1.1.2. Машинни кодове на двоичните числа

Машинните кодове са алгоритмични средства за представяне и обработка на числовите операнди в компютрите. За двоичните числа са дефинирани три машинни кода – прав, обратен и допълнителен. Обратният и допълнителният код позволяват операцията изваждане да се сведе до операция събиране. Това опростява структурата на аритметично-логическото устройство (АЛУ) на процесора, т.к. за изпълнение на двете операции ще е необходим само един суматор.

Конкретните числа в текста и примерите по-нататък се представят в разрядна решетка с 4 двоични числови разряда ($n = 4$). При дробните числа знаковият разряд е отделен от числовите с десетична запетая, а не с апостроф, за да си личи, че числото е дробно. Тук разряди за представяне на евентуална цяла част на числото липсват.

1.1.2.1. Прав код

При правия код знакът на числата се указва със съдържанието на знаковия разряд – 0 за знак "+" и 1 за знак "–". В числовите разряди на решетката се представят директно цифрите от двоичния запис на съответното число – както положително, така и отрицателно. Ако с N е означено n-разрядно число със знак $\pm N = \pm a_{n-1} a_{n-2} ... a_0 ... a_1 a_0$, то

$$N_{\text{пк}} = \begin{cases} 0,N & \text{при } N \geq 0 \\ 1,N & \text{при } N \leq 0 \end{cases} \text{ за дробни числа и } N_{\text{пк}} = \begin{cases} 0'N & \text{при } N \geq 0 \\ 1'N & \text{при } N \leq 0 \end{cases} \text{ за цели числа}$$

Примери:

$$\left.\begin{aligned} A = +101_{(2)} \rightarrow A_{\text{пк}} = 0'0101 \\ A = -101_{(2)} \rightarrow A_{\text{пк}} = 1'0101 \end{aligned}\right\} \text{ за четириразрядна решетка + знаков разряд}$$

$$\left.\begin{aligned} A = +0{,}1011_{(2)} \rightarrow A_{\text{пк}} = 0{,}1011 \quad A = +0{,}01_{(2)} \rightarrow A_{\text{пк}} = 0{,}0100 \\ A = -0{,}1011_{(2)} \rightarrow A_{\text{пк}} = 1{,}1011 \quad A = -0{,}01_{(2)} \rightarrow A_{\text{пк}} = 1{,}0100 \end{aligned}\right\} \text{ за четириразрядна}$$

решетка + знаков разряд

Ако знаковият разряд се разглежда условно като числов, правият код на отрицателните двоични числа може да се дефинира формално чрез следните зависимости:

$$\left.\begin{aligned} &\text{– цяло число} \quad N_{\text{пк}} = 2^n - X \\ &\text{– дробно число} \quad N_{\text{пк}} = 2^0 - X \end{aligned}\right| N \leq 0$$

Диапазонът на представимите числа в прав код е:

– цели положителни числа: от 0 |0'00…0| до $2^n - 1$ |0'11…1|;
– цели отрицателни числа: от –0 |1'00…0| до $-(2^n - 1)$ |1'11…1|;
– дробни положителни числа: от 0 |0,00…0| до $1 - 2^{-n}$ |0,11…1|;
– дробни отрицателни числа: от –0 |1,00…0| до $-(1 - 2^{-n})$ |1,11…1|.

Число със стойност 0 има различно представяне – като положително и като отрицателно:

$0_{пк}$ = 0'00…0 или 0,00…0;
$-0_{пк}$ = 1'00…0 или 1,00…0.

Недостатък на правия код е това, че правилата за събиране на числа с различни знаци са твърде сложни и неудобни за реализация в компютърните системи, тъй като знаковите и числовите разряди се обработват поотделно и по различен начин. Затова правият код се използва само при въвеждане и извеждане на числата, при съхраняване на числата с плаваща запетая в паметта и евентуално при умножение и деление.

