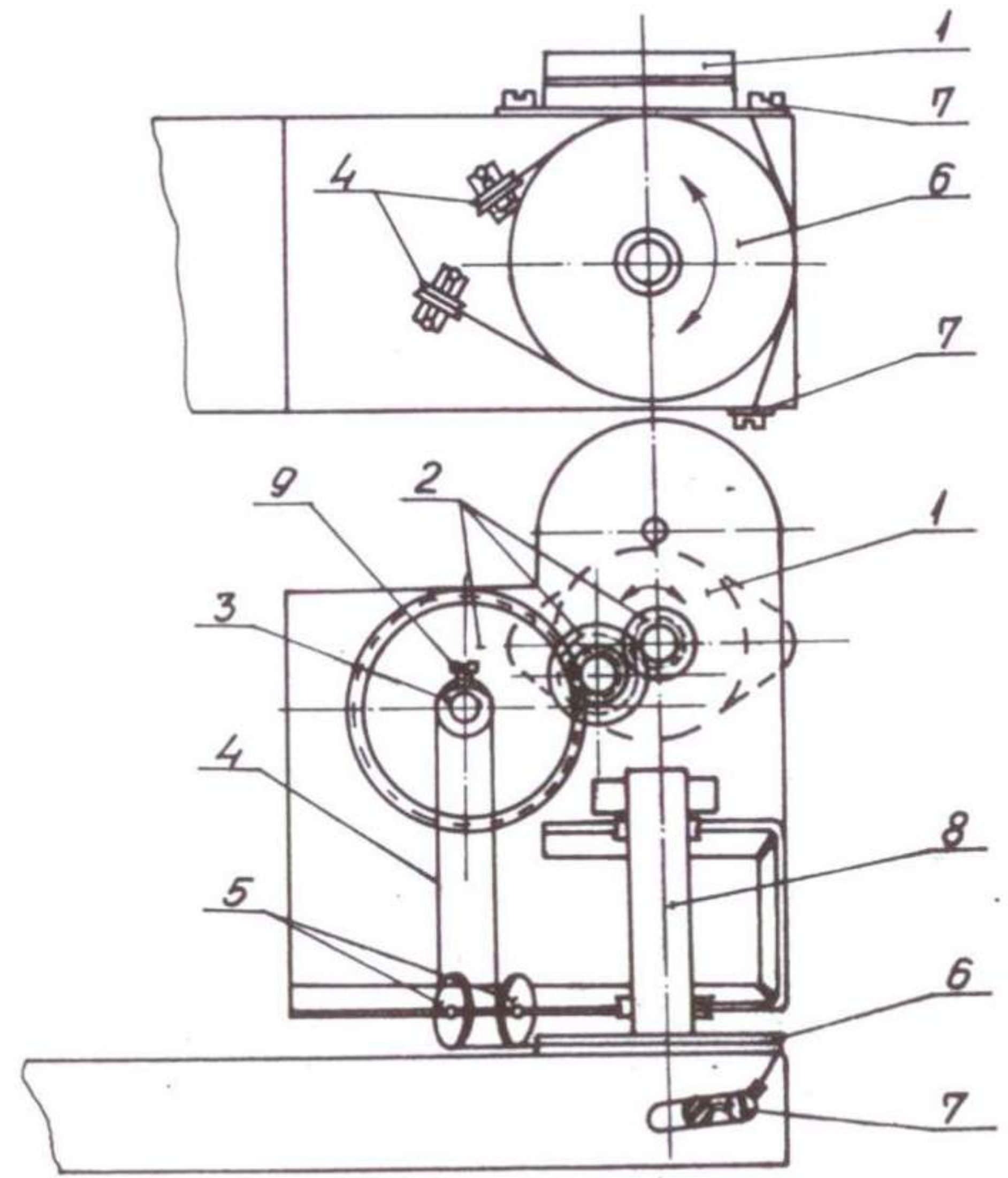
С 1 е означено първото кинематично звено. На него са монтирани всички електрозадвижвания, състоящи се от стъпковите електродвигатели 5, механичните зъбни редуктори 6 и предавателните ролки 7. Това звено е най-тежко и натоварва най-много задвижването. На фиг. 1.2 е показан начинът на механичното свързване на първото звено към базата.

Движението на оста на двигателя 1 се предава на механичния редуктор, съставен от зъбните колела 2. Около ролката 3, която е неподвижно закрепена към последното зъбно колело 2, се намотава един път въжето 4 и се фиксира с винта 9. Двата края на въжето преминават





Фиг. 1.1



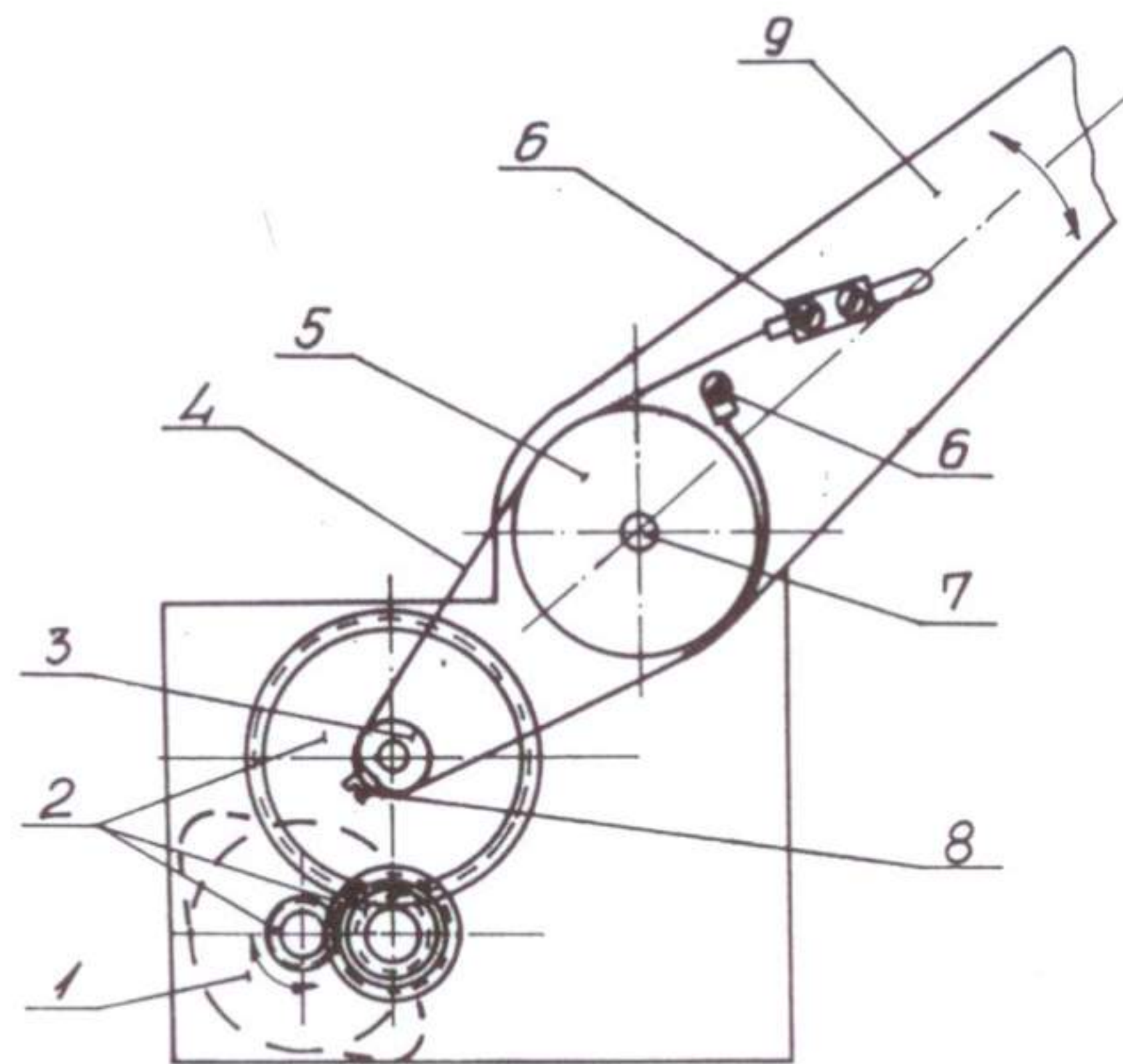
Фиг. 1.2



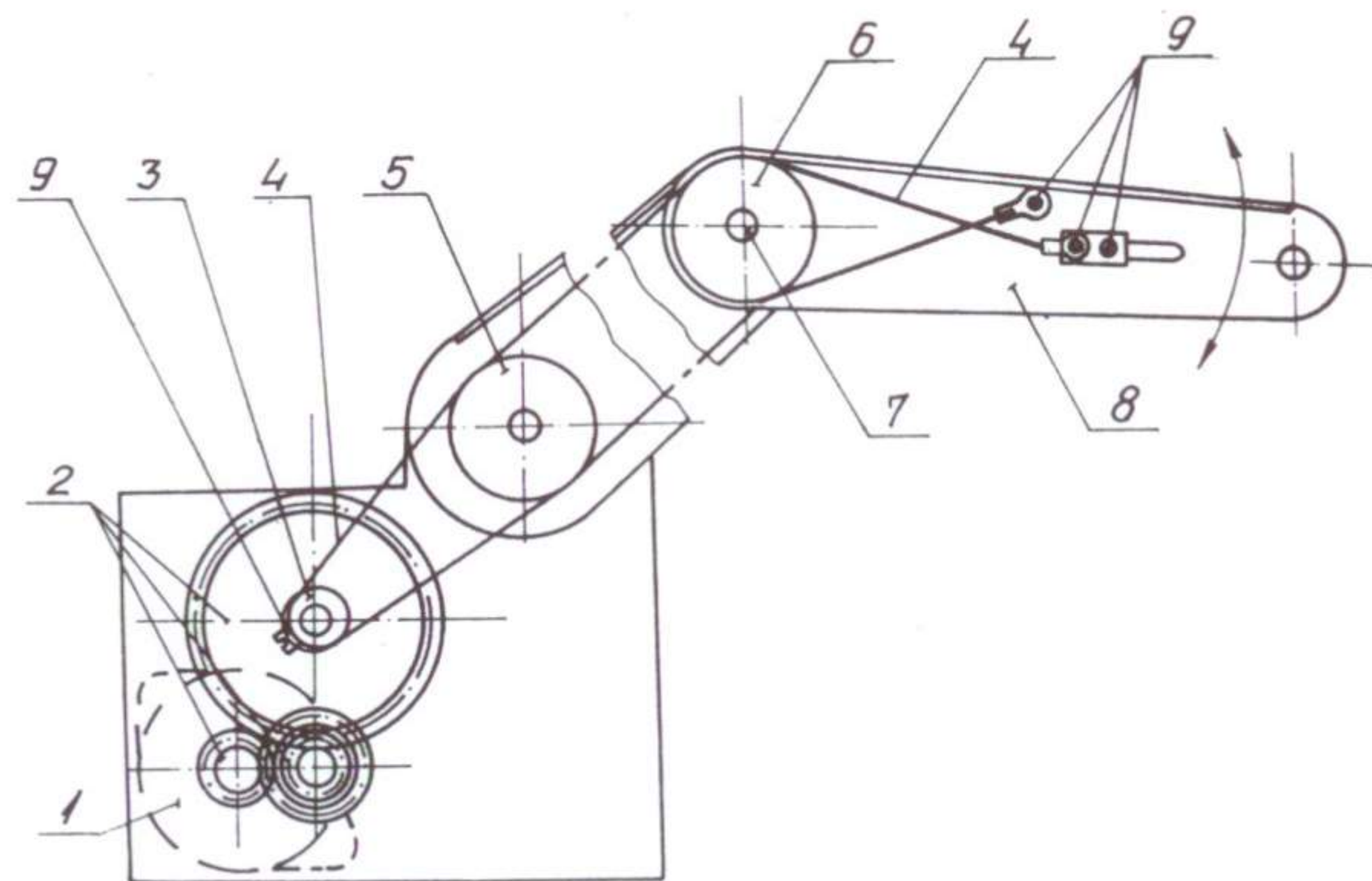
през ролките 5 и обхващайки голямата ролка 6 на базата, се закрепват с помощта на винтовете 7 към нея. По този начин при въртенето на оста на двигателя 1 в една или друга посока се осъществява движението на първото звено около вертикалната ос 8.

На фиг. 1.3 е показано механичното свързване на второто звено към първото. Движението на оста на двигателя 1 се предава на аналогичен механичен редуктор, съставен от зъбните колела 2. На последното зъбно колело (имамо най-голям диаметър) е закрепена неподвижно ролката 3, около която е намотано два пъти въжето 4 и е фиксирано посредством винта 8. Единият край на въжето обхваща отчасти голямата ролка 5, закрепена неподвижно към второто звено 9, и се фиксира към него чрез винта 6. Другият край на въжето, намотано един път около същата ролка 5, е прикрепен към второто звено с помощта на винтовете 7, като е предвидена възможност за регулиране на натягането на въжето 4. По този начин при въртенето на оста на двигателя 1 се осъществява въртенето на второто звено около оста 7.

На фиг. 1.4 е показана схемата на механичното свързване на третото звено към второто и начина на задвижването му. И в този случай се използва същият редуктор, състоящ се от комплект зъбни колела 2, които се задвижват оста на двигателя 1, монтиран на първото звено. Конструктивно последното голямо зъбно колело 2 е монтирано на обща ос с всичките еднотипни зъбни колела от досега описаните редуктори. И тук ролката 3 представлява част от зъб-



фиг. 1.3



фиг. 1.4



ното колело, и с винта 9 има задачата да фиксира въжето 4, което е намотано два пъти около нея. Една нова ролка 5 е монтирана на оста на въртене на второто звено, така че двата края на въжето 4 я обхващат и продължават към аналогична ролка 6, монтирана на оста 7 на въртене на третото звено 8. Тук единият край на въжето отчасти обхващайки ролка 6, е закрепен с помощта на винта 9 неподвижно към третото звено. Другият край е намотан един път около ролка 6 и след това също е закрепен с винтове 9 към звеното, като е предвидена възможност за регулиране на натягането на въжето 4. По този начин при въртене на оста на двигателя 1 се осъществява въртенето на третото звено 8 около оста 7. Една такава въжена предавка и наличието на ролката 5 осигуряват запазването на постоянен абсолютен ъгъл на третото звено по отношение на вертикалната ос в случай, че второто звено извършва завъртане на определен ъгъл около оста си.

Това внася определени неудобства в случай, че при координираното управление на движението на звената се използват алгоритми, отчитащи относителни ъгли на завъртане на звената. Освен това се появява нова сложна зависимост между ъгловите ограничения на движение на звеното и завъртанята на предното звено. Този ефект се нарича кинематична свързаност и трябва да бъде отчитан в програмите за управление.

Движението на хващача е реализирано като завъртаня около две взаимноперпендикулярни оси, като едната от

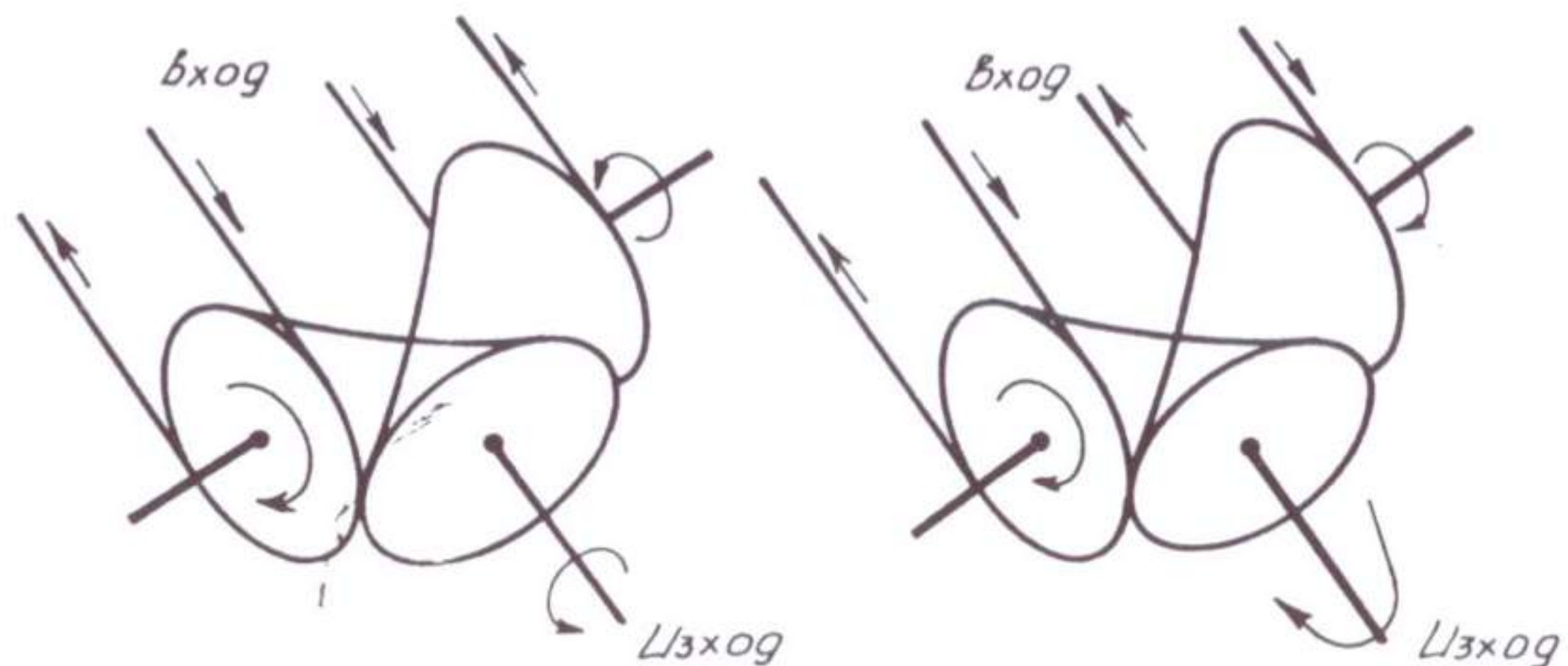
тях е успоредна на равнината ОХУ. Техническото решение представлява класическа диференциална предавка с три конусни зъбни колела. Принципът на привеждане в движение на изходното зъбно колело е пояснен на фиг. 1.5.

При въртене на зъбните колела 1 и 2 във взаимно обратни посоки около оста 4 колелото 3 се завърта около оста си 5. При въртене на колелата 1 и 2 в една посока около оста 4, оста 5 на колелото 3 се завърта около ос 4.

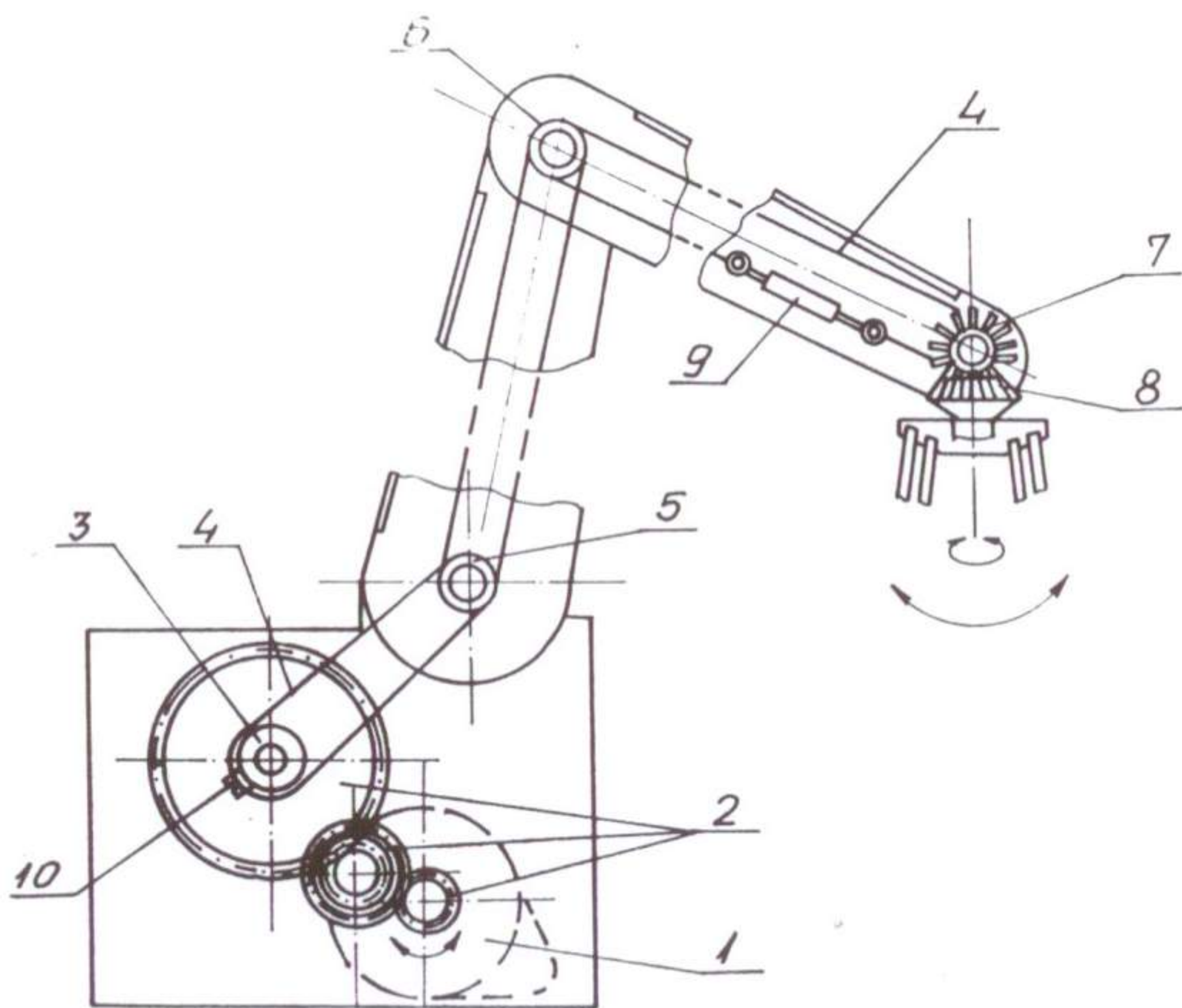
На фиг. 1.6 е показана схемата на механичното свързване на хващача към третото звено и начинът на задвижване на двойката конични зъбни колела. Всяко от тях се привежда в движение от еднотипна трансмисия. И в този случай се използва редукторът, състоящ се от зъбните колела 2, които се задвижват от оста на двигателя 1, монтиран върху първото звено. Голямото зъбно колело 2 е монтирано на общата за колелата от този тип ос. Въжето 4 е намотано два пъти около ролката 3, а винтът 10 го фиксира неподвижно към ролката. На оста на въртене на второто звено е монтирана нова ролка 5, а на оста на въртене на третото звено - ролка 6 със същите размери. Двата края на въжето 4 обхващат по един път ролките 5 и 6. Единият край, достигайки зъбното колело 7, обхваща цилиндричната му част два пъти и завършва в приспособлението 9, а другият край на въжето е присъединен непосредствено към 8, което служи за регулиране на напречността на въжето 4.

