Организация на компютъра

Лекция 6: Операционни структури за двоично-десетични и специални операции

Съдържание

- Двоично-десетична аритметика
- Машинни кодове на двоично-десетични числа
- Логическа структура на устройства за работа с двоично-десетични числа
- Операции за преобразуване на бройната система
- Операции за преобразуване на фо̀рмата на числата

Операционни структури за 2-10 аритметика

Двоично-десетична аритметика

За целите на обработката 2/10-те числа най-често се представят в прав код, в пакетиран формат.

$$89_{(10)} \rightarrow 1000\ 1001_{(2\text{-}10)}$$

Изпълнението на операциите изваждане и деление изискват прилагане на обратен и допълнителен код.

Операционни структури за 2-10 аритметика

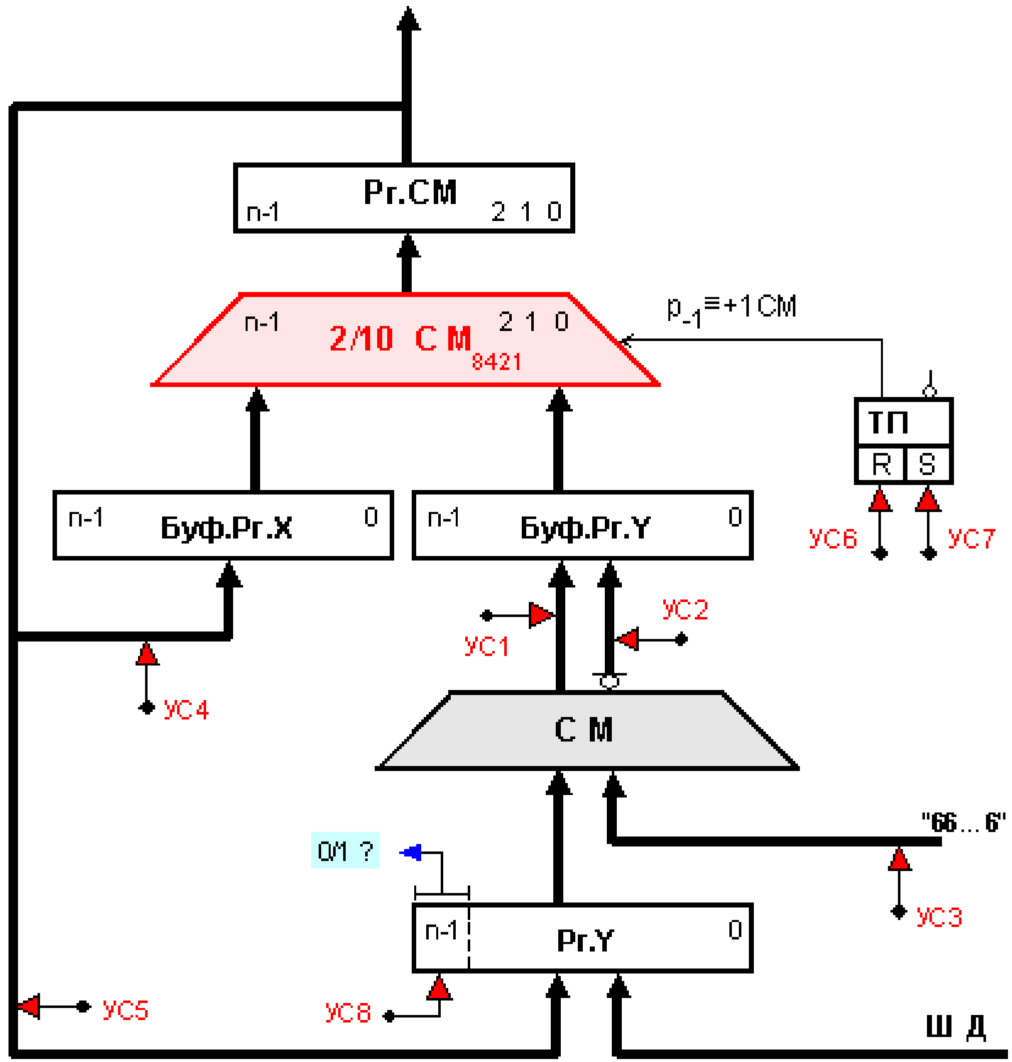
Операции **събиране и изваждане**

- ***Устройство за събиране и изваждане на десетични числа представени в код “8421”***

Общ алгоритъм за събиране (изваждане) на 2-10 числа със знак (Z=X+Y), представени в прав код:

1. Представяне на операндите в допълнителен код.
2. Изпълнение на операцията и формиране на признаците на резултата.
3. Проверка за препълване.
4. Ако резултатът е верен, той се преобразува и представя в прав код.

