Организация на компютъра

Лекция 3: Операционни структури за работа с фиксирана запетая

Операции умножение и деление. Унитарни операции

Съдържание

Операция умножение

Базова логическа структура на устройство за умножение, основана на комбинационен суматор

Базова логическа структура на устройство за умножение, основана на натрупващ суматор

Операция деление

Унитарни операции

Операционни структури

Операции върху числа с фиксирана запетая

Операция умножение

Многократно (К-кратно, където К е другият множител) натрупваща се сума на единия от множителите.

Циклически алгоритъм с предварително известен брой повторения.

Недостатък – различно времетраене (зависи от броя събирания, т. е. от модула на единия от множителите).

Операционни структури

Операции върху числа с фиксирана запетая

Операция умножение

$$Z = X.Y = X.\left(y_{n-2}.2^{n-2} + y_{n-3}.2^{n-3} + \ldots + y_1.2^1 + y_0.2^0\right) =$$

$$= X.y_{n-2}.2^{n-2} + X.y_{n-3}.2^{n-3} + \ldots + X.y_1.2^1 + X.y_0.2^0$$

y_i са цифрите на модула на множителя, записан в n-разрядна двоична мрежа като число със знак в прав код.

$X.y_i$ - *поразрядни произведения*

В двоична бройна система X.1=X и X.0=0!

Произведението при този вид на множителя може да се получи чрез *логическо изместване* на множимото наляво на i разряда.

Операционни структури

Операции върху числа с фиксирана запетая

Операция умножение

Умножение с младшите разряди напред:

$$Z = \sum_{i=0}^{n-1} \left(X . y_i . 2^i \right)$$

Умножение със старшите разряди напред:

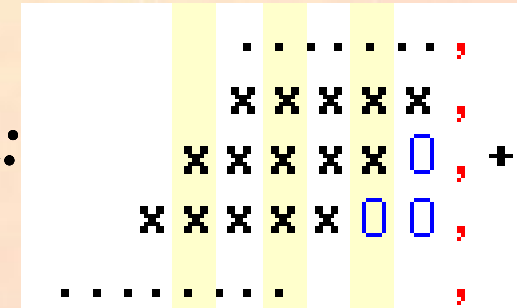
$$Z = \sum_{i=n-1}^{0} \left(X . y_i . 2^i \right)$$

Операционни структури

Операции върху числа с фиксирана запетая

Операция умножение

Умножение ***с младшите разряди напред:***



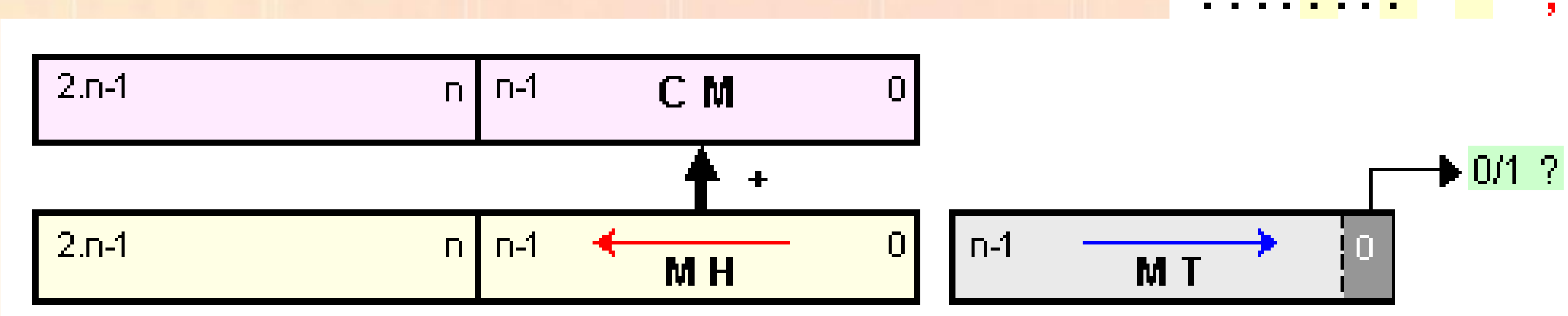


Схема за умножение **с неподвижна междинна сума**

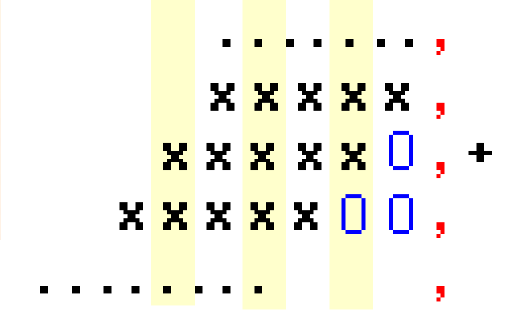
- Суматор с двойна дължина;
- Изместващ регистър на множимото с двойна дължина;
- Изместващ регистър на множителя с единична дължина.

Операционни структури

Операции върху числа с фиксирана запетая

Операция умножение

Умножение ***с младшите разряди напред:***



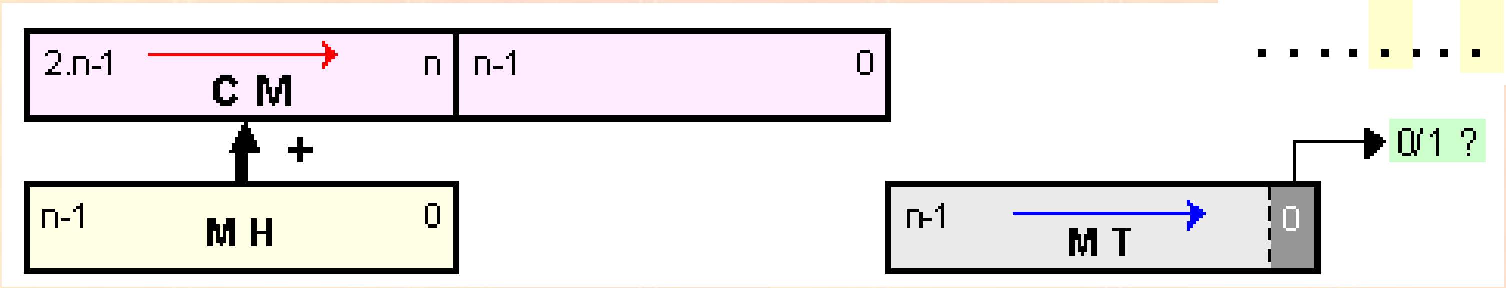
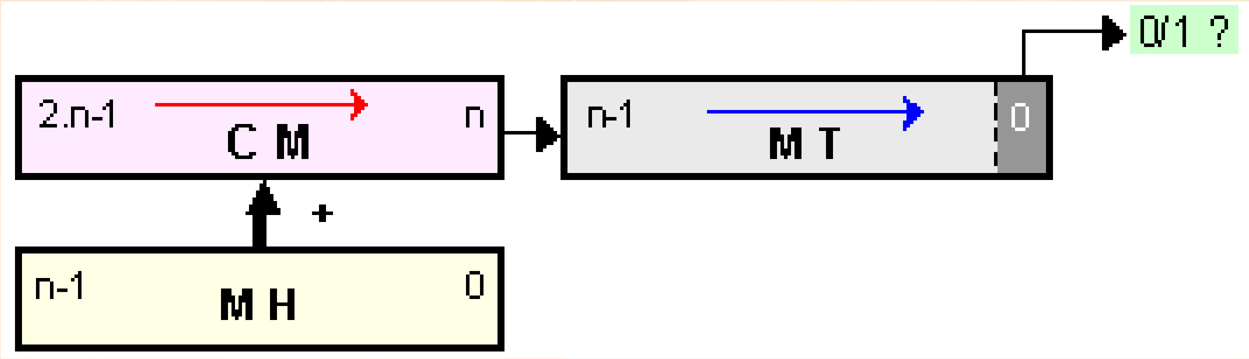


