

Лабораторно упражнение № 3

ОРГАНИЗАЦИЯ НА ОПЕРАЦИОННИ СТРУКТУРИ ЗА УМНОЖЕНИЕ НА ДВОИЧНИ ЧИСЛА С ФИКСИРАНА ЗАПЕТАЯ

3.1. Теоретична постановка

3.1.1. Алгоритмично описание на операцията умножение

Алгоритъмът за умножение на двоични числа с фиксирана запетая се състои от един основен цикъл и някои спомагателни операции, които се извършват преди и след него. Техният характер зависи от вида на числата – цели или дробни, както и от представянето и обработването на отрицателните операнди – в прав или в допълнителен код. Във всички случаи обаче основният цикъл на умножението се изпълнява по един и същи начин. По-нататък се приема, че операндите X и Y са n -разрядни цели двоични числа, които са представени в прав код. Резултатът от умножението (произведението) се получава по абсолютна стойност и се представя в прав код. Неговата максимална възможна стойност е число с двойна дължина ($2n$ разряда).

Умножението на двоични числа се основава на правилата за едноразрядно двоично умножение:

x_i	y_i	z_i
0	0	0
0	1	0
1	0	0
1	1	1

Вижда се, че то е тъждествено с операцията логическо умножение (конюнкция), поради което умножението на двоичните цифри се реализира с двуходови логически елементи „И“.

Преди основния цикъл на умножението се изпълняват следните спомагателни операции:

– Проверява се дали някой от операндите е равен на нула. Ако има такъв, основният цикъл не се изпълнява. На произведението директно се присво-

ява стойност нула (с положителен знак) и се формира признак (флаг) за нулев резултат.

– Определя се знаковият разряд на произведението Z като сума по модул 2 от знаковите разряди на двата операнда. (Логическата функция сума по модул 2 има стойност 1 тогава, когато двете ѝ входни променливи имат различни стойности – 01 или 10. Това напълно съвпада с правилото за формиране на знака на произведението на две числа – когато те са с различни знаци произведението е отрицателно. Когато те са с еднакви знаци, произведението е положително). След като знаковият разряд на резултата е определен, операндите в основния цикъл се умножават по абсолютна стойност, без да се взимат под внимание техните знакови разряди.

Основният цикъл при умножението на две n разрядни двоични числа се състои от n на брой елементарни цикъла. Във всеки един от тях се извършва умножение на множимото X с поредната двоична цифра y_i на множителя Y , при което се получава съответното частично произведение $ЧП_i = X * y_i$. Тъй като за двоична бройна система y_i може да бъде равно само на 0 или 1, частичното произведение съответно е равно на 0 или на множимото X . Това частично произведение се позиционира съобразно с двоичното тегло на разряда, който заема цифрата y_i в множителя, т.е. $ЧП_i = X * y_i * 2^i$.

Полученото частично произведение $ЧП_i$ се прибавя към сумата на частичните произведения, получени на предходните стъпки от основния цикъл. Тази сума се нарича частична (ЧС) и за нея е в сила равенството $ЧС_i = ЧС_{i-1} + ЧП_i$. След сумирането на последното частично произведение се получава крайният резултат или произведението на двете входни числа. Първото частично произведение, получавано в процеса на умножение, винаги се прибавя към частична сума, чиято стойност е нула.

Старшинството (двоичното тегло) на частичните произведения се отразява като всяко ЧП се отмества с един разряд в необходимата посока (наляво или надясно) спрямо предишната ЧС. Това относително преместване (m/y ЧП и ЧС) може да се реализира по два начина:

– при неподвижна ЧС се отмества поредното ЧП. Това се осъществява на практика чрез преместване в съответната посока на множимото X (т.к. ЧП_{*i*} е равно или на 0, или на X);

– при неподвижно множимо, респективно ЧП, предишната ЧС се премества с един разряд в обратната на горния случай посока.

Във всеки такт от основния цикъл на умножението се изпълняват по два микротакта – един за формиране и сумиране на поредното ЧП и друг за преместване на множимото или на ЧС. Последователността на двата микротакта и посоката на преместване зависят от реда, в който се взимат отделните разряди на множителя $Y(y_i)$ за формиране на съответните ЧП. Съществуват две възможности. Умножението може да започне с младшия разряд на множителя (както сме свикнали да умножаваме десетични числа), или със старшия разряд на множителя. Понеже всеки един от двата варианта може да се изпълнява при

неподвижни частични суми или при неподвижно множимо, се получават общо 4 различни варианта на операционния алгоритъм за двоично умножение.