Операционни структури за 2-10 аритметика



УС1 : Буф PrY := CM,
УС2 : Буф PrY := not(CM),
УС3 : CM + "66...6",
УС4 : Буф PrX := (PrCM),

УС5 : PrY := (PrCM),
УС6 : ТП := 0,
УС7 : ТП := 1, (+1CM),
УС8 : PrY[n-1] := not(PrY[n-1]).

Операционни структури за 2-10 аритметика

Фактическото събиране на две 2/10-чни числа в двоичното АЛУ на конвенционален процесор се постига чрез **машинна програма** от две машинни команди – една за двоично събиране и втора за 2/10-чна корекция.

За осъществяване на 2/10-чната корекция се използуват автоматично формирани и запаметени при първото двоично събиране признаци или **поразрядни (тетрадни) корекции.**

Тази втора по ред операция се осигурява със специална машинна команда и се нарича ***“двоично-десетична корекция”***.

Така при първото двоично събиране се формират два резултата – междинна двоична сума и поразрядна 2/10-чна корекция. Второто двоично събиране в същия суматор се извършва между тези две числа.

Операционни структури за 2-10 аритметика

- Поразрядната корекция зависи от 2/10-чния код.
  - ✓ Ако този код е “8421”, тетрадната корекция е числото +6. То се добавя към онези разряди, които след първото двоично събиране са генерирали тетраден пренос или като тетрадна сума представляват “забранена” комбинация.
  - ✓ В случай че кодът е “8421+3”, тетрадната корекция е **задължителна** за всеки отделен разряд и тя зависи от това, дали съответният разряд е генерирал тетраден пренос. Ако е генерирал, корекцията е числото +3, ако не – числото +13.
- В процесори, обработващи двуцифрени 2/10-чни числа, двата тетрадни преноса обикновено се фиксират в регистъра на признаците, съответно в бит C от старшата тетрада, а в бит H – от младшата

Логически операции

- Най-често реализираните логически операции са:
  - ✓ "*Логическо събиране*", (дизюнкция, ИЛИ, *OR*);
  - ✓ "*Логическо умножение*", (конюнкция, И, *AND*);
  - ✓ "*Логическа инверсия*", (НЕ, *NOT*);
  - ✓ "*Логическа неравнозначност*", (Сума по mod2, Изключващо ИЛИ, *XOR*).
- Логическите операции се реализират само върху основния формат на разрядната мрежа.
- Тъй като те са определени само върху логическите константи, в многоразрядните процесори изпълнението им е поразрядно. В този смисъл, съдържанието на участвуващите в операцията регистри се интерпретира като двоична комбинация, непритежаваща количествено значение.

Логически операции

Операции "*Сравнение*".

Сравнението е действие, с помощта на което алгоритмите "се ориентират" в хода на обработката на данните.

На сравнение се подлагат както числови, така и нечислови данни.

Операцията "*Сравнение*" има за цел да формира признаците на резултата, без да го фиксира и без да променя операндите.

Сравнението е акт на *проверка на отношение* на първия операнд спрямо втория:

$$X > Y;\ X >+ Y;\ X < Y;\ X <= Y;\ X = Y$$

Сравнението се извършва по отношение на нулата

$$(X-Y) \leftrightarrow 0$$

Логически операции

Операции "*Сравнение*".

При операция сравнение се извършва изваждане на втория операнд от първия, без да се фиксира разликата.

Фиксират се само нейните признаци C, Z, S, P и др. в регистъра на признаците на резултата.

Логически операции

Операции от тип "*Изместване*".

