

Дата 04.03.2014 год.

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 са наричани условно А, В, С и D.

Умножението по метод А е изучаваното умножение в началния курс.

Да се умножат много разредните числа 1234 по 32:

$$\begin{array}{r} Z = 1234 * 32 \\ 1234 * 2 = 2468 \\ \quad \quad \quad + \\ 1234 * 3 = \underline{37020} \\ \quad \quad \quad 39488 \end{array}$$

В двоична БС операциите са същите:

Да се умножат много разредните двоични числа $10101011_{(2)}$ по $101_{(2)}$:

$$\begin{array}{r} Z = 10101011_{(2)} * 101_{(2)} \\ 10101011 * 1 = 10101011 \\ 10101011 * 0 = 00000000 \quad + \\ 10101011 * 1 = \underline{10101011} \\ \quad \quad \quad 110101011 \end{array}$$

a	1	0	1	0	1	0	1	1	
b					1	0	1		
		1	1	1					пренос
		1	0	1	0	1	0	1	*b0 =1
+		0	0	0	0	0	0	0	*b1 =0
		1	0	1	0	1	0	1	*b2 =1
									*b3
z	1	1	0	1	0	1	0	1	1

Метод А.

При този метод произведението се представя по следния начин:

$$\begin{aligned}
 Z &= Y \cdot X = Y \cdot 0, x_1 x_2 x_3 \dots x_{n-1} x_n = Y (x_1 2^{-1} + x_2 2^{-2} + x_3 2^{-3} + \dots + x_{n-1} 2^{-(n-1)} + x_n 2^{-n}) = \\
 &= Y x_1 2^{-1} + Y x_2 2^{-2} + Y x_3 2^{-3} + \dots + Y x_{n-1} 2^{-(n-1)} + Y x_n 2^{-n} = \\
 &= 2^{-1} (Y x_1 + 2^{-1} (Y x_2 + 2^{-1} (Y x_3 + \dots + 2^{-1} (Y x_{n-1} + 2^{-1} (Y x_n + 0)) \dots))),
 \end{aligned}$$

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:

$$P_0 = 0$$

$$P_1 = 2^{-1} (Y x_n + P_0)$$

$$P_2 = 2^{-1} (Y x_{n-1} + P_1)$$

...

$$P_{i+1} = 2^{-1} (Y x_{n-i} + P_i)$$

...

$$P_n = 2^{-1} (Y x_1 + P_{n-1}) = Z$$

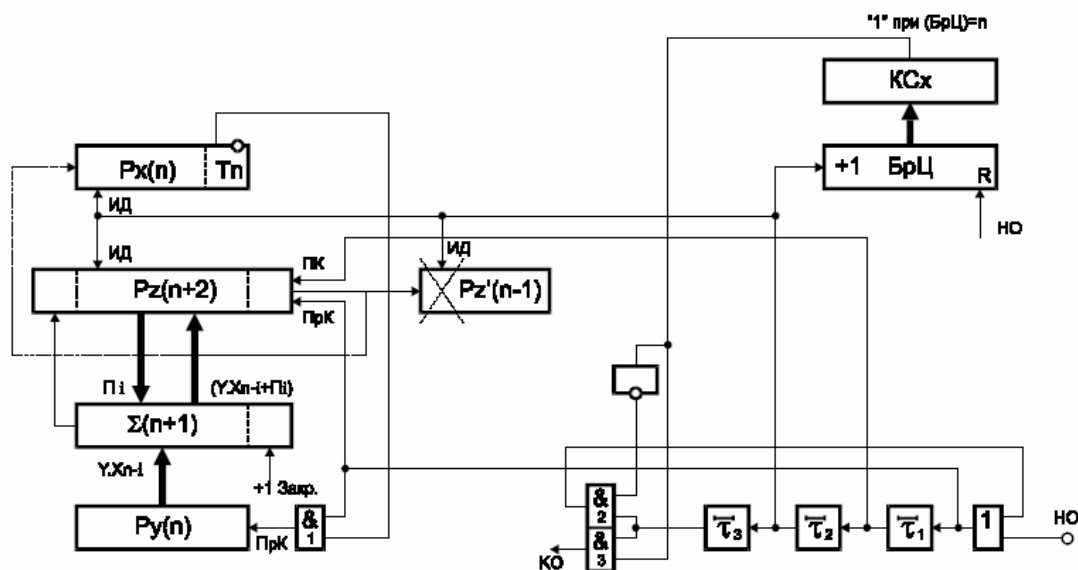
т.е. умножението се свежда към n-кратно повторение на цикъла:

$$P_{i+1} = 2^{-1} (Y x_{n-i} + P_i) \text{ при начални условия: } i = 0, P_0 = 0.$$

От тази формула се вижда, че умножението започва с младшите разряди на множителя (x_n) и че на изместване подлежи сумата на частичните произведения (P). 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 фигура 1. Действието на блока ще бъде обяснено от момента, в който в P_x и в P_y са заредени мантисите на правите кодове на двата операнда, а P_z , P_z' и БрЦ са нулирани. Подава се сигнал НО. През елемента "ИЛИ" този сигнал постъпва на шината за ПрК от P_z в Σ , при което в последния се подава $P_0 = 0$. 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 T_n на P_x е записана "1", то този елемент ще се отвори, сигналът ще премине през него и ще постъпи на шината за ПрК от P_y в Σ . В противен случай сигнал ПрК няма да се формира. Обобщено може да се приеме, че от P_y към суматора се подава $Y x_n$. След време τ_1 се получава сигнал за ПК от Σ в P_z , а след

време τ_2 - сигнал за ИД в Pz , Pz' и в Rx . При това в Pz и в Pz' се получава $\Pi_1 = 2^{-1}(Yx_n + \Pi_0)$, а в T_n се записва x_{n-1} . Същият сигнал увеличава с единица съдържанието на БрЦ. След време τ_3 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.



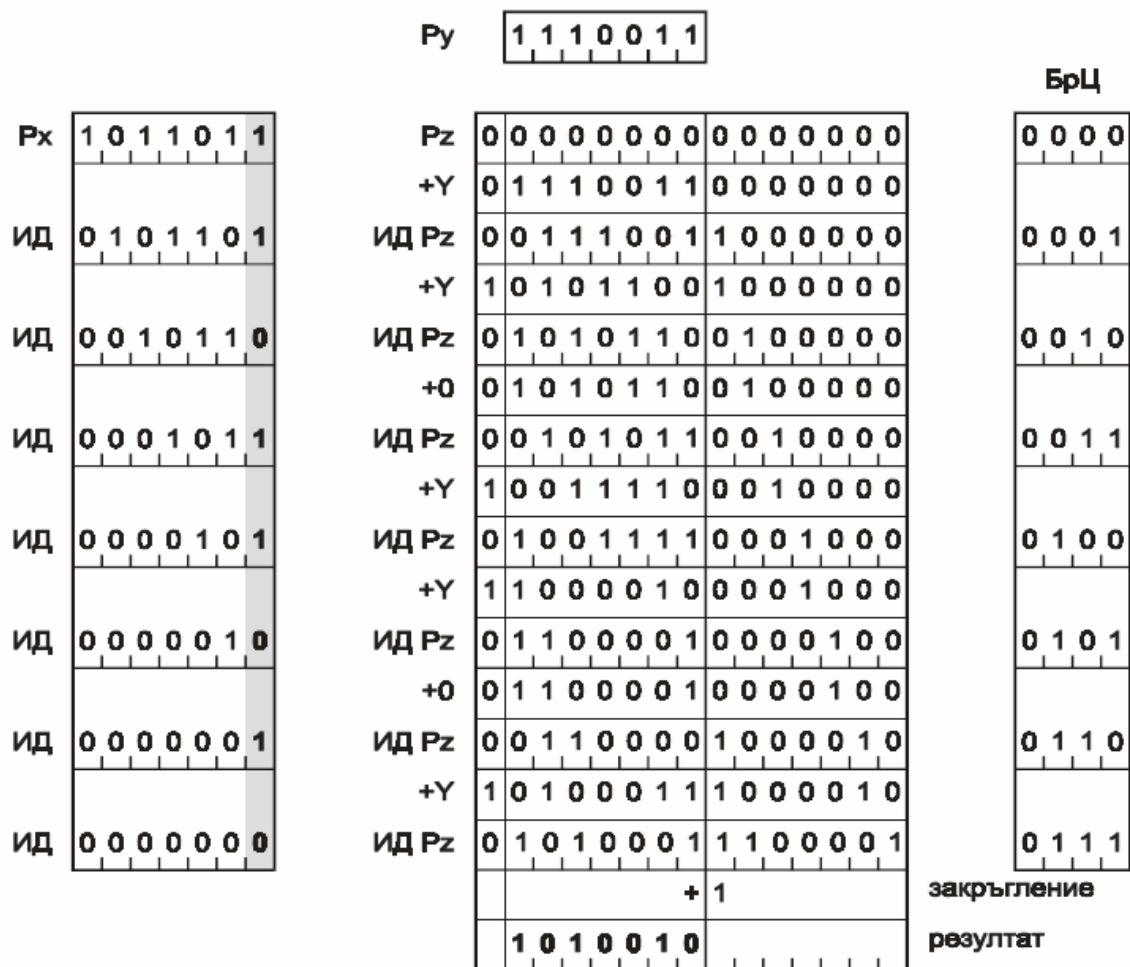
Фигура 1. Схема на операционен блок за умножение по метод А.