Представянето на двоичните числа в обратен и допълнителен код позволява операциите събиране и изваждане на числа с произволна комбинация от знаци да се реализират само с операция сумиране. Знаковите разряди на положителните и на отрицателните числа се разглеждат като числови и се обработват аритметично заедно с останалите фактически числови разряди. Двете изходни числа, подлежащи на обработка, се представят в един и същи изходен код. Резултатът се получава в същия код, като автоматично се формира и неговия знак.

1.1.2.2. Обратен код

Обратният код на положителните числа (както цели, така и дробни) съвпада изцяло с техния прав код.

Обратният код на едно отрицателно двоично число се получава от неговия прав код, като в знаковия разряд се запише 1 (знак "–"), а цифрите в числовите разреди се инвертират (0 се заменя с 1 и обратно).

$$N_{ок} = \begin{cases} 0, N \text{ при } N \geq 0 \\ 1, \underline{a_{n-1}}\ \underline{a_{n-2}} \ldots \underline{a_0} \text{ при } N \leq 0 \end{cases} \quad \text{за дробни числа}$$

$$N_{ок} = \begin{cases} 0' N \text{ при } N \geq 0 \\ 1' \underline{a_{n-1}}\ \underline{a_{n-2}} \ldots \underline{a_0} \text{ при } N \leq 0 \end{cases} \quad \text{за цели числа,}$$

където $\underline{a_i}$ е допълнение на a_i до $p - 1$ (p – основа на бройната система).

Ако отрицателните числа са представени предварително в прав код, при преобразуването им в обратен се извършва инвертиране само на числовите разряди, а знаковият разряд остава със съдържание 1. При преобразуване от обратен в прав код се извършва пак същата операция – инвертират се само числовите разряди.

Примери:

$$A = +110_{(2)} \rightarrow A_{пк} = 0'0110 \rightarrow A_{ок} = 0'0110$$
$$A = +0{,}01_{(2)} \rightarrow A_{пк} = 0{,}0100 \rightarrow A_{ок} = 0{,}0100$$
$$A = -110_{(2)} \rightarrow A_{пк} = 1'0110 \rightarrow A_{ок} = 1'1001$$
$$A = -0{,}01_{(2)} \rightarrow A_{пк} = 1{,}0100 \rightarrow A_{ок} = 1{,}1011$$

Както и при правия код, при обратния код също има положителна и отрицателна 0. Диапазоните на представимите числа са същите като при правия код.

Обратният код на отрицателните двоични числа се дефинира формално чрез зависимостите:

$$\left.\begin{array}{ll} -\text{цяло число} & N_{ок} = 2^{n+1} + N - 1 \\ -\text{дробно число} & N_{ок} = 2^{1} + N - 2^{-n} \end{array}\right| \; N \leq 0\,.$$

За обратния код са характерни следните особености:

– ако при събиране на две числа се получи пренос 1 след знаковия разряд на резултата (от знака наляво), той се прибавя към най-младшия разряд на получения до момента резултат. След извършване на новото сумиране се получава крайният резултат. Този пренос се нарича цикличен;

– ако резултатът от събирането е положително число, той е окончателен т.к. за положителните числа обратният код съвпада с правия и не е необходимо някакво допълнително преобразуване;

– ако резултатът от събирането е отрицателно число, той се преобразува в прав код, като се инвертират само числовите му разряди (преобразуването се прави само, ако е необходимо да получим правия код на резултата);

– при събиране на числа, равни по абсолютна стойност, но с различни знаци, се получава „отрицателна" нула, представена в обратен код ($-0_{ок}$ = 1'11...1 или 1,11...1). В компютърните системи тя се преобразува автоматично в „положителна" нула, представена в прав код ($+0_{пк}$ = 0'00...0 или 0,00...0).