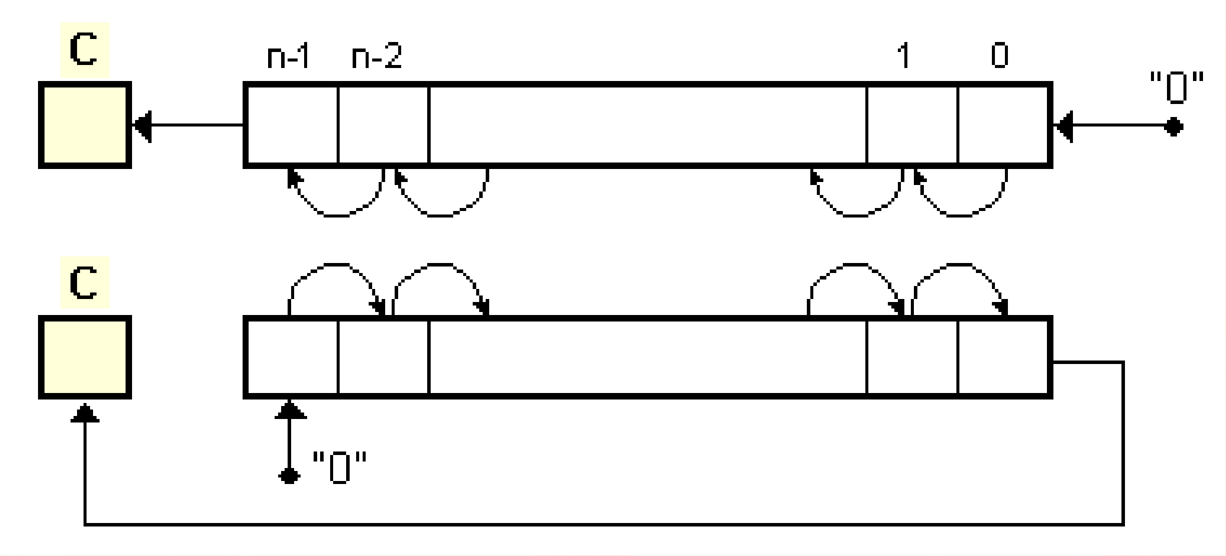
- ✓ "*Логическо изместване*" *;*
- ✓ "*Аритметическо изместване*" *;*
- ✓ "*Циклическо изместване*" *(Ротация)* .

- Според посоката, изместванията са наляво или надясно.
- Операциите от тип изместване са *едноместни*, т. е. операндът е един.

Логически операции

Операции от тип "*Изместване*".

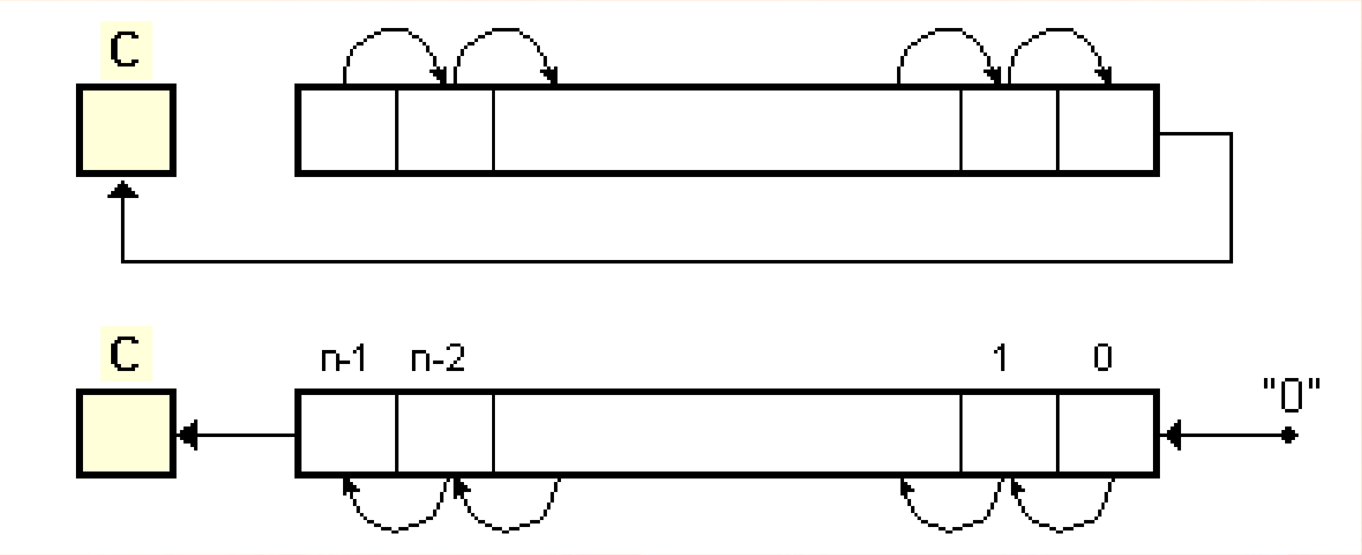
✓ "*Логическо изместване*"



Логически операции

Операции от тип "*Изместване*".