Ако умножението започва с младшия разряд на множителя, всеки следващ пореден негов разряд y_i има двойно по-голямо тегло. Затова съответното ЧП_{*i*} се отменя по отношение на предишната частична сума ЧС_{*i-1*} с един разряд наляво. (Във всички позиционни бройни системи преместването на числото с един разряд наляво и добавяне на нула в младшия разряд означава умножение на това число с основата на бройната система – в случая с 2). Това се постига чрез преместване на множимото с един разряд наляво при неподвижна ЧС, или чрез преместване на ЧС с един разряд надясно при неподвижно множимо. Във всеки елементарен цикъл най-напред се изпълнява микротакт t_1 за формиране и сумиране на ЧП, а след това микротакт t_2 за преместване на множимото или на ЧС. В t_2 се извършва и преместване на множителя с един разряд надясно. При това поредната цифра y_{i+1} става достъпна за обработка, т.е. за формиране на новото ЧП_{*i+1*}. Пример за умножение на двоични числа с младшите разряди на множителя е показан на фиг. 3.1.

$$\begin{array}{r}
 1001 \times 0101 \\
 1001 \\
 0000 \\
 1001 \\
 \hline
 0000 \\
 0101101
 \end{array}$$

Фиг. 3.1

Когато умножението започва със старшия разряд на множителя, всеки негов следващ разряд y_i има двойно по-малко тегло и затова съответното ЧП_{*i*} се отменя спрямо предишната ЧС_{*i+1*} с един разряд надясно. (Във всички позиционни бройни системи преместването на числото с един разряд надясно и добавяне на нула в старшия разряд означава делене на това число с основата на бройната система – в случая на 2). Това се осъществява или чрез преместване на множимото с един разряд надясно при неподвижна ЧС, или чрез преместване на ЧС с един разряд наляво при неподвижно множимо. Редът на действие в двата микротакта на елементарния цикъл тук е обратен – в t_1 се извършва преместване на множимото или на ЧС, а в t_2 – формиране и сумиране на ЧП. В t_1 се извършва и преместване на множителя – в този случай с един разряд наляво. С това следващият разряд на множителя y_{i-1} се подготвя за формиране на поредното ЧП_{*i-1*}. Пример за умножение на двоични числа със старшите разряди на множителя е показан на фиг. 3.2.

След изпълнението на основния цикъл произведението на двете n -разрядни числа се получава с двойна дължина – $2n$ разряда. То трябва да се представи в същия формат както входните операнди, т.е. с дължина n разряда. За целта се извършва съкращаване на произведението. При умножение на цели

числа от произведението се оставят само младшите му n разряда. При това ако поне един от изключените старши n разряда има стойност 1, се формира признак за препълване на разрядната решетка. При умножение на дробни числа с фиксирана запетая (такива са мантисите на числата с плаваща запетая) от реалното произведение се запазват само старшите n разряда. В този случай препълване на разрядната решетка не може да се получи, но съкратеното произведение се представя неточно. За да се намали размера на грешката се извършва закръгляване – ако отделените младши n разряда започват с 1 в най-левия разряд, в най-младшия от оставащите старши n разряда се прибавя 1. На съкратеното произведение се присвоява предварително определения запомнен в съответния знаков тригер знак.

$$\begin{array}{r}
 1001 \times 0101 \\
 0000 \\
 1001 \\
 0000 \\
 \underline{1001} \\
 0101101
 \end{array}$$

Фиг. 3.2

Целите двоични числа с фиксирана запетая, които се съхраняват в паметта на компютъра в допълнителен код, могат да се умножават директно в този код. В този случай основният цикъл се реализира по същия начин, но в него участват и знаковите разряди на операндите. След изпълнение на основния цикъл в произведението се внася необходимата корекция. Тя зависи от комбинацията от знаци на двата входни операнда. Резултатът със знаковия си разряд се получава представен в допълнителен код.