Свойствата на обратния код се илюстрират със следните примери:

а) Събиране на целите числа A = $0101_{(2)}$ и B = $1001_{(2)}$

$$
\begin{aligned}
A = +0101 \rightarrow A_{пк} = 0'0101 \rightarrow A_{ок} &= 0'0101 \quad + \\
B = +1001 \rightarrow B_{пк} = 0'1001 \rightarrow B_{ок} &= \underline{0'1001} \\
(A + B)_{ок} &= 0'1110 = (A + B)_{пк}
\end{aligned}
$$

$$
\begin{aligned}
A = -0101 \rightarrow A_{пк} = 1'0101 \rightarrow A_{ок} &= 1'1010 \quad + \\
B = -1001 \rightarrow B_{пк} = 1'1001 \rightarrow B_{ок} &= \underline{1'0110} \\
1| &\; 1'0000 \\
&+ \text{ цикличен пренос} \\
&\underline{\qquad 1} \\
(A + B)_{ок} &= 1'0001 \rightarrow (A + B)_{пк} = 1'1110
\end{aligned}
$$

б) Събиране на дробните числа $A = 0{,}0100_{(2)}$ и $B = 0{,}0111_{(2)}$

$$
\begin{aligned}
A = +0{,}0100 \rightarrow A_{пк} = 0{,}0100 \rightarrow A_{ок} &= 0{,}0100 \quad + \\
B = -0{,}0111 \rightarrow B_{пк} = 1{,}0111 \rightarrow B_{ок} &= \underline{1{,}1000} \\
(A + B)_{ок} &= 1{,}1100 \rightarrow (A + B)_{пк} = 1{,}0011
\end{aligned}
$$

$$
\begin{aligned}
A = -0{,}0100 \rightarrow A_{пк} = 1{,}0100 \rightarrow A_{ок} &= 1{,}1011 \quad + \\
B = +0{,}0111 \rightarrow B_{пк} = 0{,}0111 \rightarrow B_{ок} &= \underline{0{,}0111} \\
1| &\; 0{,}0010 \\
&\longrightarrow 1 + \text{ цикличен пренос} \\
(A + B)_{ок} &= 0{,}0011 \rightarrow (A + B)_{пк} = 1'1110
\end{aligned}
$$

ПК

1.1.2.3. Допълнителен код

Допълнителният код на положителните числа (както цели, така и дробни) съвпада изцяло с техния прав код.

Допълнителният код на едно отрицателно двоично число се получава от неговия прав код, като в знаковия разряд се запише 1 (знак "–"), двоичните цифри в числовите разряди се инвертират и след това в най-младшия числов разред се прибави аритметично 1.

$$
N_{дк} = \begin{cases} 0, N \text{ при } N \geq 0 \\ 1, \underline{a_{n-1}}\ \underline{a_{n-2}} \dots \underline{a_1}, \underline{\underline{a_0}} \text{ при } N < 0 \end{cases} \quad \text{за дробни числа}
$$

$$
N_{дк} = \begin{cases} 0' N \text{ при } N \geq 0 \\ 1' \underline{a_{n-1}}\ \underline{a_{n-2}} \dots \underline{a_1}, \underline{\underline{a_0}} \text{ при } N < 0 \end{cases} \quad \text{за цели числа,}
$$

където $\underline{a_i}$ е допълнение на a_i до $p - 1$, а $\underline{a_i}$ е допълнение на a_i до p (p – основа на бройната система).

Допълнителният код на отрицателните двоични числа може да се дефинира формално и чрез зависимостите:

$$\left.\begin{array}{ll} \text{– цяло число} & N_{\text{дк}} = 2^{n+1} + N \\ \text{– дробно число} & N_{\text{дк}} = 2^{1} + N \end{array}\right| N < 0$$

Примери:

$$\begin{array}{llll}
A = +110_{(2)} & \rightarrow A_{\text{пк}} = 0'0110 & \rightarrow A_{\text{дк}} & = 0'0110 \\
A = +0{,}01_{(2)} & \rightarrow A_{\text{пк}} = 0{,}0100 & \rightarrow A_{\text{дк}} & = 0{,}0100 \\
A = -110_{(2)} & \rightarrow A_{\text{пк}} = 1'0110 & \rightarrow A_{\text{дк}} & = \begin{array}{r} 1'1001 \\ \underline{1} \\ 1'1010 \end{array} + \\
A = -0{,}01_{(2)} & \rightarrow A_{\text{пк}} = 1{,}0100 & \rightarrow A_{\text{дк}} & = \begin{array}{r} 1{,}1011 \\ \underline{1} \\ 1'1100 \end{array} +
\end{array}$$