Схема за умножение **с неподвижно множимо**



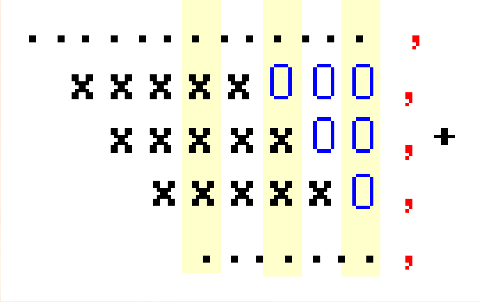
Оптимизирана схема за умножение **с неподвижно множимо**

Операционни структури

Операции върху числа с фиксирана запетая

Операция умножение

Умножение ***със старшите разряди напред:***



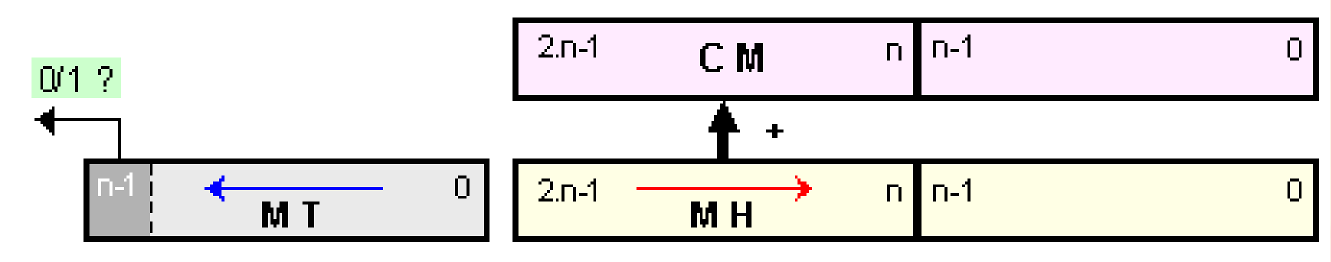


Схема за умножение **с неподвижна междинна сума**

Операционни структури

Операции върху числа с фиксирана запетая

Операция умножение

Умножение ***със старшите разряди напред:***

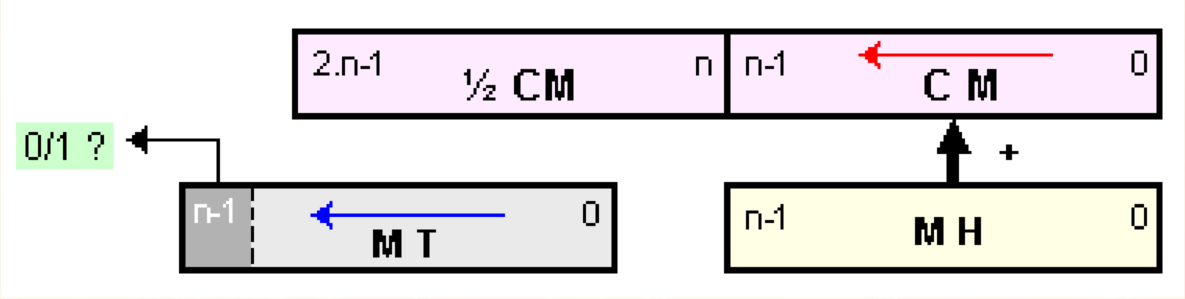


Схема за умножение **с неподвижно множимо**

Тази схема предлага удобното **ѝ** обединяване в обща логическа структура с операция *деление*.

От всички организационни схеми на алгоритми за умножение, разгледани до момента, най-икономична се оказва третата – ***оптимизирана схема за умножение с неподвижно множимо,*** ето защо тя намира най-широко практическо приложение.

Операционни структури

Операции върху числа с фиксирана запетая

Операция умножение

Произведението е число, което има **двойна** дължина и когато операция умножение се реализира в съответствие с избраната организационна схема, **препълване не е** възможно да настъпи.

Това означава, че произведението е **представимо** число и препълването е **принципно** невъзможно.

Операционни структури

Операции върху числа с фиксирана запетая

Операция умножение

*Режим на **единична** дължина на произведението*

Операндите X и Y, както и резултатът Z, се възприемат като числа, побиращи се в n-битови регистри.

*Режим на **двойна** дължина на произведението*

Операционни структури

Операции върху числа с фиксирана запетая

Операция умножение

В режим на единична дължина, при умножение на числа с дясно фиксирана запетая е възможно да настъпи **препълване**, ако старшата половина на произведението е различна от нула.

При умножение на числа с ляво фиксирана запетая положението е по-благоприятно, тъй като възможната за изхвърляне половина от произведението е младшата, т.е. във всички случаи крайният резултат, възприеман като произведение, следва да включва старшата половина от резултата.

Операционни структури

Операции върху числа с фиксирана запетая

Операция умножение

Произведение с единична дължина може да се формира чрез закръгляне до старшите n разряда, при запазване на задоволителна точност, което не е възможно при числа с дясно фиксирана запетая.

В режим на двойна дължина както при дясно, така и при ляво фиксирана запетая произведението е представимо, но то следва да се декларира като число, представено в друг формат, различен от този на множителите.

Операционни структури

Операции върху числа с фиксирана запетая

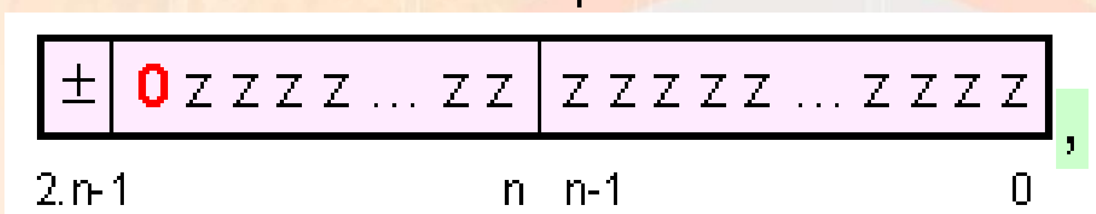
Операция умножение

Дължина на разрядната мрежа - n[b].

Двойна дължина - 2.n[b].