За ускоряване на основния цикъл в операцията умножение се използват различни алгоритмични и схемни средства. Най-простият пример за това е методът за умножение с анализ едновременно на два поредни разряда на множителя. Всяка двойка разряди може да бъде 00, 01, 10 или 11. При умножение, което започва с младшите разряди на множителя при неподвижно множимо, в тези четири случая се изпълняват следните операции:

- при 00 не се извършва формиране и сумиране на ЧП и предишната ЧС само се премества с два разряда надясно;
- при 01 се извършва прибавяне на множимото (което същност е ЧП) и след това преместване на получената нова ЧС с два разряда надясно;
- при 10 се прави преместване на старата ЧС с един разряд надясно, сумира се множимото (ЧП), след което се премества новата ЧС с един разряд надясно;
- комбинацията $11_{(2)} = 3_{(10)}$ се разглежда като разликата $(4 - 1)_{10} = (100 - 01)_2$. Тогава множимото се изважда от предишната ЧС, след което новата ЧС се премества с два разряда надясно, а следващата двойка разряди на множителя се увеличава с 1 (с което всъщност се извършва умножението по 4).

Този вариант за ускорено умножение позволява обработката на всеки два поредни разряда на множителя да се извърши за не повече от 3 елементарни такта. Вследствие на това средното време за изпълнение на операцията умножение се намалява с около 30 %. В някои процесори се реализира умножение с едновременен анализ на 4 поредни разряда на множителя, с което се постига още по-високо бързодействие.

3.1.2. Операционни схеми за умножение на двоични числа

При изпълнение на операцията умножение на двоични числа с фиксирана запетая от операционната част на процесора се използват три регистъра (P1, P2 и P3), паралелния комбинационен суматор (ПКС) и един регистър-бройч на разрядите (БрР.). Първият операнд (множимо X) се разполага в P1, а вторият операнд (множител Y) – в P2. Частичните суми се съхраняват в P3, а прибавянето на всяко частично произведение се осъществява чрез ПКС. За тази цел регистрите P1, P3 и ПКС са свързани в операционна схема за събиране в натрупващ режим, в която P3 функционира като акумулатор.

След изпълнение на основния цикъл на умножението пълното произведение на двата n разрядни операнда X и Y се получава в P3 и максималната му възможна дължина е $2n$ разряда. Това налага P3 да има съответно необходимата двойна дължина. В операционните схеми, при които частичните суми в P3 са неподвижни, се осъществява преместване на множимото в P1. За целта е необходимо P1 да има също двойно дължина. Вследствие на това изискването на двойна дължина се отнася и за комбинационния суматор.

Ако разрядната решетка на процесора има номинална дължина n разряда, операционните регистри и суматорът ще имат същата, а не двойна дължина. В такъв случай следва да се наложи съответно ограничение върху дължината на операндите – те трябва да имат двойно по-малка разрядност, т.е. $n/2$ разряда.

В операционните схеми за двоично умножение функционалните връзки между използваните блокове се установяват в съответствие с разгледаните в предишния раздел алгоритмични варианти. Така се получават четири операционни схеми (показани заедно със съответния пример за машинно умножение на двоични числа), както следва:

- за умножение с младшия разряд на множителя при неподвижни частични суми (фиг. 3.3) и при неподвижно множимо (фиг. 3.4);
- за умножение със старшия разряд на множителя при неподвижни частични суми (фиг. 3.5) и при неподвижно множимо (фиг. 3.6).

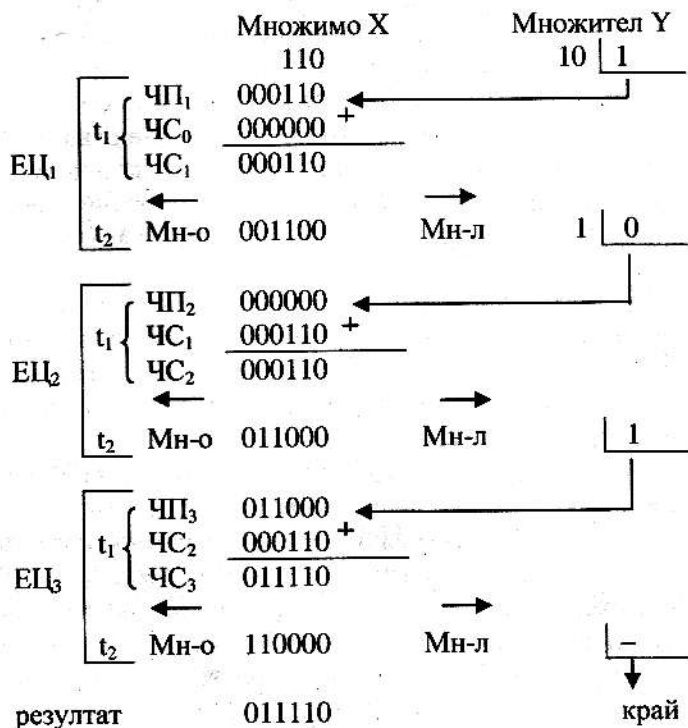
Четирите схеми имат редица общи функционални характеристики. Използват се три операционни регистъра и ПКС с дължина на разрядната решетка n разряда. Частичните произведения се получават чрез група от n на брой двуходови логически елементи „И“ (ЛЕ„И“), свързани с изходите на P1 и управлявани от съответния (младши или старши) разряд на P2. В микротакта

