BACK

Практическо упражнение No 7

МЕТОДИ И ОПЕРАЦИОННИ БЛОКОВЕ ЗА СУМИРАНЕ НА ДВОИЧНИ ЧИСЛА С ФИКСИРАНА ЗАПЕТАЯ

1. Цел на упражнението:

Целта на упражнението е, работейки с програмните модели на операционните блокове за сумиране, студентите да добият по-ясна представа за начините за реализиране на съответните методи, а също така да се установи степента на усвояване на микроалгоритмите за сумиране.

2. Теоретична част:

Всички разгледани по-долу методи и блокове работят с операнди с ляво фиксирана запетая (правилни двоични дроби). Методите и блоковете работещи с операнди с дясно фиксирана запетая (цели числа) са аналогични.

Известно е, че в ОП числата се съхраняват само в ПК или само в ДК (положителните в ПК, а отрицателните в ДК). В този си вид операндите се извличат от ОП и се изпращат в АЛУ. Ето защо, в зависимост от това в какъв код се съхраняват и получават числата от ОП, операциите в АЛУ могат да бъдат съответно в ПК или в ДК. Преди изпращане на резултата от АЛУ в ОП, той предварително трябва да се представи в същия код.

2.1. Сумиране на двоични числа с фиксирана запетая в прав код.

В този случай числата се съхраняват в ОП в ПК. В такъв код те циркулират между ОП и АЛУ.

При построяването на блока за събиране и изваждане на числа в ПК възникват трудности свързани с реализирането на операцията изваждане (a - b) на числа с еднакви знаци, т.е. алгебрическо събиране [a + (-b)] на числа с различни знаци, тъй като ПК не е удобен за извършване на тази операция. Тези трудности се преодоляват като чрез използването на ОК или ДК операцията алгебрическо събиране на числа се свежда към аритметическо събиране на техните кодове.

Предвид на това, че резултатът от операцията трябва да се върне в ОП в ПК, е целесъобразно като помощен на ПК да се използва ОК, тъй като преобразуването на отрицателния резултат от ОК в ПК става без за това да е необходимо допълнително време. На практика това става като кодът се вземе от инверсните изходи на тригерите на регистъра, в който се записва резултатът.

Използването на ДК би намалило бързодействието на блока, тъй като преобразуването на отрицателния резултата от ДК в ПК става чрез инвертиране и прибавяне на единица към младшия разряд, т.е. изисква допълнително време, равно на времето за събирането на две числа. Следователно за средното време за сумиране в ПК с използване на ДК може да се напише, че

$t_{С\ ДК} = (1 + p_{OP})\ t_{С\ ОК}$,

където p_{OP} е вероятността резултатът да е отрицателен.

Микроалгоритъм на операцията сумиране в ПК с използване на ОК :

От ОП в АЛУ се получават:

$[X]_{ПК} = x_0,x_1x_2 \ldots x_n$

и

$[Y]_{ПК} = y_0,y_1y_2 \ldots y_n$.

От АЛУ в ОП трябва да се изпрати:

$[Z]_{ПК} = [X\ S_0\ Y]_{ПК} = z_0,z_1z_2 \ldots z_n$.

По-горе с x_0, y_0, и z_0 са означени кодовете на знаците на операндите и на резултата, а със S_0 - кодът на операцията: "+" $\rightarrow$ 0, "-" $\rightarrow$ 1.

Очевидно е, че сумиращият блок ще трябва да включва в състава си следните основни възли: Px, Py, Σ, Pz и БМУ.

Микроалгоритъмът на операцията сумиране се основава на правилото за алгебрическо събиране с използване на ОК, като същото се модифицира с оглед на конкретната му схемна реализация и като се отчита, че от ОП операндите се получават в ПК и резултатът трябва да бъде върнат в ОП също в ПК.

По-долу е дадена словесната форма на микроалгоритъма:

а. Ако $x_0 = 0$, то в Σ да се предаде (Px).

