Лекция: Обработване на информацията. Аритметични операции. Логически операции.

1. Цел на занятието.

Целта на лекцията е студентите да се запознаят с аритметичните и логическите операции извършвани в компютрите.

В предходната лекция се запознахме с двоичната бройна система, представянето на числата със знак в П.К., О.К., Д.К. извършването на действията събиране и изваждане в тези кодове.

Днес ще продължим с аритметичната операция умножение и ще се запознаем с част от логическите операции реализирани в процесора.

2. Умножение на числа.

При умножение на числата в ПК знаковите разряди и разрядите на мантисите се обработват отделно. За определяне кода на знака на произведението се извършва сумиране по mod 2 на кодовете на знаците на множимото и множителя. Кодът на мантисата на резултата се получава като се умножат кодовете на мантисите на двата операнда. Умножението на мантисите може да започне с младшите или със старшите разряди на множителя, при което може да се измества сумата на частичните произведения или множимото, т.е. възможни са 4 основни метода за умножение, които по-долу ще бъдат наричани условно A, B, C и D.

Умножението по метод е изучваното умножение в началния курс.

Метод А.

При този метод произведението се представя по следния начин:

$$Z = Y.X = Y.0,x_1x_2x_3...x_{n-1}x_n = Y(x_1 2^{-1}+x_2 2^{-2}+x_3 2^{-3}+ ...+x_{n-1} 2^{-(n-1)}+x_n 2^{-n})=$$
$$= Yx_12^{-1} + Yx_22^{-2} + Yx_32^{-3} + .. + Yx_{n-1}2^{-(n-1)}+ Yx_n2^{-n} =$$
$$= 2^{-1}(Yx_1 + 2^{-1}(Yx_2 + 2^{-1}(Yx_3 + .. + 2^{-1}(Yx_{n-1} + 2^{-1}(Yx_n + 0))..))),$$

където с X, Y и Z са означени само мантисите на двата операнда и на резултата. От тук следва, че умножението може да се извърши по следните рекурентни формули:

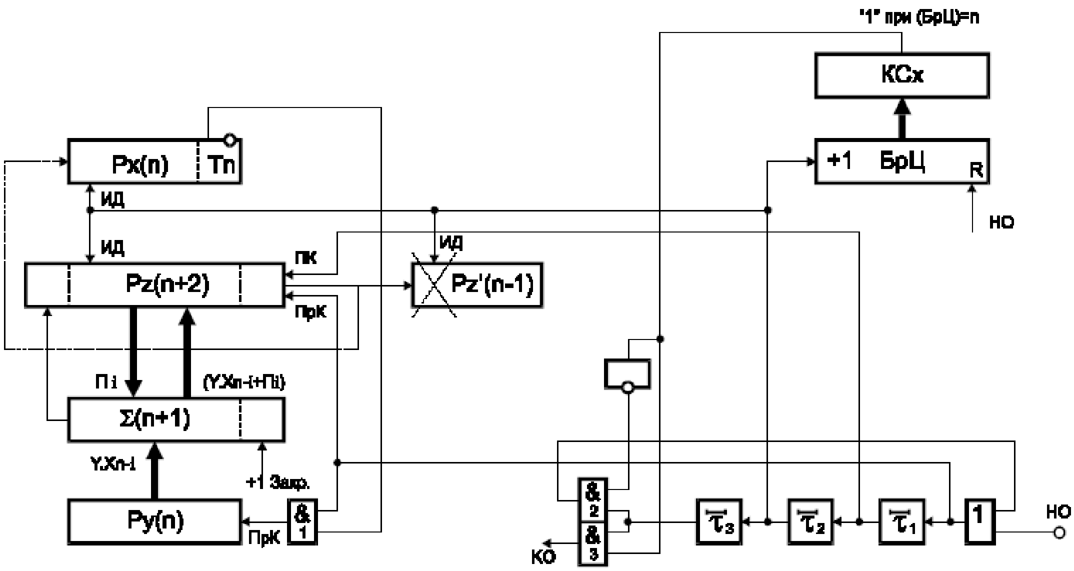
$\Pi_0 = 0$
$\Pi_1 = 2^{-1}(Yx_n + \Pi_0)$
$\Pi_2 = 2^{-1}(Yx_{n-1} + \Pi_1)$
...
$\Pi_{i+1} = 2^{-1}(Yx_{n-i} + \Pi_i)$
...
$\Pi_n = 2^{-1}(Yx_1 + \Pi_{n-1}) = Z$

т.е. умножението се свежда към n-кратно повторение на цикъла:

$\Pi_{i+1} = 2^{-1}(Yx_{n-i} + \Pi_i)$ при начални условия: $i = 0, \Pi_0 = 0$.

