

Министерство сельского хозяйства Российской Федерации
Департамент научно-технологической политики и образования
*Федеральное государственное бюджетное образовательное уч-
реждение высшего образования*
«Красноярский государственный аграрный университет»

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ

Методические указания к лабораторным работам

для студентов направления подготовки 09.03.03 «Прикладная
информатика»

Красноярск 2025

ББК 32.973.202Я7
УДК 681.324(075)

Рецензент

С.А. Бронов, д-р техн. наук, профессор кафедры «Информационные технологии и математическое обеспечение информационных систем»

Титовский, С.Н.

Вычислительные системы, сети и телекоммуникации: метод. указания к лабораторным работам для студентов подготовки 09.03.03 «Прикладная информатика» / Краснояр. гос. аграр.ун-т. – Красноярск, 2025. – 78 с.

Предназначена для студентов направлений подготовки 09.03.03 «Прикладная информатика». Содержит задания к лабораторным работам, примеры выполнения заданий и оформления отчетов.

Печатается по решению редакционно-издательского совета
Красноярского государственного аграрного университета

©Красноярский государственный
аграрный университет, 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ЛАБОРАТОРНАЯ РАБОТА № 1	6
ЛАБОРАТОРНАЯ РАБОТА № 2	9
ЛАБОРАТОРНАЯ РАБОТА № 3	16
ЛАБОРАТОРНАЯ РАБОТА № 4	25
ЛАБОРАТОРНАЯ РАБОТА № 5	30
ЛАБОРАТОРНАЯ РАБОТА № 6	37
ЛАБОРАТОРНАЯ РАБОТА № 7	44
ЛАБОРАТОРНАЯ РАБОТА № 8	48
ЛАБОРАТОРНАЯ РАБОТА № 9	53
ЛАБОРАТОРНАЯ РАБОТА № 10	59
ЗАКЛЮЧЕНИЕ	76
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	78

ВВЕДЕНИЕ

Вычислительные системы, сети и телекоммуникации являются в современном обществе самыми востребованными ресурсами. Войдя в человеческую жизнь, компьютеры сейчас стали неотъемлемой частью нашей цивилизации. И хотя первая ЭВМ с автоматическим программным управлением была создана чуть более полувека назад, к настоящему моменту уже насчитывается пять поколений вычислительных машин. Столь бурного развития, вероятно, не претерпевала ни одна технология.

Современная эпоха характеризуется стремительным процессом информатизации общества. Это сильнее всего проявляется в росте пропускной способности и гибкости информационных сетей. Полоса пропускания в расчете на одного пользователя стремительно увеличивается благодаря нескольким факторам. Логическое и физическое проектирование сети, ее нагрузка, пропускная способность и коэффициент использования сети очень важно при создании локальных сетей в больших масштабах. Во-первых, растет популярность приложений World Wide Web и количество электронных банков информации, которые становятся достоянием каждого человека. Падение цен на компьютеры приводит к росту числа домашних ПК, каждый из которых потенциально превращается в устройство, способное подключиться к сети Internet. Во-вторых, новые сетевые приложения становятся более требовательными в отношении полосы пропускания – входят в практику приложения Internet, ориентированные на мультимедиа и видеоконференцсвязь, когда одновременно открывается очень большое количество сессий передачи данных. Как результат, наблюдается резкий рост в потреблении ресурсов Internet – по оценкам средний объем потока информации в расчете на одного пользователя в мире увеличивается в 8 раз каждый год. Все это благодаря средствам телекоммуникаций.

Обучение студентов, будущих специалистов, принципам построения и функционирования вычислительных систем, организации и работы сетей ЭВМ является неотъемлемым требованием современности.

Студенты должны знать назначение и принципы построения и функционирования узлов и блоков вычислительных систем, их взаи-

модействия; способы организации сетей ЭВМ, назначение и функционирование компонент сетей

Представленное методическое указание содержит десять лабораторных работ, в ходе выполнения которых студенты изучат основные вопросы данной дисциплины. После перечня заданий к каждой лабораторной работе приводятся требования к содержанию отчетов, примеры выполнения заданий, которые одновременно являются примерами оформления отчетов.

ЛАБОРАТОРНАЯ РАБОТА № 1

Тема: Позиционные системы счисления. Перевод чисел.

Цель: Приобретение навыков перевода чисел между системами счисления с различными основаниями.

Задание:

1. Число A (табл. 1), заданное в десятичной системе счисления, перевести:

из десятичной системы в двоичную и обратно;

из десятичной системы в четверичную и обратно;

из десятичной системы в восьмеричную и обратно;

из десятичной системы в шестнадцатеричную и обратно.