Ако $x_0 = 1$, то в Σ да се предаде ($\overline{Px}$), т.е. инверсната стойност на съдържанието на Px. Инвертира се само мантисата, а в знаковите разряди на Σ се подават единици.

б. Ако $S_0 = 0$ и $y_0 = 0$ или $S_0 = 1$ и $y_0 = 1$, т.е. ако $\overline{S}_0\,\overline{y}_0 \vee S_0 y_0 = 1$, то в Σ да се предаде (Py). В знаковите разряди на Σ се подават нули. Това е така, тъй като при

(X+Y) и Y>0

последният трябва да бъде в ПК, а при

(X - Y) и Y<0, т.е. при [X + (-Y)] и (-Y)>0

последният трябва също да бъде в ПК.

Ако $S_0 = 0$ и $y_0 = 1$ или $S_0 = 1$ и $y_0 = 0$, т.е. ако $\overline{S}_0\,\overline{y}_0 \vee S_0 y_0 = 0$, то в Σ да се предаде ($\overline{Py}$), т.е. инверсната стойност на съдържанието на Py. Инвертира се само мантисата, а в знаковите разряди на Σ се подават единици. Това е така, тъй като при

(X+Y) и Y<0

последният трябва да бъде в ОК, а при

(X - Y) и Y>0, т.е. при [X + (-Y)] и (-Y)<0

последният трябва също да бъде в ОК.

в. Аритметическо събиране на модифицираните кодове на X и Y в ∑ с отчитане на цикличния пренос и приемане на кода на Z в Pz.

Формиране на признаците на резултата (OV, Z, N, C и др.)

г. Ако $z_0 = 0$, то в ОП да се предаде (Pz).

Ако $z_0 = 1$, то в ОП да се предаде ($\overline{Pz}$), т.е инверсната стойност на съдържанието на Pz. Инвертира се само мантисата, а знаковият разряд се предава директно.

Класическата схема на операционния блок за сумиране в ПК с използване на ОК е показана на фиг.1. Действието на схемата ще бъде обяснено от момента, в който в Px и в Py са заредени правите кодове на двата операнда. Подава се сигнал НО. Ако операндът X >0, т.е. ако $x_0 = 0$, респ. на нулевия изход на знаковия разряд на Px има “1”, то се отваря елементът “И-1” и сигналът постъпва на шината за ПрК от Px в ∑. В противен случай се отваря елементът “И-2” и сигналът постъпва на шината за ПрИК като същевременно се подава и към двата знакови разряда на ∑. Едновременно с това, ако $\overline{S}_0\overline{y}_0 \vee S_0y_0 = 1$, то се отваря елементът “И-3” и сигналът постъпва на шината за ПрК от Py в ∑. В противен случай се отваря елементът “И-4” и сигналът постъпва на шината за ПрИК като същевременно се подава и към двата знакови разряда на ∑. Така се изпълнява изискването положителните операнди да се представят в ПК, а отрицателните - в ОК. В ∑ се извършва аритметическо събиране на модифицираните кодове на операндите. Ако при събирането на старшите знакови разряда се получи пренос, то съгласно микроалгоритъма същият се прибавя към най-младшия разряд на ∑. В резултат се получава кодът на сумата на операндите. След време τ_1, което трябва да бъде по-голямо или равно на времето, необходимо за предаването на кодовете от Px и Py в ∑ и за тяхното аритметическо събиране, се получава сигнал за ПК от ∑ в Pz. Ако при събиране на кодовете се получи препълване на разрядната мрежа, т.е. резултат, който по модул е ≥ 1, то в двата знакови разряда на суматора ще има различни цифри (01, ... или 10, ...), на изхода на елемента “M2” ще се получи “1”, ще се отвори елементът “И-7” и ще се формира сигнал ПрРМр, който ще предизвика вдигането на флага OV в регистъра на признаците на резултата. Тъй като при препълване на разрядната мрежа за знака на резултата се съди по цифрата в старшия знаков разряд на суматора, то именно той се свързва със знаковия разряд на Pz. След време τ_2, което трябва да бъде по-голямо или равно на времето, необходимо за приемане на кода в Pz, сигналът постъпва на входовете на елементи “И-5” и “И-6” и преминава през този от тях, на който е подадена “1” от знаковия разряд на Pz. Например, ако резултатът е положителен, т.е. z_0=0, то се отваря елементът “И-6” и се формира сигналът за ПрК. В противен случай се отваря елементът “И-5” и се формира сигналът за ПрИК, като при това се инвертира само мантисата, а знаковият разряд се предава директно. Така в ОП се предава правият код на резултата и с това операцията приключва.