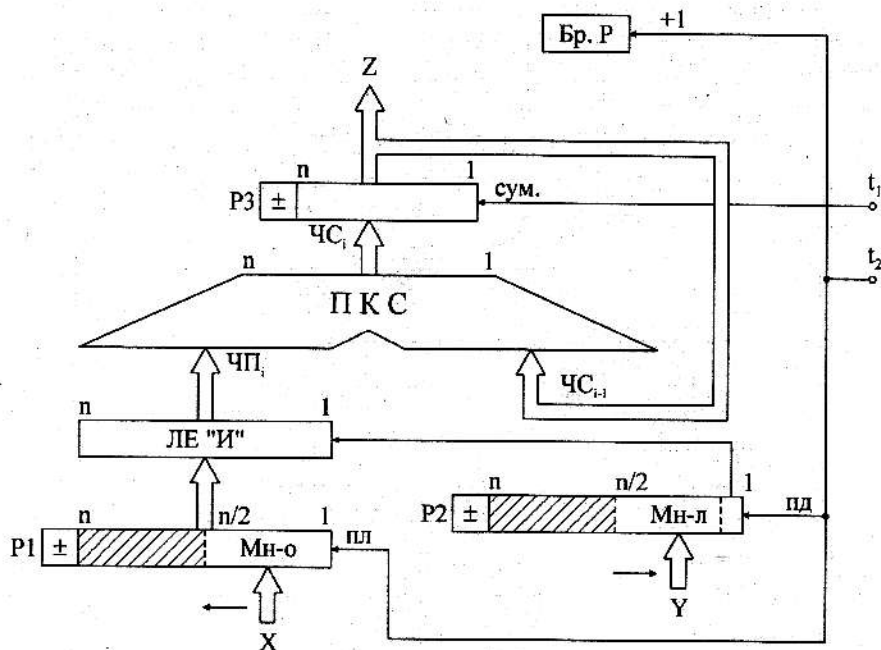
за сумиране се подава управляващ сигнал за микрооперацията „паралелен запис в P3“. Тогава резултатът от сумирането, получен на изходите на ПКС, се записва като ново съдържание (нова частична сума) в P3 ($ЧС_i := ЧС_{i-1} + ЧП_i$).

Всяко частично произведение ($ЧП_i$) се формира, като групата ЛЕ „И“ управлява предаването на съдържанието на P1 към втория групов вход на ПКС в зависимост от стойността на съответната поредна цифра y_i на множителя от P2. Ако $y_i = 1$, съдържанието на P1 се предава към суматора и се извършва сумирането $P3 := P3 + P1$, след което в P3 се установява новата частична сума. Ако $y_i = 0$, на втория групов вход на ПКС се подава число, равно на 0. В този случай сумирането е само формално, тъй като след него съдържанието на P3 не се променя ($P3 := P3 + 0$).

Във всички примери с $ЧС_0$ е означена началната частична сума, т.е. съдържанието на суматора преди започването на основния цикъл на умножение-то. Очевидно е, че винаги $ЧС_0 = 0$.

В схемите от фиг. 3.3 и 3.4 умножението започва с младшия разряд на множителя, който се намира в крайния десен тригер на P2. От неговия прав изход се подава сигнал към групата ЛЕ „И“ за формиране на ЧП. Множителя Y се разполага в дясната (младша) половина на P2. Защрихованата лява (старша) половина на P2 не се използва и в нея се записват нули.