Число разрядов в дробной части получающихся чисел:

для двоичной системы счисления – 10 разрядов;

для четверичной – 8 разрядов;

для восьмеричной – 6 разрядов;

для шестнадцатеричной – 4 разряда.

2. Число B (табл.1), заданное в десятичной системе счисления, перевести:

из десятичной системы счисления в двоичную (число разрядов в дробной части числа – 10);

из двоичной системы – в четверичную;

из двоичной – в восьмеричную;

из двоичной – в шестнадцатеричную.

Числа, полученные в двоичной, четверичной, восьмеричной и шестнадцатеричной системах счисления, перевести в десятичную систему.

Таблица 1

Номер варианта	A	B	Номер варианта	A	B
1	555,555	666,666	14	826,692	716,592
2	666,666	555,555	15	753,851	949,833
3	567,765	765,567	16	999,518	913,856

4	579,684	764,746	17	806,722	744,809
5	864,718	792,549	18	773,569	987,736
6	777,781	888,642	19	804,859	936,639
7	679,773	828,834	20	573,729	725,861
8	824,856	935,517	21	577,577	688,688
9	994,844	832,638	22	688,688	577,577
10	914,917	941,568	23	789,987	987,789
11	566,566	677,677	24	740,548	641,862
12	677,677	566,566	25	888,777	673,761
13	678,876	876,678			

Содержание отчета:

- ~ Тема работы
- ~ Условие задания
- ~ Методика перевода чисел из десятичной системы счисления в систему счисления с произвольным основанием и обратно
- ~ Методика перевода чисел между системами с кратными основаниями

Пример выполнения:

Позиционные системы счисления. Перевод чисел.

Задание:

1). Число $A_{(10)}=46,68$ перевести из десятичной системы в двоичную и обратно. Число разрядов в дробной части - 6 разрядов.

2). Число $B_{(2)}=11101,10011$ перевести в четверичную систему счисления. Числа из двоичной и четверичной систем счисления перевести в десятичную систему.

1) $A_{(10)}=46,68$

$$\begin{array}{r}
 46 \mid 2 \\
 \hline
 \overline{46} \mid \overline{23} \mid 2 \\
 \hline
 \mathbf{a}_0 = 0 \quad \overline{22} \mid \overline{11} \mid 2 \\
 \hline
 \mathbf{a}_1 = 1 \quad \overline{10} \mid \overline{5} \mid 2 \\
 \hline
 \mathbf{a}_2 = 1 \quad \overline{4} \mid \overline{2} \mid 2 \\
 \hline
 \mathbf{a}_3 = 1 \quad \overline{2} \mid \overline{1} = \mathbf{a}_5 \\
 \hline
 \mathbf{a}_4 = 0
 \end{array}$$

$$\begin{array}{r}
 0, \mid 68 \\
 \hline
 \mid 2 \\
 \hline
 \mathbf{a}_{-1} = 1, \mid 36 \\
 \hline
 \mid 2 \\
 \hline
 \mathbf{a}_{-2} = 0, \mid 72 \\
 \hline
 \mid 2 \\
 \hline
 \mathbf{a}_{-3} = 1, \mid 44 \\
 \hline
 \mid 2 \\
 \hline
 \mathbf{a}_{-4} = 0, \mid 88 \\
 \hline
 \mid 2 \\
 \hline
 \mathbf{a}_{-5} = 1, \mid 76 \\
 \hline
 \mid 2 \\
 \hline
 \mathbf{a}_{-6} = 1, \mid 52
 \end{array}$$

$$A_{(2)} = 101110,101011.$$

$$\begin{aligned}
 A_{(10)} &= 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + \\
 &0 \cdot 2^{-4} + 1 \cdot 2^{-5} + 1 \cdot 2^{-6} = 32 + 8 + 4 + 2 + 0,5 + 0,125 + 0,03125 + 0,015625 = \\
 &46,671875.
 \end{aligned}$$

$$2) \ B_{(2)} = \underset{\substack{1 \quad 3 \quad 1 \quad 2 \quad 1 \quad 2}}{011101,100110}, \quad B_{(4)} = 131,212$$

$$\begin{aligned}
 B_{(10)} &= 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5} = 16 + 8 + 4 + 1 \\
 &+ 0,5 + 0,0625 + 0,03125 = 29,59375.
 \end{aligned}$$

$$\begin{aligned}
 B_{(10)} &= 1 \cdot 4^2 + 3 \cdot 4^1 + 1 \cdot 4^0 + 2 \cdot 4^{-1} + 1 \cdot 4^{-2} + 2 \cdot 4^{-3} = 16 + 12 + 1 + 0,5 + \\
 &0,0625 + 0,03125 = 29,59375.
 \end{aligned}$$

ЛАБОРАТОРНАЯ РАБОТА № 2

Тема: Позиционные системы счисления. Арифметика.