Модулът на произведението има дължина равна на сумата от дължините на цифровите части на съмножителите, т.е. (n-1)+(n-1)=2.(n-1)[b].

Когато този модул се разположи в двойния формат, остават свободни два разряда, единият от които е предназначен за знака на произведението.

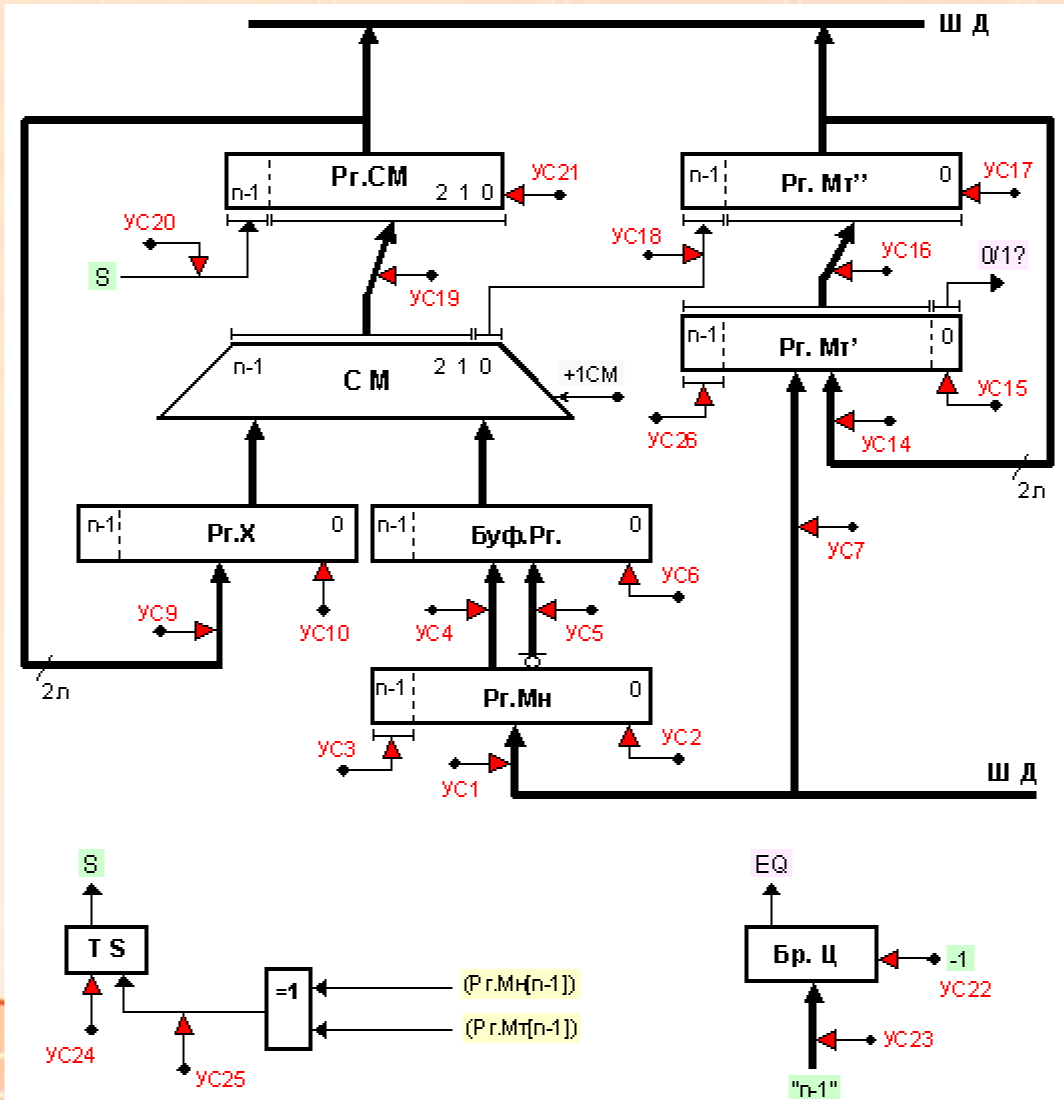


Знак Z = Знак X ⊕ Знак Y

Операционни структури

Операции върху числа с фиксирана запетая. Операция умножение

Базова логическа структура на устройство за умножение, основана на *комбинационен* суматор