Последното управлявано движение на работа е отварянето и затварянето на хващача. Схемата на трансмисията е показана на фиг. 1.7. Използува се двигателят 1 и редукторът от зъбни колела 2. Въжето 4 е намотано един





фиг. 1.5

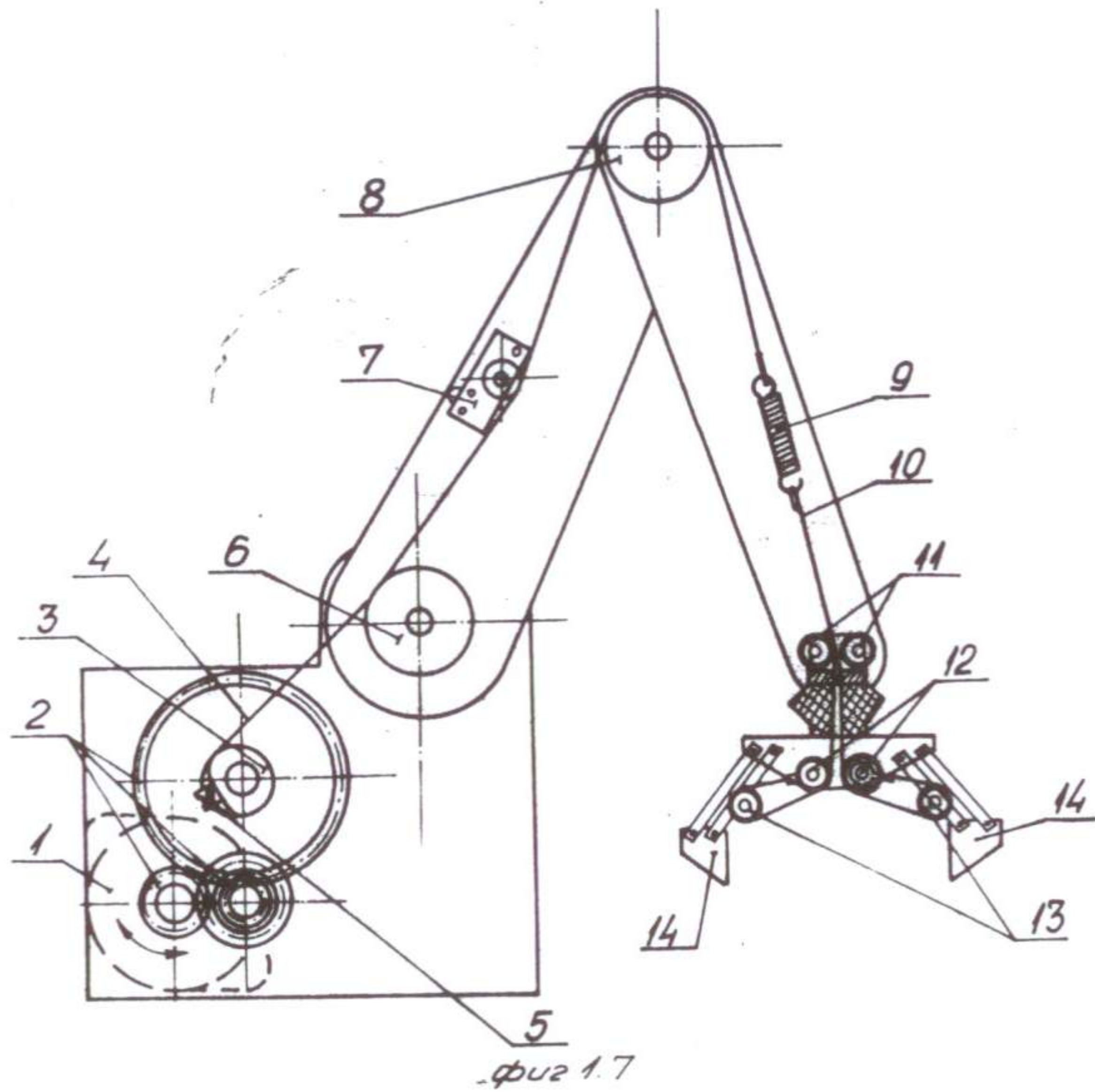


фиг. 1.6

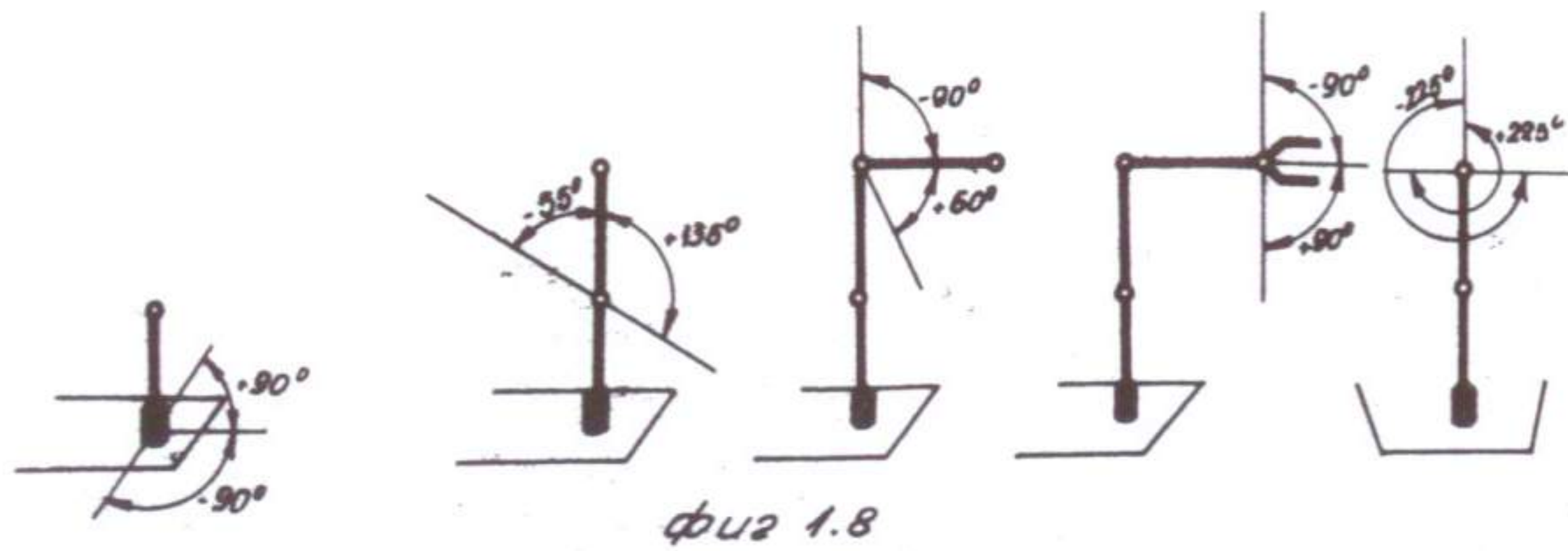
път около ролката 3, като само единият му край е застопорен чрез винта 5. Другият край обхваща един път ролките 6 и 8, монтирани съответно на осите на въртене на второто и третото звена. Между ролки 6 и 8 се намира микроизключвател 7, който подава сигнал в случай, че пръстите на хващача са затворени. Краят на въжето 4 е свързан с пружината 9, с помощта на която е възможно да се регулира силата на захвашане. Към другия край на пружината 9 е присъединено въжето 10, двата края на което, обхващайки последователно двойките ролки 11, 12 и 13, са закрепени към осите на ролки 12. При такава схема на трансмисия въртеливото движение на двигателя 1 се трансформира в постъпателно движение на пръстите 14 на хващача.

От описаните до тук трансмисии се виждат конструктивните и кинематичните ограничения, налагани върху движението на робота. Тъй като тези ограничения зависят от заеманата във всеки момент конфигурация на робота, то абсолютните им стойности (в ъгли градуси) трябва да бъдат измерени за една избрана конфигурация, наречена начално положение на робота. Началното положение е показано на фиг. 1.8 заедно със стойностите на максимално възможните ълови премествания на отделните звена.





2. СХЕМНА РЕАЛИЗАЦИЯ  
 НА РОБКО-01





## 2.1. ДРАЙВЕРНА ПЛАТКА

Управлението на движението на робота РСБКО-01 е осъществено с помощта на TTL интегрални схеми, транзистори и диоди, запоени върху платка, разположена в основата на робота (дънна платка). Платката съдържа и четири броя съединители. Три от тях са външно достъпни: съединител за входно-изходни сигнали, съединител за захранващо напрежение и съединител за връзка с компютър. Последният съединител е за вътрешна връзка между платката, стъпковите електродвигатели и датчика за контрол на хващача.

На фиг. 2.1 е представена блок-схема на платката и присъединените към нея елементи:

- 1 - адресен селектор
- 2 - регистри
- 3 - транзисторни ключове
- 4 - стъпкови електродвигатели
- 5 - формирова̀тел на магистрала за данни
- 6 - формирова̀тел на входни сигнали за външна връзка
- 7 - формирова̀тел на изходни сигнали за външна връзка.

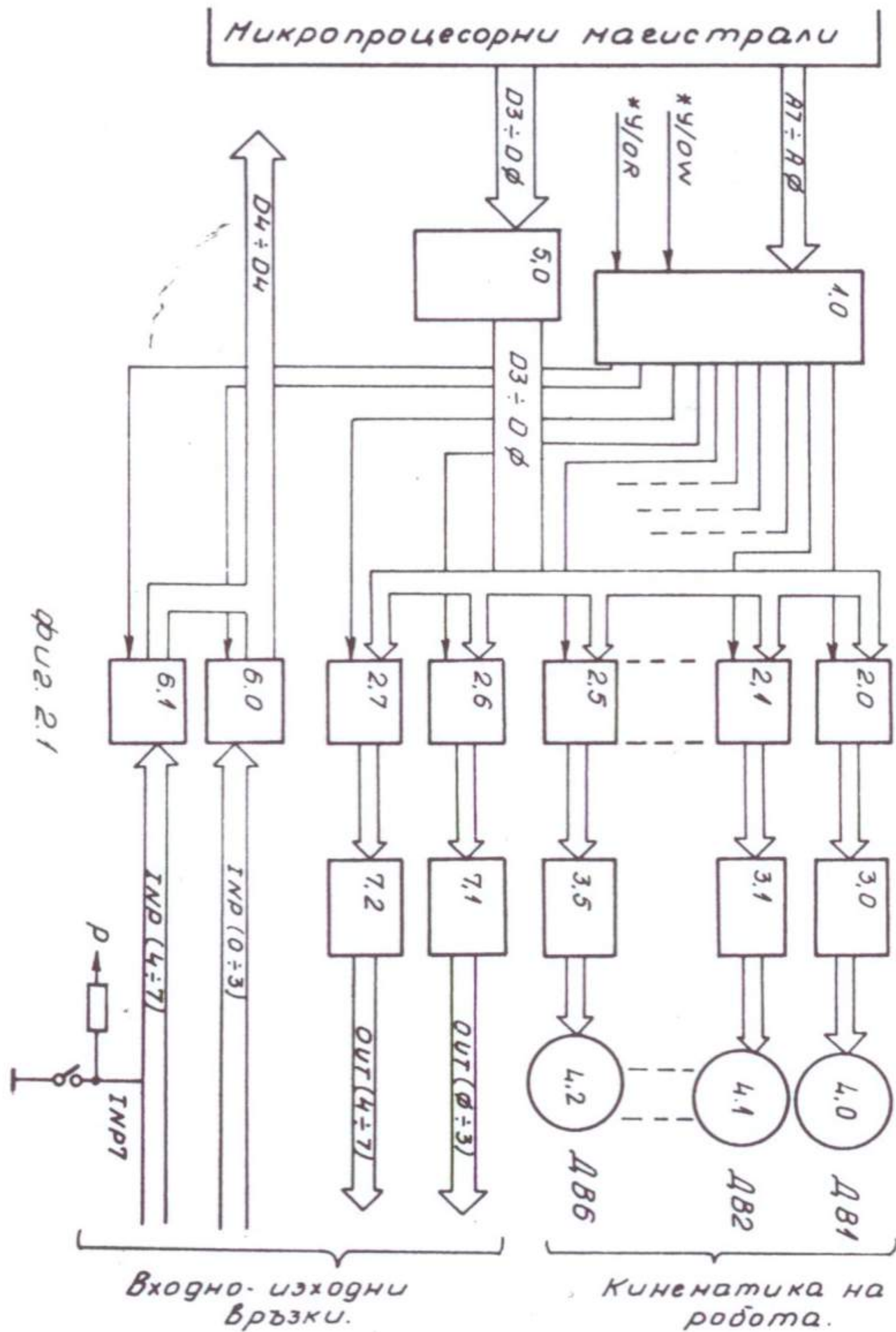
По-долу тези блокове са анализирани по реда на тяхната функционална зависимост.



## 2.1.1 СЪПКОВИ ЕЛЕКТРОДВИГАТЕЛИ

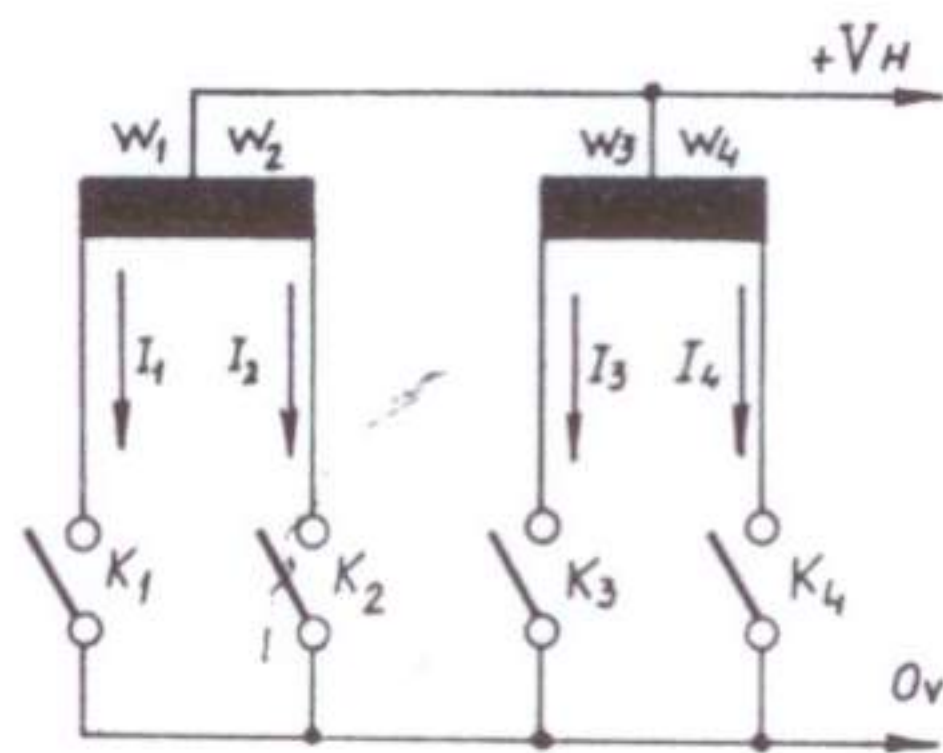
Кинематичната схема на работа се привежда в движение чрез електродвигатели, по един за всяка степен на свобода. Двигателите са стъпкови, четирифазни, с активен ротор. Роторът е постоянен магнит с множество успоредни на оста на въртене двойки магнитни полюси, равномерно разположени по цилиндричната му повърхност. Статорът е съставен от две бобини със среден извод, разположени в огънат от магнитно мек материал метален корпус. В него предварително чрез шанцоване са оформени две самостоятелни системи от множество полюсни накрайници. При протичане на постоянен ток през бобините в полюсните накрайници се образуват електромагнитни полюси, които взаимодействуват с магнитните полюси на ротора, така че той заема определено стационарно състояние. При определено превключване на изводите на бобините (фазите) в статора се създава многополюсно въртящо се магнитно поле. То обуславя въртенето на ротора, представляващо серия последователно заемани стационарни състояния (стъпки).

На фиг. 2.2 е показана принципна електрическа схема на свързване на такъв тип стъпков електродвигател. При затваряне на който и да е от ключовете К в съответната намотка се установява ток  $I$ . Ако означим наличието му с "1", а отсъствието му с "0", то задвижването на двигателя в режим "полустъпка" може да се представи чрез табл. 2.1, а в режим "цяла стъпка" - чрез табл. 2.2. В табл. 2.3 са дадени и други възможни състояния на фазо-



Фиг. 2.1





фиг. 2.2.

	$I_1$	$I_2$	$I_3$	$I_4$
	$D_0$	$D_1$	$D_2$	$D_3$
Условно избрана права посока	1	0	0	0
	1	0	1	0
	0	0	1	0
	0	1	1	0
	0	1	0	0
	0	1	0	1
	0	0	0	1
	1	0	0	1
Условно избрана обратна посока	1	0	0	0
	1	0	1	0
	0	0	1	0
	0	1	1	0
	0	1	0	0
	0	1	0	1
	0	0	0	1
	1	0	0	1

Табл. 2.1

$I_1$	$I_2$	$I_3$	$I_4$
$D_0$	$D_1$	$D_2$	$D_3$
1	0	1	0
0	1	1	0
0	1	0	1
1	0	0	1

Табл. 2.2

	$I_1$	$I_2$	$I_3$	$I_4$
	$D_0$	$D_1$	$D_2$	$D_3$
Изключен двигател.	0	0	0	0
Неразре- шено съ- стояние.	1	1	X	$\bar{X}$
	X	$\bar{X}$	1	1
	1	1	1	1

Табл. 2.3.

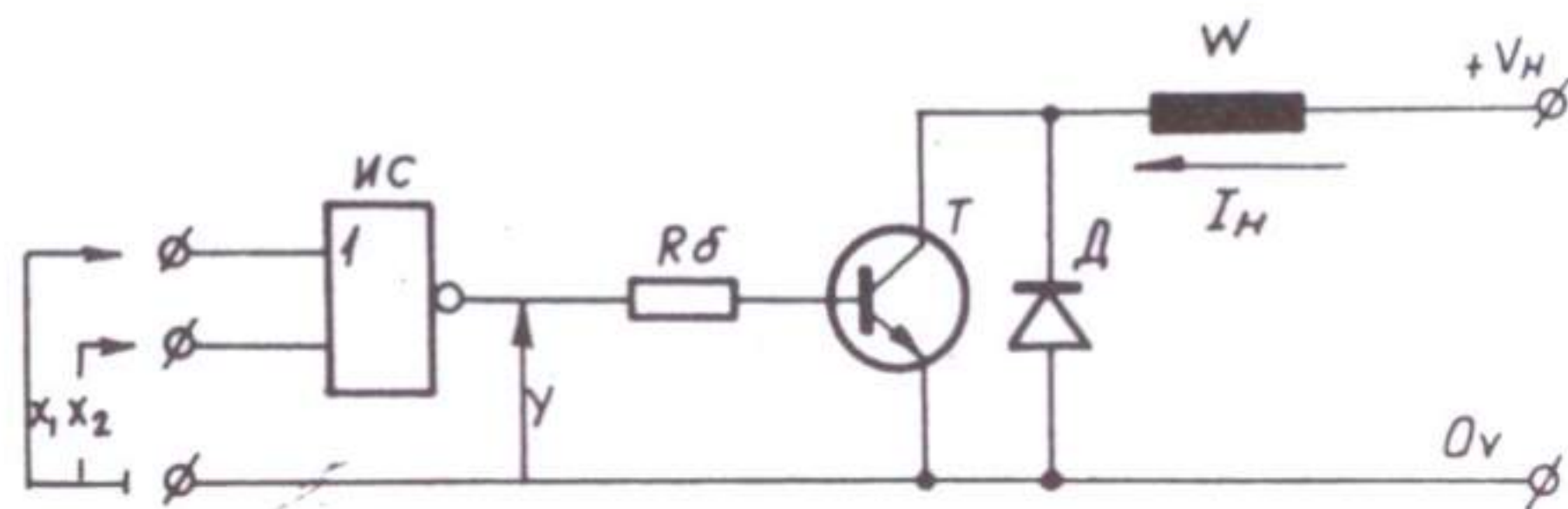
вите токове, но те показват или изключен от захранване двигател, или противоположно протичащи през бобините токове, при което не може да се създаде правилен режим на работа. Тези състояния се класифицират като неразрешени.

### 2.1.2. ТРАНЗИСТОРНИ КЛЮЧОВЕ

Ключовете К са реализирани като транзисторни стъпала с диодна защита, управлявани от логически интегрални ТТЛ схеми "ИЛИ-НЕ" (NOR). Такава схема е показана на фиг. 2.3, а зависимостта на входно-изходните ѝ сигнали е дадена в табл. 2.4, където  $X_1$  и  $X_2$  са входни ТТЛ напрежения с логически нива "0" = (0 - 0,4)V или "1" = (3,4 - 5,0)V,  $Y$  е изходно напрежение с ТТЛ ниво "0" = (0 - 0,4)V или "1" = (3,6 - 5,0)V, а фазовият ток приема две крайни стойности: нула (0) или номинална (1).

Ако присвоим ТТЛ логически стойности на четири управляващи сигнала  $D_0$ ,  $D_1$ ,  $D_2$  и  $D_3$ , съответно равни на означенията на фазовите токове  $I_1$ ,  $I_2$ ,  $I_3$  и  $I_4$ , то таблици 2.1, 2.2 и 2.3 ще бъдат и таблици на състоянията на стъпковия двигател в зависимост от сигнали, подавани от управляващо устройство (в случая микрокомпютър). За целта на входа  $X_1$  на интегралната схема се подава инверсна (обратна) стойност на управляващия сигнал, а на входа  $X_2$  - като контролиращ сигнал, правата стойност на управляващия сигнал на съответстващата фаза от другата полунамотка, както е показано в табл. 2.5. При такава постановка ред 1





Фиг. 2.3.

$N_0$	$X_1$	$X_2$	$Y$	$I_H$
1	0	0	1	1
2	x	$\bar{x}$	0	0
3	1	1	0	0

Табл. 2.4

Упр. С-л		Контр. С-л		Управление
$X_1$	$X_2$	$D_1$	$D_0$	
$\bar{D}_0$	$D_1$			фаза 1
$\bar{D}_1$	$D_0$			фаза 2
$\bar{D}_2$	$D_3$			фаза 3
$\bar{D}_3$	$D_2$			фаза 4

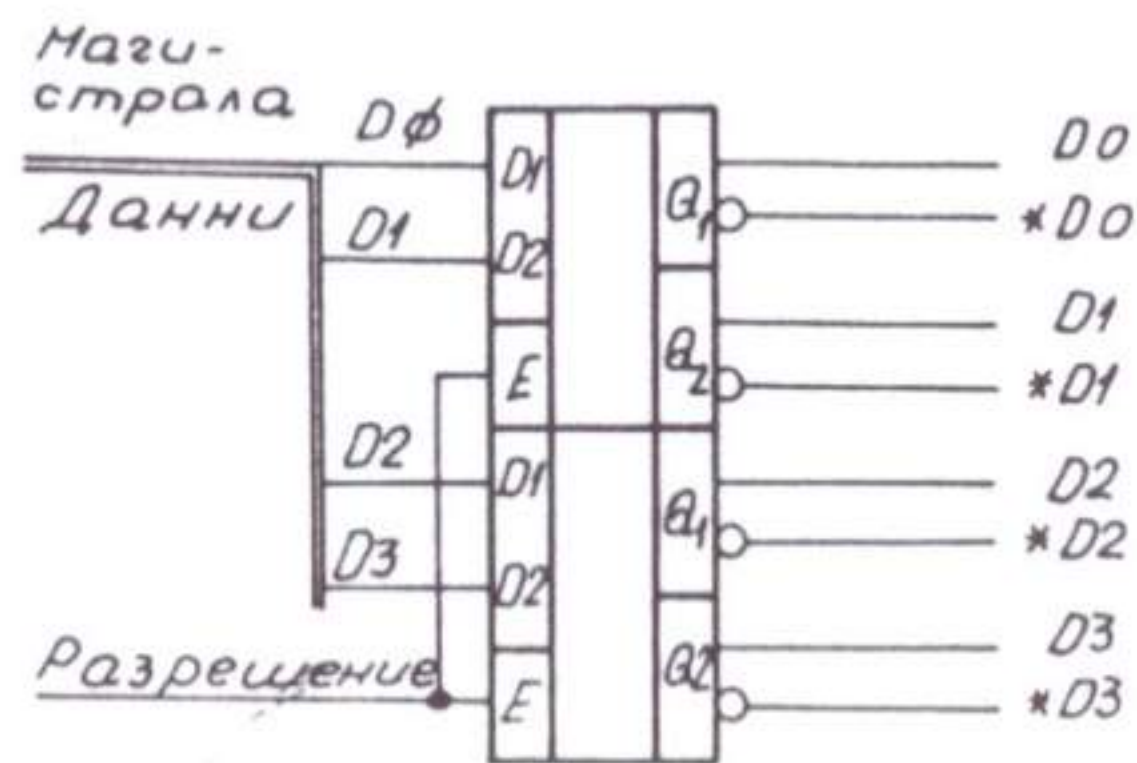
Табл. 2.5

от табл. 2.4 изразява което и да е работно състояние на двигателя, дефинирано от табл. 2.4, а ред 2 и 3 от табл. 2.4-съответно неразрешено и изключено състояние на двигателя съгласно табл. 2.3.

### 2.1.3. РЕГИСТРИ

З предходните точки беше демонстрирана една от възможните схеми за управление на стъпков двигател чрез система от управляващи сигнали D. Когато тези сигнали се подават от микрокомпютър чрез магистрала за данни, възниква необходимост от тяхната продължителна регистрация. Една такава възможност дават схемите с регистри тип LATCH, показани на фиг. 2.4. От таблицата на истинност (табл. 2.6) се вижда, че при високо ниво на сигнал "разрешение", подадено на вход E, входната информация D управлява състоянието на изходите Q и  $\bar{Q}$ , а при ниско ниво-изходната информация не се променя и има такава стойност, каквато е била непосредствено преди прехода на входа E към ниско ниво. Ако сигналът "разрешение" се дефинира от адресната магистрала на компютъра, то въпросите за управлението на двигателите се свеждат до зареждане на данни в регистри с предварително известен адрес, като посоката на ротация се определя от предварително установена последователност на смяна на тези данни. Освен регистрите за управление на шестте двигателя, платката съдържа и два други аналогични 4-битови регистъра за управление на осем броя изходни сигнали (вж. също т. 2.5).