Цель: Приобретение навыков выполнения арифметических действий с числами в позиционных системах счисления с произвольными основаниями.

Задание:

1. Числа A и B (табл. 2), заданные в десятичной системе счисления, перевести в
 - ~ двоичную (в дробной части 8 разрядов),
 - ~ четверичную (в дробной части 4 разряда),
 - ~ восьмеричную (в дробной части 3 разряда),
 - ~ шестнадцатеричную (в дробной части 2 разряда)системы счисления.
2. Выполнить сложение $A+B$ в десятичной, двоичной, четверичной, восьмеричной и шестнадцатеричной системах счисления. Полученные результаты из двоичной, четверичной, восьмеричной и шестнадцатеричной систем счисления перевести в десятичную систему счисления.
3. Выполнить вычитание $A-B$ в десятичной, двоичной, четверичной, восьмеричной и шестнадцатеричной системах счисления. Полученные результаты из двоичной, четверичной, восьмеричной и шестнадцатеричной систем счисления перевести в десятичную систему счисления.
4. Выполнить умножение $A*B$ в десятичной, двоичной, четверичной, восьмеричной и шестнадцатеричной системах счисления. Полученные результаты из двоичной, четверичной, восьмеричной и шестнадцатеричной систем счисления перевести в десятичную систему счисления.
5. Выполнить деление A/B в десятичной, двоичной, четверичной, восьмеричной и шестнадцатеричной системах счисления. В дробной части частного получить
 - ~ для двоичной системы счисления – 8 разрядов
 - ~ для четверичной системы счисления – 4 разряда
 - ~ для восьмеричной системы счисления – 3 разряда
 - ~ для шестнадцатеричной системы счисления – 2 разряда.

Полученные результаты из двоичной, четверичной, восьмеричной и шестнадцатеричной систем счисления перевести в десятичную систему счисления.

Таблица 2

Номер варианта	<i>A</i>	<i>B</i>	Номер варианта	<i>A</i>	<i>B</i>
1	555,55	23,45	14	826,69	12,98
2	666,66	35,45	15	753,85	13,56
3	567,76	34,23	16	999,51	16,78
4	579,68	32,54	17	806,72	27,57
5	864,71	32,34	18	773,56	29,34
6	777,78	33,76	19	804,85	26,23
7	679,77	43,67	20	573,72	24,78
8	824,85	34,87	21	577,57	28,56
9	994,84	41,57	22	688,68	25,34
10	914,91	44,67	23	789,98	27,23
11	566,56	31,45	24	740,54	26,98
12	677,67	24,67	25	888,77	29,99
13	678,87	34,76			

Содержание отчета:

- ~ Тема работы
- ~ Условие задания
- ~ Методика сложения чисел в позиционных системах счисления с произвольным основанием.
- ~ Методика вычитания чисел в позиционных системах счисления с произвольным основанием.
- ~ Методика умножения чисел в позиционных системах счисления с произвольным основанием
- ~ Методика деления чисел в позиционных системах счисления с произвольным основанием.

Пример выполнения:

Позиционные системы счисления. Арифметика.

Задание:

1. Числа $A_{(10)}=46,68$ и $B_{(10)}=9,6$ перевести в
 - ~ двоичную (в дробной части 4 разряда),
 - ~ четверичную (в дробной части 3 разряда),
 - ~ восьмеричную (в дробной части 2 разряда),
 - ~ шестнадцатеричную (в дробной части 1 разряд)системы счисления.
2. Выполнить сложение $A+B$ в десятичной, двоичной, четверичной, восьмеричной и шестнадцатеричной системах счисления. Полученные результаты из двоичной, четверичной, восьмеричной и шестнадцатеричной систем счисления перевести в десятичную систему счисления.
3. Выполнить вычитание $A-B$ в десятичной, двоичной, четверичной, восьмеричной и шестнадцатеричной системах счисления. Полученные результаты из двоичной, четверичной, восьмеричной и шестнадцатеричной систем счисления перевести в десятичную систему счисления.
4. Выполнить умножение $A*B$ в десятичной, двоичной, четверичной, восьмеричной и шестнадцатеричной системах счисления. Полученные результаты из двоичной, четверичной,

5. Выполнить деление **A/B** в десятичной, двоичной, четверичной, восьмеричной и шестнадцатеричной системах счисления.