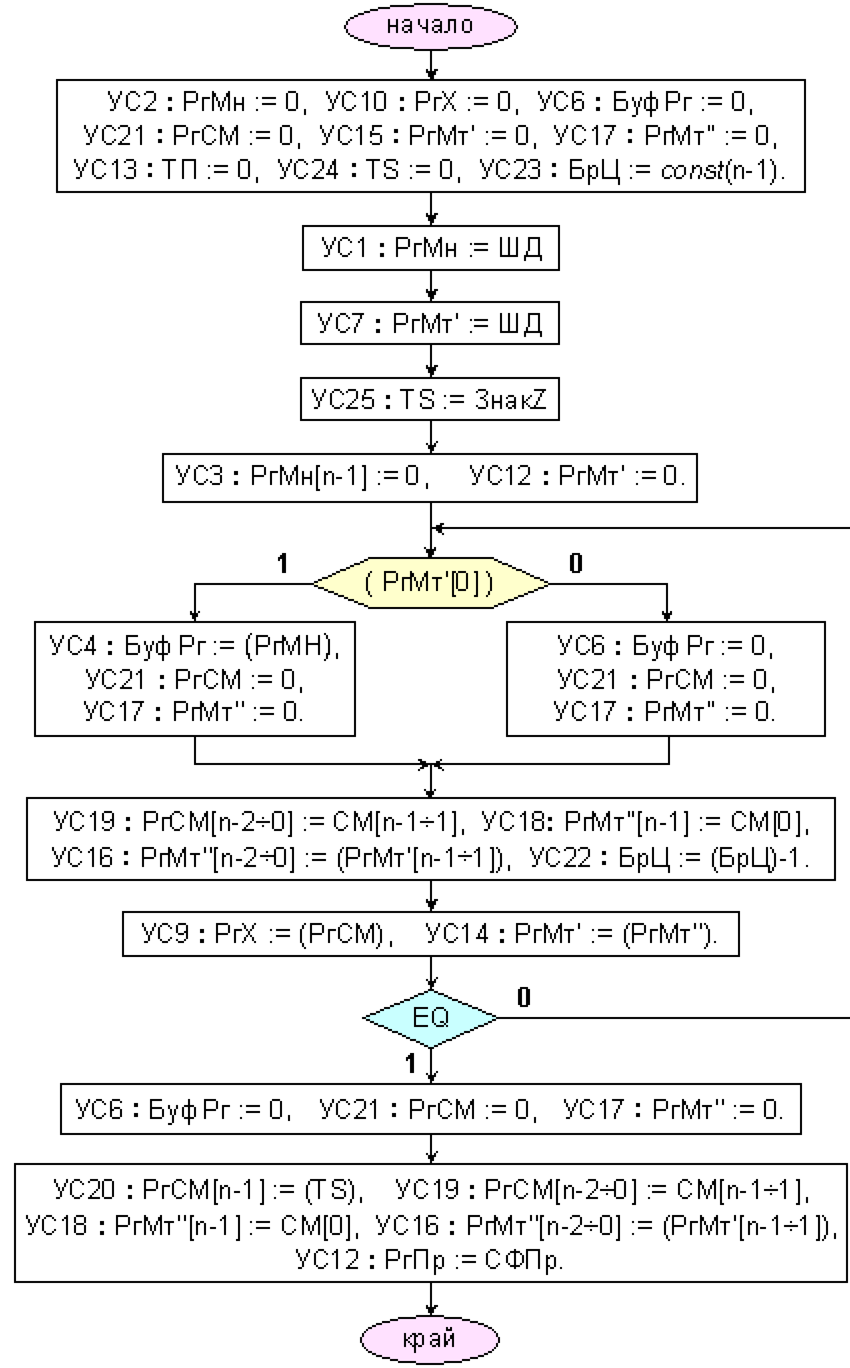


УС1 : РгМн := ШД,
УС2 : РгМн := 0,
УС3 : РгМн[n-1] := 0,
УС4 : Буф Рг := (РгМн),
УС5 : Буф Рг := not(РгМн),
УС6 : Буф Рг := 0,
УС7 : РгМт' := ШД,
УС8 : ТП := 1 (+1СМ),
УС9 : РгХ := (РгСМ),
УС10 : РгХ := 0,
УС11 : ТП := (РгПр[С]),
УС12 : РгПр := СФПр,
УС13 : ТП := 0,
УС14 : РгМт' := (РгМт''),
УС15 : РгМт' := 0,
УС17 : РгМт'' := 0,
УС18 : РгМт'' [n-1] := СМ[0],
УС20 : РгСМ[n-1] := (TS)=S,
УС21 : РгСМ := 0,
УС22 : БрЦ := (БрЦ) −1,
УС23 : БрЦ := *const*(n-1),
УС24 : TS := 0,
УС26 : РгМт'[n-1] := 0.
УС16 : РгМт'' [n-2÷0] := (РгМт' [n-1÷1]),
УС19 : РгСМ[n-2÷0] := СМ[n-1÷1]
УС25 : TS := (РгМн[n-1]) ⊕ (РгМт' [n-1]).

Операционни структури

Операции върху числа с фиксирана запетая
Операция умножение

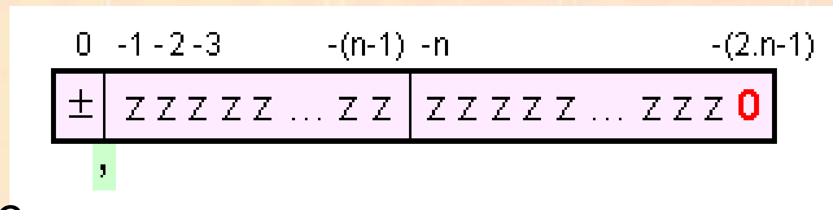
Микропрограма за умножение на числа с ДФЗ, представени в **ПК**, в устройство с **комбинационен суматор**



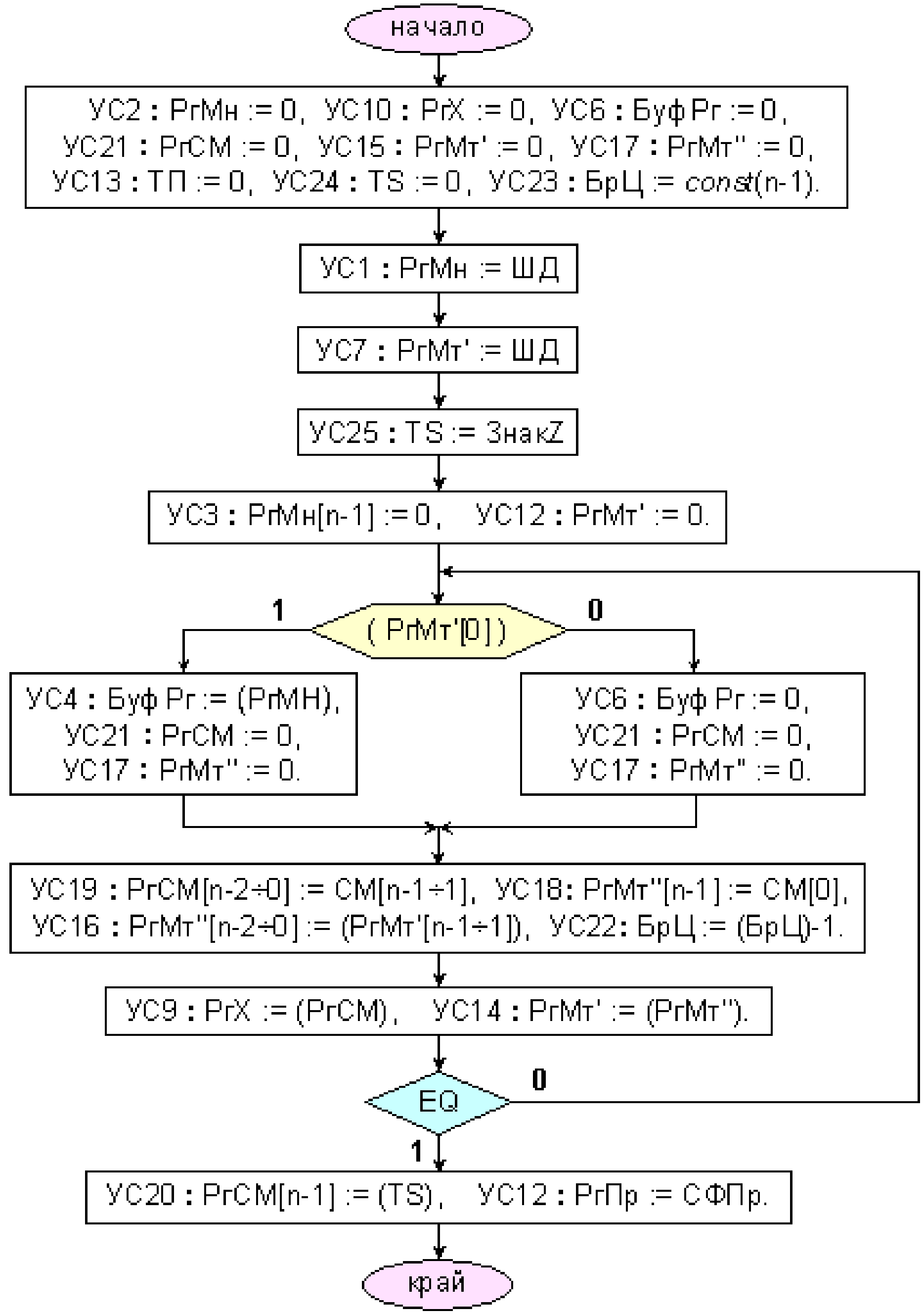
Операционни структури

Операции върху числа с фиксирана запетая Операция умножение

Микропрограма за умножение на числа с ЛФЗ, представени в **ПК**, в устройство с **комбинационен суматор**