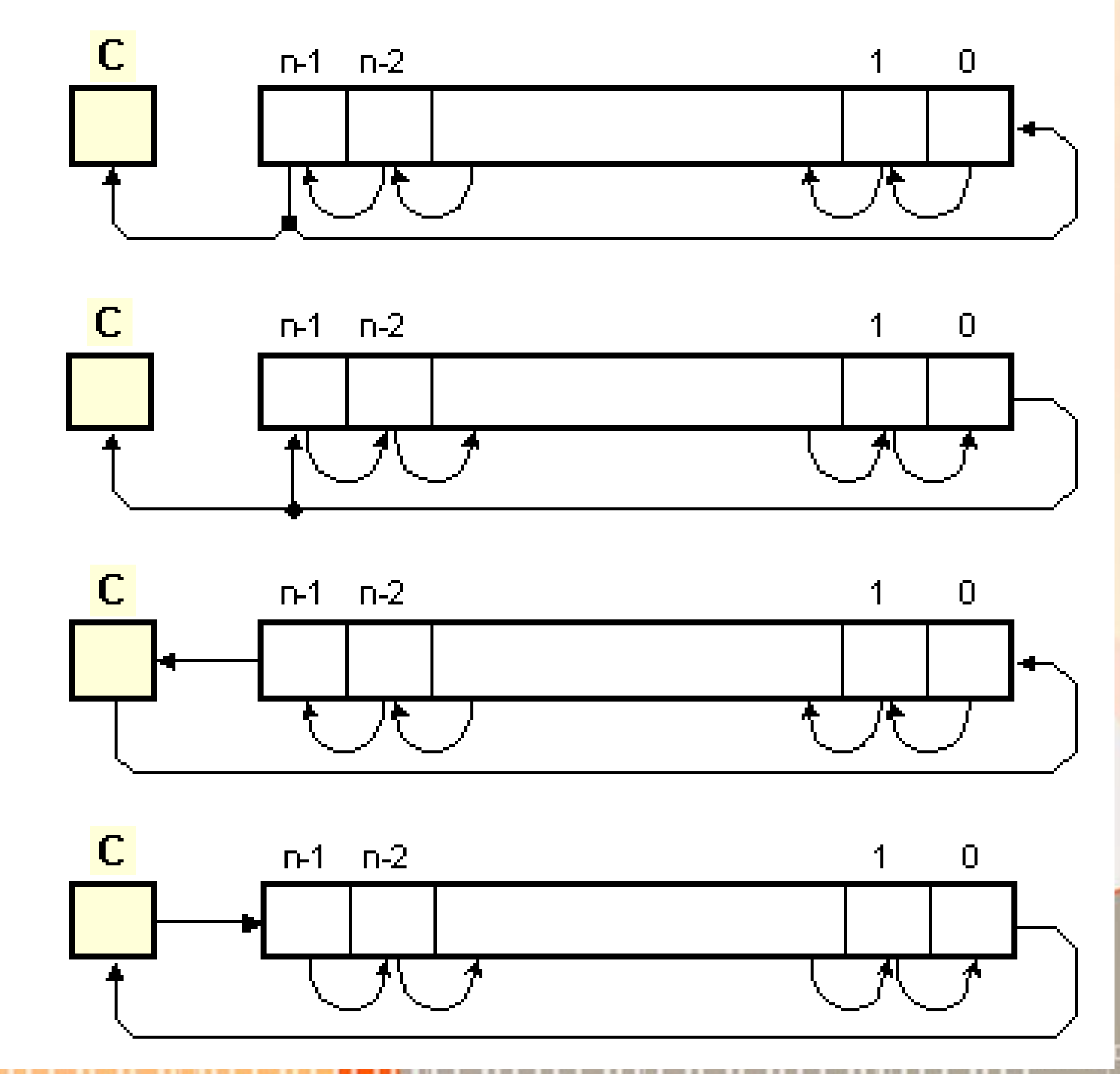
- ✓ *Аритметическо изместване*



Логически операции

Операции от тип "*Изместване*".