- ~ для двоичной системы счисления – 4 разряда
- ~ для четверичной системы счисления – 3 разряда
- ~ для восьмеричной системы счисления – 2 разряда
- ~ для шестнадцатеричной системы счисления – 1 разряд.

1.

$A_{(10)}=46,68$

46	2							68		
0	23	2					1	36		
	1	11	2					2		
		1	5	2			0	72		
			1	2	2				2	
				0	1	2			1	44
					1	0				2
									0	88
									1	76
										2
									1	52

$B_{(10)}=9,6$

9	2							6
1	4	2					1	2
	0	2	2					2
		0	1	2			0	4
			1	0				2
							0	8
								2
							1	6
								2
							1	2
								2
							0	4

$$A=46,68_{(10)}=101110,1010_{(2)}=232,223_{(4)}=56,53_{(8)}=2E, A_{(16)}$$

$$2. A + B = 46,68_{(10)} + 9,6_{(10)} = 56,28_{(10)}$$

$$+ \begin{array}{cccccccccc} & 1 & 1 & 1 & 1 & 1 & & & & \\ 1 & 0 & 1 & 1 & 1 & 0, & 1 & 0 & 1 & 0_{(2)} \\ & & 1 & 0 & 0 & 1, & 1 & 0 & 0 & 1_{(2)} \\ \hline 1 & 1 & 1 & 0 & 0 & 0, & 0 & 0 & 1 & 1_{(2)} \end{array}$$

$$+ \begin{array}{cccccc} 1 & 1 & 1 & 1 & 1 & \\ 2 & 3 & 2, & 2 & 2 & 3_{(4)} \\ & 2 & 1, & 2 & 1 & 2_{(4)} \\ \hline 3 & 2 & 0, & 1 & 0 & 1_{(4)} \end{array}$$

$$+ \begin{array}{cccc} 1 & 1 & 1 & \\ 5 & 6, & 5 & 3_{(8)} \\ 1 & 1, & 4 & 6_{(8)} \\ \hline 7 & 0, & 2 & 1_{(8)} \end{array}$$

$$+ \begin{array}{ccc} 1 & 1 & \\ 2 & E, & A_{(16)} \\ & 9, & 9_{(16)} \\ \hline 3 & 8, & 3_{(16)} \end{array}$$

$$111000,0011_{(2)} = 2^5 + 2^4 + 2^3 + 2^{-3} + 2^{-4} = 56,1875_{(10)}$$

$$320,101_{(4)} = 3 \cdot 4^2 + 2 \cdot 4^1 + 4^{-1} + 4^{-3} = 56,265625_{(10)}$$

$$70,21_{(8)} = 7 \cdot 8^1 + 2 \cdot 8^{-1} + 8^{-2} = 56,265625_{(10)}$$

$$38,3_{(16)} = 3 \cdot 16^1 + 8 + 3 \cdot 16^{-1} = 56,1875_{(10)}$$

$$3. A - B = 46,68_{(10)} - 9,6_{(10)} = 37,08_{(10)}$$

$$- \begin{array}{cccccccccc} & & & & & & & & 2 & \\ 1 & 0 & 1 & 1 & 1 & 0, & 1 & 0 & 1 & 0_{(2)} \\ & & 1 & 0 & 0 & 1, & 1 & 0 & 0 & 1_{(2)} \\ \hline 1 & 0 & 0 & 1 & 0 & 1, & 0 & 0 & 0 & 1_{(2)} \end{array}$$

$$- \begin{array}{cccccc} 2 & 3 & 2, & 2 & 2 & 3_{(4)} \\ & 2 & 1, & 2 & 1 & 2_{(4)} \\ \hline 2 & 1 & 1, & 0 & 1 & 1_{(4)} \end{array}$$

$$- \begin{array}{cccc} 8 & & & \\ 5 & 6, & 5 & 3_{(8)} \\ 1 & 1, & 4 & 6_{(8)} \\ \hline 4 & 5, & 0 & 5_{(8)} \end{array}$$

$$- \begin{array}{ccc} 2 & E, & A_{(16)} \\ & 9, & 9_{(16)} \\ \hline 2 & 5, & 1_{(16)} \end{array}$$

$$100101,0001_{(2)} = 2^5 + 2^2 + 2 + 2^{-4} = 37,0625_{(10)}$$

$$211,011_{(4)} = 2 \cdot 4^2 + 4^1 + 1 + 4^{-2} + 4^{-3} = 37,078125_{(10)}$$

$$45,05_{(8)} = 4 \cdot 8^1 + 5 \cdot 8^{-1} + 5 \cdot 8^{-2} = 37,078125_{(10)}$$