От тази формула се вижда, че умножението започва с младшите разряди на множителя (x_n) и че на изместване подлежи сумата на частичните произведения (П). Поради това методът се нарича "**умножение с младшите разряди на множителя, с изместване сумата на частичните произведения надясно**".

Схемата на операционния блок за умножение по метод А е показана на фигура1. Действието на блока ще бъде обяснено от момента, в който в Px и в Py са заредени мантисите на правите кодове на двата операнда, а Pz, Pz' и БрЦ са нулирани. Подава се сигнал НО. През елемента "ИЛИ" този сигнал постъпва на шината за ПрК от Pz в $\sum$, при което в последния се подава $П_0 = 0$. Същият сигнал се подава и на един от входовете на елемента "И-1". При това, ако x_n е равно на "1", т.е. ако в тригера Tn на Px е записана "1", то този елемент ще се отвори, сигналът ще премине през него и ще постъпи на шината за ПрК от Py в $\sum$. В противен случай сигнал ПрК няма да се формира. Обобщено може да се приеме, че от Py към суматора се подава Yx_n. След време τ1 се получава сигнал за ПК от $\sum$ в Pz, а след време τ2 - сигнал за ИД в Pz, Pz' и в Px. При това в Pz и в Pz' се получава
$П_1 = 2\text{-}1(Yx_n+П_0)$, а в T_n се записва x_{n-1}. Същият сигнал увеличава с единица съдържанието на БрЦ. След време τ3 се проверява съдържанието на този брояч. Тъй като в края на първия цикъл (БрЦ) = 1, то на изхода на КСх ще има "0", която затваря елемента "И-3" и не позволява формирането на сигнала КО, а след инвертиране отваря елемента "И-2" и разрешава връщането на сигнала на входа на блока за местно управление. Така описаните действия се повтарят общо n пъти. В края на n-тия цикъл в Pz и в Pz' се получава 2n-разрядната мантиса на произведението на двата операнда, след което се формира сигналът КО.



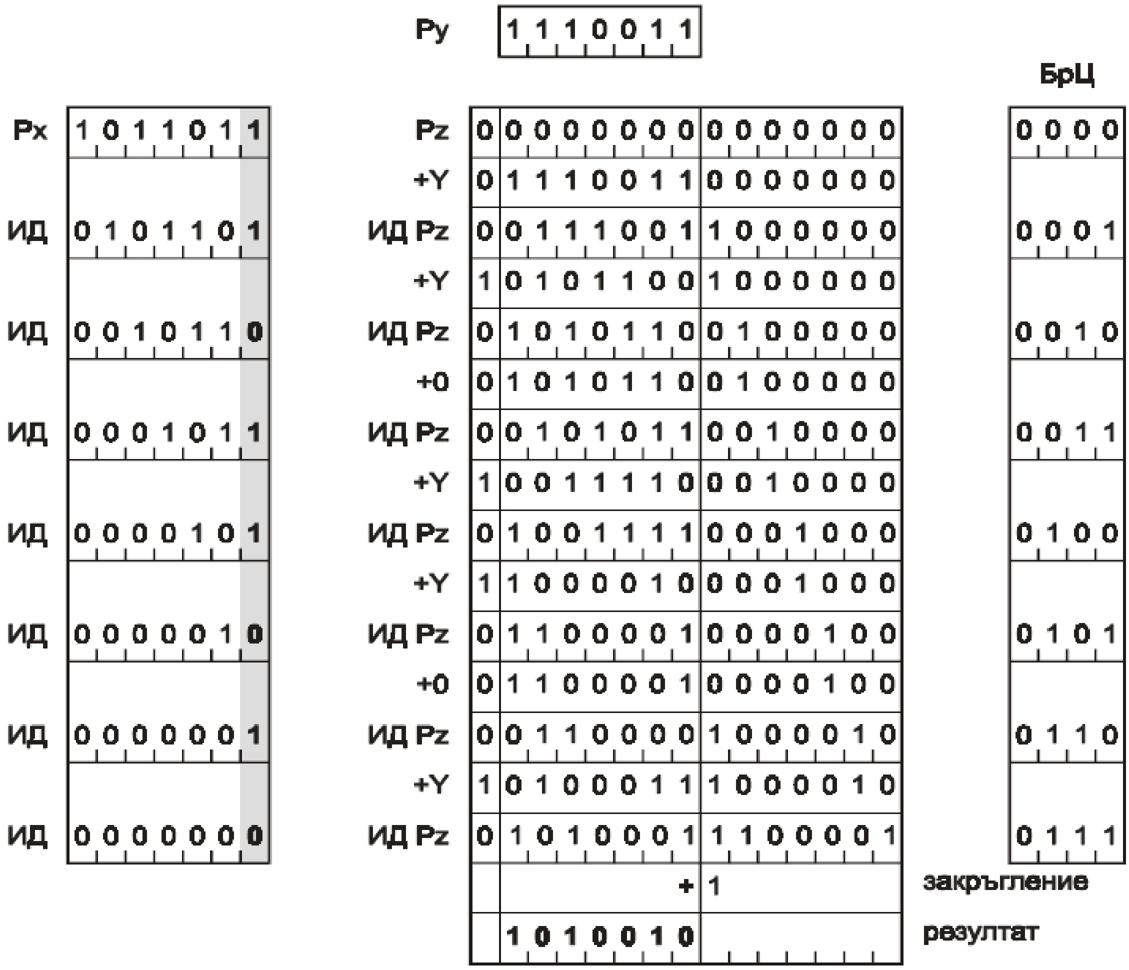
Фигура 1. Схема на операционен блок за умножение по метод А.