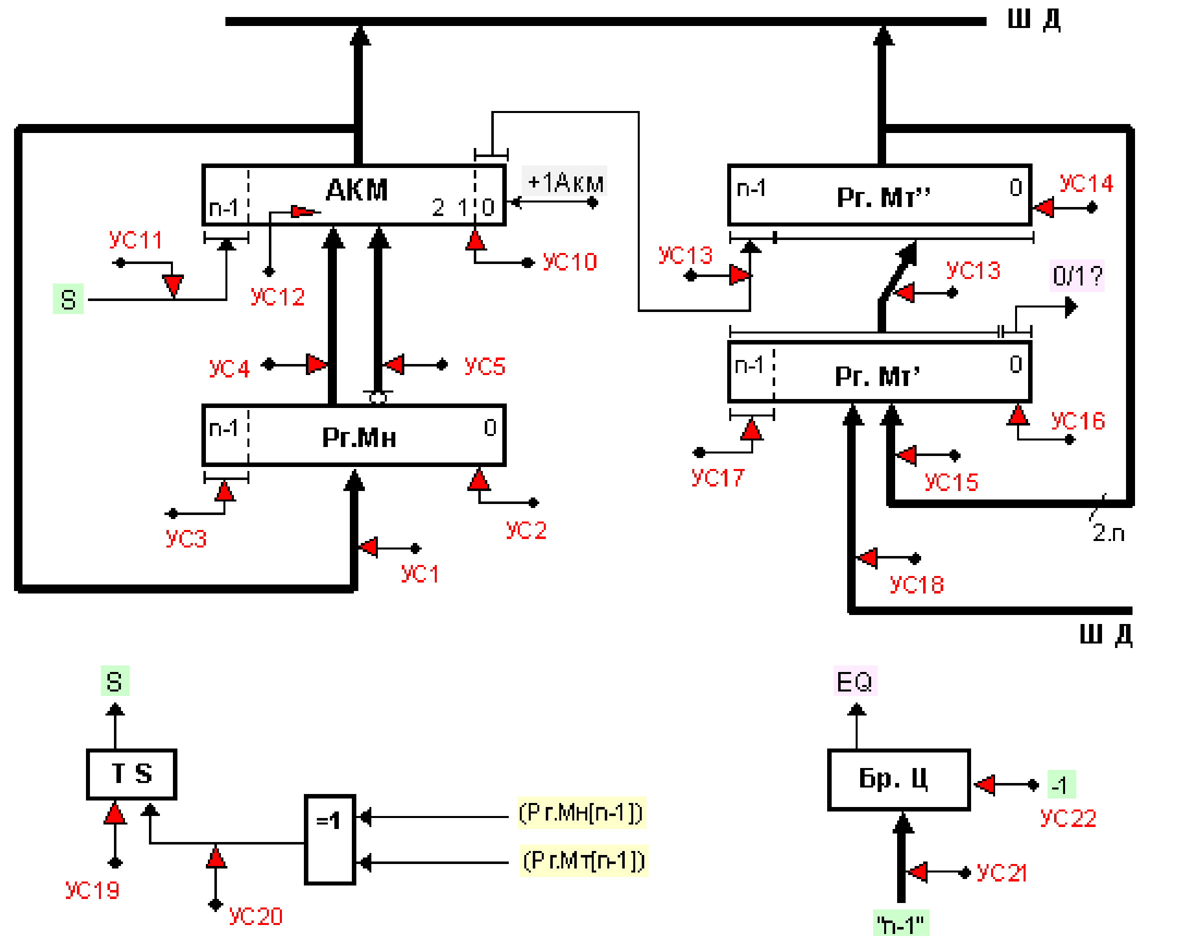
За да се осигури задължителната незначеща цифра, която се намира в най-младшия разряд, при числа с ляво фиксирана запетая е необходимо да се **нулират знаците на операндите преди началото на същинското умножение.**



Операционни структури

Операции върху числа с фиксирана запетая
Операция умножение

Базова логическа структура на устройство за умножение на числа **в прав код по метода с младшите разряди напред** по схемата с **неподвижно множимо** и с **натрупващ суматор**



УС1 : РгМн := ШД,
УС2 : РгМн := 0,
УС3 : РгМн[n-1] := 0,
УС4 : АКМ := (АКМ)+(РгМн),
УС5 : АКМ := (АКМ)+not(РгМн),
УС6 : РгПр := СФПр,
УС7 : ТП := 0,

УС8 : ТП := 1 (+1Акм),
УС9 : ТП := (РгПр[C]),
УС10 : АКМ := 0,
УС11 : АКМ[n-1] := (TS)=S,
УС12 : АКМ := ЛИД1(АКМ),
УС14 : РгМт'' := 0,
УС15 : РгМт' := (РгМт''),

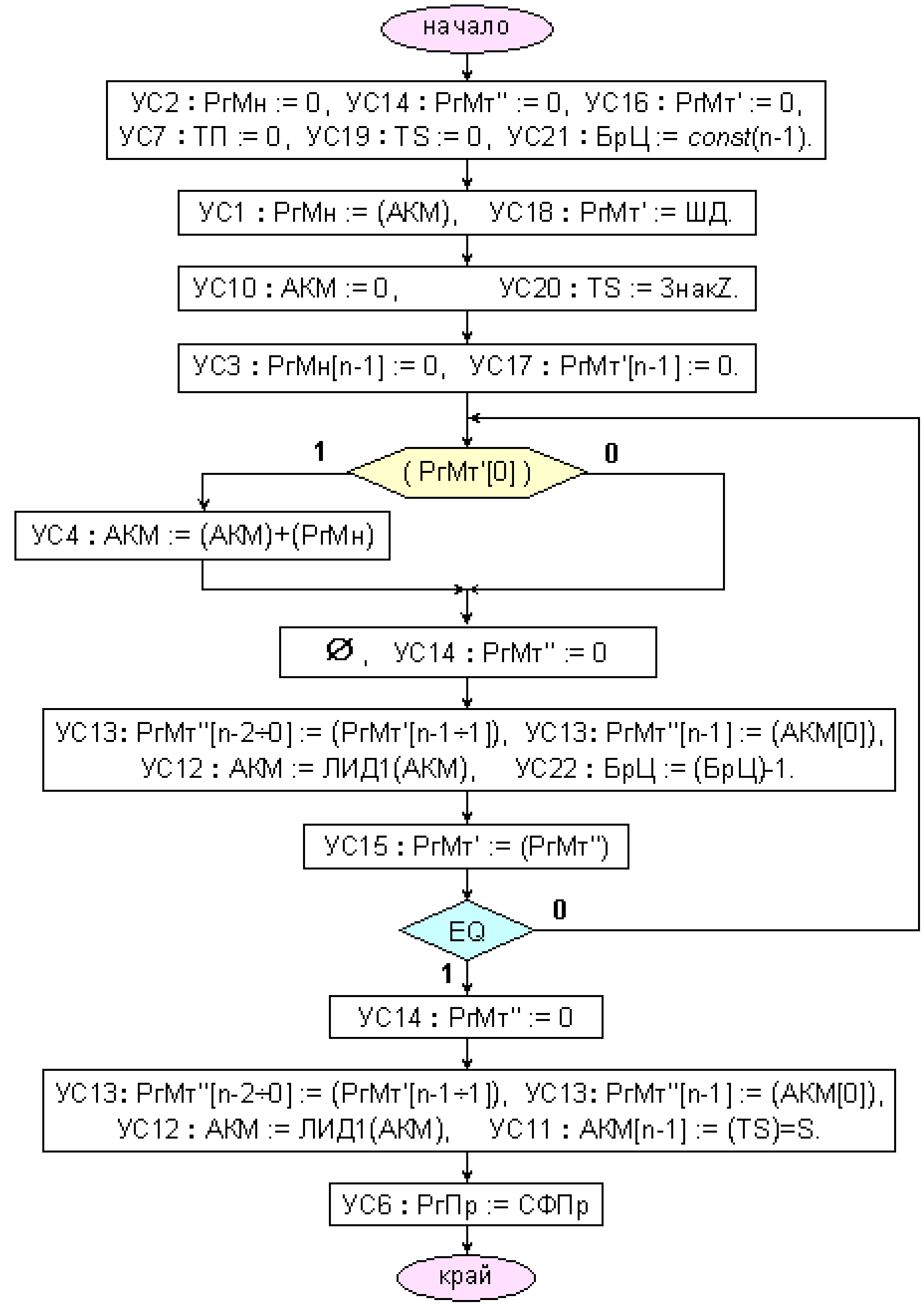
УС16 : РгМт' := 0,
УС17 : РгМт'[n-1] := 0,
УС18 : РгМт' := ШД,
УС19 : TS := 0,
УС21 : БрЦ := *const*(n-1),
УС22 : БрЦ := (БрЦ)-1,