$$25,1_{(16)} = 2 \cdot 16^1 + 5 + 16^{-1} = 37,0625_{(10)}$$

4. $A * B = 46,68_{(10)} * 9,6_{(10)} = 448,128_{(10)}$

$$\begin{array}{cccccccccccccc}
 & & & & & & x & 1 & 0 & 1 & 1 & 1 & 0, & 1 & 0 & 1 & 0_{(2)} \\
 & & & & & & & & & & 1 & 0 & 0 & 1, & 1 & 0 & 0 & 1_{(2)} \\
 & & & & & & & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
 & & & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & & & \\
 & & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & & & & \\
 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & & & & & & & \\
 \hline
 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1, & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0_{(2)}
 \end{array}$$

$$\begin{array}{cccccccc}
 & & & & & x & 2 & 3 & 2, & 2 & 2 & 3_{(4)} \\
 & & & & & & & 2 & 1, & 2 & 1 & 2_{(4)} \\
 & & & & & 1 & 1 & 3 & 1 & 1 & 1 & 2 \\
 & & & & & 2 & 3 & 2 & 2 & 2 & 3 & \\
 & & 1 & 1 & 3 & 1 & 1 & 1 & 2 & & & \\
 & & 2 & 3 & 2 & 2 & 2 & 3 & & & & \\
 1 & 1 & 3 & 1 & 1 & 1 & 2 & & & & & \\
 \hline
 1 & 2 & 3 & 3 & 3, & 3 & 0 & 0 & 2 & 0 & 2_{(4)}
 \end{array}$$

		x	5	6,	5	3 ₍₈₎
			1	1,	4	6 ₍₈₎
		4	3	0,	0	2
2	7	2	5	4		
5	6	5	3			
5	6	5	3			
6	7	7,	6	0	4	2 ₍₈₎

	x	2	E,	A ₍₁₆₎
			9,	9 ₍₁₆₎
	1	A	3	A
1	A	3	A	
1	B	D,	D	A ₍₁₆₎

$$110111101,11011010_{(2)} = 2^8 + 2^7 + 2^5 + 2^4 + 2^3 + 2^2 + 1 + 2^{-1} + 2^{-2} + 2^{-4} + 2^{-5} + 2^{-7} = 445,8515625_{(10)}$$

$$12333,300202_{(4)} = 4^4 + 2*4^3 + 3*4^2 + 3*4^1 + 3 + 3*4^{-1} + 2*4^{-4} + 2*4^{-6} = 447,7583_{(10)}$$

$$677,6042_{(8)} = 6 \cdot 8^2 + 7 \cdot 8^1 + 7 + 6 \cdot 8^{-1} + 4 \cdot 8^{-3} + 2 \cdot 8^{-4} = 447,7583_{(10)}$$

$$1BD,DA_{(16)} = 16^2 + 11*16^1 + 13 + 13*16^{-1} + 10*16^{-2} = 445,8515625_{(10)}$$

5. $A / B = 46,68_{(10)} / 9,6_{(10)} = 4,8625_{(10)}$

1	0	1	1	1	0	1	0	1	0 ₍₂₎	1	0	0	1	1	0	0	1 ₍₂₎
1	0	0	1	1	0	0	1										
			1	0	0	0	0	1	1	0	0						
			1	0	0	1	1	0	0	1							
			1	1	1	0	0	1	1	0							
			1	0	0	1	1	0	0	1							
			1	0	0	1	1	0	1	0							
			1	0	0	1	1	0	1	0	1	0					
			1	0	0	1	1	0	0	1							
											1	0					

$$\begin{array}{r}
 \begin{array}{cccccc|cccc}
 2 & 3 & 2 & 2 & 2 & 3_{(4)} & 2 & 1 & 2 & 1 & 2_{(4)} \\
 2 & 1 & 2 & 1 & 2 & & 1 & 0 & 3 & 1 & 3_{(4)} \\
 \hline
 & 2 & 0 & 1 & 0 & 3 & 0 & & & & \\
 & 1 & 3 & 0 & 3 & 0 & 2 & & & & \\
 \hline
 & & 1 & 0 & 1 & 2 & 2 & 0 & & & \\
 & & & 2 & 1 & 2 & 1 & 2 & & & \\
 \hline
 & & & 2 & 0 & 0 & 0 & 2 & 0 & & \\
 & & & 1 & 3 & 0 & 3 & 0 & 2 & & \\
 \hline
 & & & & 3 & 1 & 1 & 2 & & &
 \end{array}
 \end{array}$$