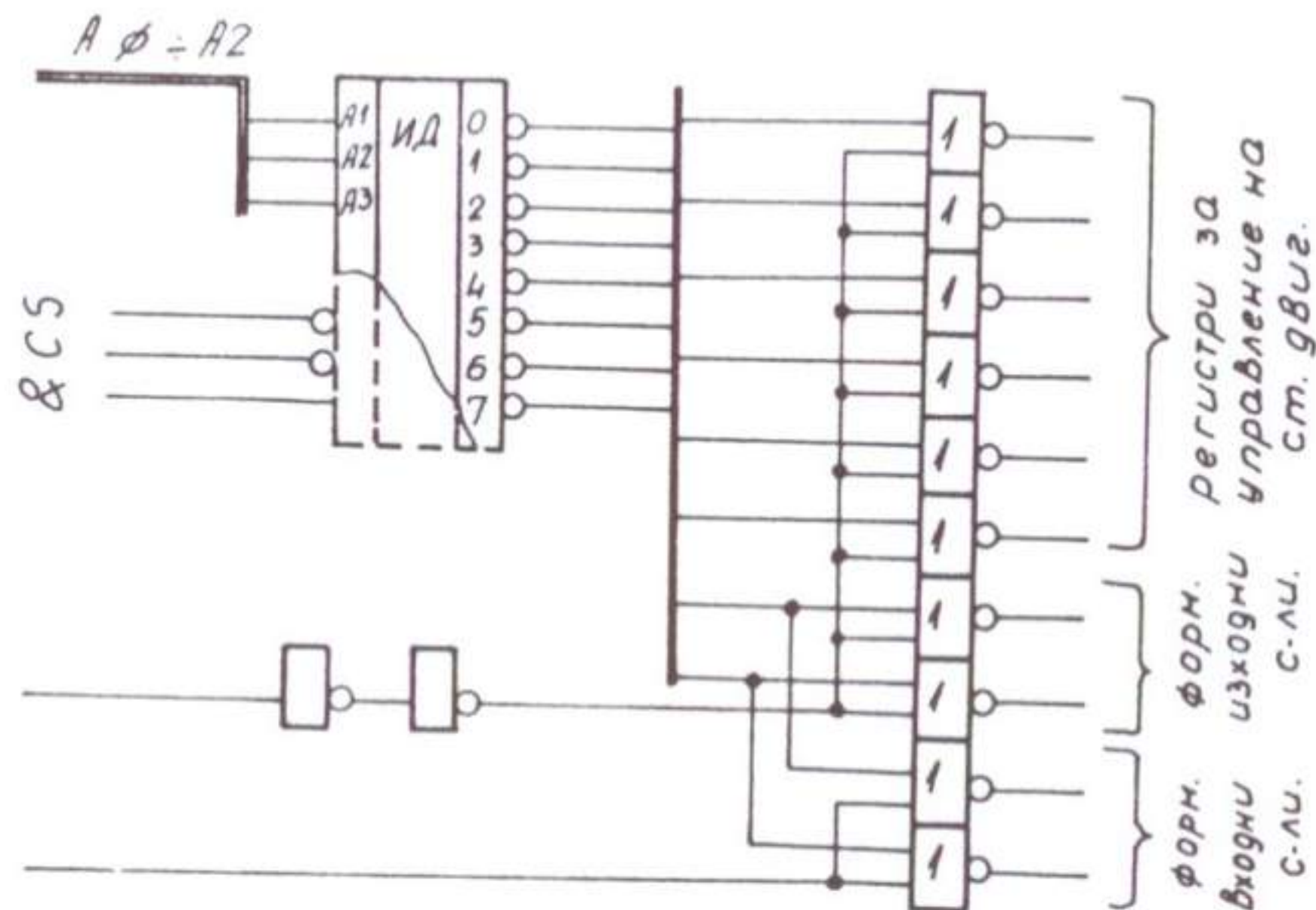




Фиг. 2.4

Входове		Изходи	
$D_i$	$E$	$Q_i$	$\bar{Q}_i$
0	1	0	1
1	1	1	0
x	0	$Q_{in}$	$Q_{in}$

Табл. 2б.



Фиг. 2.5

#### 2.1.4. АДРЕСЕН СЕЛЕКТОР

Присвояването на определени адреси на регистрите от т. 2.3 се извършва от блок адресен селектор. Условно блокът може да се раздели на две части:

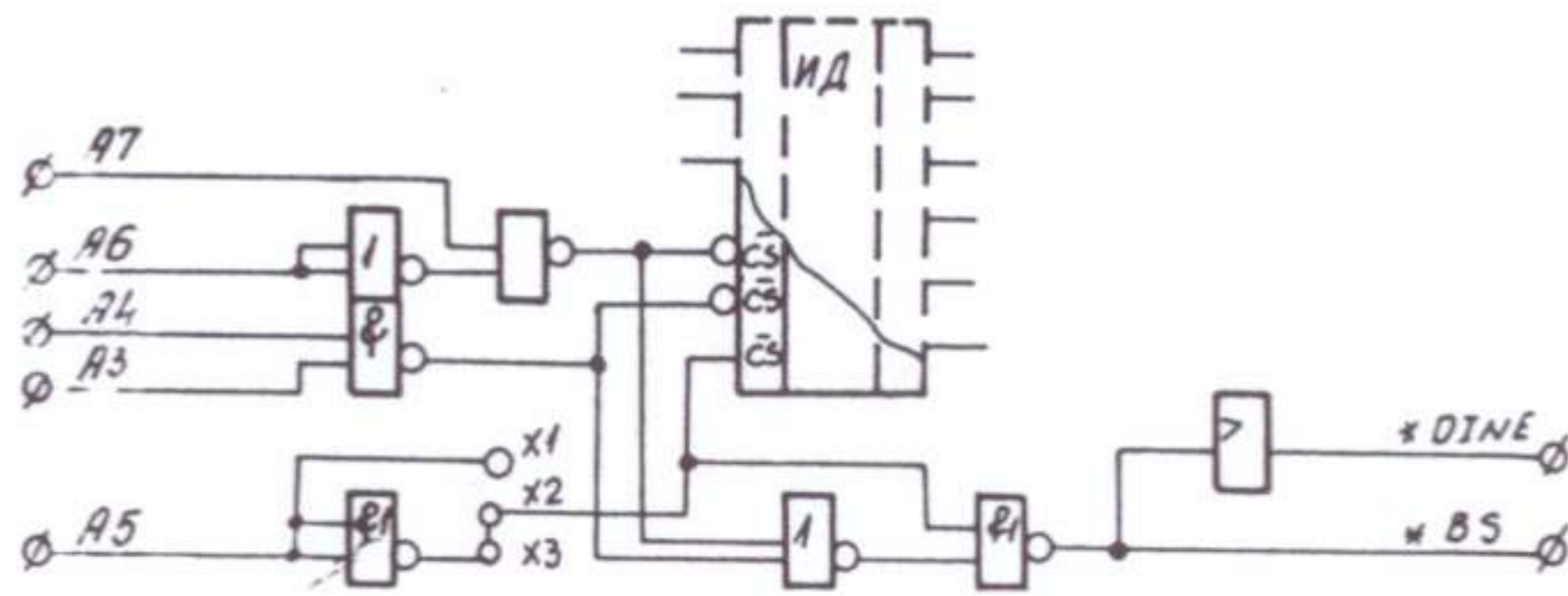
- селекция на един измежду осем адреса за запис и четене (фиг. 2.5);

- трансляция на осемте адреса в определена област от адресното пространство (фиг. 2.6).

Дешифрирането на един измежду осемте адреса се извършва от показаната на фиг. 2.5 интегрална схема ИД (дешифратор от двоичен код). Когато са изпълнени условията  $\&CS$ , вследствие подадените на входовете  $A_0, A_1$  и  $A_2$  адресни сигнали, на един от изходите 0, 1, 2, ..., 7 се дефинира ниско ниво, като при това поредният номер на изхода е равен на стойността на числото, образувано от входните сигнали в двоичен код, като разрядът  $A_2$  се счита за старши. При наличие на ниско ниво на  $*I/O W$  сигнал, една от присъединените към него и към дешифратора интегрални схеми ИЛИ-НЕ изработва сигнал "разрешение" към съответния регистър (вж. т. 2.3). Аналогично, при ниско ниво на сигнал  $*I/O R$  се изработват сигнали "разрешение" към формирозателите на входни сигнали.

Схемата за трансляция в адресното пространство (фиг. 2.6) удовлетворява изискванията за селекция на дешифратора ИД от фиг. 2.5. На пет броя входове на взаимно свързаните интегрални схеми И-НЕ (NAND) и ИЛИ-НЕ са подадени условно означени адресни сигнали  $A_7 - A_3$ , като





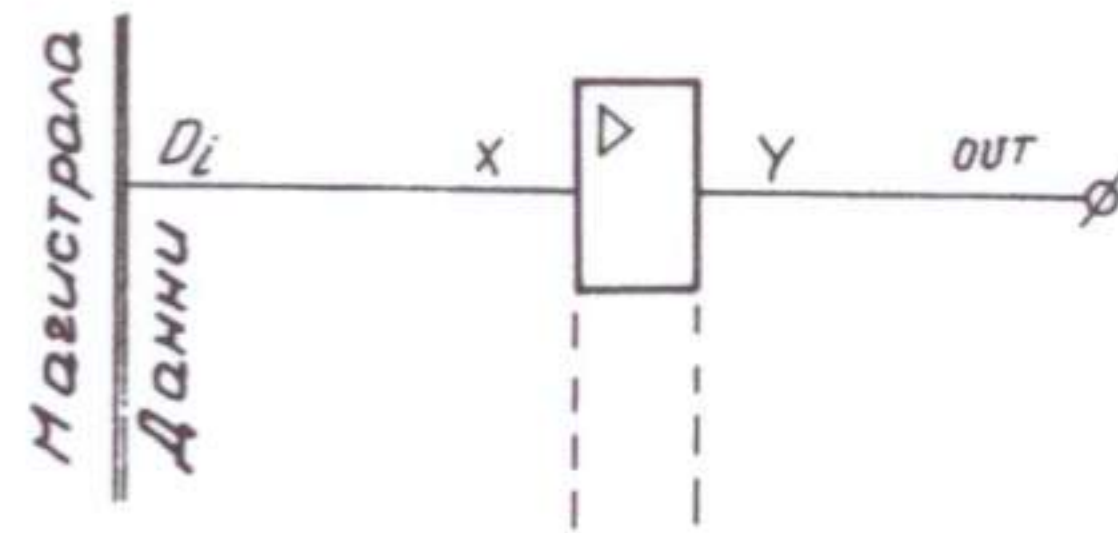
Фиг. 2.6

Nº	A7	A6	A5	A4	A3	$\bar{B5}$	$\overline{DINE}$
1	1	0	0(1)*	1	1	0	0**
2	0	x	x	x	x	1	V**
3	x	1	x	x	x	1	V**
4	x	x	1(0)*	x	x	1	V**
5	x	x	x	0	x	1	V**
6	x	x	x	x	0	1	V**

Табл. 2.7

Заб\*: Съгласно мостова връзка  $X_1 - X_2 - X_3$

Заб\*\*: Изходът е тип „отворен колектор“



Фиг. 2.7.

x	Y
0	0**
0	V**

Табл. 2.8

Адресът  $A_5$  може да се подаде без или посредством инверсия в зависимост от предварителното запояване на мостовата връзка с координати върху платката  $X_1$  (09 x 13),  $X_2$  (09 x 14) и  $X_3$  (09 x 15). Първоначално платките са изработени с адресна селекция при  $A_5 = 0$ . При необходимост от  $A_5 = 1$ , пътеката между  $X_2$  и  $X_3$  се разрушава и се запойва мостова връзка между  $X_1$  и  $X_2$ . Друга възможност за трансляция е присвояването на адресите към адресните входове  $A_7 - A_3$  в друг произволен ред. Едно примерно адресно пространство с така означените адресни изводи  $A_7 - A_3$  е дадено в табл. 2.7.

#### 2.1.5. ФОРМИРОВАТЕЛ НА ИЗХОДНИ СИГНАЛИ

Формирователят дава възможност за управление на евентуално включени външни елементи (релета, транзисторни ключове, индикатори и др.). Съставен е от интегрални схеми "усилвател с отворен колекторен изход" (BUFFER) с високоволтово захранване на изходите ( $V_{\max} = 30 V$ ) и максимален ток  $I_{CCO} = 30 mA$  при логическа нула на изхода. Електрическата схема на формирователя е дадена на фиг. 2.7.

#### 2.1.6. ФОРМИРОВАТЕЛ НА МАГИСТРАЛА ДАННИ

Необходимостта от четири броя управляващи сигнали към стъпковите електродвигатели дефинира 4 битова структура на магистралата за данни. Останалата част от



байта данни не се оползотворява, но това дава възможност за разделно обособяване на входна ( $D_3 - D_0$ ) и изходна ( $D_7 - D_4$ ) магистрали за данни, или едновременно двупосочна 4 битова магистрала данни. В последния случай е необходимо паралелно свързване на съответните битове  $D_7 - D_4$  с  $D_3 - D_0$  извън съединителя на платката, както това е показано на фиг. 2.8

Магистралата  $D_3 - D_0$  след буфериране и инвертиране в схеми НЕ образува магистрала данни "РОБКО".

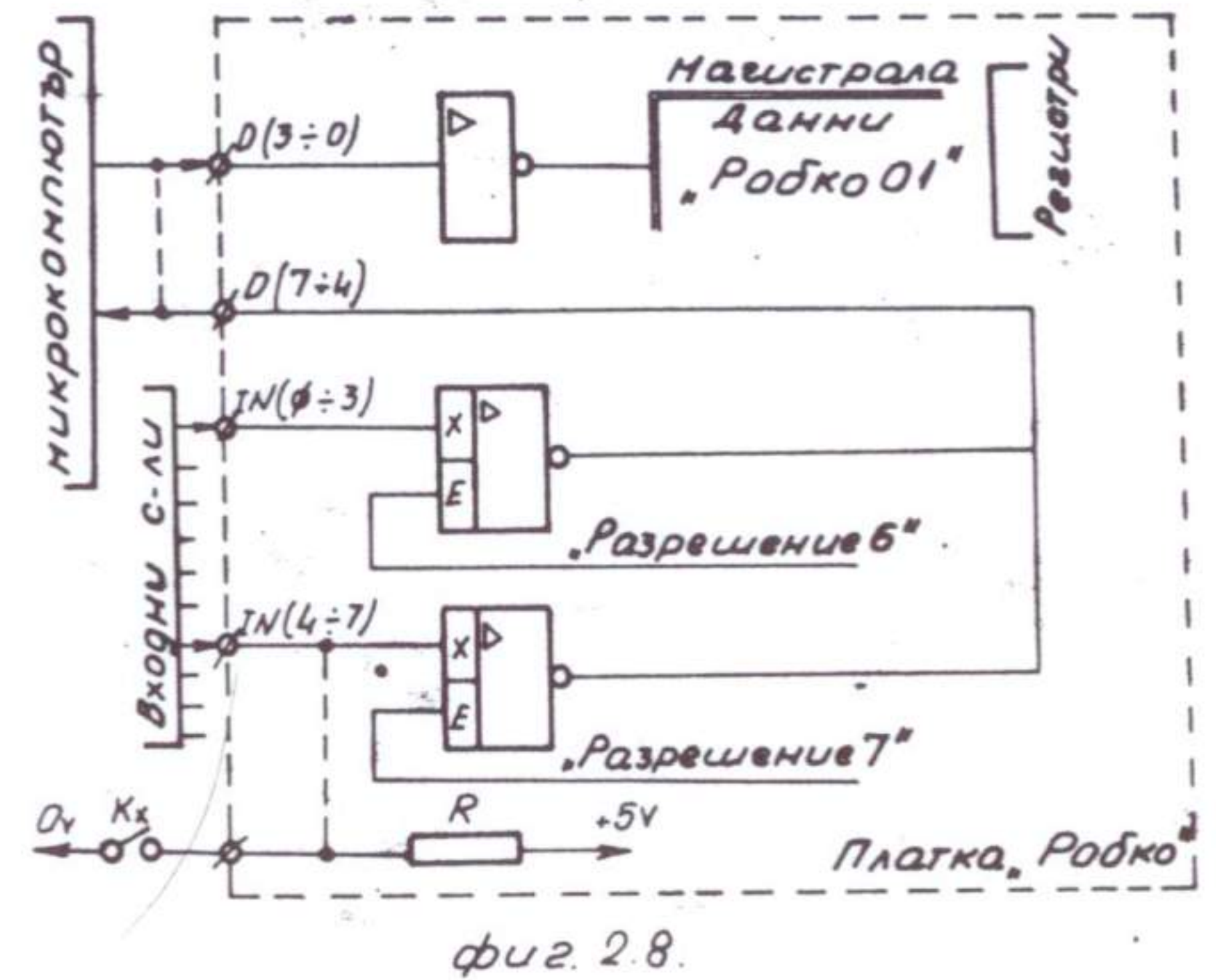
Магистралата  $D_7 - D_4$  се образува от изходи на интегрални схеми "усилвател с високоимпедансно състояние на изхода" (5-STATE BUFFER).

Паралелно на входен сигнал  $IN7$  е присъединен датчикът за контрол на хващача. Входните сигнали се задействуват при логически входен TTL сигнал.

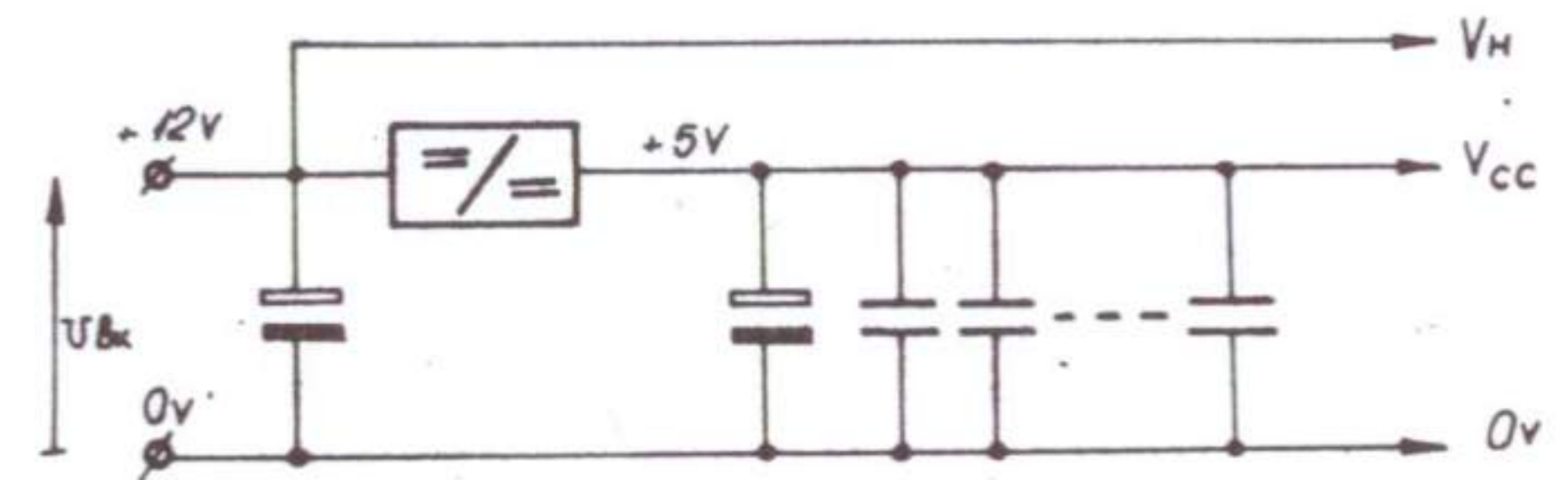
#### 2.1.7. ЗАХРАНВАНЕ

Електрическата схема на блока за захранване е представена на фиг. 2.9. Тя съдържа стабилизатор на напрежение и филтриращи кондензатори. Входното захранващо напрежение постъпва на филтриращ кондензатор, на средните изводи на намотките на двигателите и на входа на стабилизатора на захранващото напрежение за интегралните схеми, където се стабилизира на ниво  $V_{CC} = 5 V$ .

На изхода на стабилизатора също са включени филтриращи кондензатори.



фиг. 2.8.



фиг. 2.9.



3. БАЗОВО ПРОГРАМНО

ОСИГУРЯВАНЕ

НА РОБКО-01



### 3.1. ПРОГРАМА "УПРАВЛЕНИЕ"

Програмата "Управление" е една от основните в програмното осигуряване на РОБКО-01. Тя е предназначена за управление на работа посредством персоналния компютър ИМКО-2. Тази програма изпълнява много от функциите на управляващо устройство (контролер) за промишлен робот.

В настоящото ръководство за работа са описани основните режими на програмата "Управление" и правилата за работа с тях, така че дори начинаещите потребители на РОБКО-01 да усвоят някои от неговите възможности.

Работата с програма "Управление" не изисква предварителни познания. За по-задълбочено усвояване, обаче, се препоръчва познаване на ръководството за работа с ИМКО-2.

Роботът РОБКО-01 е съставен от 6 независимо движещи се стави, които общо дават 6 възможности за движение на работа (степени на свобода), включително тази на кранча. По-нататък понятията "стави" и "степени на свобода" употребяват като синоними.

Посредством подаване на стъпки към всеки от шестте двигателя, програмата изменя положението на ставите на работа, и той заема определена конфигурация в пространството.

Точка от програмата на работа се нарича матриц, съдържащ ъглите (броя стъпки) за всеки двигател на работа, необходими неговите стави да заемат определена конфигурация в пространството.

#### 3.1.1. РЕЖИМИ НА РАБОТА С ПРОГРАМАТА.



- o Начало. Инициализиране на работа. Поставяне на работа в начално положение, нулиране на датчиците и т.н.
- o Запис на точка. Обучение на точките от програмата на работа.
- с Автоматично движение. Изпълнение на обучените точки.
- с Движение по точки. Движение от текущата точка към желана точка от програмата.
- o Редактиране. Извършва се редактиране на точките от програмата. Това включва:
  - смяна на точката и скоростта;
  - смяна само на скоростта;
  - вмъкване на точка;
  - изтриване на точка.
- o Освобождаване от напрежение. Изключва се напрежението към двигателите и роботът може да бъде движан ръчно.
- o Движение без запис. Движение на работа с помощта на бутоните за ръчно управление, без да става записване на точки от програмата.
- o Спасяване на точките върху касета и четене на точките от касета. Когато точките от програмата са записани, настроени и тествани, те следва да бъдат съхранени върху магнитна лента за по-нататъшна употреба.
- o Продължение. Този режим осигурява продължаване на прекъснатата програма от точката на прекъсване.

- o Записване на точките върху диск и четене на точките от диск. Режимът е актуален при дисковия вариант на програмата, т.е., когато тя е записана на минифлопи диск.

### 3.1.2. РАБОТА С ПРОГРАМАТА.

#### А. Включване на системата.

- 1° Изключва се компютърът ИМКО-2 от захранване.
- 2° Куплунгът на лентовия кабел, излизащ от интерфейсната платка, се присъединява към куплунг В на РОБКО-01, намиращ се от задната страна на работа.

3° Отваря се капакът на ИМКО-2 и се поставя интерфейсната платка в някой от интерфейсните куплунзи (слотове) на компютъра. Платката може да се постави във всеки куплунг, с изключение на този с номер нула.

При поставянето на платката страната с лентовия кабел трябва да се падне от задната страна на компютъра, като лентовият кабел свободно излиза през специалния прорез.

4° Ако се използва дисковият вариант на програмата "Управление", тогава в интерфейсния куплунг №6 на компютъра се поставя контролерът на минифлопидисковото устройство.

Затваря се капакът на ИМКО-2.

Поставя се дискетата с програма "Управление" в минифлопидисково устройство № 1.

Включва се компютъра. На екрана на монитора се появява заглавието на програмата.

Подава се напрежение 12 V към дънната платка.



5.0. Ако се използва касетният вариант на програмата "Управление", тогава се процедурира по-нататък по следния начин:

5.1.0. Включва се компютърът.

5.2.0. Подава се напрежение 12 V към дънната платка на робота.

5.3.0. Свързва се касетофонът към ИМКО-2. Поставя се в касетофона касетката с програмата "Управление" и се намира началото на програмата (чрез брояча или чрез прослушване).

5.4.0. Написва се LOAD.

Пуска се касетофонът в режим на възпроизвеждане.

Малко след това се натиска клавишът RETURN на ИМКО-2 (продълговатият клавиш под червения бутон в десния горен ъгъл на клавиатурата. По-нататък за краткост ще го означаваме с (R)).

5.5.0. Когато се достигне началото на записа върху касетата, компютърът сигнализира с "бип".

Ако сигналът "бип" е последван от изписване на съобщението ERR, значи е станала грешка при четенето. Спира се касетофонът, а компютърът се изключва. След пауза от около 5 секунди ИМКО-2 се включва отново и вкарването на програмата започва отначало.

При нов неуспешен опит за прочитане следва да се измени нивото на възпроизвеждане на касетофона и/или да се потърси точното начало на записа, и отново да се опита.

Ако четенето е протекло нормално, след завършването му се чува отново сигналът "бип". Спира се касетофонът.

5.6.0. Поставя се касетата с програмата RO BASIC в касетофона, и се намира началото на програмата.

5.6.1.0. Написва се следното:

CALL -151 (R)

Символът вляво на екрана се изменя от " ] " на " \* ".

5.6.2.0. Написваме

8D00.9310R

и поставяме касетофона в режим на възпроизвеждане. Малко след това натискаме клавиша (R).

5.6.3.0. Ако четенето е протекло нормално, компютърът сигнализира с "бип" и символът " \* " отново се появява на екрана. Спираме касетофона.

Ако след сигнала "бип" на екрана се появи съобщение "ERR", тогава при четенето е станала грешка. Спира се касетофонът и се опитва ново зареждане на програмата.

Качествено четене от касетофона може да бъде постигнато посредством изменяне на нивото на сигнала и/или посредством намиране на точното начало на записа върху лентата.

5.6.4.0. Натискаме комбинацията "CTRL-C" и след това (R). Символът " ] " отново се появява на екрана.

5.7.0. Написваме

RUN (R)

Заглавието на програма Управление се появява на екрана.