- ✓ *Циклическо изместване (ротация)*



Операции за преобразуване на бройните системи

Преобразуване на числа с ДФЗ от десетична в двоична бройна система

$$X = d_4 d_3 d_2 d_1 d_0$$

$$X_{(10)} = d_4.10^4 + d_3.10^3 + d_2.10^2 + d_1.10^1 + d_0.10^0$$

По схемата на Хорнер:

$$X_{(10)} = ((((0 + d_4).10 + d_3).10 + d_2).10 + d_1).10 + d_0$$

Операции за преобразуване на бройните системи

Преобразуване на числа с ДФЗ от десетична в двоична бройна система

Алгоритъмът за това преобразуване е циклически - повторенията ще бъдат толкова, колкото са цифрите на изходното десетично число.

$$S_0 = 0\ ;$$

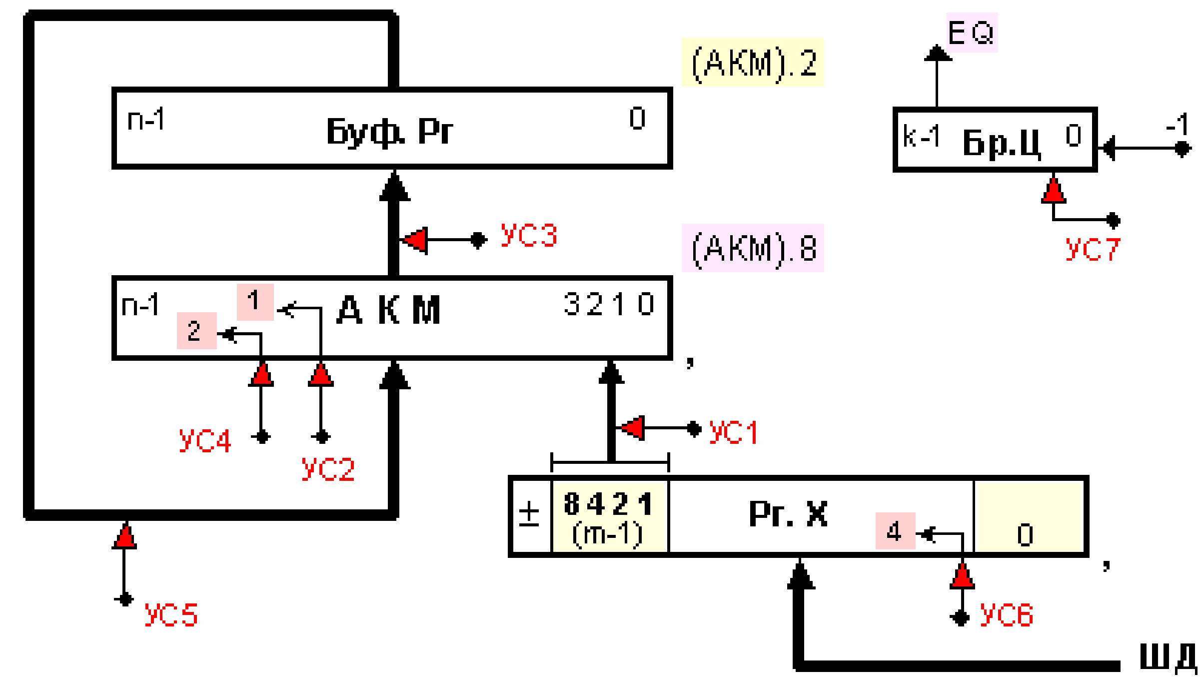
$$S_{i+1} = \left(S_i + d_{(m-1)-i}\right).10\ ,\quad i = \overline{0\,,m-2}\ ;$$

$$S_m = S_{m-1} + d_0\ ;$$

В променливата S се натрупва стойността на търсения еквивалент като двоична сума.

Операции за преобразуване на бройните системи

Преобразуване на числа с ДФЗ от десетична в двоична бройна система



УС1 : АКМ := (АКМ)+(РгХ[m]) ,
УС2 : АКМ := АИЛ1(АКМ)=(АКМ).2 ,
УС3 : Буф Рг := (АКМ) ,
УС4 : АКМ := АИЛ2(АКМ)=(АКМ).4 ,
УС5 : АКМ := (АКМ)+(Буф Рг) ,
УС6 : РгХ := ЛИЛ4(РгХ)=(РгХ).10 ,
УС7 : БрЦ := m-1 .