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 Σ с разрядност $(n+1)$, за да не се загуби съответно единицата на преноса, който евентуално би могъл да възникне при събиране на Π_1 и Yx_{n-1} и за да може да се извърши закръгляване на резултата чрез прибавяне на "1" в $(n+1)$ -вия разряд на $2n$ -разрядното произведение. Тъй като изместването в Pz' и Rx се осъществява в една и съща посока, при което старшите разряди на Pz' се запълват, а на Rx се освобождават, то вместо Pz' може да се използва Rx .

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

$$t_y A = n(\tau_1 + \tau_2 + \tau_3) = n(t_{\text{ПрК}} + t_{\Sigma} + t_{\text{ПК}} + t_{\text{ИК}}) = n(t_c + t_{\text{ИК}})$$



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 използва 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.

Логически функции

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

Функционална зависимост:

$Y = f(x_1, x_2, \dots, x_n)$, 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

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

Логическа функция **без променливи**: Y

Y	Наименование на състоянието
0	константа нула
1	константа едно

Логическа функция **с една променлива**: Y

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

Логическо отрицание (NO): $Y = \text{NO } X1 = \overline{X1}$

X	Y
0	1
1	0

Логически функции на **две променливи**:

Логическо сумиране (OR): $Y = X1 \text{ OR } X2 = X1 \vee X2 = (X1 + X2)$
(X1 + X2) този запис може да се използва само когато се работи с булевата алгебра, т.е. само с логически функции уравнения.

Логическо сумиране (OR): $Y = X1 \text{ OR } X2 = X1 \vee X2 = (X1 + X2)$

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

Логическо произведение (AND): $Y = X1 \text{ AND } X2 = X1 \& X2 = (X1 \cdot X2)$

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

Логическо сумиране с отрицание (NOR): $Y = \overline{X_1 \text{ OR } X_2} = \overline{X_1 \vee X_2}$

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

Логическо произведение с отрицание (NAND): $Y = \overline{X_1 \text{ AND } X_2} = \overline{X_1 \& X_2}$

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

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

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

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

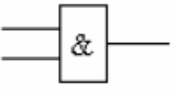
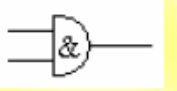
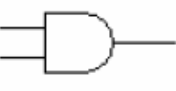
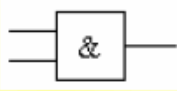
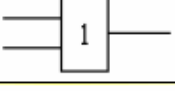