## Б. Обучение.

В началото се появява въпрос SLOT NR ? Отговаря се с N (R) , където N е номерът на слота (интерфейсния куплунг) на ИМКО-2, където е поставена интерфейлната платка на РОБКО-01.

След стартирането на програмата се появява таблица, която показва номерата на началната, последната (крайната) и текущата точки, и се задава въпрос:

РЕЖИМ ?

Ако потребителят не знае какъв режим да избере, отговаря с "ПОМОЩ (R) ", при което се появява таблица с изброяване на режимите и начина на извикването им. Натиска се клавишът за интервал, след което въпросът РЕЖИМ ? се появява отново.

Запис на точка

В този режим се извършва обучение на работа по точки в ставна (релативна) координатна система. Всяка от степените на свобода на работа се движи индивидуално посредством съответните бутони от клавиатурата, като броят стъпки, които всяка става е изминала, се запомня като елемент на масива - точка от програмата.

Трябва да се има предвид, че точките от програмата ще се изпълняват в същия ред, в който са обучени.

На въпроса РЕЖИМ ? се отговаря с ЗАП (R) . Появява се заглавието ЗАПИС НА ТОЧКА и въпрос СКОРОСТ ЗА ОБУЧЕНИЕ ?. След отговора програмата отново пита СКОРОСТ ?, тъй като съществува възможност точките да бъдат обучавани с една скорост, а да се възпроизвеждат с друга скорост. Обикновено това се налага при точки, които е необходимо да се обучат прецизно - с ниска скорост, докато при възпроизвеждането им скоростта може да бъде по-голяма.

На първо време на въпросите за скорост може да се отговаря с 500 (R) . Скоростта (така, както се задава) всъщност представлява закъснение, и колкото съответното число е по-голямо, толкова по-бавно ще се движи роботът.

След като се отговори на въпросите за скорост, роботът може да бъде управляван с помощта на клавиатурата на ИМКО - 2. Използват се горните леви 12 бутона. Двойката клавиши цифра и буквата под нея управляват една степен на свобода на работа, като всеки клавиш отговаря за съответната посока на въртене на ставата. Например въртенето на базата на работа се осъществява с помощта на бутоните 1 и Я, на рамото - с 2 и В и т.н. , до бутоните 6 и Ъ, които управляват отварянето и затварянето на хващача.

При еднократно натискане на някой от бутоните съответната става се задвижва. Спирането на нейното



движение става чрез еднократно натискане на другия съответстващ ѝ бутон.

Движението на всички степени на свобода се преустановява посредством натискане на клавиша за интервал.

Допуска се движение на всички стави едновременно

С помощта на бутоните роботът се довежда в желаното положение. Натиска се бутонът нула. С това точката е записана.

На екрана се появява отново въпросът за режим. Показалците за текуща и крайна точка са нараснали с единица.

Ако е необходимо да се запише следваща точка, отговаря се отново с ЗАП (R) и горната процедура се повтаря.

### В. Изпълнение

Изпълнението на програмата се състои във възпроизвеждане на обучените точки в определена последователност. Програмата предвижда две възможности за изпълнение: движение по точки и автоматично движение.

Автоматично движение

В този режим записаните точки се възпроизвеждат по реда на обучението, т.е. циклично. Съществува

възможност за задаване както на точката, която е начална за цикъла, така и на крайната точка от цикъла.

За преминаване в режим автоматично движение, на въпроса РЕЖИМ? се отговаря с АВТ (R). Появява се заглавие АВТОМАТИЧНО ДВИЖЕНИЕ и въпрос ОТ ТОЧКА К ДО ТОЧКА?. Тук К е номерът на текущата точка (т.е. точката, в която роботът се намира в момента). На въпроса се отговаря с номера на точката, до която трябва да свършва цикълът. Нека например това е точката М. След като се отговори на въпроса и се натисне клавишът RETURN, роботът започва автоматично движение, като изпълнява последователно точките К, К + 1, ..., М - 1, М. След достигане на точката М роботът се връща в точката К и цикълът се повтаря.

По време на автоматичното движение на екрана се изписват номерът на точката, която е изпълнена, скоростта, с която роботът се е движил към тази точка, и след изпълнението на всеки цикъл се изписва неговият пореден номер.

Автоматичното движение може да бъде преустановено по три начина:

- Спиране в края на цикъла - чрез натискане на клавиша за интервал. Ако по време на автоматичното движение се натисне клавиш ИНТЕРВАЛ, роботът завършва текущия цикъл и спира. Появява се въпросът РЕЖИМ? и изпълнението продължава по избор на оператора.



- Спиране след изпълнение на точката - чрез натискане на бутона нула. Ако по време на автоматично движение натиснем бутон нула, роботът завършва движението си към текущата точка, и спира. Появява се отново въпросът РЕЖИМ?

- Моментално спиране (за аварийни ситуации) - чрез натискане на бутона RESET (червеният бутон в горния десен край на клавиатурата). Този начин не се препоръчва за нормална употреба, тъй като се губят обучените точки от програмата. След аварийно спиране се препоръчва да се стартира програмата "Управление" отначало.

#### Движение по точки

В този режим се осъществява движение от текущата към желана точка от числото на обучените.

За да извикаме този режим, на въпроса РЕЖИМ? отговаряме с ТОЧ (R). Появява се съобщение ДВИЖЕНИЕ ПО ТОЧКИ. ОТ ТОЧКА k ДО ТОЧКА ?. Тук k е текущата точка. Отговаряме с номера на точката, в която желаем роботът да се придвижи, и натискаме RETURN. Роботът извършва указаното движение и спира. Появява се обичайният въпрос РЕЖИМ?

#### Г. Настройка и експлоатация.

Предвидени са няколко възможности за настройка и използване на програмата, а именно:

- редактиране
- движение без запис

- освобождаване от захранване
- спасяване и четене на точките от програмата.

#### Редактиране

Извикваме този режим, като на въпроса РЕЖИМ? отговорим с РЕД (R). Програмата изрежда възможностите, които дава редактирането:

- смяна на точката и скоростта
- смяна на скоростта
- вмъкване на точка
- изтриване на точка

СМЯНА НА ТОЧКАТА И СКОРОСТТА се налага обикновено, когато желаем да направим корекция на някоя вече обучена точка от програмата.

За да извършим такава корекция, отговаряме с 1 (R). Програмата пита за скорост при обучение и при изпълнение. Отговаряме на въпросите, както при запис на точка, и получаваме възможност да движим робота с помощта на клавиатурата.

Като приведем робота в желано положение, натискаме бутона нула. Появява се таблица на точките, която показва точките от програмата, заедно с техните скорости. След като разгледаме таблицата, натискаме клавиша ИНТЕРВАЛ и се връщаме в "главата" на програмата.

СМЯНА НА СКОРОСТТА можем да направим, като в режима редактиране отговорим с 2 (R).



Появява се таблицата на точките, а след нея въпрос: "Коя скорост желаете да измените?" Отговаряме с номера на точката, в която желаем изменението, след което отговаряме на въпроса за скорост с желаната стойност и натискаме RETURN. Програмата се връща в главата за режим.

ВМЪКВАНЕ НА ТОЧКА можем да осъществим, като отговорим с 3 (R). Появява се въпрос "вмъкване на точка след:" на който отговаряме с номера на точката, след която желаем да вмъкнем новата точка.

Роботът следва да се намира в точката, след която желаем да вмъкнем новата точка. Ако това не е така, появява се съобщение: "придвижете робота в точка k; върнете се с ПРОД". Появява се въпросът РЕЖИМ?

Тук следва да изберем режим ТОЧ - движение по точки, и да придвижим робота от текущата точка към точката с номер k. След това, в отговор на обичайния въпрос РЕЖИМ? изписваме ПРОД (R), за да продължим в редакторския режим оттам, откъдето сме прекъснали.

Нататък програмата дава възможност да запишем новата точка. Отговаряме на въпросите за скорост; записваме точката и натискаме бутон нула. Появява се таблицата на точките.

От таблицата се вижда, че точките от програмата на робота са се увеличили с една. Следва да се има предвид, че точките до вмъкнатата точка получават същите номера, както преди корекцията. След тях следва новата вмъкната точка, а след нея - старите точки с номера, увеличени с единица.

ИЗТРИВАНЕ НА ТОЧКА можем да направим, като изберем 4 (R). Програмата пита "Изтриване на точка?" Отговаряме с номера на точката, която желаем да изтрием. Програмата изтрива точката, и показва таблицата на точките. Виждаме, че броят им е намалял с единица.

Следва да се има предвид, че точките преди изтриваната точка остават със старите си номера, а онези след нея получават номер, с единица по-малък.

Натискаме клавиш ИНТЕРВАЛ и се връщаме в "главата" на програмата.

Спасяване на точките върху касетофон

Четене на точките от касетофон

След като сме съставили някаква програма, желанието на робота, естествено е да съществува възможност тя да бъде съхранена и след това многократно използвана. За това служат режимите за спасяване и четене на точките от програмата.

СПАСЯВАНЕ НА ТОЧКИТЕ ОТ КАСЕТОФОН. На въпроса РЕЖИМ? отговаряме със СПК (R). Навиваме лентата на касетофона до мястото, където ще започнем записа. Пускаме касетофона на запис, като запомняме показанието на неговия брояч, и малко след това натискаме RETURN.



При записа компютърът сигнализира с "бип" 6 пъти. При завършване на записа програмата се връща в главата за избор на режим. Тогава изключваме касетофона и си записваме показанията на неговия брояч.

ЧЕТЕНЕ НА ТОЧКИТЕ ОТ КАСЕТОФОН. На въпроса РЕЖИМ? отговаряме с ЧЕК (R). Намираме началото на записания масив от точки върху касетата. Пускаме касетофона на възпроизвеждане и малко след това натискаме бутона RETURN. По време на възпроизвеждането компютърът сигнализира с "бип" 6 пъти. Когато е готово, програмата се връща в главата за режим. Изключваме касетофона.

Много е важно, преди да се започне обучението на програмата, да бъде извършена инициализация на работата. Тя се състои в привеждането на работата в една специална конфигурация, точка, в която регистрите, съответстващи на всеки двигател, биват нулирани. Тази точка, обикновено с номер нула, служи за отправна точка при обучението и при възпроизвеждането на програмата.

Роботът следва да бъде приведен в тази точка преди да започне обучението му. След спасяване на точките върху касетофона, върху магнитната лента се намират точки, тоест масиви от стъпки, изброени спрямо нулевата точка.

Ако желаем след известно време да възпроизведем някоя програма, задължително трябва преди да започне процедурата на четене от касетофон, роботът да бъде приведен в нулевата точка и да бъде извършена инициализация. Оставяйки работата в това положение, можем да прочетем точките от касетофона. След това можем да придвижваме работата до

където желаем точка от въведената програма, да изпълняваме автоматично движение, редактиране и т.н.

Процедурата на инициализация на работата е описана по-нататък.

Спасяване на точките върху диск

Четене на точките от диск

СПАСЯВАНЕ НА ТОЧКИТЕ ВЪРХУ ДИСК. Това е възможно, ако работим с дисковия вариант на програмата Управление. На въпроса РЕЖИМ? отговаряме със СПД (R). Появява се съобщението. . . . . ни известява режима и въпрос: DRIVE No?. Отговаряме с номера на минифлопидисковото устройство, в което се намира дискетата за спасяване на точките. След това се появява въпрос, дали желаем да дадем име на наименование на файла (масива) с точките. Ако желаем, отговаряме с "ДА (R)" и след това написваме името на файла. Ако отговорим с "НЕ (R)", тогава на файла с точките от програмата автоматично се присвоява име ТОЧКИ.

ВНИМАНИЕ: Ако на дискетата съществува друг файл с наименование ТОЧКИ, той ще бъде изтрит, ако отговорим, че не желаем ново наименование на файла с точките. Затова следва да се внимава при отговора на този въпрос.

По-нататък програмата записва точките от програмата на дискета и дава каталог на дискетата. Виждаме, че точките от програмата са спасени в текстов файл. Натискаме клавиша за ИНТЕРВАЛ и се връщаме в "главата" на програмата Управление.



ЧЕТЕНЕ НА ТОЧКИТЕ ОТ ДИСК. На въпроса РЕЖИМ?

отговаряме с ЧЕД (R). Появява се съобщението за режима и въпросът DRIVE NO ?. Отговаряме с номера на устройството, в което се намира дискетата, откъдето точките ще бъдат прочетени.

След това се появява въпрос за името на файла с точките. Ако не знаем точно това име, отговаряме с въпросителен знак: ? (R). Тогава се появява каталог на файловете, намиращи се на диска. Програмата отново пита за наименованието на файла с точките. Отговаряме. Файлът се прочита и програмата се връща в главата за режим. Точките са вече в програмата и това може да се види от показалците за текуща и крайна позиция.

Движение без запис

Предвидена е възможност за движение на робота с помощта на бутоните от клавиатурата, без при това да се записват конфигурациите на ръката като точки. Това движение без запис може да бъде от полза, ако желаем да изменим положението на робота спрямо няколко (или всички) точки от програмата (т.н. "повдигане" или "сваляне" на точките), за компенсация при прескачане на стъпки и пр.

Следва да сме особено внимателни при този режим, тъй като той засяга всички точки в програмата, включително и нулата. Тоест, при него се измества центърът на

цялата координатна система на робота, което е равносилно на изместване на самия робот физически в пространството.

Препоръчва се този режим да не се използва, ако не се притежава достатъчен опит в програмирането на робота.

Режимът се извиква, като на въпроса РЕЖИМ? отговорим с ДВИЖ (R). След като с помощта на бутоните от клавиатурата на ИМКО извършим необходимите движения, натискаме бутона нула и се връщаме в главата на програмата.

Ако сега преминем в автоматичен режим или режим движение по точки, ще видим, че точките от програмата са изместени.

Освобождение от захранване

Този режим се практикува, когато желаем да придвижим робота ръчно, чрез механично въртене на ръка на неговите степени на свобода.

ВНИМАНИЕ: Въртенето на робота на ръка следва да става само чрез въртене на големите зъбни колела, разположени в задната част на робота, при изключено напрежение към двигателите. В никакъв случай не бива да се прилагат външни усилия към ставите и рамената на робота, тъй като това може да доведе до неговото механично повреждане.

За да преминем в режима, на въпроса РЕЖИМ? отговаряме с ОСВ (R). Сега напрежението към двигателите е изключено. Като завършим придвижването, натискаме клавиша ИНТЕРВАЛ и се завръщаме в главата на програмата.



ЧЕТЕНЕ НА ТОЧКИТЕ ОТ ДИСК. На въпроса РЕЖИМ? отговаряме с ЧЕД (R). Появява се съобщението за режима и въпросът DRIVE NO ?. Отговаряме с номера на устройството, в което се намира дискетата, откъдето точките ще бъдат прочетени.

След това се появява въпрос за името на файла с точките. Ако не знаем точно това име, отговаряме с въпросителен знак: ? (R). Тогава се появява каталог на файловете, намиращи се на диска. Програмата отново пита за наименованието на файла с точките. Отговаряме. Файлът се прочита и програмата се връща в главата за режим. Точките са вече в програмата и това може да се види от показаните за текуща и крайна позиция.

Движение без запис

Предвидена е възможност за движение на робота с помощта на бутоните от клавиатурата, без при това да се записват конфигурациите на ръката като точки. Това движение без запис може да бъде от полза, ако желаем да изменим положението на робота спрямо няколко (или всички) точки от програмата (т.н. "повдигане" или "сваляне" на точките), за компенсация при прескачане на стъпки и пр.

Следва да сме особено внимателни при този режим, тъй като той засяга всички точки в програмата, включително и нулата. Тоест, при него се измества центърът на

цялата координатна система на робота, което е равносилно на изместване на самия робот физически в пространството.

Препоръчва се този режим да не се използва, ако не се притежава достатъчен опит в програмирането на робота.

Режимът се извиква, като на въпроса РЕЖИМ? отговорим с ДВИЖ (R). След като с помощта на бутоните от клавиатурата на ИМКО извършим необходимите движения, натискаме бутона нула и се връщаме в главата на програмата.

Ако сега преминем в автоматичен режим или режим движение по точки, ще видим, че точките от програмата са изместени.

Освобождение от захранване

Този режим се практикува, когато желаем да придвижим робота ръчно, чрез механично въртене на ръка на неговите степени на свобода.

ВНИМАНИЕ: Въртенето на робота на ръка следва да става само чрез въртене на големите зъбни колела, разположени в задната част на робота, при изключено напрежение към двигателите. В никакъв случай не бива да се прилагат външни усилия към ставите и рамената на робота, тъй като това може да доведе до неговото механично повреждане.

За да преминем в режима, на въпроса РЕЖИМ? отговаряме с ОСВ (R). Сега напрежението към двигателите е изключено. Като завършим придвижването, натискаме клавиша ИНТЕРВАЛ и се завръщаме в главата на програмата.



Начало

Този режим се използва за инициализация на робота. На въпроса РЕЖИМ? отговаряме с НАЧ (R). Появява се съобщението за режима. Отговаряме на въпроса за скорост, и можем да движим робота с помощта на бутоните от клавиатурата на ИМКО-2.

Привеждаме робота в нулево положение. Това е някакво произволно, но еднозначно повторимо положение на ставите на робота.

След това натискаме бутона нула. Програмата се връща в "главата" за режим. Роботът се намира в своята начална, изходна точка.

### 3. АЛГОРИТМИЧЕН ЕЗИК ROBASIC

#### 3.2.1 Предназначение

Специализираният драйверен език ROBASIC е предназначен за задвижване на двигателите на робота РОБКО-01, както и за изпълнение на някои други функции на управление. Езикът представлява допълнение към алгоритмичния език за програмиране BASIC. ROBASIC дава възможност програмите за управление на робота да бъдат записвани изцяло на език от високо ниво, в случая BASIC. При необходимост могат да бъдат съставяни и програми за управление на робота на асемблер или на машинен език.

#### 3.2.2 Описание на командите на ROBASIC

1/\* MANUAL

- ръчно движение с помощта на бутони от клавиатурата на ИМКО-2. Изменя съдържанието на вътрешните регистри, съответстващи на двигателите.



- 2/\*GRASP - затваряне на хващача до включване на специален микропревключвател.
- 3/\*INIT - нулиране на вътрешните регистри и изключване на захранването на двигателите.
- 4/\*FULL - управление на двигателите в режим "пълна стъпка".
- 5/\*HALF - управление на двигателите в режим "полустъпка".
- 6/\*READ - прехвърляне на изминатия брой стъпки за всеки двигател в масив на BASIC.
- 7/\*ROBOT - избор на робот (при едновременно управление на два робота от един компютър).
- 8/\*MOVE - движение по точки при некоординирано управление на ставите.
- 9/\*MOV P - движение по точки при координирано управление на ставите.

### 1. \*MANUAL

Командата е предназначена за ръчно управление на двигателите на робота (движение ръчно по стави). Тя работи в два режима: директен и програмен.

Форматът на командата в директен режим е

\* MANUAL S ,

където S е цяло число, изразяващо желаната честота на въртене на двигателите по време на изпълнение на командата. Колкото S е по-голямо, толкова скоростта на въртене на двигателите е по-малка.

Движението се извършва с помощта на горните леви 12 бутона от клавиатурата на ИМКО-2. Двойката клавиши цифра и буквата под нея управляват една степен на свобода на робота, като всеки клавиш отговаря на съответната посока на въртене на ставата. Например въртенето на основата на робота се осъществява с помощта на бутоните 1 и Я, на рамото - с 2 и Б и т.н. до бутоните 6 и Ъ, които управляват съответно отварянето и затварянето на хващача.

При еднократно натискане на някои от бутоните съответната става на робота се задвижва. Спирането на движението става чрез еднократно натискане на другия съответстващ ъ бутон.

Движението на всички двигатели се преустановява посредством натискане на клавиша за ИНТЕРВАЛ. Допуска се движение на всички двигатели едновременно.

Форматът на командата в програмен режим е

10 PRINT "\*\*MANUAL" S

където S има същото значение както по-горе.

### 2. \*GRASP

Командата е предназначена за затваряне на хващача до момента на включване на специален микропревключвател. Това осигурява надеждно хващане на детайла.



Форматът на командата в директен режим е

\* GRASP S

където S е скоростта на затваряне на хващача.

Форматът на командата в програмен режим е

10 PRINT "\*\*GRASP" S

### 3. \*INIT

При движение на работа броят на инкрементите (стъпките), подавани към всеки негов двигател, се записва във всеки момент в специални вътрешни регистри. Командата \*INIT нулира тези 6 работни регистъра и изключва подаването на напрежение към всички двигатели.

Тази команда обикновено се използва за инициализиране на работа, за освобождаване на двигателите от заранване (при ръчно движение) и т.н.

Форматът на командата в директен режим е

\* INIT

Форматът в програмен режим е

10 PRINT "\*\*INIT"

### 4. \*FULL

Двигателите на работа могат да се управляват в различни режими, два от които са: цяла стъпка и полустъпка. Режимът \*FULL позволява движение с цяла стъпка.

Форматът на командата в директен режим е

\* FULL

Форматът в програмен режим е

10 PRINT "\*\*FULL"

Ако програмистът не специфицира режим, т.е. не избере нито \*HALF нито \*FULL, програмата автоматично установява режим полустъпка (\*HALF).

### 5. \*HALF

Форматът на тази команда в директен режим е

\* HALF

Форматът в програмен режим е

10 PRINT "\*\*HALF"

### 6. \*READ

Командата е предназначена за представяне във вид на променливи от BASIC на стойностите на шестте вътрешни брояча на стъпки (които се изменят от командите \*MANUAL и \*INIT). Чрез командата \*READ стойностите на тези броячи се прехвърлят в произволно дефинирани променливи на BASIC.

Инструкцията работи само в програмен режим.

Форматът ѝ е

10 PRINT "\*\*READ"

20 INPUT W(1,I),W(2,I),W(3,I),W(4,I),W(5,I),W(6,I)

където W(1,I) - W(6,I) са променливите от BASIC, на които се приписват стойностите на съответните регистри



(броячи) на двигателите. Тук те са показани като двумерен масив, единият индекс на който показва номера на двигателя, а другият (I) - поредния номер на точката от програмата, която масивът представлява.

#### 7. \*ROBOT

Както бе споменато по-горе, съществува възможност за едновременно управление на два робота от един и същ компютър.

За целта следва на дънните платки на роботите да се извърши превключване на адресите, както беше упоменато в описанието на дънната платка. Фабрично адресът е установен за робот № 1.

След хардуерното установяване на адресите, с помощта на командата \*ROBOT се дава възможност за програмно избиране на желаня робот, като всички команди в програмата, намиращи се след командата \*ROBOT,N, се отнасят към робота с номер N

Форматът на командата в директен режим е  
\* ROBOT,N

където N = 1 или 2 е номерът на избрания робот.

В програмен режим форматът е  
10 PRINT "\*ROBOT",N

#### 8. \*MOVE

Командата осигурява движение на робота по точки, като движението на ставите му не се координира, т.е. всички стави започват движението си едновременно, а го завършват когато всяка става поотделно изпълни зададения брой стъпки.

При този режим движението между две точки не е гладко като хващачът описва начупена линия в пространството. Всички стави на робота се движат с една и съща скорост, равна на зададената.

Командата се изпълнява само в програмен режим.

Форматът ѝ е

```
10 PRINT "*MOVE" S,W1,W2,W3,W4,W5,W6
```

където S е скоростта на движение, а W1 - W6 е броят инкременти (стъпки), които всеки двигател трябва да изпълни. Инкрементите могат да се задават и във вид на математически изрази, най-често като разлика в броя стъпки за две точки от програмата на робота, например

```
10 PRINT "*MOVE" S,W(1,I)-W(1,I-1),  
W(2,I)-W(2,I-1),...,W(6,I)-W(6,I-1)
```

При изпълнение на тази команда роботът ще се придвижи от точка I-1 към точка I.

#### 9. \*MOVp

Командата има същия формат като \*MOVE, но при изпълнението ѝ ставите на робота се движат синхронизирано: всички стави започват и завършват движението си едновременно.



Така се получава гладко движение между две точки като хващачът на работата описва гладка крива (дъга) в пространството. При този режим ставите се движат изобщо с различни скорости, като най-бързо се движи ставата, която трябва да отработи най-голям брой инкременти.

### 3.2.3 Работа с ROBASIC

Работата с ROBASIC се изпълнява в следната последователност:

1<sup>o</sup> Присъединява се куплунгът на лентовия кабел от интерфейсната платка към куплунга В на дънната платка на РОБКО-01.

2<sup>o</sup> Поставя се интерфейсната платка в някой от интерфейсите куплунзи (слотове) на ИМКО-2. Платката може да бъде поставена във всеки куплунг, с изключение на този с номер 0.

При поставяне на платката страната с лентовия кабел следва да се падне от задната страна на компютъра.

3<sup>o</sup> Включва се мониторът (телевизорът).

4<sup>o</sup> Включва се ИМКО-2.

5<sup>o</sup> Подава се напрежение 12 V към дънната платка на работата.

6<sup>o</sup> Зарежда се програмата ROBASIC в паметта на компютъра.

Това може да стане по следните начини:

6.1<sup>o</sup> Ако се работи с дисково устройство, тогава се поставя дискетата с програмата ROBASIC в дисковото устройство и се изписва:

```
BLOAD ROBASIC 10.2.OBJO (R)
```

където с R е означен клавишът RETURN.