УС20 : TS := (РгМн[n-1]) ⊕ (РгМт' [n-1]),
УС13 : РгМт'' [n-2÷0] := (РгМт' [n-1÷1]), УС13 : РгМт'' [n-1] := (АКМ[0]),

Операционни структури

Операции върху числа с фиксирана запетая Операция умножение

Микропрограма за умножение на числа с ДФЗ в **ПК** в устройство с **натрупващ суматор**



Операционни структури

Операции върху числа с фиксирана запетая

Умножение на числа, представени в допълнителен код

$$[X]_{ДК} \cdot [Y]_{ДК} = ?$$

Две групи според знака на множителя:

- В ***първа група*** са случаите, когато **множителят е положителен,** т.е. Y>0

Всички поразрядни произведения се изчисляват правилно. Всички междинни суми се натрупват също правилно. Произведението се получава по естествен път в **допълнителен код.**

Операционни структури

Операции върху числа с фиксирана запетая

Умножение на числа, представени в допълнителен код

$$[X]_{ДК} \cdot [Y]_{ДК} = ?$$

Две групи според знака на множителя:

- ***Втората група*** съдържа всички случаи, в които **множителят е отрицателен, т.е. Y<0.**

Цифрите на множителя не съответстват на тези в неговия модул – той е представен в допълнителен код.

Прилагайки същия алгоритъм за умножение, получаваме по същество **псевдопроизведението** на множимото с онова, което стои в цифровата част на множителя.

Операционни структури

Операции върху числа с фиксирана запетая

Умножение на числа, представени в допълнителен код

Ако множителят Y е представен в допълнителен код, то: $Y_{ДК} = 2^n - |Y|$

Умножението с него се извършва само с цифрите от цифровата му част - битове [n-2¸0], където се намира числото $2^{n-1} - |Y|$.

Това число е получено от стойността на кода чрез премахване на теглото на знаковия разряд, т.е.

$$\left(2^n - |Y|\right) - 2^{n-1} = \left(2^{n-1} + 2^{n-1} - |Y|\right) - 2^{n-1} = 2^{n-1} - |Y|.$$

Тогава стойността на псевдопроизведението може да бъде записана така:

$$\tilde{Z} = X.\left(2^{n-1} - |Y|\right) = X.2^{n-1} - X.|Y|.$$

След анализ записваме: $\tilde{Z} = X.2^{n-1} + Z$, т. е. $Z = \tilde{Z} - X.2^{n-1}$.

Този израз изразява начина, по който може да се получи истинското произведение, ако вече е получено псевдопроизведението. Тази последна операция изваждане се нарича ***корекция.***

Операционни структури

Операции върху числа с фиксирана запетая

Операция деление

Операция деление Z=X/Y е производна от операция изваждане.

Частното се търси като число, което показва *колко пъти делителят се съдържа в делимото* (за числа с ДФЗ).

Операцията е ***изпълнима***, ако е изпълнено съотношението:

$$|X| \geq |Y|$$

Ако съотношението не е изпълнено, операцията е **неизпълнима**, а резултатът е нула (Z = 0).

Операционни структури

Операции върху числа с фиксирана запетая

Операция деление

Ако делителят е равен на нула, операцията е не само **неизпълнима, но и недефинирана.** Резултатът се сигнализира чрез установяване на признака за препълване V = 1.

Операция деление може да бъде ***неопределена, неизпълнима, точна или неточна*** и в общия случай **частното е *реално* число** (т.е. число с цяла и дробна част).

Операционни структури

Операции върху числа с фиксирана запетая

Операция деление

Между операндите и резултатите съществува следната зависимост:

$$X = Y.Z + R_X$$

R_X е *остатъкът* от делението и е *цяло* число. Z (резултатът) е също *цяло* число.

Разглеждаме резултата (и остатъка) като число от същия тип, какъвто е типът на операндите (в случая – цели числа) и говорим за ***целочислено*** деление.

Операционни структури

Операции върху числа с фиксирана запетая

Операция деление

Както за операция умножение, така и за операция деление, се търсят алгоритми за реализация, които **гарантират предварително нейното времетраене.**

За техния синтез следва да бъдат взети под внимание всички характеристики на числата, на първо място трябва да бъде отчетен фактът, че числата са представени в позиционна бройна система, а на второ място - че те са в двоична бройна система.

Операционни структури

Операции върху числа с фиксирана запетая

Операция деление

Ако делимото е полином от k-ти ред - $P_x^{(k)}$, делителят е също полином, но от l-ти ред - $P_y^{(l)}$, то частното е полином от ред (k-l), т.е. частното има (k-l+1) на брой неизвестни цифри в цялата си част.

Като се има предвид, че модулите и на двата операнда X и Y са записани в поле с константна дължина (n разряда – разрядната мрежа е с дясно фиксирана запетая), то чрез една лява нормализация на операндите тази величина може да бъде определена.

$$2^{n-2} \leq X;Y < 2.2^{n-2}$$

Операционни структури

Операции върху числа с фиксирана запетая

Операция деление

Тогава:

$$Ly - Lx = (k - l)$$

Lx, Ly – брой на изместванията на делимото и делителя съответно.