$$\begin{array}{cccc|cccc} 5 & 6 & 5 & 3_{(8)} & 1 & 1 & 4 & 6_{(8)} \\ 4 & 6 & 3 & 0 & 4, & 6 & 7_{(8)} & \\ 1 & 0 & 2 & 3 & 0 & & & \\ \hline & 7 & 1 & 4 & 4 & & & \\ & 1 & 0 & 6 & 4 & 0 & & \\ & 1 & 0 & 3 & 1 & 2 & & \\ \hline & & & 3 & 2 & 6 & & \end{array}$$

$$\begin{array}{r|l} 2 & \text{E} \quad \text{A}_{(16)} \\ 2 & 6 \quad 4 \\ & 8 \quad 6 \\ & 8 \quad 5 \\ \hline & 2 \end{array} \quad \begin{array}{l} 9 \quad 9_{(16)} \\ 4, \quad \text{E}_{(16)} \\ 0 \\ \text{E} \\ 2 \end{array}$$

$$100,1110_{(2)} = 2^2 + 2^{-1} + 2^{-2} + 2^{-3} = 4,875_{(10)}$$

$$10,313_{(4)} = 4^1 + 3 \cdot 4^{-1} + 1 \cdot 4^{-2} + 3 \cdot 4^{-3} = 4,859375_{(10)}$$

$$4,67_{(8)} = 4 + 6 \cdot 8^{-1} + 7 \cdot 8^{-2} = 4,859375_{(10)}$$

$$4, E_{(16)} = 4 + 14 \cdot 16^{-1} = 4,875_{(10)}$$

ЛАБОРАТОРНАЯ РАБОТА № 3

Тема: Аксиомы и свойства алгебры логики.

Цель: Приобретение навыков тождественных преобразований функций алгебры логики.

Задание:

1. Пользуясь аксиомами и свойствами алгебры логики, упростить выражение:

$$1. F = x_1 \cdot x_2 \cdot x_4 + x_1 \cdot x_2 + x_2 \cdot x_3 \cdot \overline{x_2} \cdot x_4 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 + (x_2 + x_1 \cdot x_2 \cdot x_3) \cdot (\overline{x_2} \cdot \overline{x_3} + \overline{x_2} \cdot x_3 \cdot \overline{x_4} + \overline{x_2} \cdot \overline{x_3} + \overline{x_2} \cdot x_3 \cdot x_4) + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_3 \cdot \overline{x_4} + \overline{x_1} \cdot \overline{x_3}.$$

$$2. F = x_2 \cdot x_4 \cdot x_5 + x_2 \cdot x_3 + x_2 \cdot \overline{x_3} \cdot x_4 + (x_2 \cdot x_3 \cdot x_4 + x_3) \cdot (\overline{x_3} \cdot x_4 \cdot \overline{x_5} + \overline{x_3} \cdot \overline{x_4} + \overline{x_3} \cdot \overline{x_4} + \overline{x_3} \cdot x_4 \cdot x_5) + x_2 \cdot x_3 + x_2 \cdot \overline{x_3} \cdot \overline{x_4} + \overline{x_1} \cdot x_2 \cdot x_1 \cdot x_3 + x_2 \cdot \overline{x_4} + x_2 \cdot x_4 \cdot \overline{x_5}.$$

$$3. F = \overline{x_1} \cdot x_2 + x_1 \cdot x_3 + x_1 \cdot x_2 \cdot x_3 \cdot \overline{x_2} + x_1 \cdot x_2 \cdot x_4 + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_4} + \overline{x_1} \cdot \overline{x_2} \cdot x_3 + (\overline{x_1} \cdot \overline{x_2} + \overline{x_1} \cdot x_2 \cdot x_3 + \overline{x_1} \cdot x_2 \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2}) \cdot (x_1 + x_1 \cdot x_3 \cdot x_4).$$

$$4. F = \overline{x_2} \cdot x_3 \cdot x_4 + x_1 \cdot x_2 \cdot x_5 + x_3 \cdot x_1 \cdot \overline{x_4} \cdot \overline{x_1} + x_2 \cdot x_4 + \overline{x_2} \cdot \overline{x_3} \cdot x_4 + x_2 \cdot x_4 + (x_1 \cdot x_2 \cdot x_3 + x_1) \cdot (\overline{x_1} \cdot \overline{x_3} + \overline{x_1} \cdot x_3 \cdot x_4 + \overline{x_1} \cdot x_3 \cdot \overline{x_4} + \overline{x_1} \cdot \overline{x_3}) + \overline{x_1} \cdot \overline{x_2} + x_1 \cdot x_2 \cdot \overline{x_5}.$$