6.2<sup>o</sup> Ако се работи с касетофон, тогава се прави следното:

6.2.1<sup>o</sup> Присъединява се касетофонът към ИМКО-2. В касетофона

се поставя касетата с програмата ROBASIC и се намира началото на програмата (с помощта на брояч или чрез прослушване).

6.2.2<sup>o</sup> Пишем

```
CALL -151 (R)
```

Символът в лявата част на екрана се изменя от "J" на "\*".

6.2.3<sup>o</sup> Пишем

```
8D00.9310R
```

и включваме касетофона на възпроизвеждане. Малко след това натискаме клавиша (R).

6.2.4<sup>o</sup> Ако четенето е преминало нормално, компютърът дава сигнал "бип" и символът "\*" отново се появява на екрана.

Ако след сигнала "бип" се изписва съобщение "ERR", значи е станала грешка при четенето от касетофона. Спираме касетофона и опитваме да заредим програмата отново.

Качествено прочитане на програмата от касетофона се достига посредством изменение на нивото на възпроизвеждане на касетофона и/или посредством намиране на точното начало на записа върху касетката.

6.2.5<sup>o</sup> Натискаме комбинацията CTRL-C и след това (R). Символът "J" отново се появява на екрана.

7<sup>o</sup> Изписваме

```
POKE 828, N (R)
```

```
CALL 37120 (R)
```

```
PRINT
```

където N е номерът на интерфейсия куплунг, в който е поставена платката с лентовия кабел.

Ако всичко е нормално, на екрана се изписва заглавието на програмата ROBASIC.

8<sup>o</sup> Оттук нататък може да се работи с ROBASIC в директен режим. Например, с помощта на командата



\* MANUAL 500 (R)

роботът може да се движи ръчно посредством бутоните на компютъра, с желана скорост 500.

За тестване на изправността на работа може да се подаде команда

\* INIT (R)

след което да се завъртят с ръка големите зъбни колела в задната част на робота. Те трябва да могат да се въртят свободно, тъй като в този момент към двигателите не е подадено напрежение.

9<sup>o</sup> Работата с ROBASIC в програмен режим по нищо не се отличава от програмирането на BASIC, като командите на ROBASIC се използват в описания по-горе формат.

10<sup>o</sup> Ако желаем да излезем от програмата ROBASIC и се върнем в основната конфигурация на компютъра, следва да бъде изпълнена командата

CALL 37489 (R)

*(Връщане с 20G)*

или просто да бъде натиснат клавишът RESET.

#### 4. ПРОГРАМНО ОСИГУРЯВАНЕ

О Т П О - В И С О К О Н И В О



В тази глава са разгледани въпросите за създаване на алгоритми и на програмно осигуряване за решаване на сложни манипулационни задачи.

#### 4.1. КООРДИНАТНИ ПРЕОБРАЗОВАНИЯ

В редица приложения е необходимо конфигурацията на ГОБКО - 1 да се опише в различни координатни системи. Най-често използваните координатни системи са:

- Ставна координатна система. Тази система включва ъглите на завъртане на отделните звена, и е най-удобна при директното управление на ГОБКО - 1 с помощта на персонален компютър.

- Декартова координатна система. Тази система включва декартовите (пространствени) координати  $X, Y, Z$  на избрана точка от хващача. Тя е удобна при описание на движението на хващача в тримерното пространство.

За ефективно управление на работа при решаване на различни манипулационни задачи е необходимо да се познават зависимостите, по които ставните координати се преобразуват в пространствени и обратно.

Намирането на декартовите координати по зададени ставни координати е предмет на правата задача на кинематиката на робота. На свой ред намирането на ставните координати, съответстващи на зададени декартови координати, е обект на обратната задача на кинематиката. Необходимо е да се отбележи, че решението на правата задача е единствено, докато това на обратната задача в общи случаи е неединствено.



Намирането на ефективни методи за решаване на обратната задача на кинематиката в реално време е един от актуалните проблеми на роботиката. В останалата част от тази точка ще покажем как става решаването на правата и на обратната задачи за робот с кинематичната структура на РОБКО - 1.

За целта ще изведем явни формули за получаване на:  
 - правото решение (т.е. координатите  $X, Y, Z$  като функция на ставните координати);

- обратното решение (т.е. ставните координати като функция на декартовите координати  $X, Y, Z$ ).

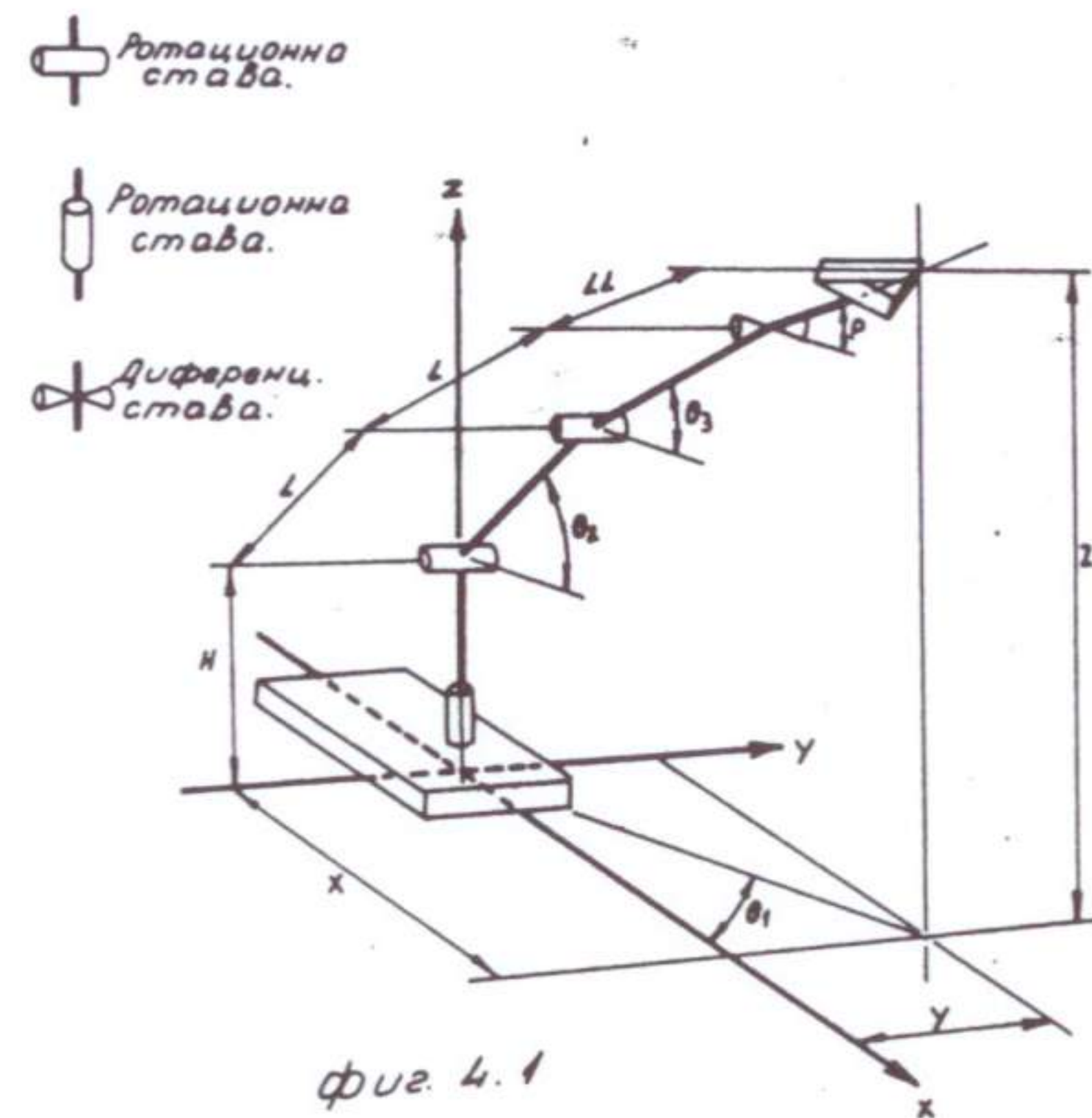
За получаване и използване на правото и обратното решения е необходимо да се познават някои елементарни факти от тригонометрията, кратки сведения за които са представени по-долу.

#### 4.1.1. Кинематичен модел на ръката

За решаване на задачите на кинематиката е необходимо да се състави кинематичен модел на ръката на РОБКО - 1 (фиг. 4.1).

Ъглите  $\theta_1 - \theta_5$  съответствуват на изразите  $V$   $W(1,1)$   $W(5,1)$  използвани в *ROBASIC*. Ъглите  $\theta$ , измерени в градуси или радиани, отговарят на величините, измерени в брой стъпки на двигателите, съгласно табл. 4.1.

Ще напомним, че едно пълно завъртане съответствува на  $360^\circ$  или на  $2\pi$  ( $\approx 6,283$ ) радиана.





На фиг. 4.1 разстоянията между ставите са означени с  $H$ ,  $L$  и  $LL$ :  $H$  е разстоянието от основата до оста на раменната става,  $L$  е разстоянието между раменната става и лакетната става (равно на разстоянието от лакетната става до ставата на китката), а  $LL$  е разстоянието от ставата на китката до централната точка между върховете на пръстите. Стойностите на тези параметри са дадени в табл. 4.2.

Движението на диференциалната става на китката (фиг. 4.2) може да бъде описано по следния начин.

Нека  $P$  е ъгълът на изправяне на хващача (т.е. ъгълът между хващача и хоризонта), а  $R$  - ъгълът на завъртане на хващача спрямо оста му на симетрия. Тогава

$$(4.1) \quad P = \frac{1}{2} (\theta_5 + \theta_4),$$

$$(4.2) \quad R = \frac{1}{2} (\theta_5 - \theta_4),$$

където  $\theta_4$  и  $\theta_5$  са ъглите на завъртане на дясната и на лявата китки (фиг. 4.2).

#### 4.1.2 Права задача на кинематиката

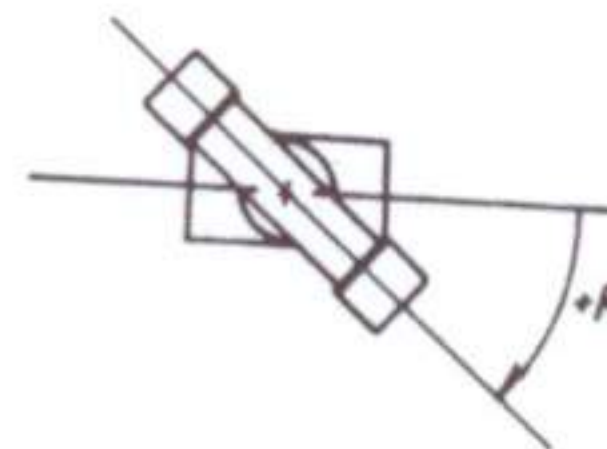
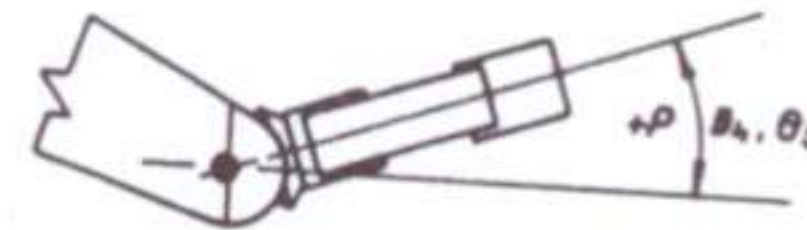
В тази подточка е описано получаването на декартовите координати  $X$ ,  $Y$ ,  $Z$  като функция на ставните ъгли  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ ,  $\theta_4$  и  $\theta_5$  (фиг. 4.1). По този начин се решава правата задача на кинематиката. За справка на фиг. 4.3 са представени основните зависимости при решаване на правоъгълни триъгълници.

Коефициент	S1	S2	S3	S4	S5	S6
Steps/rad	1222.7	1161.4	680.1	244.7	244.7	27 steps/m

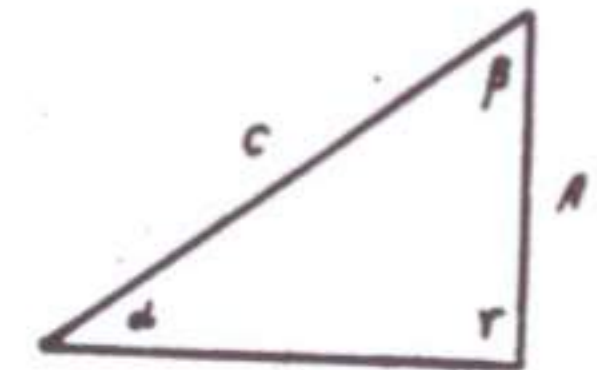
Табл. 4.1.

Рамо	H	L	LL
Дължина, mm	190	178	92

Табл. 4.2.



фиг. 4.2



$$\begin{aligned} \gamma &= 90^\circ, \quad d + \beta = 90^\circ \\ c^2 &= a^2 + b^2 \\ A &= C \sin d \\ B &= C \cos d \\ \frac{A}{B} &= \operatorname{tg} d \end{aligned}$$

фиг. 4.3



Първоначално се определя височината  $Z$  на крайната точка спрямо хоризонталната равнина на основата. За целта се въвежда спомагателната величина  $RR$ , равна на разстоянието от основата до хоризонталната проекция на крайната точка (фиг. 4.4). Сумирането на хоризонталните и вертикални проекции на всички стави дава

$$(4.3) \quad Z = H + L \sin \theta_2 + L \sin \theta_3 + LL \sin P,$$

$$(4.4) \quad RR = L \cos \theta_2 + L \cos \theta_3 + LL \cos P,$$

където ъгълът на издигане на хващача  $P$  е

$$(4.5) \quad P = \frac{1}{2} (\theta_5 + \theta_4).$$

На свой ред координатите  $X$ ,  $Y$  на крайната точка се определят с помощта на спомагателната величина  $RR$  съгласно фиг. 4.5.

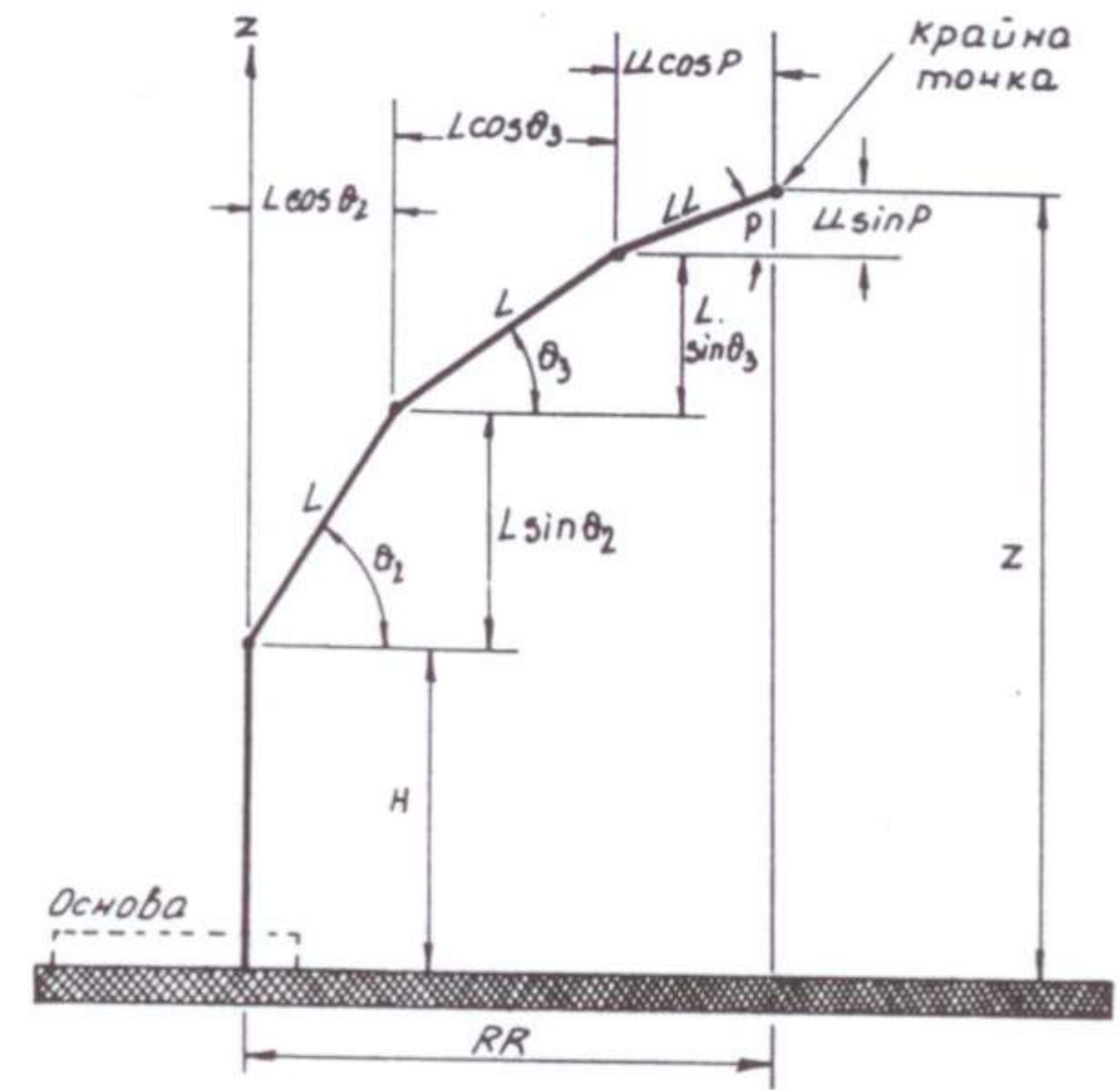
$$(4.6) \quad X = RR \cos \theta_1,$$

$$(4.7) \quad Y = RR \sin \theta_1.$$

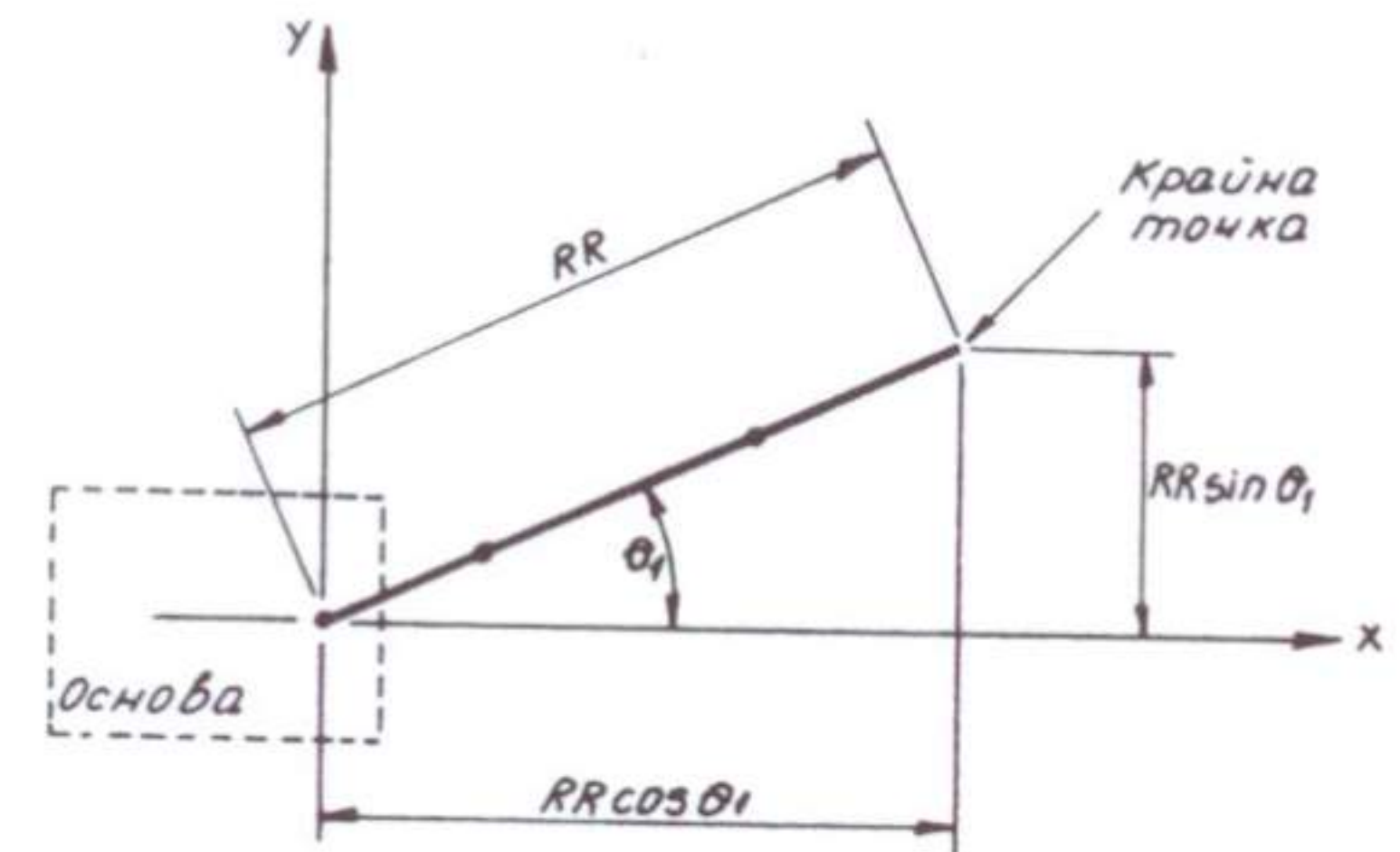
В резюме решението на правата задача е дадено в табл. 4.3. Реализация на това решение на *ROBASIC* е показана в табл. 4.5. (оператори от 2200 до 3310).

#### 4.1.3. Обратна задача на кинематиката

В тази подточка е показано как се определят став-



фиг. 4.4

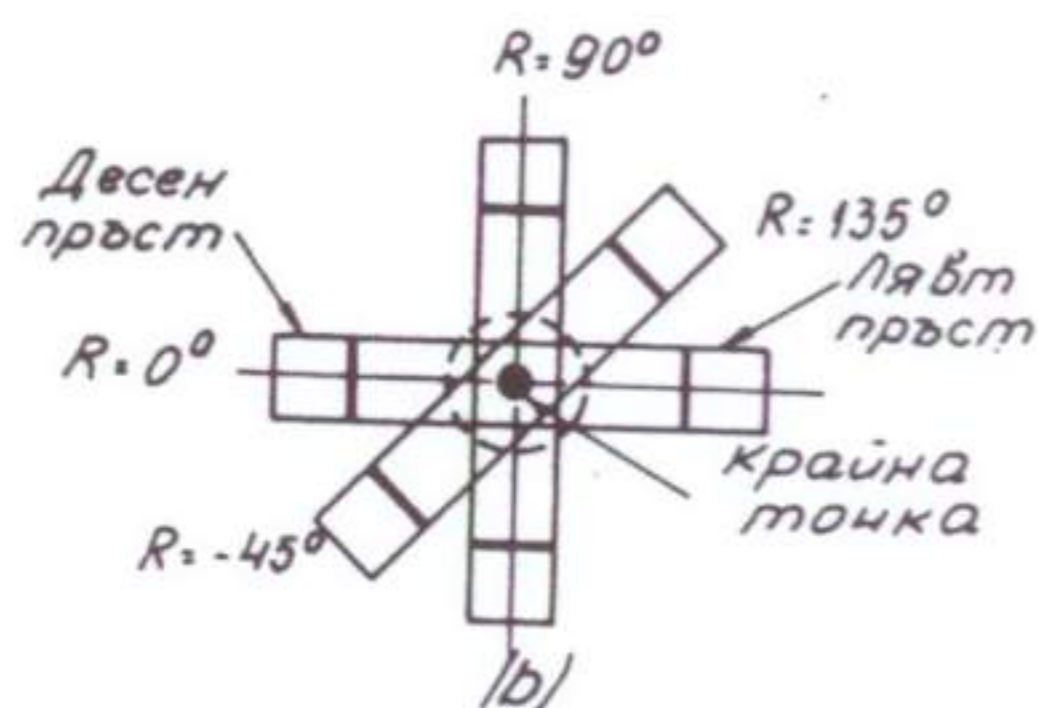
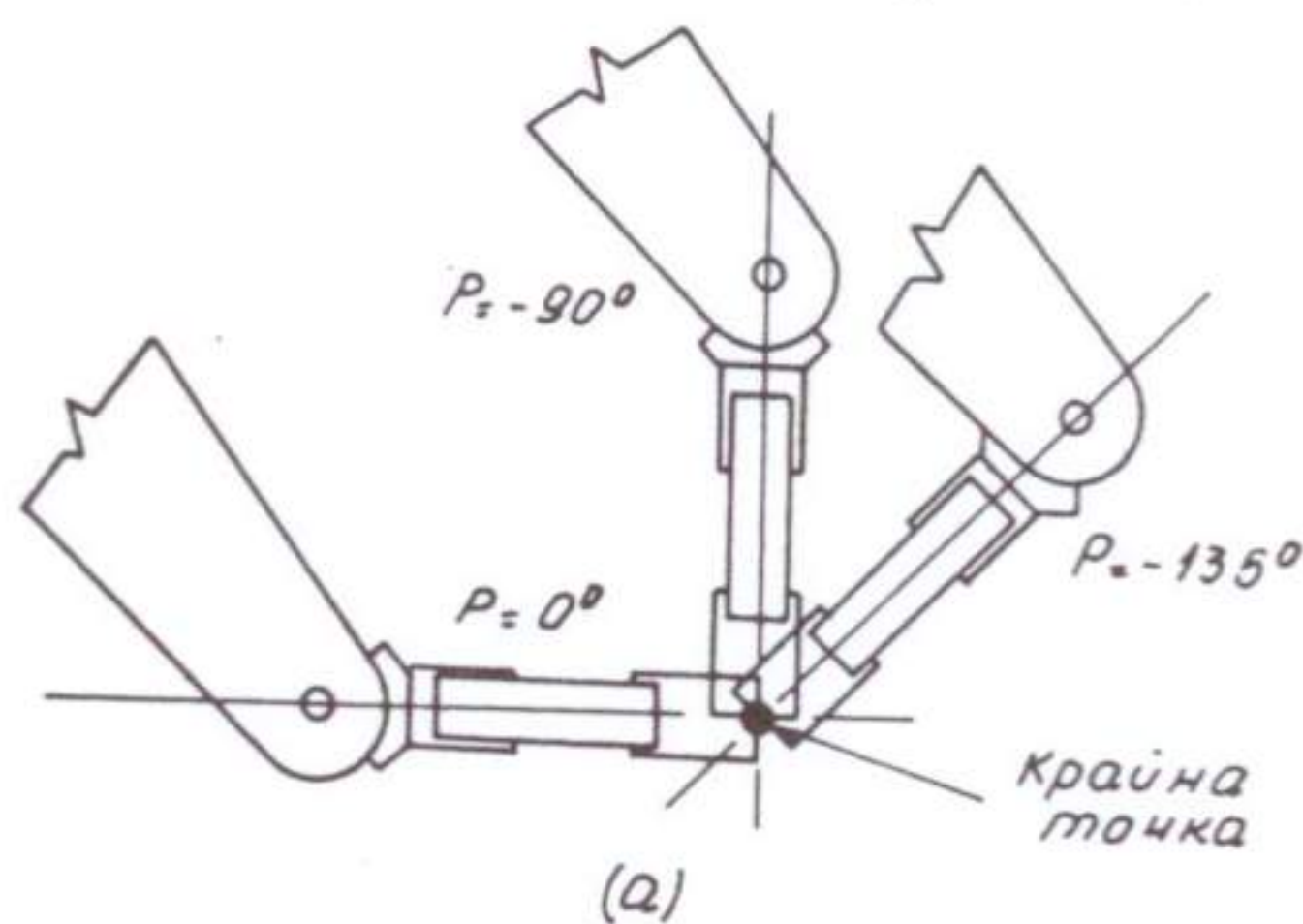


фиг. 4.5



Табл. 4.3

$$\begin{aligned}
 1^{\circ} \quad & P = (\theta_5 + \theta_4)/2 \\
 2^{\circ} \quad & R = (\theta_5 - \theta_4)/2 \\
 3^{\circ} \quad & RR = L \cos \theta_2 + L \cos \theta_3 + LL \cos P \\
 4^{\circ} \quad & X = RR \cos \theta_1 \\
 5^{\circ} \quad & Y = RR \sin \theta_1 \\
 6^{\circ} \quad & Z = H + L \sin \theta_2 + L \sin \theta_3 + LL \sin P .
 \end{aligned}$$



Фиг. 4.6.