Действието на блока за умножение е пояснено и чрез цифровата диаграма на фигура 2.

Както се вижда от схемата на фигура 1, Pz е с разрядност (n+2), а $\sum$ с разрядност (n+1), за да не се загуби съответно единицата на преноса, който евентуално би могъл да възникне при събиране на Пi и Yx_{n-i} и за да може да се извърши закръгляване на резултата чрез прибавяне на "1" в (n+1)-вия разряд на 2nразрядното произведение. Тъй като изместването в Pz’ и Px се осъществява в една и съща посока, при което старшите разряди на Pz’ се запълват, а на Px се освобождават, то вместо Pz’ може да се използва Px .

Времето за умножение по този метод с показаната схема може да се определи по следната формула:

$t_{y}A = n(\tau_1+\tau_2+\tau_3) = n(t_{ПрК} + t_{\Sigma} + t_{ПК} + t_{ИК}) = n(t_C + t_{ИК})$



Фигура 2. Цифрова диаграма на операционен блок за умножение по метод А.

3. Логически операции.

Логически функции

Думата ***логика*** има гръцки произход (от ***логос*** – дума, мисъл, понятие, разум).

В качеството си на философски термин се използува за ***означаване на общите закономерности*** на света и ***мисленето***.

Тя се ***оформя*** като наука ***в дълбока древност*** в трудовете ***на Аристотел*** (384-322 год. преди н.е.) – ***аристотелова логика***.

Основен интерес в тази наука ***са съжденията***.

Математическата логика се оформя като наука през 19 век, от ирландския математик Джордж Бул (1815-1864), които поставя началото на математическата логика, по-късно наречена на него булева алгебра.

Булевата алгебра (или алгебра на съжденията) е специална алгебрична структура, която съдържа съждения, логическите оператори И, ИЛИ, НЕ, както и множествените функции: сечение, обединение, допълнение.

Въвеждат се понятията Вярно и Невярно (Истина и Лъжа), които се означават с 1 и 0.

Логическите функции съществуват без променливи, с една променлива, с две променливи, с три и повече променливи.

Изходната стойност на логическата функция ще я отбелязваме с Y. Тя има едно от двете логически състояния: ”0” – лъжа или ”1” – истина. Наричат се още логическа нула и логическа единица.

<u>Логически функции</u>

а) логически константи: лог. 0 и лог. 1

б) логически променливи всяка величина, която може да приема само две стойности лог. 0 или лог. 1

в) логическа функция на краен брой логически променливи, която може да приема само две стойности лог. 0 или лог. 1

<u>Функционална зависимост:</u>

$Y = f(x1,x2,,xn)$, където $x_1, x_2, \dots , x_n$ са променливи

Набор от логически променливи – всяка комбинация от стойности на определен брой променливи, за да се получи номера на набора, той се разглежда като двоично число и се преобразува в число от ДБС. Броят на всички възможни набори от n - променливи е: $N=2^n$.