Действието на блока за сумиране е пояснено и чрез цифровата диаграма на фиг.2.

Времето за сумиране на двата операнда с показаната схема може да се определи по следната формула:

$t_C = \tau_1+\tau_2 = t_{ПрК} + t_\Sigma + t_{ПК}$.

За да се намали t_C , трябва да се използват синхронни суматори с ускорен пренос, при които t_Σ е най-малко.

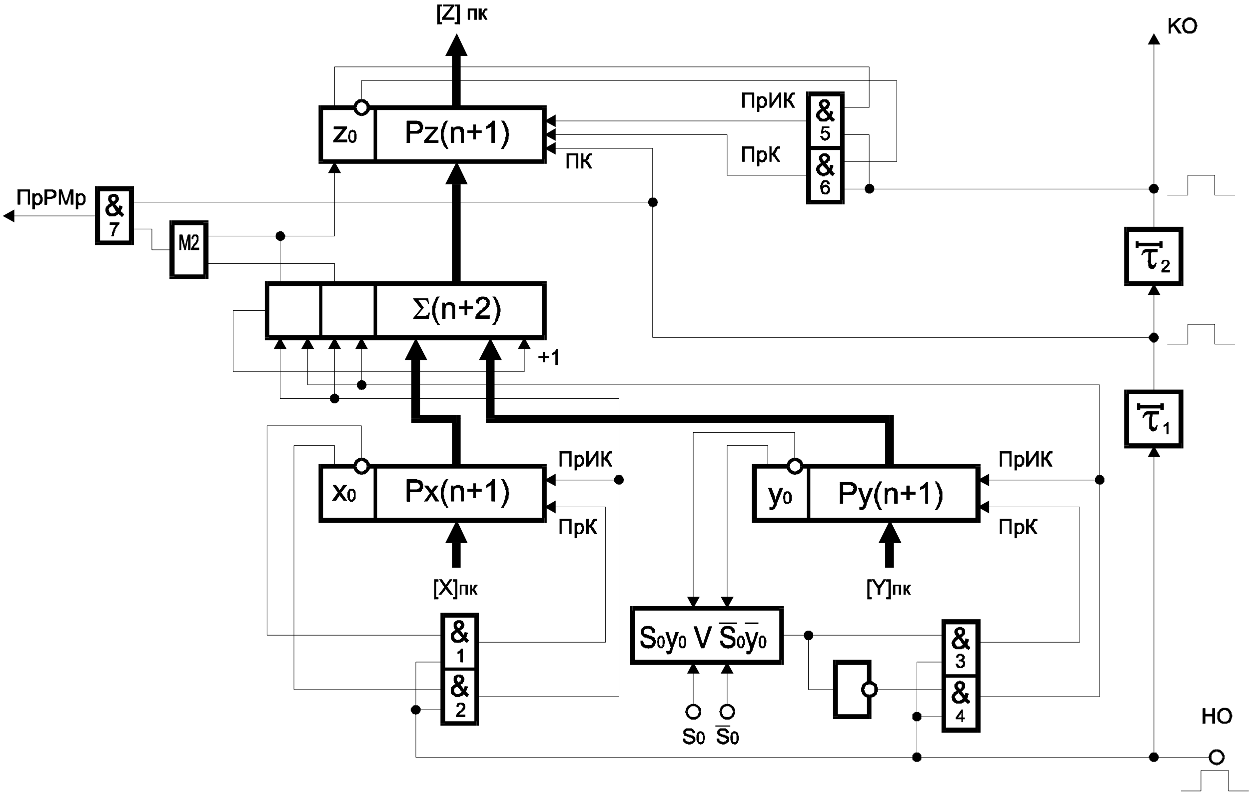
От гледна точка на теорията на автоматите този блок може да се разглежда като операционен блок (ОБ), състоящ се от операционна част (ОЧ) и управляваща част (УЧ), наричан още блок за местно управление (БМУ).