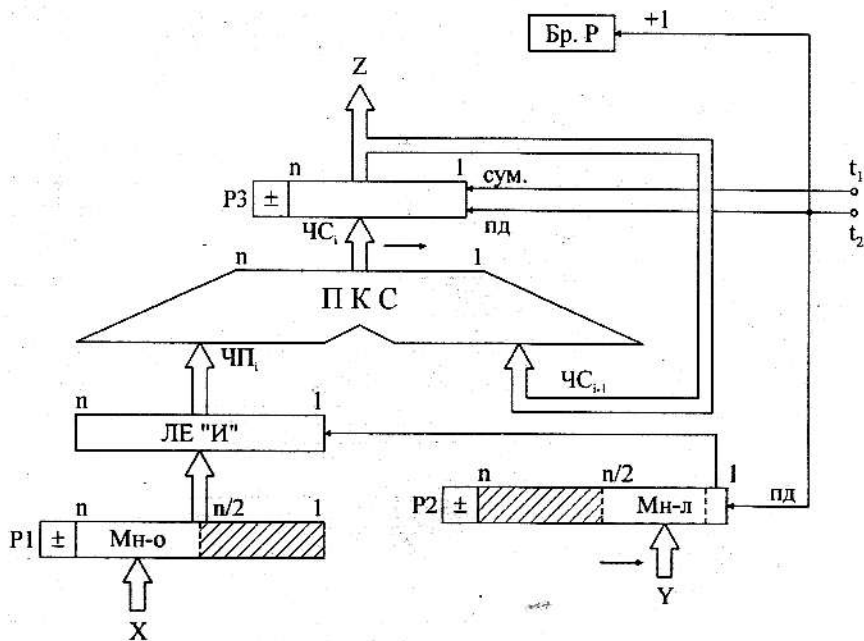




Фиг. 3.3

	Множимо X	Множител Y
	110	10 1
ЕЦ ₁	t ₁ { ЧП ₁ 110000	←
	ЧС ₀ 000000 +	
	ЧС ₁ 110000	
t ₂ → ЧС ₁ 011000	→ МН-1	1 0
ЕЦ ₂	t ₁ { ЧП ₂ 000000	←
	ЧС ₁ * 011000 +	
	ЧС ₂ 011000	
t ₂ → ЧС ₂ 001100	→ МН-1	1
ЕЦ ₃	t ₁ { ЧП ₃ 110000	←
	ЧС ₂ * 001100 +	
	ЧС ₃ 111100	
t ₂ → ЧС ₃ 011110	→ МН-1	-
результат	011110	край

(с ЧС₁* е означена преместената с един разряд надясно ЧС₁)



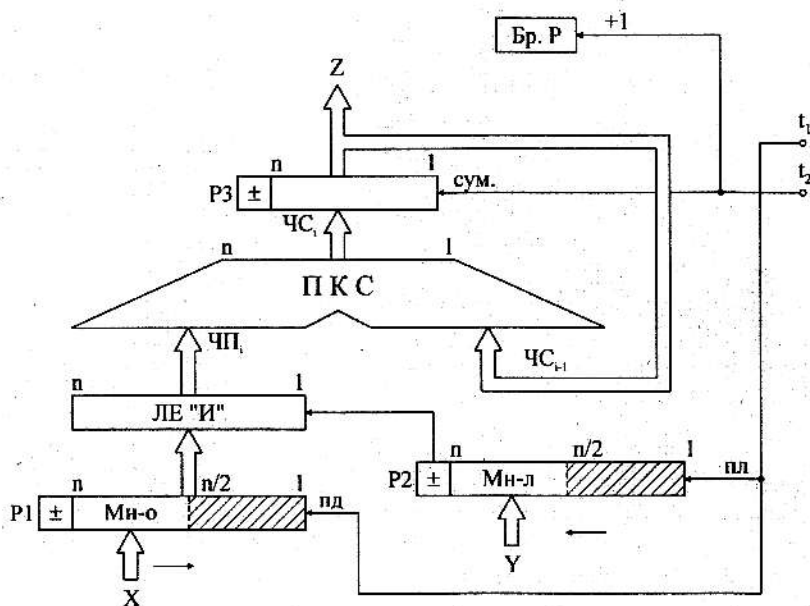
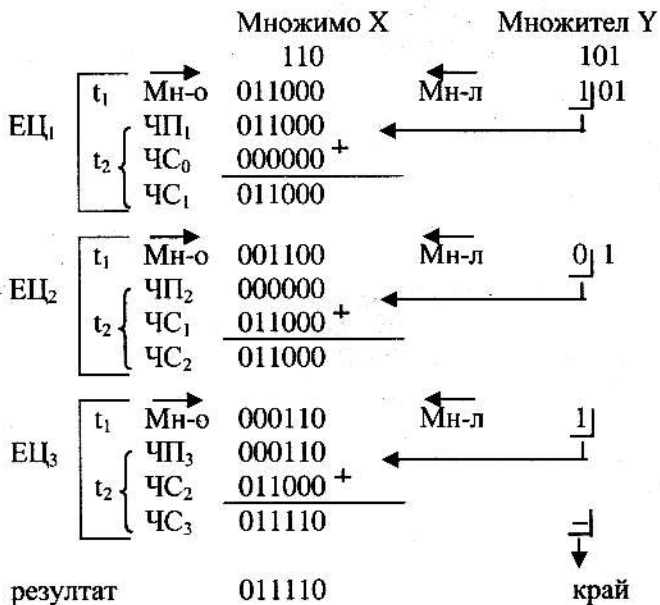
Фиг. 3.4