ните ъгли (координати)  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ ,  $\theta_4$  и  $\theta_5$ , отговарящи на зададени декартови координати  $X$ ,  $Y$ ,  $Z$  на крайната точка (в случая за крайна точка е избрана централната точка на хващача между двата пръста).

Определяне на положението и ориентацията на хващача

Положението на хващача се определя от координатите  $X$ ,  $Y$ ,  $Z$  на крайната точка.

Ориентацията на хващача зависи от ъглите (фиг. 4.6)  $P$  (изправяне спрямо хоризонта) и  $R$  (завъртане спрямо оста на симетрия). Ако направим аналогия с движението на летателен апарат, то  $P$  отговаря на ъгъла на атака, а  $R$  - на ъгъла на крена на апарата.

Поради наличието само на два управляеми ъгъла на ориентация, хващачът на РОЕКС - 1 не може да бъде произволно ориентиран (както е известно, за постигане на произволна ориентация на твърдо тяло в пространството е необходимо да се разполага с три свободни ъгови параметъра). Това се отразява във факта, че хоризонталната проекция на оста на симетрия на хващача винаги съвпада с хоризонталната проекция на ръката.

Определяне на ъгъла на крена в тримерното пространство

В случая, когато хващачът е ориентиран право надолу ( $P = 90^\circ$ ) кренът  $R$  може да бъде измерен спрямо ръката (както беше правено досега), или спрямо избрана равнина в



тримерното пространство. В последния случай кренът ще бъде означаван с  $R'$ .

Ако хващачът е насочен точно надолу, изваждането на  $\theta_1$  от зададения крен ще гарантира запазване на ориентацията на хващача спрямо координатните оси (фиг. 4.7).

$$(4.8) \quad R' = R - \theta_1.$$

Така например  $R' = 0^\circ - \theta_1$  запазва ориентацията спрямо оста X, докато  $R' = 90^\circ - \theta_1$  запазва ориентацията спрямо оста Y.

При получаване на обратното решение е удобно да се въведе специална целочислена променлива  $R1$  за индикиране на разликата между стандартния крен  $R$  и крена  $R'$  спрямо избрана равнина в тримерното пространство:  $R1 = 0$  за стандартния крен, и  $R1 = 1$  в противен случай. С помощта на  $R1$  уравнение (4.8) може да се запише като

$$(4.9) \quad R' = R - \theta_1 R1,$$

откъдето

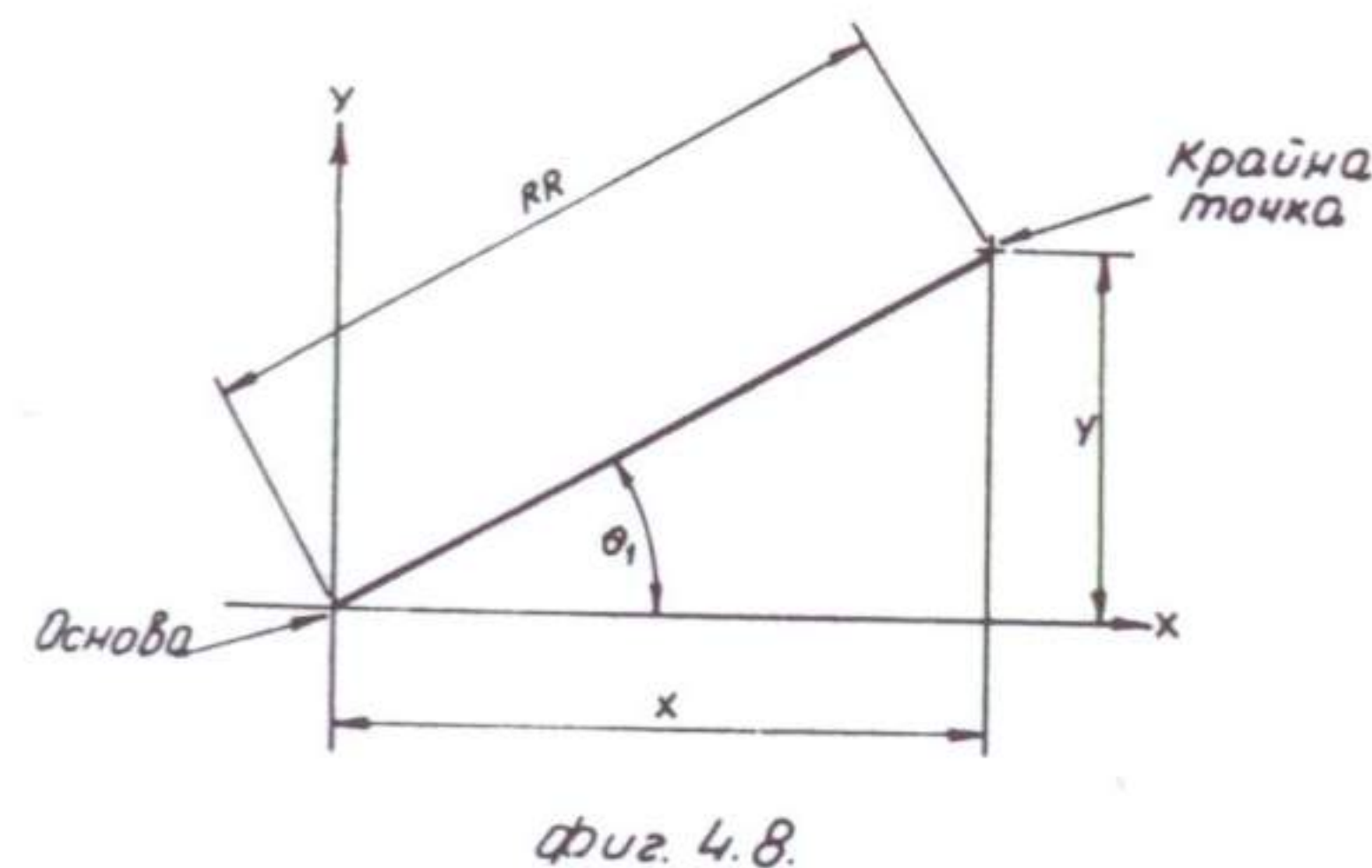
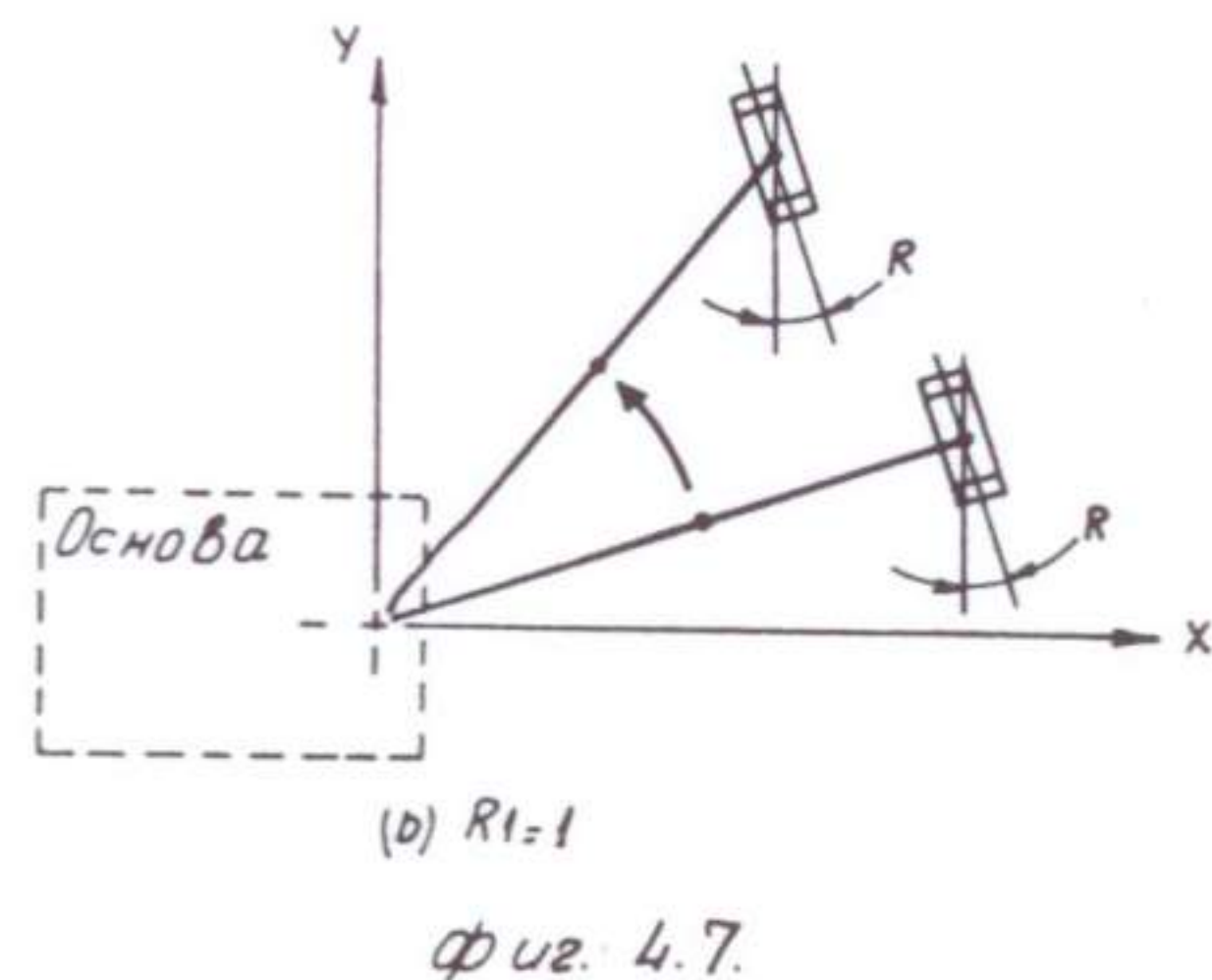
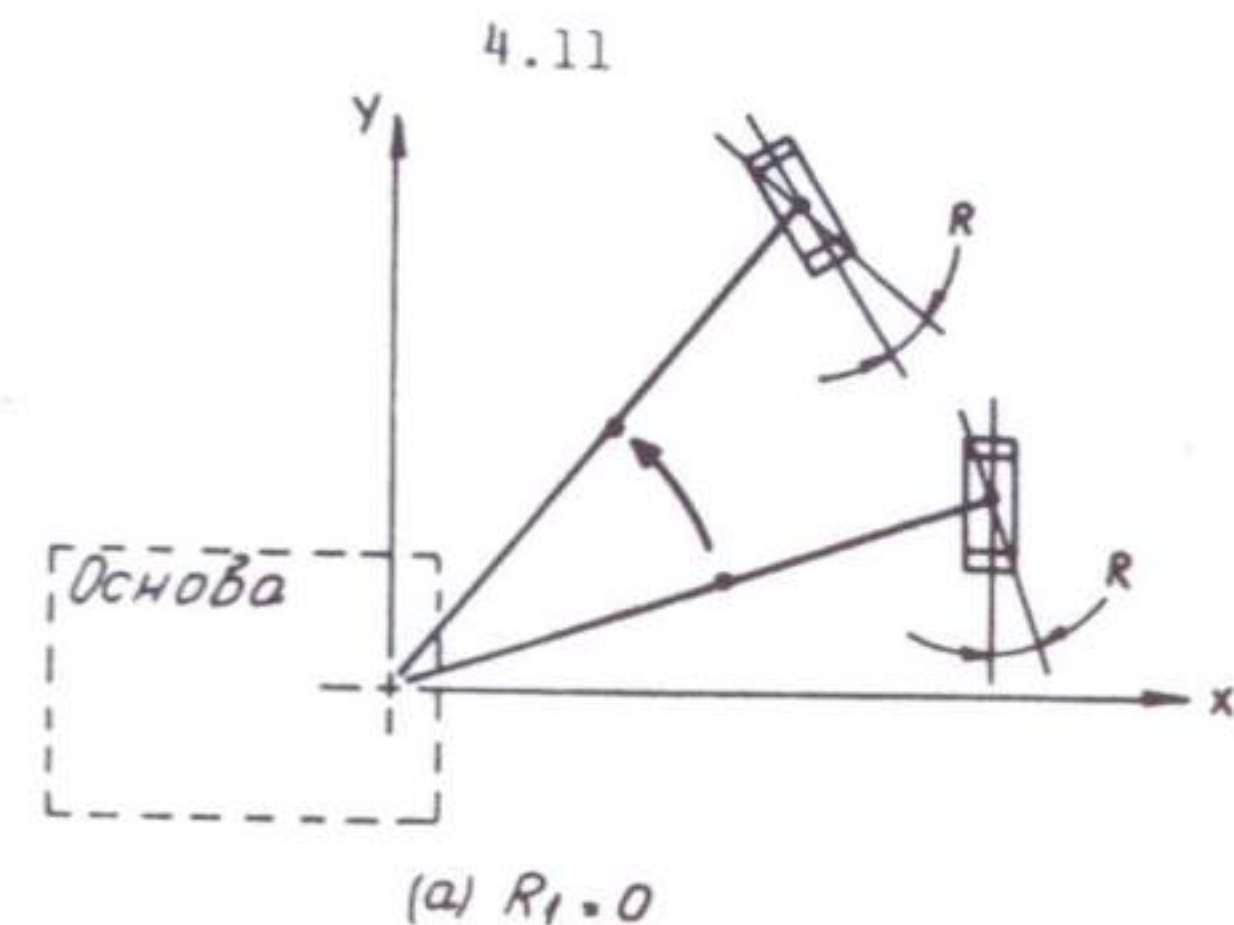
$$(4.10) \quad R = R' + \theta_1 R1.$$

Алгоритъм за получаване на обратното решение

Алгоритъмът за решаване на обратната задача на кинематиката се състои от следните стъпки:

1<sup>o</sup> Изчисляване на  $\theta_1$  и  $RR$  (фиг. 4.8)

$$(4.11) \quad RR = \sqrt{X^2 + Y^2},$$





$$(4.12) \quad \theta_1 = \arctg(Y/X).$$

2° Изчисляване на  $\theta_4$  и  $\theta_5$

От уравнения (4.1), (4.2) и (4.10) получаваме

$$(4.13) \quad \theta_4 = P + R' + \theta_1 R_1,$$

$$(4.14) \quad \theta_5 = P - R' - \theta_1 R_1.$$

Ще напомним, че нормалният крен се получава при

$$R_1 = 0.$$

3° Изчисляване на координатите на китката

Както и при решаване на правата задача, тук ще използваме страничен поглед към кинематичния модел (фиг. 4.4). Нека  $R_e$  и  $Z_e$  са координатите на крайната точка в равнината  $(R, Z)$  на чертежа. Тогава координатите

$R_w$ ,  $Z_w$  на китката са

(фиг. 4.9)

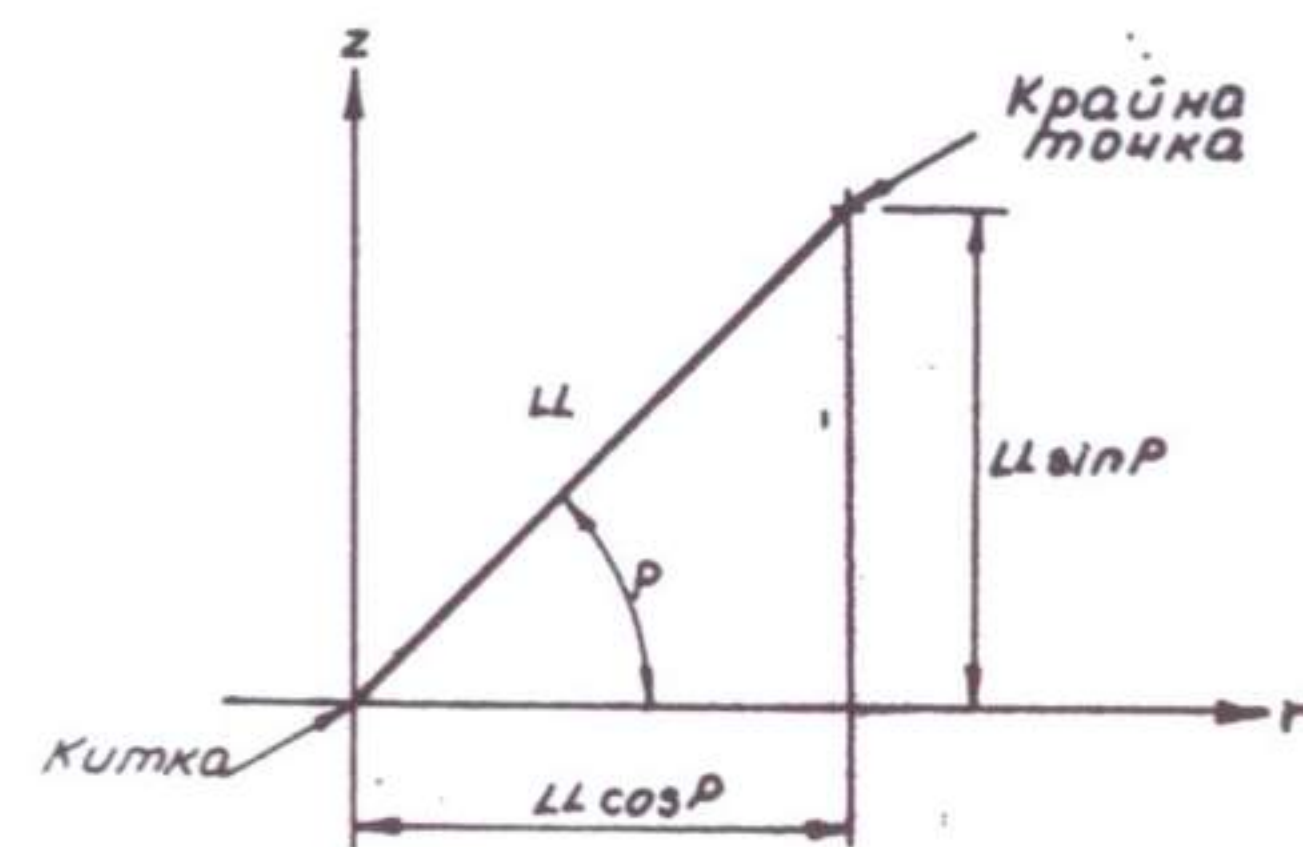
$$(4.15) \quad R_w = R_e - LL \cos P,$$

$$(4.16) \quad Z_w = Z_e - LL \sin P.$$

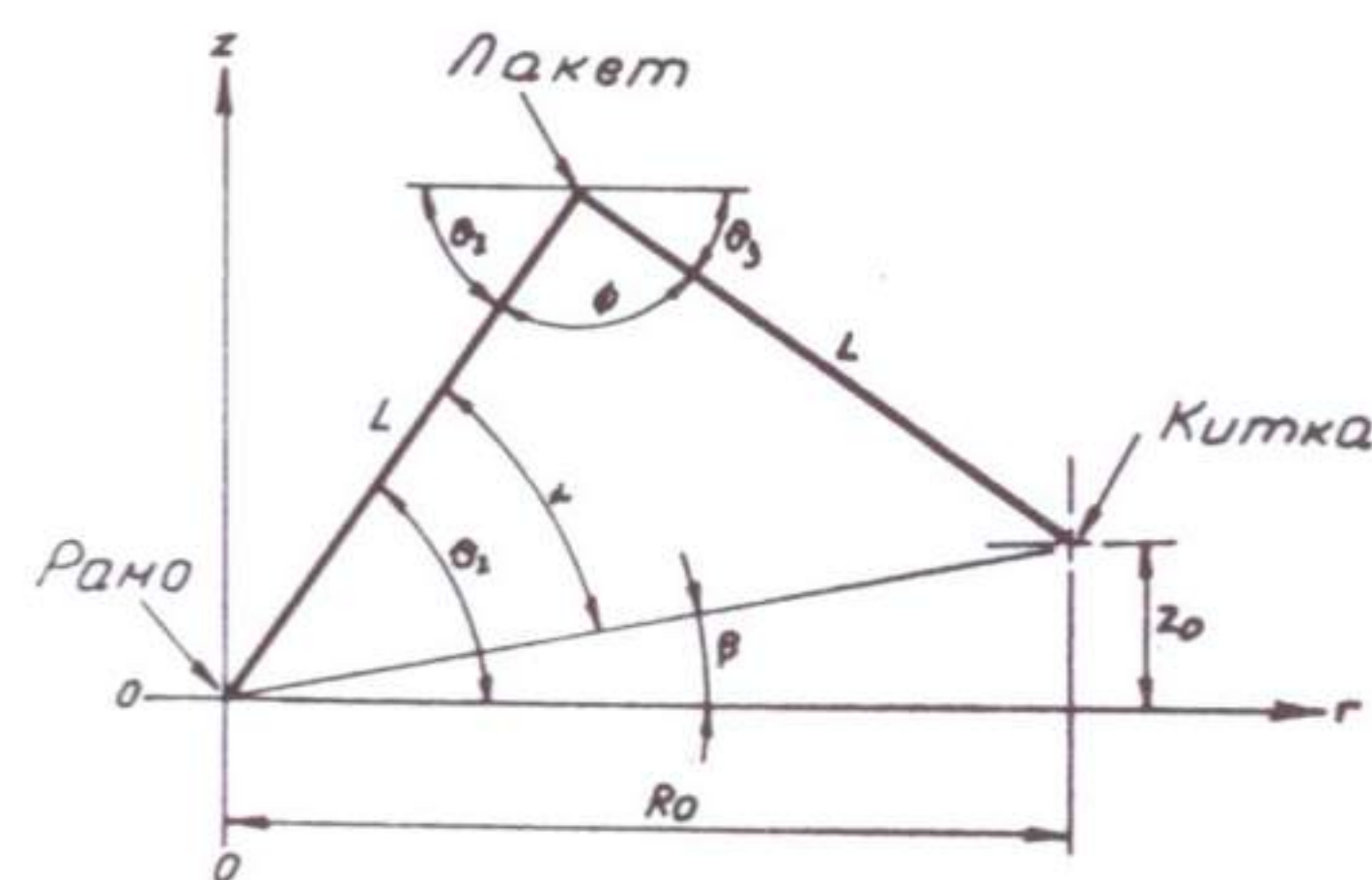
4° Изчисляване на  $\theta_2$  и  $\theta_3$

За изчисляване на ъглите  $\theta_2$  и  $\theta_3$  ще използваме транслираната координатна система, показана на фиг. 4.10. Началото  $(0, 0)$  е разположено в рамото, като координатите на китката в тази система са  $(R_o, Z_o)$ . Разстоянието  $R_o$  от рамото до китката е равно на вече изчисленото в (4.15) разстояние  $R_w$ :

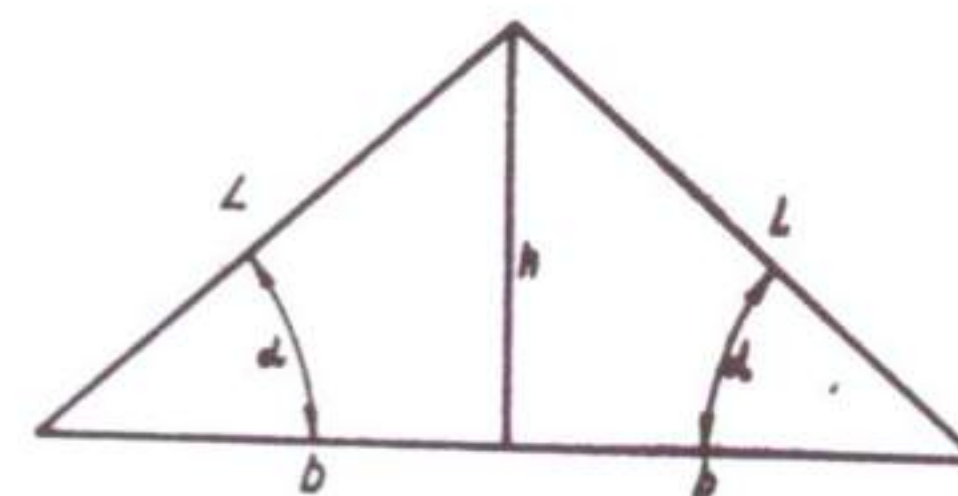
$$(4.17) \quad R_o = R_e - LL \cos P.$$



фиг. 4.9



фиг. 4.10



фиг. 4.11



Височината  $Z_w$  на китката спрямо основата е равна на сумата от височината  $H$  на рамото и величината  $Z_o$ , равна на относителното извисяване на китката спрямо рамото:

$$Z_w = H + Z_o, \text{ т.е.}$$

$$(4.18) \quad Z_o = Z_w - H.$$

Оттук и от (4.16) получаваме

$$(4.19) \quad Z_o = Z_e - LL \sin P - H.$$

За определяне на  $\theta_2$  и  $\theta_3$  въвеждаме спомагателните ъгли  $\alpha$ ,  $\beta$  и  $\phi$  (фиг. 4.10).