Броя на всички възможни лог. функции на n променливи

$$M = 2^N = 2^{2n}$$

Логическа функция **без променливи**: Y

Y	Наименование на състоянието
0	константа нула
1	константа едно

Логическа функция **с една променлива**: Y

променлива X	изходна функция Y	Наименование на състоянието
x	0	константа нула
0	0	променливата X
1	1	
1	0	NO X (отрицание на променливата X, инверстна стойност)
0	1	
x	1	константа едно

Логически функции на **две променливи:**

Логическо сумиране (OR): Y = X1 OR X2 = X1 V X2

Логическо произведение (AND): Y = X1 AND X2 = X1 & X2

Логическо отрицание (NO): Y = NO X1

Сума по модул 2 (mod 2) (логическа не равнозначност) $Y = X_1 \overline{X_2}\ V\ \overline{X_1} X_2$

Логическо отрицание (NO): Y = NO X1

X	Y
0	1
1	0

Логическо сумиране (OR): Y = X1 OR X2 = X1 V X2 = X1 + X2

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

Логическо произведение (AND): Y = X1 AND X2 = X1 & X2

X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

Сума по модул 2 (mod 2) (логическа не равнозначност) $Y = X_1 \overline{X_2} \ V \ \overline{X_1} X_2$

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0

Логическо сумиране с отрицание (NOR): $Y = \overline{\mathrm{X1\,OR\,X2}} = \overline{\mathrm{X1\,V\,X2}}$

X1	X2	Y
0	0	1
0	1	0
1	0	0
1	1	0

Логическо произведение с отрицание (NAND): $Y = \overline{\mathrm{X1\,AND\,X2}} = \overline{\mathrm{X1\,\&\,X2}}$

X1	X2	Y
0	0	1
0	1	1
1	0	1
1	1	0

[Нова тема 12.03.2013 г.:]

Лекция: Базови компоненти. Комбинационни логически схеми.
Логически схеми с памет.

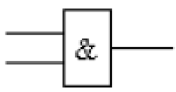
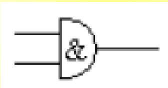
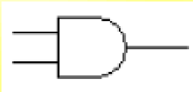
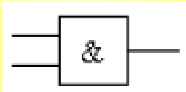
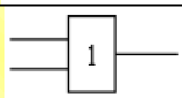
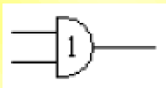
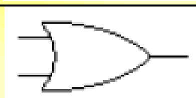
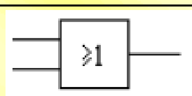
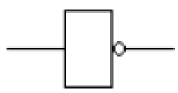
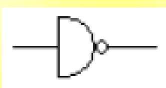
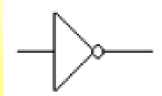
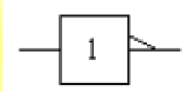
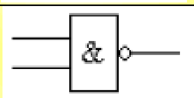
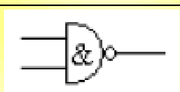
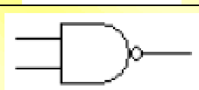
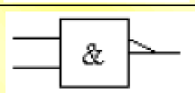
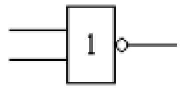
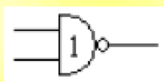
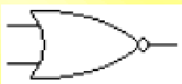
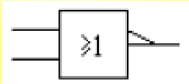
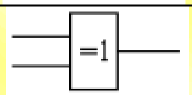
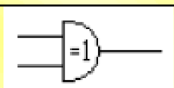
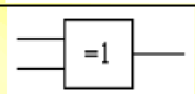
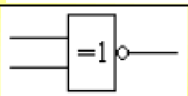
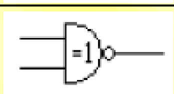
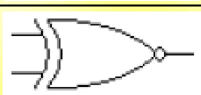
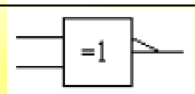
Базови компоненти.

Схемната реализация на логическите функции от логически аргументи е логическа схема (логически елемент).

Базови елементи са логически елементи с помощта, на които могат да се реализират всички функции. Например:

- Логическо произведение (*AND*): Y = X1 AND X2 = X1 & X2
- Логическо сумиране (*OR*): Y = X1 OR X2 = X1 V X2
- Логическо отрицание (*NO*): $Y = NO\ X1 = \overline{X1}$
- Сума по модул 2 (*mod 2*) (логическа не равнозначност) $Y = X_1\overline{X_2}\ V\ \overline{X_1}X_2$

Условните означения на логическите елементи (логическите схеми) по най-разпространените стандарти за изчертаване са дадени в следващата таблица.