Към ОЧ се отнасят регистрите и суматорът.

Към УЧ се отнасят закъснителните вериги (τ_1 и τ_2), логическите схеми, които формират управляващите сигнали - ПрК, ПрИК, признаците на резултата и др.

ОЧ може да се разглежда и синтезира като автомат на Мур, но на практика се проектира с използване на готови схеми на регистри и суматори.

УЧ може да бъде реализирана схемно или микропрограмно. В първия случай УЧ може да се разглежда и синтезира като автомат на Мур или Мили като възможните структури са две - регистър на състоянието и комбинационна схема или датчик на управляващи импулси и комбинационна схема. В горната схема е използвана втората от двете структури, тъй като това съществено облекчава обяснението на действието й.



Фиг.1. Схема на операционен блок за сумиране в ПК

X = + 0,0011
Y = - 0,1001
Z = X + Y

в ОП	
$[X]_{ПК}$	0,0011
$[Y]_{ПК}$	1,1001

в АЛУ		
в Px	$[X]_{ПК}$	0,0011
в Py	$[Y]_{ПК}$	1,1001
в $\sum$	$[X]_{МПК}$	00,0011
	$[Y]_{МОК}$	11,0110
	$[Z]_{МОК}$	11,1001
в Pz	$[Z]_{ОК}$	1,1001

Признаци на резултата	Z=0, N=1 C=0 OV=0

в ОП	
$[Z]_{ПК}$	1,0110

X = - 0,1010
Y = + 0,0111
Z = X - Y =
= X + (-Y)

в ОП	
$[X]_{ПК}$	1,1010
$[Y]_{ПК}$	0,0111

в АЛУ		
в Px	$[X]_{ПК}$	1,1010
в Py	$[Y]_{ПК}$	0,0111
в $\sum$	$[X]_{МОК}$	11,0101
	$[-Y]_{МОК}$	11,1000
	$[Z]_{МОК}$	10,1110
в Pz	$[Z]_{ОК}$	1,1110

Признаци на резултата	Z=0, N=1 C=1 OV=1

в ОП	
$[Z]_{ПК}$	1,0001

Фиг.2. Цифрова диаграма на блока за сумиране в ПК

2.2. Сумиране на двоични числа с фиксирана запетая в допълнителен код.

В този случай числата се съхраняват в ОП в ДК (положителните в ПК, а отрицателните в ДК). В такъв код те циркулират между ОП и АЛУ.

Микроалгоритъм на операцията сумиране в ДК :

Както и в предния случай, в основата на този микроалгоритъм стои правилото за алгебрическо събиране с използване на ДК.

а. В $\sum$ да се предаде (Px).

б. Ако $S_0 = 0$, то в $\sum$ да се предаде (Py).

Ако $S_0 = 1$, то в $\sum$ да се предаде ($\overline{Py}$), т.е. инверсната стойност на Py и да се добави "1" в младшия разряд на $\sum$. Това е така, тъй като при (X - Y) = [X + (-Y)], ако

Y > 0, то той е получен от ОП в ПК, а

(-Y) < 0 и следователно трябва да се представи в ДК, т.е. налага се преминаване от ПК на положително число към ДК на симетричното му отрицателно число, а ако

Y < 0, то той е получен от ОП в ДК, а

(-Y) > 0 и следователно трябва да се представи в ПК, т.е. налага се преминаване от ДК на отрицателно число към ПК на симетричното му положително число, което става по същия начин, както преминаването от ПК в ДК;

в. Аритметическо събиране на модифицираните кодове на X и Y в $\sum$ и приемане на кода на Z в Pz.