Броят на неизвестните цифри на частното е:

$$N = k - l + 1 = Ly - Lx + 1$$

Операционни структури

Операции върху числа с фиксирана запетая

Операция деление

Делението на два полинома се дефинира математически еднозначно по следния начин:

$$\frac{P_x^{(k)}}{P_y^{(l)}} = \frac{x_k \cdot 2^k + x_{k-1} \cdot 2^{k-1} + \ldots + x_0}{y_l \cdot 2^l + y_{l-1} \cdot 2^{l-1} + \ldots + y_0} =$$

$$= \underbrace{(z_{k-l} \cdot 2^{k-l} + z_{k-l-1} \cdot 2^{k-l-1} + \ldots + z_0 \cdot 2^0)}_{\text{цяла част}} + \underbrace{\frac{R_x}{P_y^{(l)}}}_{\text{дробна част}}$$

Последователността от двоични цифри $\{z_{k-l}, z_{k-l-1}, \ldots, z_0\}$ (коефициенти на полинома на цялата част на частното) представляват неизвестните цифри на частното.

Операционни структури

Операции върху числа с фиксирана запетая

Операция деление

Изравнявайки реда на полинома на делителя до този на делимото, се получава:

$$\frac{P_x^{(k)}}{P_y^{(l)} \cdot 2^{k-l}} = z_{k-l} \cdot 2^0 + \left(z_{k-l-1} \cdot 2^{-1} + \ldots + z_0 \cdot 2^{-(k-l)} \right) + Q$$

z_{k-l} е най-старшата цифра на търсеното частно

$$z_{k-l} = \begin{cases} 1, & \text{ако } R_1 \geq 0; \\ 0, & \text{ако } R_1 < 0. \end{cases}$$

R_1 - първата разлика между делимото и нормализирания делител.

Операционни структури

Операции върху числа с фиксирана запетая

Операция деление

За определяне на следващата цифра z_{k-l-1} в частното е необходимо да се **възстанови нормализираният вид** на разликата спрямо нормализирания делител.

След възстановяване на съвместния нормализиран вид на двата полинома, те се подлагат отново на деление, което се реализира отново по същия начин, т.е. по определение. Резултатът, в смисъла на целочисленото деление,

$$\frac{R_1.2}{P_y.2^{k-l}} = \underbrace{z_{k-l-1}.2^0}_{\text{цяла част}} + \underbrace{z_{k-l-2}.2^{-1} + \ldots + z_0.2^{-(k-l-1)} + Q.2}_{\text{дробна част}}$$

е цифрата z_{k-l-1} и новата разлика R_2.

Операционни структури

Операции върху числа с фиксирана запетая

Операция деление

Процедирайки по този начин (с изравняване на реда на двоичните полиноми и делейки ги по определение), могат да бъдат определени последователно всички останали младши цифри на частното.

Продължителността на алгоритъма **не зависи** от модулите на операндите - той има **толкова стъпки, колкото са неизвестните цифри на частното.** И тъй като те не могат да бъдат повече от (n-1), то максималният брой тактове няма да надхвърля това число.

Операционни структури

Операции върху числа с фиксирана запетая

Операция деление

За реализиране на операция деление е необходимо устройство, в което да се изпълняват операции **събиране, изваждане и изместване.**

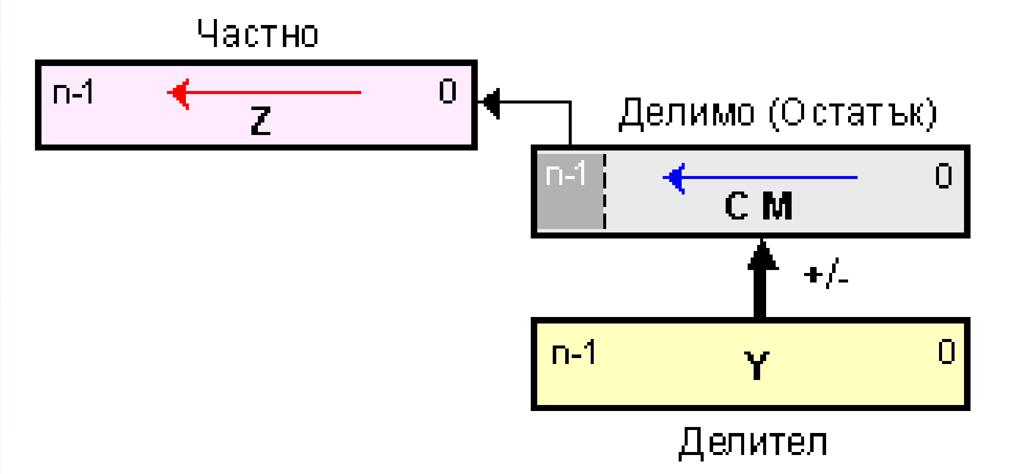
Възможни са две схеми за организация:

- С неподвижен делител и изместващ се наляво частичен остатък;
- С неподвижен частичен остатък и изместващ се надясно делител.