Тъй като  $\operatorname{tg} \beta = Z_o / R_o$ , то

$$(4.20) \quad \beta = \operatorname{arctg} (Z_o / R_o).$$

Триъгълникът, образуван от рамото, лакета и китката е показан отделно на фиг. 4.11. Височината  $h$  разделя този равнобедрен триъгълник на два еднакви триъгълника с основа  $b$ . При това

$$(4.21) \quad b = \frac{1}{2} \sqrt{Z_o^2 + R_o^2}$$

и

$$(4.22) \quad h = \sqrt{L^2 - b^2}.$$

Тъй като

$$(4.23) \quad \operatorname{tg} \alpha = h / b,$$

то

$$(4.24) \quad \alpha = \operatorname{arctg} (h / b) = \operatorname{arctg} (\sqrt{L^2 - b^2} / b).$$

След заместване на (4.21) и (4.22) в (4.24) получаваме

$$(4.25) \quad \alpha = \operatorname{arctg} \sqrt{4L^2 / (R_o^2 + Z_o^2) - 1}.$$

От друга страна от чертежа на фиг. 4.10 се вижда, че

$$(4.26) \quad \theta_2 = \alpha + \beta,$$

$$(4.27) \quad \theta_2 + \phi + \theta_3 = 180^\circ.$$

$$\text{Съгласно фиг. 4.11} \quad \phi + 2\alpha = 180^\circ$$

т.е.

$$(4.28) \quad \phi = 180^\circ - 2\alpha.$$

От (4.26) - (4.28) следва

$$(4.29) \quad \theta_3 = \alpha - \beta.$$

В действителност ъгълът  $\theta_3$  на фиг. 4.10 е равен на отрицателната стойност на ъгъла  $\theta_3$  от фиг. 4.1. Следователно

$$(4.30) \quad \theta_2 = \alpha + \beta,$$

$$(4.31) \quad \theta_3 = \beta - \alpha.$$

По този начин получихме решението на обратната задача в явен вид: това са зависимостите (4.12) - (4.14), (4.30) и (4.31) с отчитане на (1.25), (4.20), (4.19) и (4.17).



Алгоритъмът за получаване на обратното решение е резюмиран в табл. 4.4.

Реализация на алгоритъма на *ROBASIC* е дадена в табл. 4.5 (оператори от 3310 до 4110).

Важна характеристика на алгоритмите за решаване на обратната задача на кинематиката е тяхното бързодействие, което се измерва с броя на изчислителните операции (в аритметика с плаваща запетая), необходими за изпълнението им.

Предложеният по-горе алгоритъм изисква 12 събирания (изваждания), 10 умножения, 3 деления, 3 пъти изчисляване на *arctg*, 2 пъти изчисляване на *sin* и *cos* и 2 пъти изчисляване на квадратен корен. За езика *BASIC* на персоналния компютър ИМКО - 2 това изисква по-малко от 1 секунда. При това за извършване на изчисленията е достатъчна машинна аритметика с относителна точност  $10^{-6}$ . На потребителите, които се нуждаят от по-бързи алгоритми, предлагаме да съставят собствени такива, основани на целочислена аритметика, табулиране на тригонометричните функции (вместо изчисляването им), и т.н.

Табл. 4.4

1°	Задаване на константите $H, L, LL$
2°	Задаване на величините $X, Y, Z, R, P, R1$
3°	$\theta_1 = \text{arctg}(Y/X)$
4°	$RR = (X^2 + Y^2)^{1/2}$
5°	$\theta_4 = P + R + R1 \theta_1$
6°	$\theta_5 = P - R - R1 \theta_1$
7°	$R_0 = RR - LL \cos P$
8°	$z_0 = Z - H - LL \sin P$
9°	$\beta = \text{arctg}(z_0/R_0)$
10°	$= \text{arctg}(4L^2/(R_0^2 + z_0^2) - 1)^{1/2}$
11°	$\theta_2 = \alpha + \beta$
12°	$\theta_3 = \beta - \alpha$



#### 4.1.4 Кинематичен модел на хващача

Отварянето на хващача е пропорционално на броя на стъпките, зададени на неговия двигател, като коефициентът на пропорционалност е  $S_6 =$  стъпки/мм.

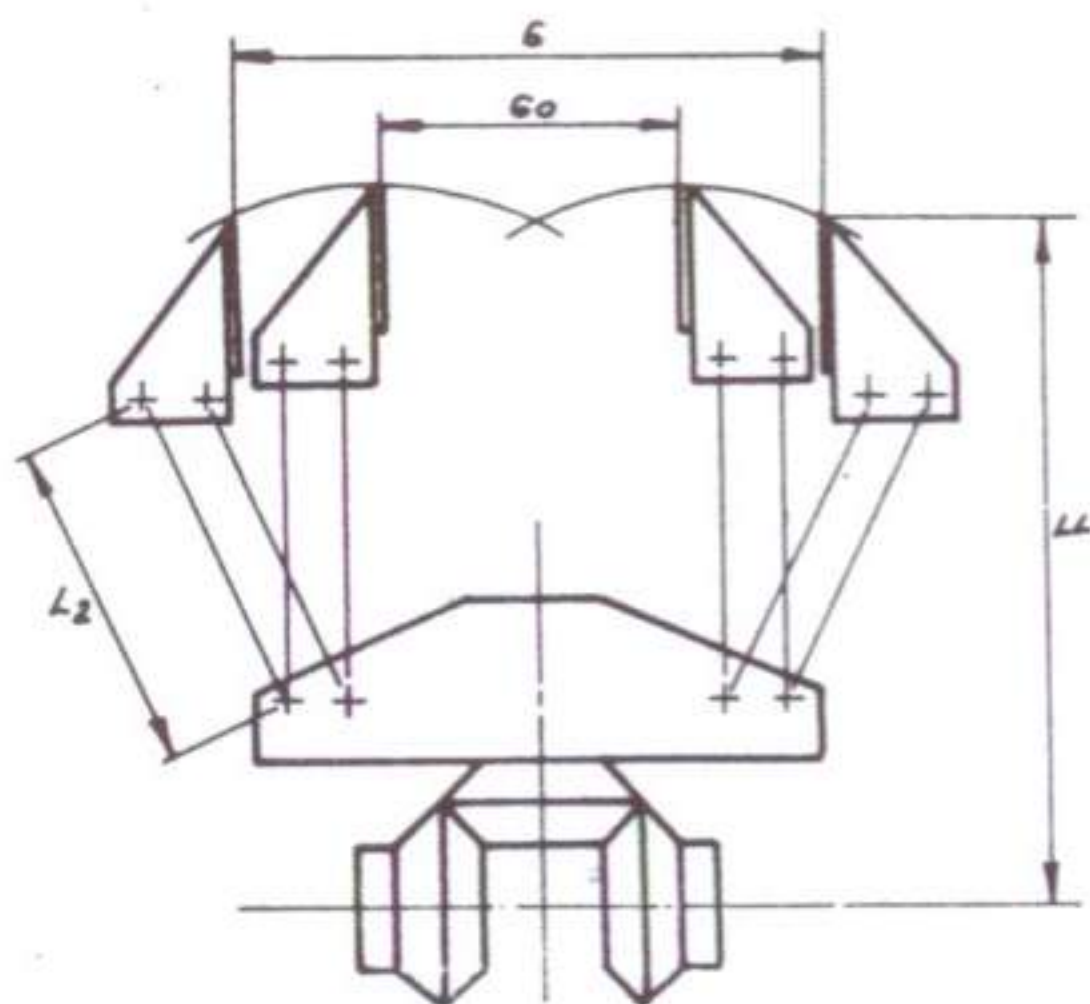
До този момент дължината на хващача  $LL$  беше разглеждана като постоянна величина. В действителност тази дължина се изменя слабо с отварянето на хващача (фиг. 4.12). Макар че ефектът от това изменение е малък (около 2,5 мм), той трябва да се отчита при изпълнение на по-прецизни манипулационни задачи.

Дължината  $LL$  може да се представи като

$$(4.32) \quad LL = L_1 + \sqrt{L_2^2 + (G - G_0)^2 / 2},$$

където  $L_1$ ,  $L_2$  и  $G_0$  са константи, а  $G$  е разтворът на хващача,  $L_1 =$  мм,  $L_2 =$  мм,  $G_0 =$  мм. Измерването на  $G$  в брой стъпки на двигателя става чрез коефициента  $S_6$ .

По-точната формула (4.32) (вместо  $LL = const$ ) може да се използва както при правото, така и при обратното решение.



фиг. 4.12

#### LIST 2200-3310

```

2200 REM
3000 REM "обратна кинематична задача"
3010 REM INPUT VARIABLES X0,Y0,Z0,P0,R0 (MM,DEG)
3015 REM OUTPUT VARIABLES T1, ..., T5 (RAD)
3017 IF Z0 < 0 THEN Z0 = 0
3020 IF Z0 < 300 AND X0 < 0 THEN X0 = 100
3025 RR = SQR (X0 * X0 + Y0 * Y0): LF = 2 * L1 + LG
3030 IF Z0 = H THEN RM = LF: GOTO 3060
3040 IF Z0 = 0 THEN RM = SQR (LF * LF - H * H): GOTO 3060
3050 RM = SQR (LF * LF - (H - Z0) * (H - Z0))
3060 IF RR > RM THEN RR = RM
3070 P0 = P0 / C: R0 = R0 / C
3080 R0 = RR - LG * COS (P0): Z0 = H - Z0 - LG * SIN (P0)
3090 IF R0 = 0 THEN GA = SGN (Z0) * PI / 2: GOTO 3110
3100 GA = ATN (Z0 / R0)
3110 AL = SQR (R0 * R0 + Z0 * Z0) / 2: AL = ATN (SQR (L1 * L1 - AL * AL) / AL)
3120 IF X0 = 0 THEN T1 = SGN (Y0) * PI / 2: GOTO 3140
3130 T1 = ATN (Y0 / X0)
3140 T2 = GA - AL: T3 = GA + AL
3145 T4 = P0 + R0 + R1 * T1
3150 T5 = P0 - R0 - R1 * T1
3300 RETURN
3310 REM
    
```

#### LIST 3310 - 4140

```

3310 REM
4000 REM права кинематична задача
4010 REM INPUT VARIABLES T1, ..., T5 (RAD)
4020 REM OUTPUT VARIABLES XX,YY,ZZ,PP,RR (MM, DEG)
4030 REM
4040 PP = (T4 + T5) / 2: RR = (T4 - T5) / 2 - R1 * T1
4050 RP = L1 * COS (T2) + L2 * COS (T3) + LG * COS (PP)
4060 XX = RP * COS (T1): YY = RP * SIN (T1)
4070 ZZ = H - L1 * SIN (T2) - L2 * SIN (T3) - LG * SIN (PP)
4080 PP = PP * C: RR = RR * C
4100 RETURN
4110 REM
    
```

Табл. 4.5



5. ВРЪЗКА НА РОБКО - 01

С РАБОТНАТА СРЕДА



## 5.1. ВХОДОВЕ И ИЗХОДИ ЗА РЕЛЕИНА ВРЪЗКА С РАБОТНАТА СРЕДА

При практически приложения на РОБКО-01 често се налага съвместна работа на робота с други устройства, с различно периферийно оборудване : конвейри, бункери, машини и др. системи, с които роботът образува т.н. "роботизирано работно място"(RRM).

Сигналообменът в RRM обикновено е от релеен (цифров) вид. Такива са сигналите, получавани от крайни изключватели, релета, позиционни ключове или други датчици, задействувани от пристигнал детайл в позиция за вземане, сигнали за начало на цикъл и т.н., които за програмата на робота представляват цифрови входове, а цифрови изходи са сигналите, чрез които роботът задействува от своя страна съответно машина, друг механизъм или подава към други системи електрически сигнал за начало на изпълняван цикъл.

РОБКО-01 притежава 8 цифрови входа и 8 цифрови изхода, като е това той е не само пасивно изпълнително устройство, а и диспечер и синхронизатор в RRM.

В настоящата глава ще бъдат описани апаратната реализация и програмното осигуряване на връзката на РОБКО-01 с работната среда, някои практически реализации на тази връзка и др.

### 5.1.1. ВРЪЗКА С РАБОТНАТА СРЕДА - АПАРАТНА РЕАЛИЗАЦИЯ

Свързването на РОБКО-01 с работната среда се осъществява с помощта на съединител А, разположен на задната страна в основата на робота. Изводите за връзка с работната среда са представени на фиг. 5.1.1.1 , а функционалното им предназначение - в таблица 5.1.1.А - Б.



Наименование	ВХОДОВЕ							
	1	2	3	4	5	6	7	8
Означеніе	IN1	IN2	IN3	IN4	IN5	IN6	IN7	IN8
Извод на съединител А	C1	C2	C3	C4	C5	C6	C7	C8
Разряд данни	D4	D5	D6	D7	D4	D5	D6	D7
Адресно пространство A3 ÷ A0	IOR & E <sub>(16)</sub>				IOR & F <sub>(16)</sub>			

Табл. 5.1.1.-А

Наименование	ИЗХОДИ							
	1	2	3	4	5	6	7	8
Означеніе	OUT1	OUT2	OUT3	OUT4	OUT5	OUT6	OUT7	OUT8
Извод на съединител А	A1	A2	A3	A4	A5	A6	A7	A8
Разряд данни	D0	D1	D2	D3	D0	D1	D2	D3
Адресно пространство A3 ÷ A0	IOW & E <sub>(16)</sub>				IOW & F <sub>(16)</sub>			

Табл. 5.1.1.-Б

Наименование	НУЛЕВ ПОТЕНЦИАЛ (ШАСИ)
Извод на съединител А	A9 ÷ A13 ; C9 ÷ C13 ; B1 ÷ B13

Табл. 5.1.1.-В

Входните сигнали следва да притежават стандартни TTL напрежения :

- логическа нула :  $U_{bx(0)} = (0 \div 0,3) \text{ V}$  при  $I \leq 12 \text{ mA}$
- логическа единица :  $U_{bx(1)} = (2,0 \div 5,0) \text{ V}$  при  $I \leq 1 \text{ mA}$

Напреженията, подадени на входа, не трябва да надвишават стойността на захранващото напрежение на входните интегрални схеми  $V_{cc} = 5 \text{ V}$  или да имат отрицателна стойност, тъй като това води до тяхното разрушаване, а напреженията в зоната между логическата нула и единица - до непредсказуеми състояния на разрядите за данни.

При номинални входни напрежения, логическите нива на разрядите за данни съвпадат с нивата на съответните входни сигнали, т.е. те са взаимно неинверсни.

Две примерни схеми за подаване на входни сигнали към РОБКО-01 са показани на фиг. 5.1.1.2 и фиг. 5.1.1.3, където, в първата от тях, за източник е включена батерия с електродвижимо напрежение (ЕДН) = 4,5 V, а във втората - потенциометричен делител  $R_{1p}$ ,  $R_{2p}$  към по-високо напрежение. Стойностите на резисторите R можем да изчислим от съотношенията :

$$R \leq \frac{E - U_{bx(1)min}}{I_{bx(1)}} = \frac{4,5 - 2}{1 \cdot 10^{-3}} = 2,5 \cdot 10^3$$

Приблизителна стандартна стойност  $R = 1,5 \text{ K}$  .

На фиг. 5.1.1.3. е показана възможност за смяна на резистора  $R_{2p}$  с ценер-диод, при което се получава стабилизатор на напрежение. Изчисляването на такава схема изисква ползуването на по-подробна справочна литература и данни за стабилитрона, което е извън задачите на настоящето ръководство.

Изходните сигнали на РОБКО-01 се получават от интегрални схеми с отворен колекторен изход и повишено напрежение. Имат следните логически нива :



- логическа нула :  $U_{\max(0)} = 0,4V$  при  $I_{(0)} = 40 \text{ mA}$
- логическа единица :  $U_{\min(1)} = E_{\text{захр}}$  при  $I_{(1)} = 0,25 \text{ mA}$

Нивата на изходните сигнали са неинвертирани по отношение на определящите ги разряди за данни.

На фиг. 5.1.1.4 е показано в общ вид свързване на външно устройство към изхода на интегрална схема с отворен колектор. Това свързване на външното устройство е възможно, ако електрическите параметри на товара не превишават максималните стойности на изходните сигнали. В противен случай се налага включване на товара посредством транзисторно стъпало или реле.

На фиг. 5.1.1.5 е представено включване на транзистор от NPN тип в схеми ОЕ и ОК, а на фиг. 5.1.1.6 - на транзистор от PNP.

При избора на транзистор и изчисляване на съпротивленията, включени в базата му, трябва да се има в предвид стойността на базовия ток при насищане и захранващото напрежение на стъпалото в сравнение с максималните стойности на изходните сигнали. Едно такова примерно изчисляване на NPN стъпало, схема ОЕ, изисква следните съображения при изчисляване на  $R_b$  :

$$R_b \leq \frac{E}{I_{bn}} - \frac{U}{I_{bn}} ; \quad R_b \geq \frac{E}{I_{\text{изх}(0)}}$$

#### 5.1.2. ВРЪЗКА С РАБОТНАТА СРЕДА - ПРОГРАМНО ОСИГУРЯВАНЕ

При програмиране с микрокомпютър ИМКО-2 входно-изходните сигнали на работа в РРМ са инверсни, в сравнение с данните на програмата. Това се дължи на инверсията, въведена от адаптерната платка за връзка с РОБКО-01. Тази особеност трябва да се има в предвид при реализацията на програми за входно-изходни сигнали.

#### ВХОДОВЕ КЪМ РОБКО-01

На фиг. 5.1.2.1 е представена програма "ВХОД" - програма за четене на входни сигнали. В нея се използва инструкцията РЕЕК за четене на входните сигнали, които са разделени на две групи по четири. Всяка група представлява старши полубайт данни, намиращи се на адреси АА и АВ (виж табл. 5.1.1.-А).

Целесъобразно е посочените адреси да се изчисляват в началото на програмата за връзка с работната среда, като им се присвояват символични имена (АА и АВ). Към тях програмата ще се обръща при нужда.

Програмата ВХОД разпечатва (маскира) всеки от четирите бита на зададения адрес. Тя може да се ползува за тестване работоспособността на цифровите входове, като на съединител А (фиг. 5.1.1.1) свържем показаната на фиг. 5.1.1.2 изпитателна схема и задействуваме програмата. При отворени ключове, на екрана следва да се разпечатва периодично по един нов ред, съдържащ осем нули, съответстващи на входните сигнали в избраната група. При затваряне на кой и да е от ключовете, следва да се промени съответстващата нула в единица и обратно.