$$5. F = x_2 \cdot \overline{x_3} + x_4 \cdot x_5 + (x_1 + x_1 \cdot x_2 \cdot x_5) \cdot (\overline{x_1} \cdot x_4 \cdot x_5 + \overline{x_1} \cdot x_4 \cdot \overline{x_5} + \overline{x_1} \cdot \overline{x_4} + \overline{x_1} \cdot \overline{x_4}) + x_4 \cdot x_5 + x_2 \cdot \overline{x_4} \cdot x_5 + x_2 \cdot x_3 \cdot x_4 + \overline{x_1} \cdot x_2 \cdot x_1 \cdot \overline{x_4} \cdot \overline{x_5} + \overline{x_2} \cdot \overline{x_4} \cdot x_5 + x_2 \cdot x_3 \cdot \overline{x_5}.$$

$$6. F = \overline{x_1} \cdot x_2 + \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_3 \cdot x_4 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} \cdot x_2 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot x_2 + \overline{x_1} \cdot \overline{x_3} + (\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_2}) \cdot (x_2 \cdot x_3 \cdot x_4 + x_2 \cdot \overline{x_3} + x_2 \cdot x_3 \cdot \overline{x_4}) + \overline{x_1} \cdot x_3 \cdot \overline{x_4}.$$

$$7. F = x_2 \cdot x_4 + \overline{x_2} \cdot x_3 + \overline{x_2} \cdot x_4 \cdot x_5 + x_1 \cdot \overline{x_3} \cdot x_4 \cdot \overline{x_1} + \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + (x_3 \cdot x_4 \cdot x_5 + x_3 \cdot \overline{x_4} + x_3 \cdot \overline{x_4} + x_3 \cdot x_4 \cdot \overline{x_5}) \cdot (\overline{x_3} + x_2 \cdot \overline{x_3} \cdot x_4) + \overline{x_2} \cdot x_3 + \overline{x_2} \cdot \overline{x_3} \cdot x_4 + \overline{x_2} \cdot x_4 \cdot \overline{x_5}.$$

$$8. F = x_1 \cdot x_2 + (x_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} + x_1 \cdot \overline{x_2}) \cdot (\overline{x_1} + \overline{x_1} \cdot x_3 \cdot x_4 \cdot x_5) + x_1 \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot x_4 + \overline{x_1} \cdot \overline{x_3} \cdot \overline{x_4} \cdot x_1 \cdot x_5 + x_1 \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot \overline{x_4} + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}.$$

$$9. F = x1 \cdot \overline{x2} \cdot \overline{x4} \cdot x2 \cdot x5 + \overline{x1} \cdot \overline{x2} \cdot \overline{x5} + \overline{x2} \cdot \overline{x3} \cdot \overline{x4} + x2 \cdot \overline{x4} + (x1 + x1 \cdot \overline{x2} \cdot \overline{x3}) \cdot (\overline{x1} \cdot \overline{x3} \cdot \overline{x4} + \overline{x1} \cdot x3 + \overline{x1} \cdot \overline{x3} \cdot x4 + \overline{x1} \cdot x3) + x2 \cdot \overline{x4} + \overline{x2} \cdot x3 \cdot \overline{x4} + \overline{x1} \cdot \overline{x2} \cdot x5 + \overline{x1} \cdot \overline{x2}.$$

$$10. F = \overline{x2} \cdot x3 + x4 \cdot \overline{x5} + x1 \cdot \overline{x4} \cdot \overline{x5} + \overline{x2} \cdot \overline{x3} \cdot \overline{x5} + x4 \cdot \overline{x5} + x2 \cdot x3 \cdot \overline{x3} \cdot \overline{x2} + (x3 + x2 \cdot x3 \cdot x4) \cdot (\overline{x2} \cdot x3 \cdot \overline{x4} + x3 \cdot x4 + x2 \cdot x3 \cdot \overline{x4} + x3 \cdot x4) + \overline{x1} \cdot \overline{x4} \cdot \overline{x5} + \overline{x2} \cdot \overline{x3} \cdot x5.$$