Операции за преобразуване на бройните системи

Преобразуване на числа с ДФЗ от десетична в двоична бройна система

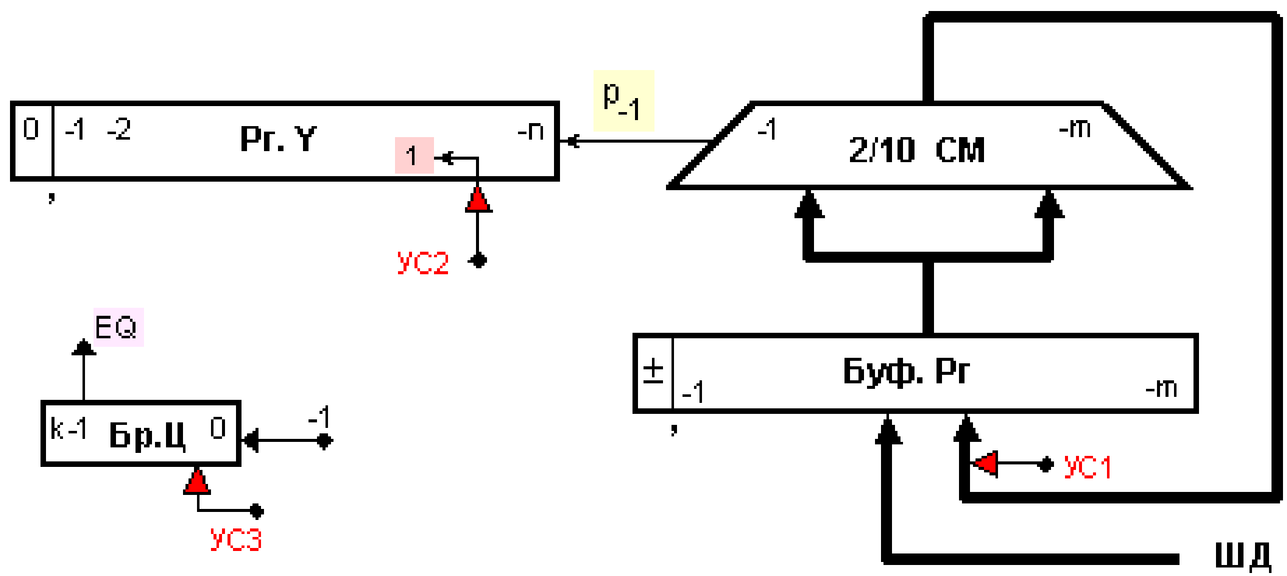
- За да може да бъде получен двоичният еквивалент на всяко m-разрядно десетично число, следва дължината на двоичните възли (акумулатор и буферен регистър) да бъде достатъчна, в противен случай ще се получи препълване в двоичното поле.
- Двоичната дължина от n бита ще бъде:

$$n \geq \log_2(10^m) = m.\log_2 10$$ т. е. $n \geq m.(3,3222591)$

- Дължината на двоичното поле от n бита, е 3,3222591 пъти по-голяма от дължината m на десетичното поле, т. е. *един десетичен разряд е еквивалентен на 3,3222591 двоични разряда.*
- 19 десетични разряда са еквивалентни на 64 двоични, тъй като 19.(3,3222591) ≈ 64 [b]

Операции за преобразуване на бройните системи

- ***Преобразуване на числа с ЛФЗ от десетична в двоична бройна система***



УС1 : БуфРг := СМ , УС2 : РгY := ЛИЛ1(РгY) , УС3 : БрЦ := n .

Преобразуване на фо̀рмата на числата

Разрядна мрежа с дължина **n**[b]

Порядъкът се определя по формулата

$$\mathbf{p} = (n-1) - L\ ,$$

L - броят на изместванията на числото в разрядната мрежа при неговата лява нормализация.

Числото L е число със знак, тъй като изместванията са възможни в посока както наляво, така и надясно. Случаят с изместване надясно се отнася единствено за най-малкото число, представимо в допълнителен код в разрядната мрежа с фиксирана запетая.

Преобразуване на фо̀рмата на числата

Известно е, че:

- Всяко число с фиксирана запетая е *представимо* във форма с плаваща запетая.
- Всяко число с фиксирана запетая, чийто модул има дължина по-малка от дължината на полето, определено за мантисата, се *представя точно*.
- Всяко число с фиксирана запетая, чийто модул има дължина по-голяма от дължината на полето, определено за мантисата, се *представя неточно*.