#### ИЗХОДИ ОТ РОБКО-01

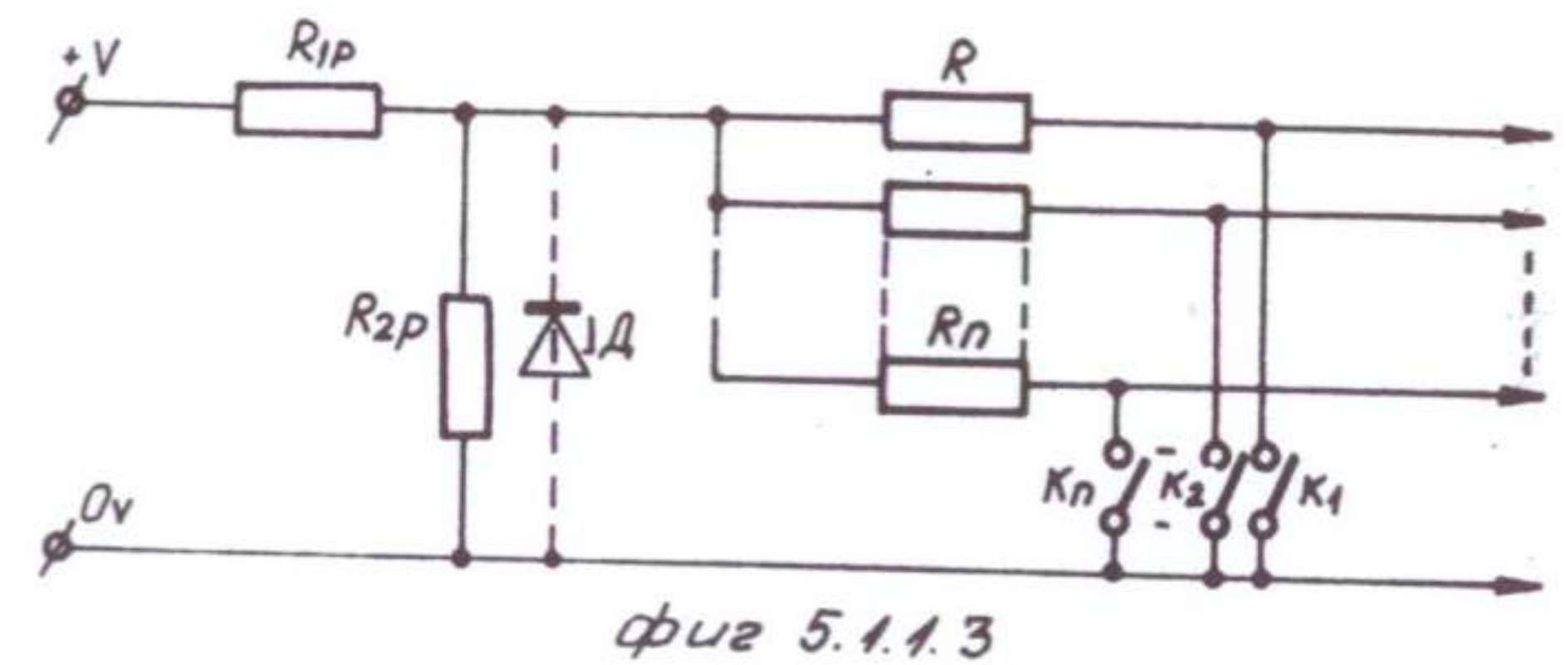
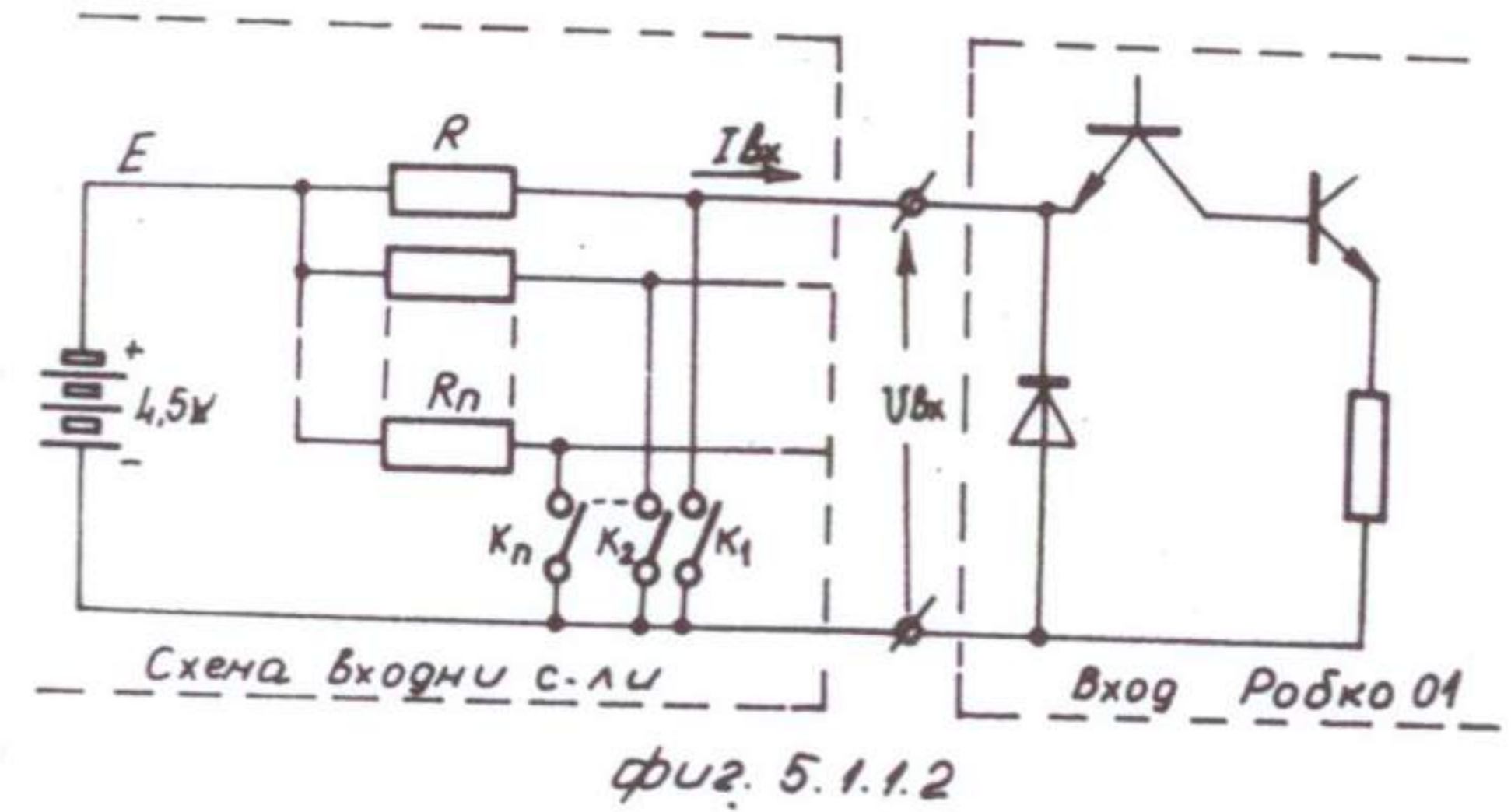
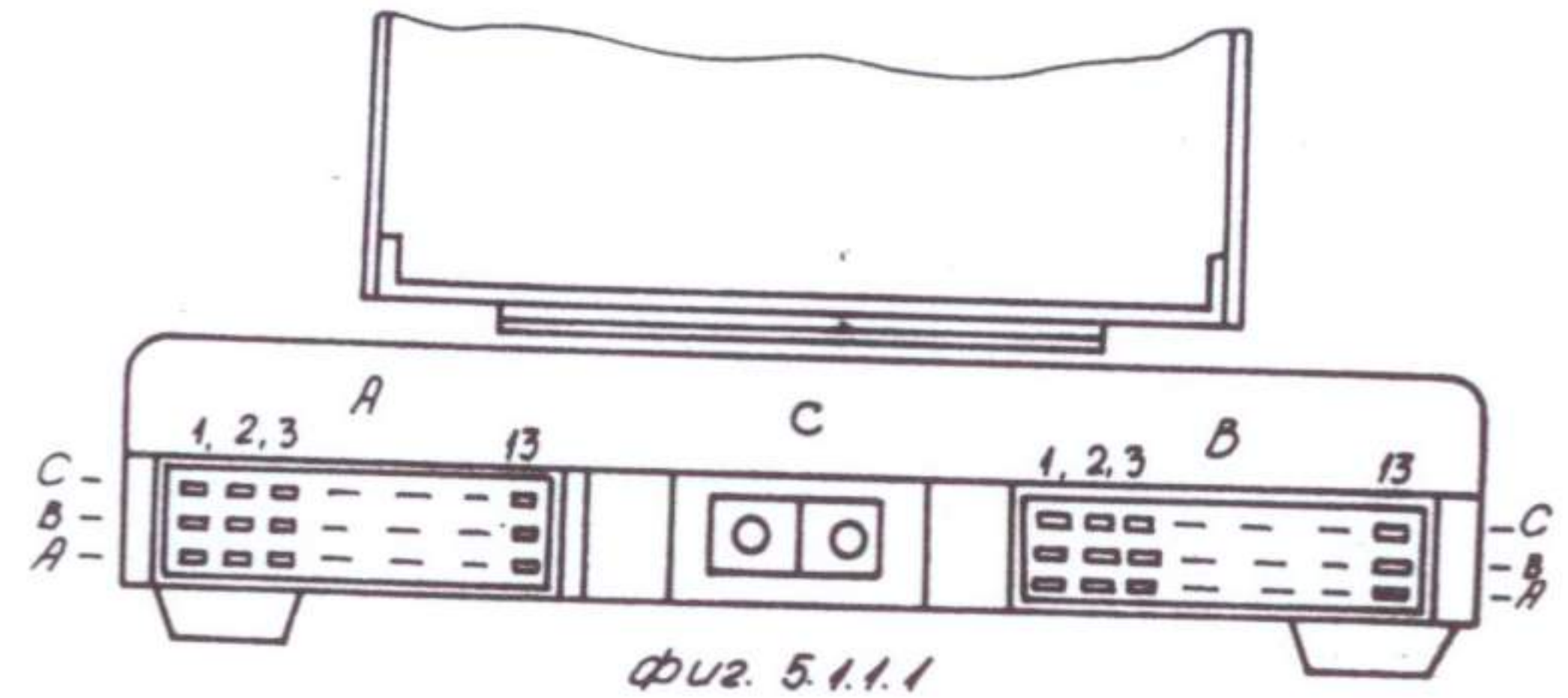
На фиг. 5.1.2.2 е представена програма "ИЗХОД". Тя е предназначена за подаване на изходни сигнали от РОБКО-01. Използува се инструкцията РОКЕ за запис на изходните сигнали на две групи по четири, като младши полубайт данни, намиращи се съответно на адреси АА и АВ (виж табл. 5.1.1.1-Б).

При работа с програмата следва да свържем към всеки един от изходите на избраната група, по едно транзисторно стъпало (фиг. 5.1.1.5) с включен светодиод или друг индикатор.

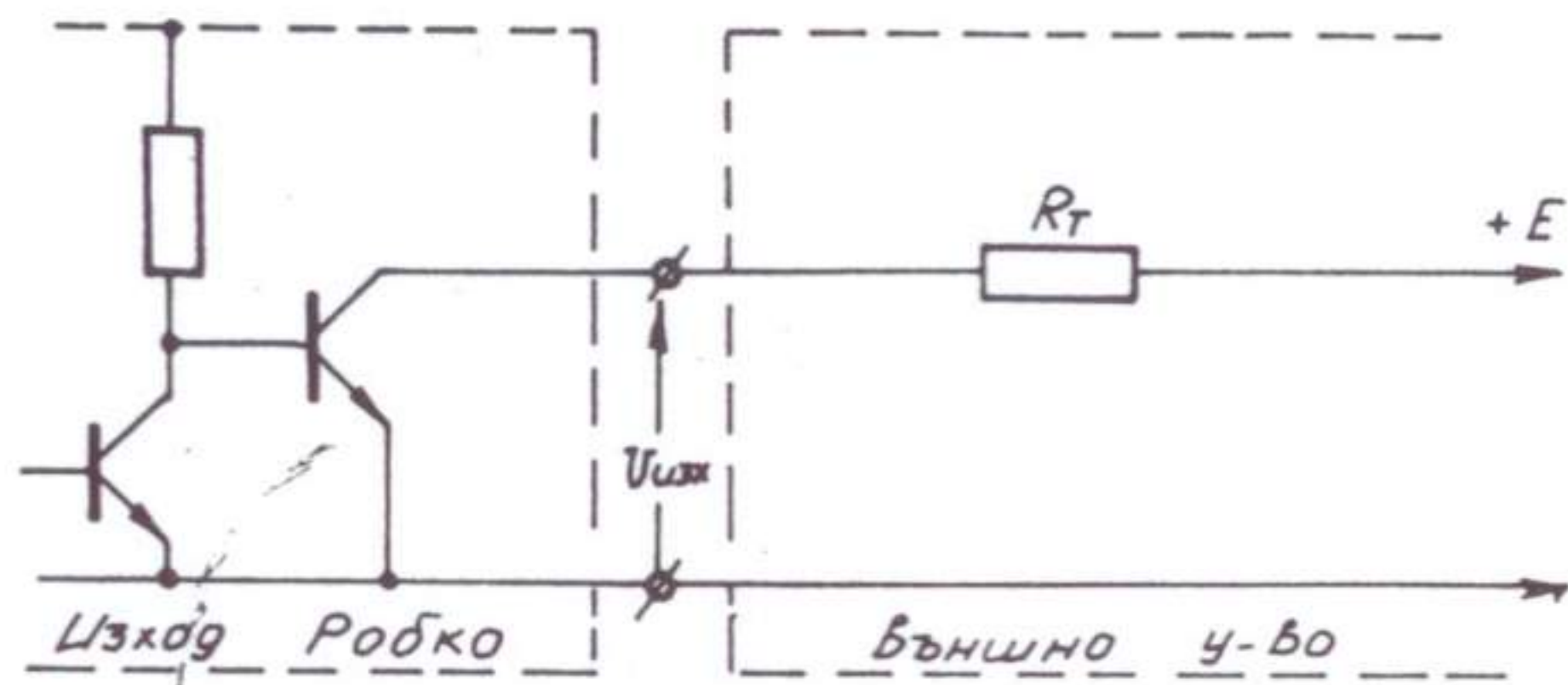


След задействане на програмата, на екрана на монитора се разпечатват последователно редове, които съдържат по осем елемента, представляващи нули, единици или произволна комбинация от тях. Едновременно с това се задействуват и индикаторите, включени към селектираната група изходни сигнали.

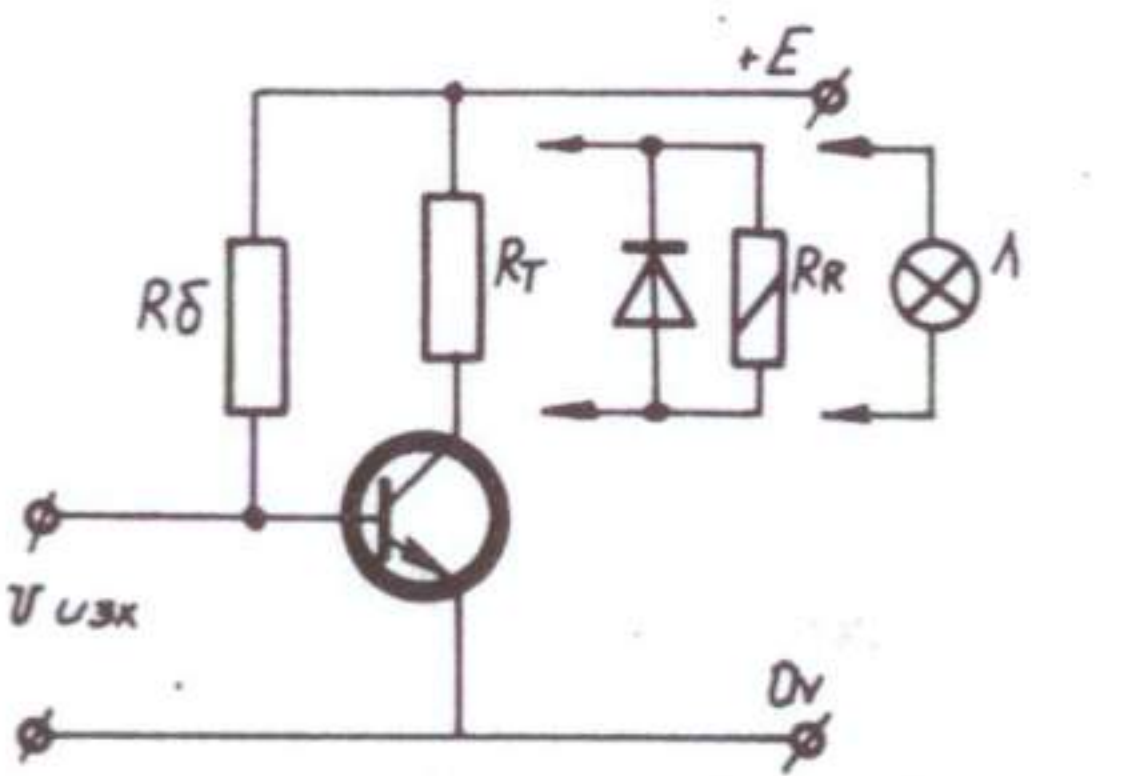
Описаните програми ВХОД и ИЗХОД са само примерни. Те могат да бъдат включвани в програми за осъществяване на връзка на работа с работна среда, посредством входно-изходни сигнали.



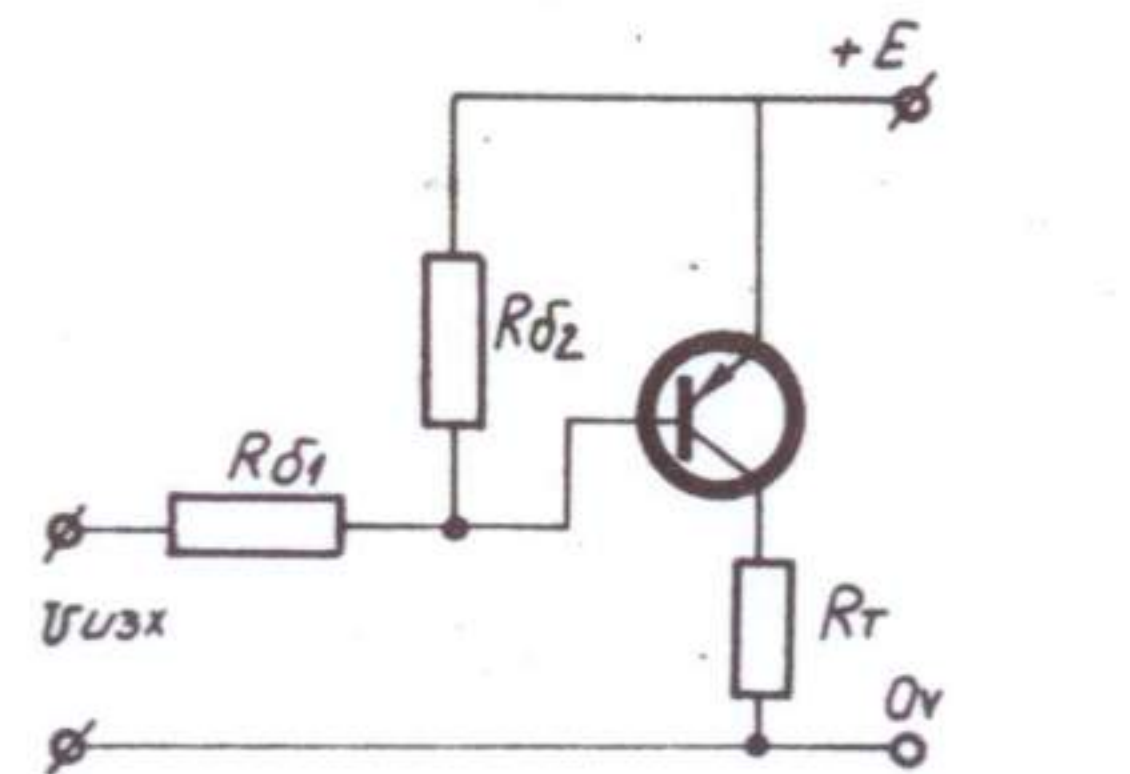
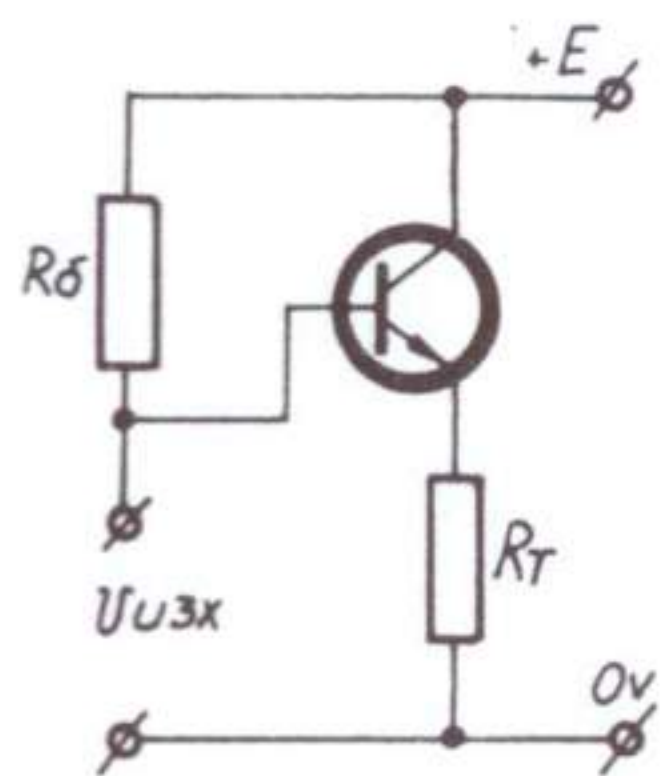




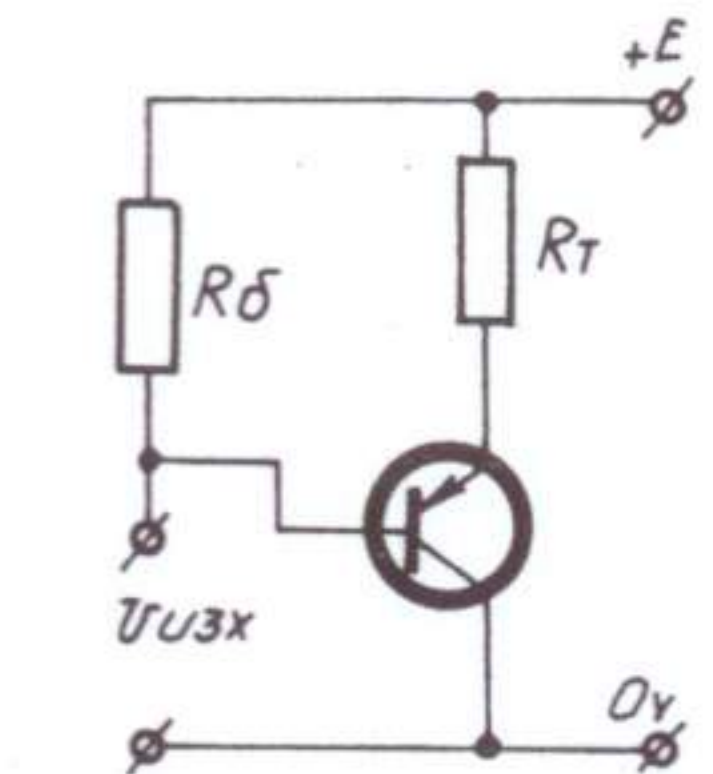
Фиг. 5.1.1.4



Фиг. 5.1.1.5



Фиг. 5.1.1.6



Фиг. 5.1.2.1. Програма ВХОД

LIST

```

9 HOME : VTAB 5
10 PRINT TAB( 8) "*****"
15 PRINT
20 PRINT TAB( 8) " ПРОГРАМА ВХОД 1.3 "
25 PRINT
30 PRINT TAB( 8) "*****"
35 VTAB 16: PRINT "-----"
40 VTAB 18
50 PRINT "COPYRIGHT (C)      В.ГЕОРЧЕВ,"
60 PRINT "                      К.БЕЛОВ": PRINT
70 PRINT "                      1985      ИТКР - БАН"
80 VTAB 23: PRINT "-----"
90 FOR X = 1 TO 2000: NEXT X
100 HOME : INPUT "SLOT NR ? "; SL
110 IF SL < 1 OR SL > 7 THEN 100
120 A1 = 49280 + SL * 16 + 6: REM ADR OF PORTS
130 A2 = A1 + 1
135 PRINT : PRINT TAB( 10) "СЪСТОЯНИЕ НА ВХОДОВЕТЕ": PRINT
140 I1 = PEEK (A1): REM FIRST ADRS
150 GOSUB 10000: REM SUB MASK
160 N3 = Z3:N2 = Z2:N1 = Z1:N0 = Z0
170 I1 = PEEK (A2): REM SECOND ADRS
180 GOSUB 10000: REM SUB MASK
190 N7 = Z3:N6 = Z2:N5 = Z1:N4 = Z0
195 HTAB 17
200 PRINT N7;N6;N5;N4; " ";N3;N2;N1;N0
210 ST = PEEK ( - 16384): IF ST = 160 THEN 230
220 GOTO 140
230 END
10000 REM SUBR <MASK>
10030 ZZ = I1
10040 Z3 = INT (ZZ / 128)
10050 IF Z3 = 0 THEN 10070
10060 ZZ = ZZ - 128
10070 Z2 = INT (ZZ / 64)
10080 IF Z2 = 0 THEN 10100
10090 ZZ = ZZ - 64
10100 Z1 = INT (ZZ / 32)
10110 IF Z1 = 0 THEN 10130
10120 ZZ = ZZ - 32
10130 Z0 = INT (ZZ / 16)
10150 RETURN
    
```



## LIST

```

9 HOME : VTAB 5
10 PRINT TAB( 8) "*****"
15 PRINT
20 PRINT TAB( 8) " ПРОГРАМА ИЗХОД 1.3 "
25 PRINT
30 PRINT TAB( 8) "*****"
35 VTAB 16: PRINT "-----"
40 VTAB 18
50 PRINT "COPYRIGHT (C)      В.ГЕОРЧЕВ,"
60 PRINT "                  К.БЕЛОВ": PRINT
70 PRINT "                  1985      ИТКР - БАН"
80 VTAB 23: PRINT "-----"
90 FOR X = 1 TO 2000: NEXT X
95 HOME
100 INPUT "SLOT NR ? "; SL
110 A1 = 49280 + SL * 16 + 6
120 A2 = A1 + 1
125 PRINT : PRINT TAB( 10) "СЪСТОЯНИЕ НА ИЗХОДИТЕ": PRINT
130 FOR J = 0 TO 3
140 N = 15 - 2 * J
150 POKE A1, N
160 POKE A2, 15
170 I1 = N
180 GOSUB 10000
190 N3 = Z3: N2 = Z2: N1 = Z1: N0 = Z0
200 PRINT TAB( 16) "1111 "; N3; N2; N1; N0
210 FOR X = 1 TO 600: NEXT X
220 NEXT J
230 FOR J = 0 TO 3
240 N = 15 - 2 * J
250 POKE A2, N
260 POKE A1, 15
270 I1 = N
280 GOSUB 10000
290 N7 = Z3: N6 = Z2: N5 = Z1: N4 = Z0
300 PRINT TAB( 16) N7; N6; N5; N4; " 1111"
310 FOR X = 1 TO 600: NEXT X
320 NEXT J
330 ST = PEEK ( - 16384): IF ST = 160 THEN 9000
340 PRINT
350 GOTO 130
9000 END
10000 REM SUBR <MASK>
10030 ZZ = I1
10040 Z3 = INT (ZZ / 8)
10050 IF Z3 = 0 THEN 10070
10060 ZZ = ZZ - 8
10070 Z2 = INT (ZZ / 4)
10080 IF Z2 = 0 THEN 10100
10090 ZZ = ZZ - 4
10100 Z1 = INT (ZZ / 2)
10110 IF Z1 = 0 THEN 10130
10120 ZZ = ZZ - 2
10130 Z0 = INT (ZZ / 1)
10150 RETURN

```

Фиг. 5.1.2.2. Програма ИЗХОД

RUN

\*\*\*\*\*

ПРОГРАМА ИЗХОД 1.3

\*\*\*\*\*

```

-----
COPYRIGHT (C)      В.ГЕОРЧЕВ,
                  К.БЕЛОВ

```

```

1985      ИТКР - БАН
-----

```

SLOT NR ? 4

СЪСТОЯНИЕ НА ИЗХОДИТЕ

```

1111 1110
1111 1101
1111 1011
1111 0111
1110 1111
1101 1111
1011 1111
0111 1111

1111 1110
1111 1101
1111 1011
1111 0111
1110 1111
1101 1111
1011 1111
0111 1111

1111 1110
1111 1101
1111 1011
1111 0111
1110 1111
1101 1111
1011 1111
0111 1111

```

Фиг. 5.1.2.3. Примерно изпитание на програма ИЗХОД