Правилото за преобразуване на отрицателни двоични числа в допълнителен код може да се формулира и по следния начин: допълнителният код се получава, като към обратния код на същото число се прибави 1 в младшия разряд. При преобразуване от допълнителен в прав код на отрицателни числа се изпълняват същите операции и в същия ред, както при преобразуване от прав в допълнителен код – числовите разряди в записа на допълнителния код се инвертират, след което към младшия разряд се прибавя 1.

Допълнителният код има следните особености:

– ако при събиране на две числа се получи пренос 1 след знаковия разряд на резултата (от знаковия разред наляво), той се елиминира (отпада, не се отчита);

– ако резултатът от събирането е положително число, той е окончателен т.к. за положителните числа допълнителният код съвпада с правия и не е необходимо някакво преобразуване;

– ако резултатът от събирането е отрицателно число, той се получава в допълнителен код и ако е необходимо да се преобразува в прав код, се прилага описаното по-горе правило;

– при събиране на еднакви по абсолютна стойност числа с различни знаци се получава винаги „положителна“ нула;

– число със стойност 0 и отрицателен знак не може да се представи в допълнителен код. Действително, $+0 \rightarrow [+0]_{\text{пк}} = 0'00...0 = [+0]_{\text{дк}}$

$$-0 \rightarrow [-0]_{пк} = 1'00...0 \rightarrow [-0]_{дк} = 1'11...1$$
$$\underline{\qquad\qquad +1}$$
$$отпада \rightarrow \underline{1}|\ 0'00...0 = [+0]_{дк}$$

Записът 1'00...0, респективно 1,00...0, който в прав код означава –0, а в обратен код представя числата $-(2^n - 1)$, респ. $-(1 - 2^{-n})$, в допълнителен код представя отрицателното число -2^n, респ. –1. Това е възможно, защото цифрата 1 в знаковия разред изпълнява едновременно две функции – показва знак минус и представлява значеща цифра в числов разряд с тегло 2^n, респ. $2^0 = 1$. Вследствие на тази особеност диапазоните на представимите отрицателни числа в допълнителен код са от –1|1'11…11| до $-(2^n)$|1'00…00| за цели числа и от $-(2^{-n})$|1,11…11| до –1|1,00…00| за дробни числа, т.е. те са различни от диапазоните на отрицателните числа в прав и обратен код. Диапазоните на представимите положителни числа в допълнителен код съвпадат с тези на числата в прав и обратен код.

Свойствата на допълнителния код се илюстрират със следните примери:

а) Събиране на целите числа $A = 0101_{(2)}$ и $B = 1001_{(2)}$.

$$A = +0101 \rightarrow A_{пк} = 0'0101 \rightarrow A_{дк} = 0'0101 \;+$$
$$B = +1001 \rightarrow B_{пк} = 0'1001 \rightarrow B_{дк} = \underline{0'1001}$$
$$(A + B)_{дк} = 0'1110 = (A + B)_{пк}$$

$$A = -0101 \rightarrow A_{пк} = 1'0101 \rightarrow A_{дк} = 1'1010 \;+ \qquad A_{дк} = 1'1011 \;+$$
$$\underline{\qquad 1} \qquad\qquad B_{дк} = \underline{1'0111}$$
$$1'1011 \qquad отпада \rightarrow \underline{1}|\ 1'0010$$

$$B = -1001 \rightarrow B_{пк} = 1'1001 \rightarrow B_{дк} = 1'0110 \;+$$
$$\underline{\qquad 1}$$
$$1'0111$$

$$(A + B)_{дк} = 1'0010 \rightarrow (A + B)_{пк} = 1'1101 \;+$$
$$\underline{\qquad 1}$$
$$1'1110$$