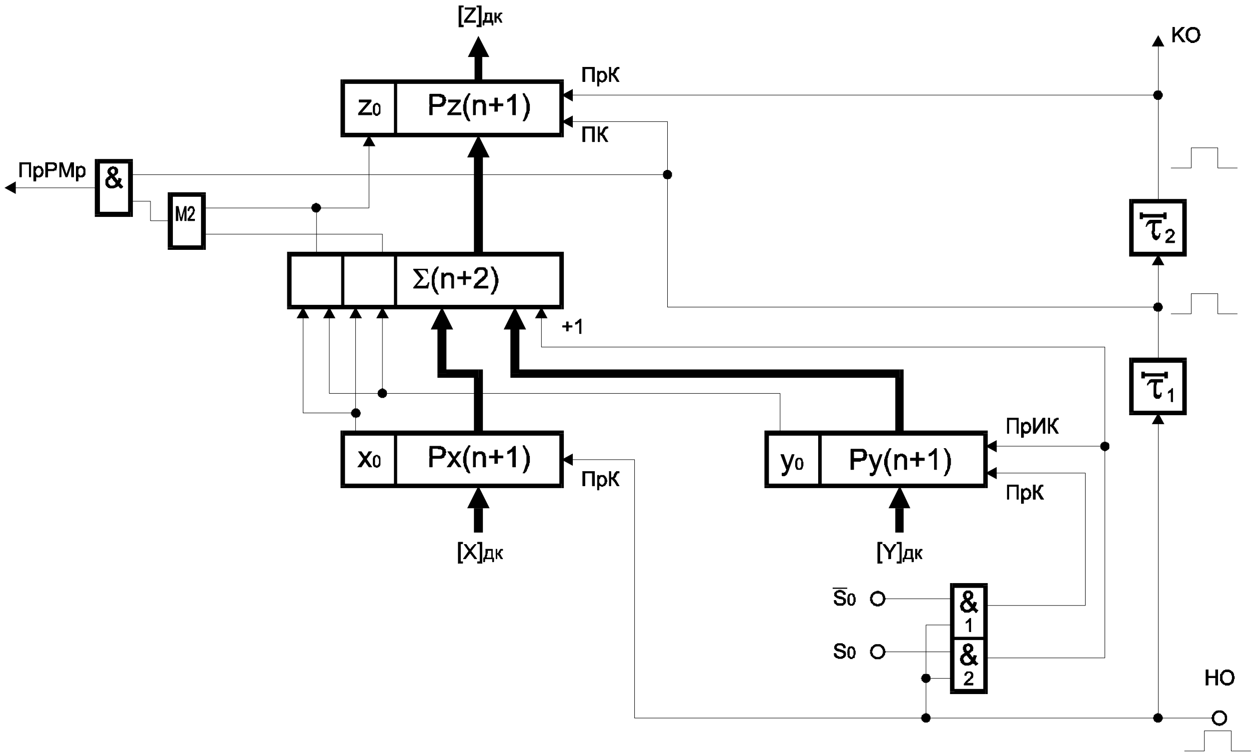
Формиране на признаците на резултата (OV, Z, N, C и др.)

г. Предаване на (Pz) в ОП.

Очевидно е, че този алгоритъм е по-прост от предния, което води до опростяване и на схемата на сумиращия блок. Това е една от причините в някои компютри да се работи само в ДК включително и при умножение и деление, въпреки трудностите, свързани с извършването на тези две операции в такъв код.

Класическата схема на операционния блок за сумиране в ДК е показана на фиг.3. Действието на блока е пояснено чрез цифровата диаграма на фиг.4.

Бързодействието на този блок е същото като на предния, но разходът на апаратура е по-малък поради отсъствието на логически елементи за ПрИК Px и ПрИК Pz, а също и поради отпадането на някои компоненти на УЧ.