При всеки елементарен цикъл в микротакта t_1 се извършва сумиране на поредното ЧП и в R_3 се формира новата ЧС. В микротакта t_2 множителят се премества с един разряд надясно (ПД – преместване надясно) и следващият му разряд попада в крайния десен тригер на R_2 . При това преместване надясно младшите разряди на множителя последователно напускат R_2 и се губят (но те вече са използвани и не са нужни), а в старшия разряд на R_2 влизат нули. В схемата от фиг. 3.3 частичните суми в R_3 са неподвижни, а множимото в R_1 при всеки микротакт t_2 се премества с един разряд наляво (ПЛ – преместване наляво). Затова преди началото на основния цикъл множимото се разполага в дясната половина на R_1 .

В схемата от фиг. 3.4 множимото е неподвижно и заема лявата половина на R_1 . Защрихованата дясна половина на R_1 не се използва, но участва в сумирането и затова се запълва с нули. В микротакта t_2 частичните суми в R_3 се преместват с един разряд надясно.

В схемите от фиг. 3.5 и 3.6 умножението започва със старшия разряд на множителя Y . Множителят се разполага в лявата половина на R_2 , а защрихованата дясна част на R_2 не се използва и се запълва с нули. В първия микротакт t_1 от всеки елементарен цикъл се извършва преместване на множителя в R_2 с един разряд наляво. При това най-напред старшия а след това и всеки следващ пореден разряд на множителя попада в знаковия тригер на R_2 , който се използва като буферен разряд. Сигналят от неговия прав изход управлява групата ЛЕ „И“ за формиране на ЧП. При преместването наляво множителят последователно губи старшите си разряди и R_2 се запълва с нули, които влизат в неговия младши разряд. Сумирането на ЧП се осъществява във втория микротакт t_2