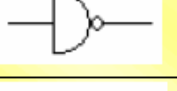



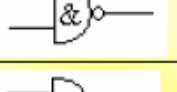


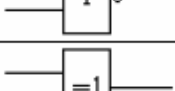
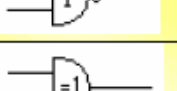

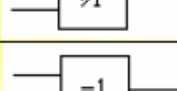
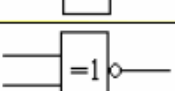
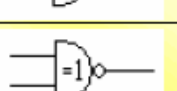






Базови компоненти.

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 с отрицание (NAND):
 $Y = \overline{X_1 \text{ AND } X_2} = \overline{X_1} \& \overline{X_2}$
- Логическо произведение (AND): $Y = X_1 \text{ AND } X_2 = X_1 \& X_2$
- Логическо сумиране (OR): $Y = X_1 \text{ OR } X_2 = X_1 \vee X_2$
- Логическо отрицание (NO): $Y = \text{NO } X_1 = \overline{X_1}$
- Сума по модул 2 (*mod* 2) (логическа не равнозначност)
 $Y = \overline{X_1} \overline{X_2} \vee X_1 X_2$
- Логическо сумиране с отрицание (NOR): $Y = \overline{X_1 \text{ OR } X_2} = \overline{X_1} \vee \overline{X_2}$
- Част от тези базисни функции са реализирани с три, четири или осем променливи.

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				
ИЗКЛЮЧАЩО ИЛИ				
ИЗКЛЮЧАЩО ИЛИ - НЕ				

Комбинационни логически схеми

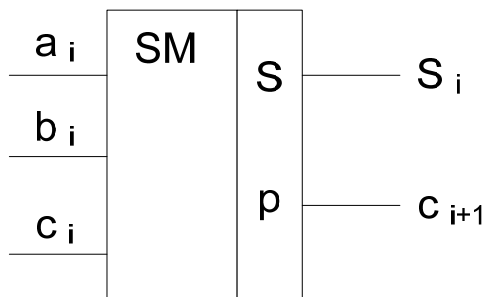
Суматори

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

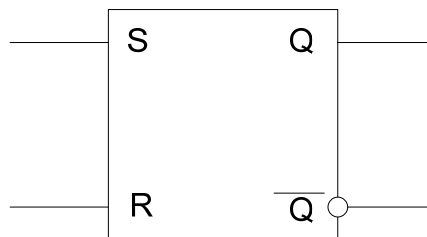
Тригери

Тригери ($\bar{R}-\bar{S}$ 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

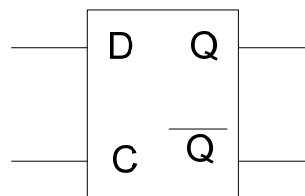


$\overline{R}-\overline{S}$ Тригер

\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	x	Q_t
\wedge	0	0
\wedge	1	1



T Тригер

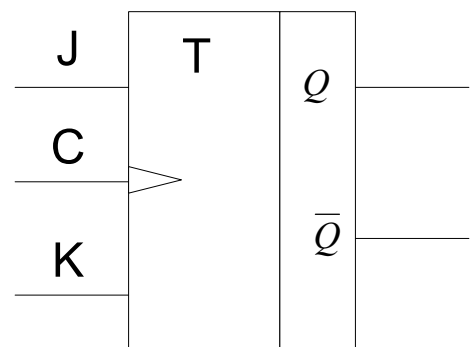
Тригер с броячен вход. На всеки тактов импулс си променя състоянието.

T	Q_{t+1}
\wedge	\overline{Q}_t

J-K Тригер

Вход J установява в 1, а K – в 0.

C	J	K	Q_{t+1}
\wedge	0	0	Q_t
\wedge	0	1	1
\wedge	1	0	0
\wedge	1	1	\overline{Q}_t
0	x	x	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.