Фиг.3. Схема на операционен блок за сумиране в ДК

X = + 0,0011
Y = - 0,1001
Z = X + Y

в ОП	
$[X]_{ПК}$	0,0011
$[Y]_{ДК}$	1,0111

в АЛУ		
в Px	$[X]_{ПК}$	0,0011
в Py	$[Y]_{ДК}$	1,0111
в $\sum$	$[X]_{МПК}$	00,0011
	$[Y]_{МДК}$	11,0111
	$[Z]_{МДК}$	11,1010
в Pz	$[Z]_{ДК}$	1,1010

Признаци на резултата	Z=0, N=1 C=0 OV=0

в ОП	
$[Z]_{ДК}$	1,1010

X = - 0,1010
Y = + 0,0111
Z = X - Y =
= X + (-Y)

в ОП	
$[X]_{ДК}$	1,0110
$[Y]_{ПК}$	0,0111

в АЛУ		
в Px	$[X]_{ДК}$	1,0110
в Py	$[Y]_{ПК}$	0,0111
в $\sum$	$[X]_{МДК}$	11,0110
	$[-Y]_{МДК}$	11,1001
	$[Z]_{МДК}$	10,1111
в Pz	$[Z]_{ДК}$	1,1111

Признаци на резултата	Z=0, N=1 C=1 OV=1

в ОП	
$[Z]_{ДК}$	1,1111

Фиг.4. Цифрова диаграма на блока за сумиране в ДК

3. Задачи за изпълнение:

3.1. Да се преобразуват в двоичната система десетичните числа X = 0,...... и Y = 0,...... с точност 7 знака след запетаята.

3.2. Да се попълнят цифровите диаграми за ПК и ДК съответно за събиране и изваждане.

3.3. Да се извърши събиране и изваждане в ПК със съответния програмен модел (фиг.5) като междинните резултати се сравняват с цифровите диаграми.

3.4. Да се извърши събиране и изваждане в ДК със съответния програмен модел (фиг.6) като междинните резултати се сравняват с цифровите диаграми.

4. Контролни въпроси:

4.1. В какъв код е прието да се съхраняват операндите в ОП на компютъра?

4.2. Какво гласи правилото за алгебрическо събиране с използване на ОК?

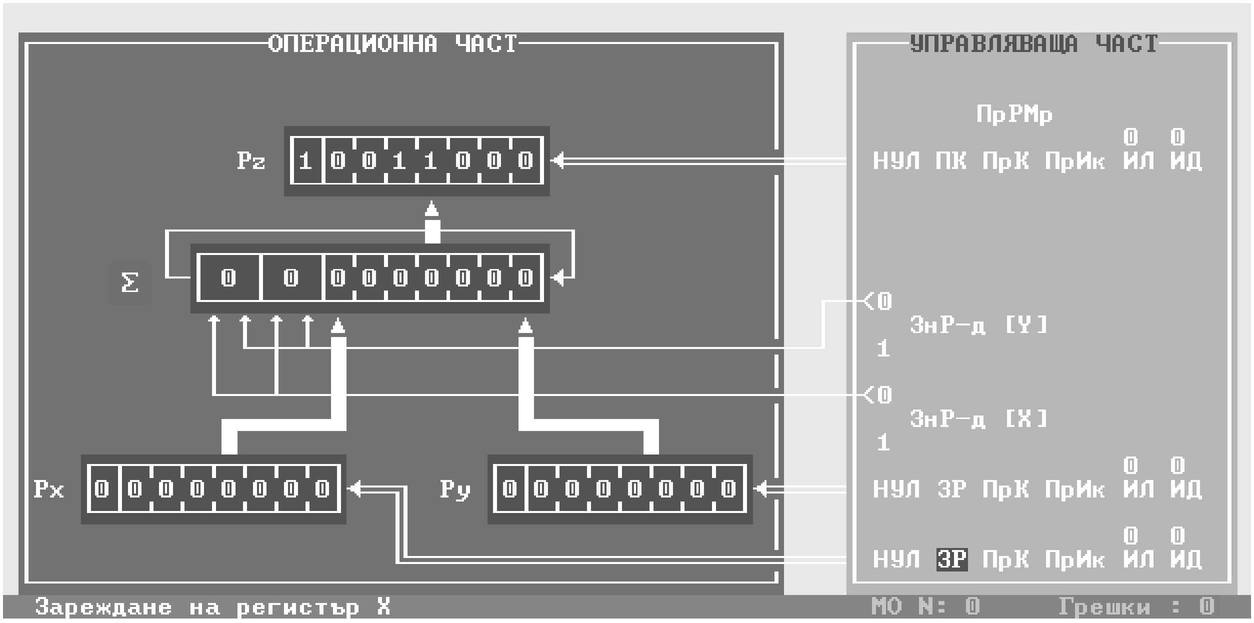
4.3. Какво гласи правилото за алгебрическо събиране с използване на ДК?