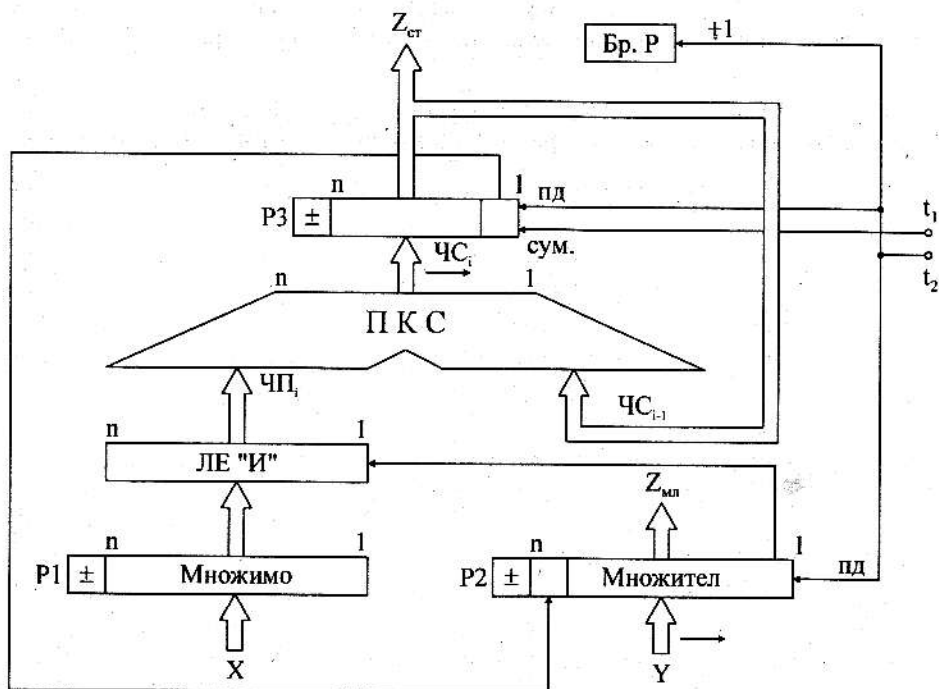
от елементарните цикли. В схемата от фиг. 3.5 частичните суми в Р3 са неподвижни. Множимото се разполага в лявата половина на Р1 и при всеки микротакт t_1 се премества с един разряд надясно. В схемата от фиг. 3.6 частичните суми в Р3 се преместват в микротакт t_1 с един разряд наляво. Множимото е неподвижно и се намира в дясната половина на Р1. Лявата половина на Р1, която не се използва, но участва в сумирането, се запълва с нули.



Фиг. 3.5

В четирите операционни схеми регистърът-бройч на разрядите (Бр.Р.) следи броя на елементарните цикли. В началото на основния цикъл той се нулира, след което при всеки втори микротакт t_2 увеличава съдържанието си с 1. В разгледаните случаи операндите са с дължина $n/2$ бита и основният цикъл на умножението завършва, когато Бр.Р. достигне съдържание $n/2$.

Нека да разгледаме отново операционната схема от фиг. 3.4. При нея неподвижното множимо заема лявата половина на Р1 и следователно сумирането на всички ЧП се осъществява само в тази част на разрядната решетка на ПКС. Дясната (младша) половина на суматора не се използва, т.к. от Р1 към нея се подават нули, а преносите при сумиране на поредното ЧП се разпространяват наляво, т.е. само в старшата половина на ПКС. Вследствие на това при преместването на ЧС надясно, в дясната половина на Р3 попадат последователно само готови младши разряди на окончателното произведение, които до края на основния цикъл не се променят, а само се преместват. Тази особеност позволява схемата от фиг. 3.4 да се оптимизира и да се използва за умножение на операнди с пълна дължина от n разряда. За целта се реализира само една допълнителна връзка – правият изход на младшия тригер на Р3 се свързва с информационния вход на старшия числов тригер на Р2. Така се получава операционната схема от фиг. 3.7.



Фиг. 3.7

Тук при преместване на ЧС надясно „изтичащите“ от младшия разряд на Р3 готови цифри на произведението се въвеждат в Р2, който при преместване надясно на множителя се освобождава откъм старшия си разряд. Това е най-ефективната схема за двоично умножение, в която всички операционни блокове – трите регистъра и ПКС – имат единична дължина n разряда, но произведението се получава с двойна дължина – $2n$ разряда. След края на основния цикъл старшите n разряда на произведението остават в Р3, а младшите n разряда, се намират в Р2. Основният цикъл на умножението завършва, когато Бр.Р. достигне съдържание n .

3.2. Задание за лабораторна работа

1. Да се реализират последователно четирите операционни схеми за двоично умножение (фиг. 3.3 ÷ 3.6). Да се проследи и опише по тактове действието им при умножение на четириразрядните операнди 1011 и 1101.

2. Да се умножат в четирите алгоритмични варианта за умножение дробните числа 0,1011 и 0,0111; 0,1001 и 0,1101. Да се представят получените произведения с единична дължина 4 разряда.

3. Да се реализира операционната схема за умножение от фиг. 3.7 и да се проследи действието ѝ при умножение на 8 разрядните операнди 11101011 и 10001100.

4. Да се изпълни ускорено умножение на двоичните числа 11111001 и 10001110, като се обработват едновременно по два младши разряда на множителя.