Операционни структури

Операции върху числа с фиксирана запетая

Операция деление

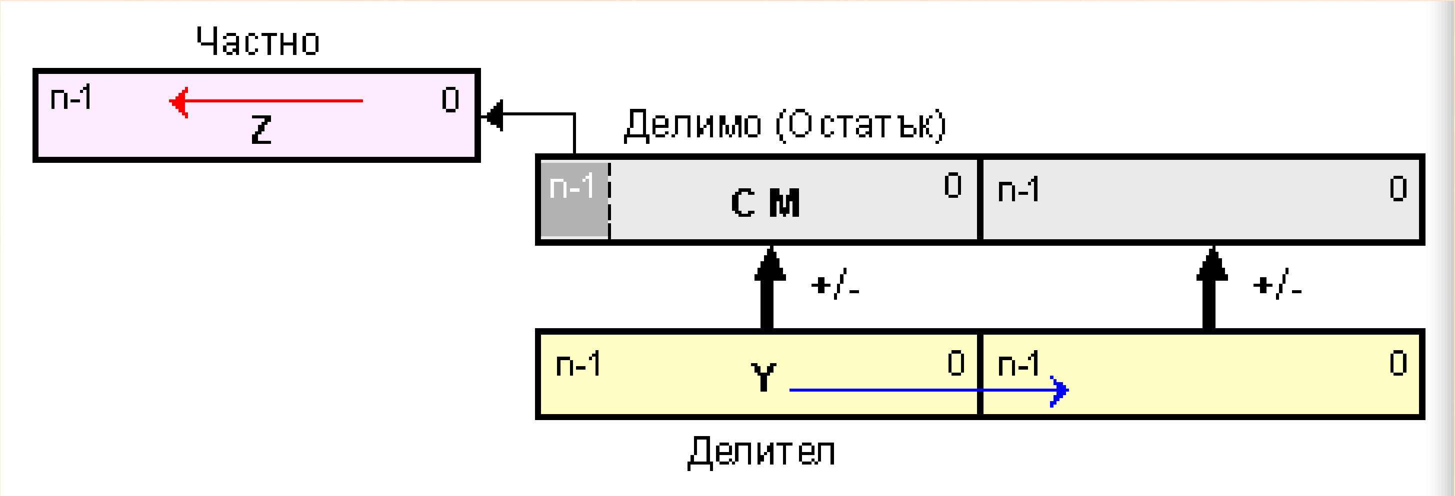


Организационна схема за деление с неподвижен делител

Операционни структури

Операции върху числа с фиксирана запетая

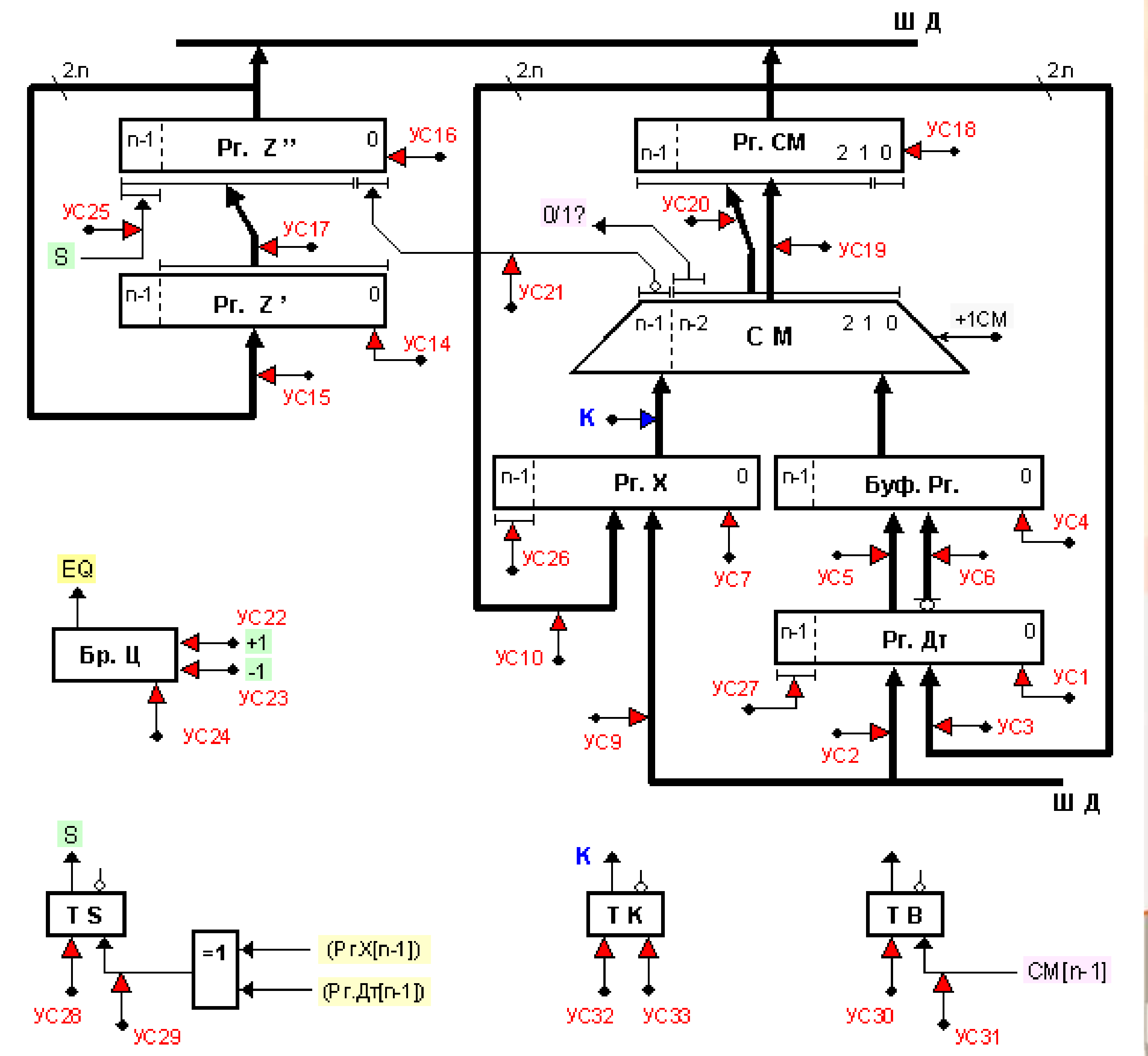
Операция деление



Организационна схема за деление с неподвижен частичен остатък

Операционни структури

Операция деление - Базова логическа структура на устройство за деление на числа, представени в прав код, по схемата с неподвижен делител



УС1 : РгДт := 0,
УС2 : РгДт := ШД,
УС3 : РгДт := (РгСМ),
УС4 : БуфРг := 0,
УС5 : БуфРг := (РгДт),
УС6 : БуфРг := not(РгДт),
УС7 : РгХ := 0,
УС8 : ТП := 1 (+1СМ),
УС9 : РгХ := ШД,
УС10 : РгХ := (РгСМ),
УС11 : ТП := (РгПр[С]),
УС12 : РгПр:=СФПр,
УС13 : ТП := 0,
УС14 : РгZ' := 0,
УС15 : РгZ' := (РгZ"),
УС16 : РгZ" := 0,
УС17 : РгZ"[n-1÷1] := (РгZ'[n-2÷0]),
УС18 : РгСМ := 0,
УС19 : РгСМ := СМ,
УС20 : РгСМт[n-1÷1] := СМ[n-2÷0],
УС21 : РгZ"[0] := notСМ[n-1],
УС22 : БрЦ := (БрЦ)+1,
УС23 : БрЦ := (БрЦ)-1,
УС24 : БрЦ := 0,
УС25 : РгZ"[n-1] := (TS)=S,
УС26 : РгХ[n-1] := 0,
УС27 : РгДт[n-1] := 0,
УС28 : TS := 0,

УС29 : TS := ЗнакZ=(РгХ[n-1])⊕(РгДт[n-1]),
УС30 : ТВ := 0,
УС31 : ТВ := СМ[n-1],
УС32 : ТК := 0,
УС33 : ТК := 1,

Операционни структури

Унитарни операции

Събиране и изваждане с единица: "***Увеличи***", "***Намали***"

$$Z = X + 1, \qquad Z = X - 1.$$

- Най-често тези операции се изпълняват върху резултат от предходно действие

$$Z := Z + 1, \qquad Z := Z - 1\ ,$$

и служат за организиране на броячи.

- При изпълнение на този вид операции се формират и в регистъра на признаците се фиксират **всички** признаци на резултата, **с изключение** на тригера за пренос С, ***който остава непроменен.***

Операционни структури

Унитарни операции

- ***Аритметическо допълнение***

По същество това е операция, при която операндът сменя знака си, а цифровата му част се инвертира и увеличава с единица, т.е. тя се преобразува по правилото за получаване на допълнителен код. В устройството за събиране тази операция се изпълнява като операция изваждане:

$$X := 0 - X ,$$

в резултат на което форматът на резултата не се удължава, а признаците му се формират както при операция изваждане.

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.