4.4. Кое от двете правила е по-просто за практическа реализация и защо?

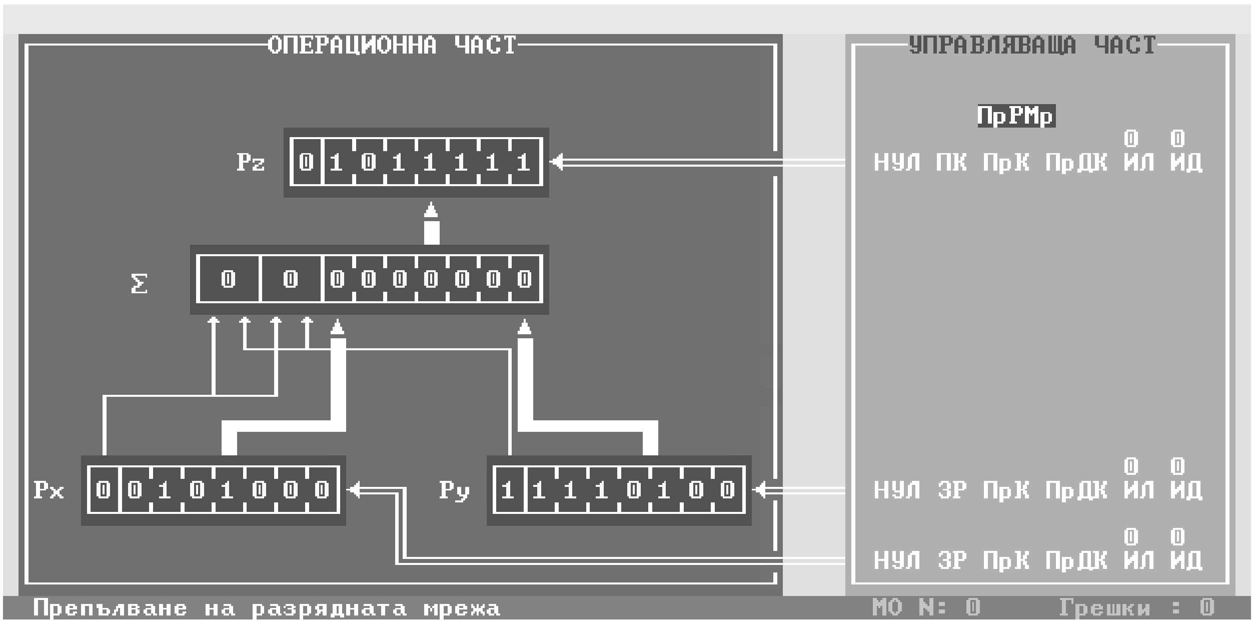
4.5. Каква разновидност на машинните кодове се използва при събирането им и защо?

4.6. Кои са признаците на резултата?

4.7. Как може да се намали времето за сумиране?



Фиг.5. Програмен модел на блока
за събиране и изваждане в ПК



Фиг.6. Програмен модел на блока
за събиране и изваждане в ДК