б) Събиране на дробните числа $A = 0{,}0100_{(2)}$ и $B = 0{,}0111_{(2)}$

$$A = +0{,}0100 \rightarrow A_{пк} = 0{,}0100 \rightarrow A_{дк} = 0{,}0100$$

$$B = -0{,}0111 \rightarrow B_{пк} = 1{,}0111 \rightarrow B_{дк} = \begin{array}{r} 1{,}1000 \\ \underline{1} \\ 1{,}1001 \end{array} +$$

$$(A+B)_{дк} = \begin{array}{r} 0{,}0100 \\ \underline{1{,}1001} \\ 1{,}1101 \end{array} + \qquad (A+B)_{пк} = \begin{array}{r} 1{,}0010 \\ \underline{1} \\ 1{,}0011 \end{array} +$$

$$A = -0{,}0100 \rightarrow A_{пк} = 1{,}0100 \quad \rightarrow \quad A_{дк} = \begin{array}{r} 1{,}1011 \\ \underline{1} \\ 1{,}1100 \end{array} +$$

$$B = +0{,}0111 \rightarrow B_{пк} = 0{,}0111 \quad \rightarrow \quad B_{дк} = 0{,}0111$$

$$(A+B)_{дк} = \begin{array}{r} 1{,}1100 \\ \underline{0{,}0111} \end{array}$$

$$\text{отпада} \rightarrow \underline{1}|0{,}0011 = (A+B)_{пк}$$

Предимството на обратния код пред допълнителния е по-лесното преобразуване от и в прав код (числовите разряди само се инвертират, без да се прибавя 1). Предимството на допълнителния код пред обратния е по-бързото извършване на сумирането т.к. при него няма цикличен пренос. В реалните компютърни системи се предпочита за използване допълнителният код т.к. в много случаи над дадени числа се извършват многократно аритметични действия, в резултат на което бързодействието ще бъде по-голямо. За да се намали влиянието на недостатъка на допълнителния код, числата се преобразуват от прав код в допълнителен и обратно само два пъти – при въвеждане в компютъра на входните данни и при извеждане на резултатите. По време на обработката числата се съхраняват в паметта на компютъра в допълнителен код.

1.1.3. Препълване на разрядната решетка

При събиране на двоични числа с фиксирана запетая може да се получи резултат S, който по абсолютна стойност е по-голям от най-голямото представимо число в разрядната решетка, т.е. при цели числа $|S| \geq 2^n$, а при дробни $|S| \geq 1$. Това явление се нарича препълване на разрядната решетка при фиксирана запетая. Правилното представяне и обработване на полученото при препълване число е невъзможно и затова в такъв случай изпълнението на компютърната програма се прекратява, т.е. изпълнява се програмно прекъсване по препълване на разрядната решетка. Препълване при фиксирана запетая може да се получи само при събиране на числа с еднакви знаци.

В случай на препълване на разрядната решетка при събиране на две числа с фиксирана запетая, към сумата от знаковите разряди се прибавя и „паразитен" пренос 1 от числовите разряди, т.к. сумата на последните надхвърля определените за тях n разряда. Това води до получаване на недействителен резултат с грешен знак – отрицателен при събиране на положителни числа и положителен при събиране на отрицателни числа. Не е вярна и числовата част на резултата, защото тя е загубила преноса с тегло 2^n (за цели числа) или $2^0 = 1$ (за дробни числа), който е попаднал в знаковия разряд. Същевременно полученият фиктивен резултат, разглеждан самостоятелно, с нищо не показва, че има препълване.

Дефиниционните условия за препълване при събиране на отрицателни операнди в обратен и в допълнителен код са различни. Граничните числа -2^n и респ. -1 не допускат представяне в обратен код и затова при получаването им се установява препълване. Същите числа в допълнителен код се представят и обработват коректно и при тях не се проявява препълване.