Преобразуване на фо̀рмата на числата

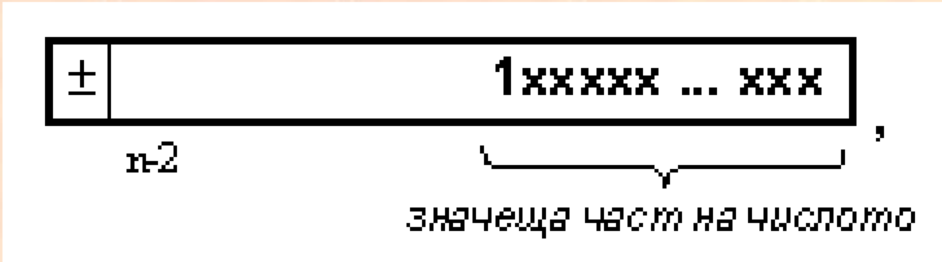
Алгоритъмът на това преобразуване изисква изходното число да бъде представено в **прав код.**

Значещата част на числото, намираща се в разрядната мрежа, следва да се превърне в **нормализирана мантиса.** За целта цялото число трябва да се нормализира спрямо левия край на разрядната мрежа.

В процеса на нормализация следва да се определи и порядъкът на числото.

Преобразуване на фо̀рмата на числата

Нека си представим числото X в разрядната мрежа по следния начин:



- Изместването на числото X наляво го превръща в полином от (n-2)-ри ред. Ако се приеме, че редът на полинома на изходното число е k, където k<(n-2), то нормализираният вид на числото се получава след [(n-2)-k] на брой едноразрядни измествания наляво, т.е.

$$P_x^{(n-2)} = X.2^{(n-2)-k}$$

Преобразуване на фо̀рмата на числата

- Тъй като редът k на изходното число е неизвестен, необходимо е в процеса на нормализиране да бъдат *преброени* изместванията наляво.

Ако броят на изместванията се означи с h, то от уравнението $(n-2) - k = h$

се определя $k = (n-2) - h$

Тъй като порядъкът на цифрата 1 е +1, то окончателният израз за порядъка на числото ще бъде

$$p_X = (n-2) - h + 1$$

В крайна сметка числото X ще има вида: $X = M_X . 2^{p_X}$,

където $|M_X| = 0,1xxx \ldots xx$.

Преобразуване на фо̀рмата на числата

Преобразуване на двоични числа с плаваща запетая в двоични с фиксирана запетая

Трябва да се реши уравнението

$$(n - 2) - h + 1 = p_x ,$$

по отношение на h.

$$h = (n - 2) - p_x + 1$$

Неизвестното h изразява броя на изместванията на мантисата спрямо десния край на разрядната мрежа.

Изместванията трябва да компенсират и k-битовата дължината на полето за порядък.

Тогава

$$h = (n - 2) - p_x + 1 + k$$

Преобразуване на фо̀рмата на числата

Преобразуване на двоични числа с плаваща запетая в двоични с фиксирана запетая

Стойността h може да се изчисли, като от константата [(n-2)+1] (или респективно от константата [(n-2)+1+k]) се извади стойността на порядъка P^x на изходното число.

Ако получената разлика е положително число (h>0), посоката за изместване на мантисата е надясно, ако обаче е отрицателно (h<0) - наляво.

Броят на изместванията е |h| , стойност която се явява начална за изваждащ брояч, организиращ циклическия алгоритъм.

Препълването, което е възможно да настъпи в разрядната мрежа с ФЗ, се открива чрез отношението

h > (n-1).

Преобразуване на фо̀рмата на числата

Преобразуване на двоични числа с плаваща запетая в двоични с фиксирана запетая

Не всяко число, представено във форма с плаваща запетая, може да се преобразува в число, представено във форма с фиксирана запетая.

Представимите с фиксирана запетая числа след преобразуване са значително по-малко.

При загуба на дробната част цялата част, която остава, е точна.

Литература

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

[2]. http://tyanev.com/ - On-line книги – ОРГАНИЗАЦИЯ НА КОМПЮТЪРА – упражнения книга [2];

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

[4]. *Димитър Тянев, ОРГАНИЗАЦИЯ НА КОМПЮТЪРА - упражнения,* ISBN 978-954-20-0258-0, Варна 2007г.