Име на функцията	Английски стандарт	Стар Английски стандарт	Американски стандарт	ANSI / IEEE стандарт (91 – 1984)
И				
ИЛИ				
НЕ				
И – НЕ				
ИЛИ – НЕ				
ИЗКЛЮЧВАЩО ИЛИ				
ИЗКЛЮЧВАЩО ИЛИ - НЕ				

Комбинационни логически схеми

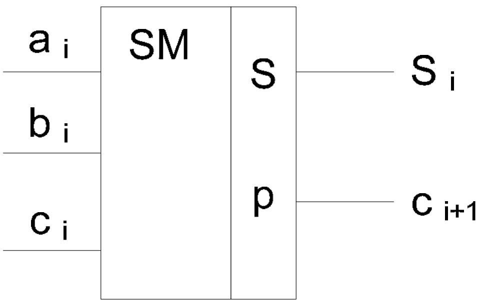
Суматори

Суматорите са устройства, изпълняващи операцията аритметическо събиране на кодовете на числата. Тъй като в компютрите, чрез въвеждането на т.нар. машинни кодове, всички аритметични операции с числа се свеждат към аритметично събиране на техните кодове, то суматорът се оказва една от най-важните съставни части на централния процесор и по-точно на неговото аритметико-логическо устройство.

Суматорът и по-точно неговото бързодействие е един от основните фактори, от които зависи производителността на компютърната система.

Пълен суматор

a_i	b_I	c_i	S_i	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Логически схеми с памет

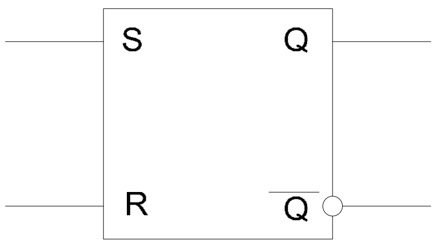
Тригери

Тригери ($\overline{R}-\overline{S}$ R-S D J-K)

Логически елементи с памет. Имат състояние, в което съхраняват вече записаната информация.

R-S Тригер (S – Set, R – Reset)

R	S	Q_{t+1}
0	0	Q_t
0	1	1
1	0	0
1	1	X

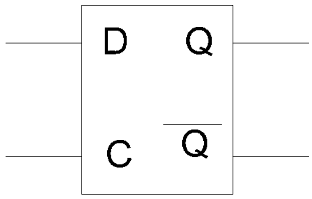


$\overline{R}-\overline{S}$ Тригер

$\overline{R}$	$\overline{S}$	Q_{t+1}
0	0	X
0	1	0
1	0	1
1	1	Q_t

D Тригер

C	D	Q_{t+1}
0 ^	0 0	Q_t 0
^	1	1



T Тригер

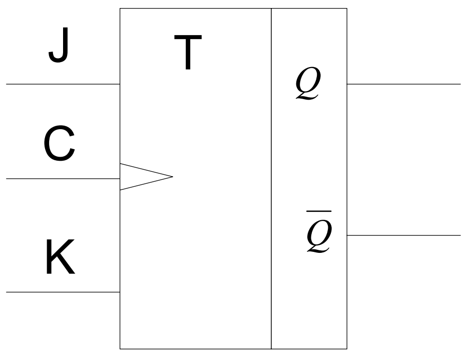
Тригер с броячен вход. На всеки тактов импулс си променя състоянието.

T	Q_{t+1}
^	$\overline{Q}_t$

J-K Тригер

Вход J установява в 1, а K – в 0.

C	J	K	Q_{t+1}
^	0	0	Q_t
^	0	1	1
^	1	0	0
^	1	1	$\overline{Q}_t$
0	*	*	Q_t



Регистри

Елементи с памет, които записват подадената им на входовете информация при активиране на тактовия сигнал.

Елементи с памет съставени от *n* броя тригери, свързани в определена схема. За реализиране на:

- паралелен запис;
- последователен запис;
- преместващи регистри;
- паралелно четене (извеждане на информацията);
- последователно четене;
- комбинирано записване и/или четене.

Броят на разрядите им е от два до шестнадесет.