$$
\begin{array}{l}
A = -0110_{(2)} \rightarrow A_{\text{пк}} = 1'0110 \rightarrow A_{\text{ок}} = 1'1001 \\
B = -1010_{(2)} \rightarrow B_{\text{пк}} = 1'1010 \rightarrow B_{\text{ок}} = \underline{1'0101} \\
\hfill \underline{1}|\ 0'1110 \quad + \\
\hfill \underline{\quad\quad\rightarrow 1} \\
\hfill (A + B)_{\text{ок}} \neq 0'1111
\end{array}
$$

$$
\begin{array}{l}
A = -0110 \rightarrow A_{\text{пк}} = 1'0110 \rightarrow A_{\text{дк}} = 1'1010 \\
B = -1010 \rightarrow B_{\text{пк}} = 1'1010 \rightarrow B_{\text{дк}} = \underline{1'0110} \quad + \\
\hfill \text{отпада } \underline{1}|\ 1'0000 \\
\hfill (A + B)_{\text{дк}} = 1'0000
\end{array}
$$

Получаването на препълване при събиране числа с фиксирана запетая може да се установи по един от следните начини:

– чрез сравняване на знаковите разряди на двете числа, участващи в събирането, със знаковия разряд на резултата. Препълване има само ако знаковите разряди на числата са еднакви, а знаковият разряд на резултата е противоположен;

– чрез сравняване на преноса към знаковия разряд и преноса след знаковия разряд в процеса на формиране на резултата. Ако двата преноса имат еднакви значения 0 или 1, препълване няма. Препълване има само ако единият от двата преноса е 1 (без значение кой), а другият е 0, т.е. двата преноса имат различни стойности;

– чрез използване на така наречените модифицирани машинни кодове – прав, обратен и допълнителен. При тях знакът на числата се представя не с един, а с два разряда – "+" → 00 и "–"→ 11. И двата знакови разряда се приемат условно за числови и участват в операцията двоично сумиране. В случай на препълване двата знакови разряда на резултата се получават различни – 01 при събиране на положителни числа и 10 при събиране на отрицателни числа. При това старшият (левият) знаков разряд запазва верния знак (0 или 1) на действителния резултат. Признак за препълване се формира чрез логическата функция „сума по модул 2“ над знаковите разряди на резултата. Всички останали правила и особености на обикновените кодове (с един знаков разред) са валидни и за съответните модифицирани кодове. При събиране в модифициран обратен код цикличен пренос се взема след старшия знаков разряд на резултата.

***Примери*:**

$$A = -1001_{(2)} \to A_{мпк} = 11'1001 \to A_{мок} = 11'0110$$
$$B = -1110_{(2)} \to B_{мпк} = 11'1110 \to B_{мок} = \underline{11'0001}$$
$$1|\;10'0111 \; +$$
$$\to 1$$
$$(A + B)_{мок} \neq \underbrace{10}_{\text{препълване}}'1000$$

$$A = -0{,}0100_{(2)} \to A_{мпк} = 11{,}0100 \to A_{мдк} = 11{,}1100$$
$$B = +0{,}0111_{(2)} \to B_{мпк} = 00{,}0111 \to B_{мдк} = \underline{00{,}0111} \; +$$
$$\text{отпада } \underline{1}|\,00{,}0011$$

$$(A + B)_{мдк} = 00{,}0011 = (A + B)_{мпк}$$

1.2. Задание за лабораторна работа

1. Десетичните числа 5; 7; 11; 14; 0,25; 0,375; 0,625 и 0,8125 да се преобразуват в двоична бройна система и да се представят в разрядната решетка на макета с отрицателен знак в прав код. Да се проследи и обясни преобразуването им в ОК, в ДК и обратно.

2. Двойката числа 5 и 7 да се съберат в двоична бройна система при четирите възможни комбинации от знаци на числата в обратен код и в допълнителен код. Получените резултати да се преобразуват в прав код. Да се провери тяхната истинност.

3. Заданието от точка 2 да се изпълни с числата 0,25 и 0,375.

4. Двойките числа 11 и 14; 11 и 5; 0,625 и 0,8125; 0,375 и 0,625 да се съберат като положителни и като отрицателни в обратен код и в допълнителен код.

5. Двойките числа от точка 4 да се съберат като положителни и като отрицателни в модифициран обратен код и в модифициран допълнителен код. Да се сравнят и обяснят получените резултати.