$$11. F = (x1 \cdot \overline{x2} \cdot x3 + x1 \cdot x2 + x1 \cdot \overline{x2} \cdot \overline{x3} + x1 \cdot x2) \cdot (x3 + x2 \cdot x3 \cdot x4) \cdot (\overline{x3} \cdot x4 \cdot x5 + \overline{x3} \cdot \overline{x4} + \overline{x3} \cdot x4 \cdot \overline{x5} + \overline{x3} \cdot \overline{x4}) + x1 \cdot x3 \cdot x4 + x2 \cdot x3 \cdot \overline{x2} \cdot \overline{x4} + x1 \cdot x3 \cdot \overline{x4} + \overline{x1} \cdot x3.$$

$$12. F = (x2 \cdot x4 \cdot x5 + \overline{x1} \cdot x2 \cdot x1 \cdot x3 + x2 \cdot \overline{x4} + x2 \cdot x4 \cdot \overline{x5}) \cdot (\overline{x2} \cdot x3 \cdot x4 + \overline{x2} \cdot \overline{x3} + \overline{x2} \cdot x3 \cdot \overline{x4} + \overline{x2} \cdot \overline{x3}) \cdot (x2 + x1 \cdot x2 \cdot \overline{x3}) + x1 \cdot x2 + x1 \cdot \overline{x2} \cdot x3 + x1 \cdot \overline{x2} \cdot \overline{x3} + x1 \cdot x2.$$

$$13. F = x1 \cdot \overline{x2} + x1 \cdot x2 \cdot x4 + x1 \cdot x2 \cdot \overline{x3} \cdot \overline{x2} + x1 \cdot x3 + \overline{x1} \cdot x2 \cdot x3 + (\overline{x1} \cdot x2 \cdot x3 + \overline{x1} \cdot \overline{x2} + \overline{x1} \cdot x2 \cdot \overline{x3} + x1 \cdot \overline{x2}) \cdot (x3 \cdot x1 \cdot x4 + x1) + x1 \cdot x3 + x1 \cdot x2 \cdot \overline{x4} + \overline{x1} \cdot \overline{x2} \cdot x3.$$

$$14. F = x1 \cdot \overline{x2} + x2 \cdot x4 + x1 \cdot x2 \cdot x5 + \overline{x1} \cdot x3 \cdot \overline{x4} \cdot x1 + (x2 \cdot x1 \cdot x3 + x1) \cdot (\overline{x1} \cdot x3 \cdot x4 + \overline{x1} \cdot \overline{x3} + \overline{x1} \cdot x3 \cdot \overline{x4} + \overline{x1} \cdot \overline{x3}) + x2 \cdot x4 + \overline{x2} \cdot \overline{x3} \cdot x4 + x1 \cdot x2 \cdot \overline{x5} + \overline{x2} \cdot x3 \cdot x4.$$

$$15. F = (x2 \cdot \overline{x3} + x4 \cdot x5) \cdot (x1 + x1 \cdot x2 \cdot x5) \cdot (x4 \cdot x5 + x2 \cdot \overline{x4} \cdot x5 + x2 \cdot x3 \cdot x5 + \overline{x1} \cdot x2 \cdot x1 \cdot \overline{x4} \cdot \overline{x5} + \overline{x2} \cdot \overline{x4} \cdot x5 + x2 \cdot x3 \cdot \overline{x5}) \cdot (\overline{x1} \cdot x4 \cdot x5 + \overline{x1} \cdot x4 \cdot \overline{x5} + \overline{x1} \cdot \overline{x4} + \overline{x1} \cdot \overline{x4}).$$

$$16. F = x1 \cdot x3 + \overline{x1} \cdot x2 + \overline{x1} \cdot \overline{x2} \cdot x3 + \overline{x1} \cdot x3 \cdot x4 + \overline{x1} \cdot x2 + \overline{x1} \cdot \overline{x2} \cdot \overline{x3} + (\overline{x1} \cdot \overline{x2} \cdot \overline{x3} + \overline{x2}) \cdot (x2 \cdot \overline{x3} + x2 \cdot x3 \cdot x4 + x2 \cdot x3 \cdot \overline{x4} + x2 \cdot \overline{x3}) + \overline{x1} \cdot x3 \cdot \overline{x4} + x1 \cdot \overline{x2} \cdot \overline{x3} \cdot \overline{x4} \cdot x2.$$

$$17. F = \overline{x2} \cdot \overline{x4} + \overline{x2} \cdot x4 \cdot \overline{x5} + \overline{x2} \cdot x3 + \overline{x2} \cdot \overline{x3} \cdot x4 + \overline{x2} \cdot x3 + x1 \cdot \overline{x3} \cdot x4 \cdot \overline{x1} + \overline{x2} \cdot x4 \cdot x5 + \overline{x2} \cdot \overline{x3} \cdot \overline{x4} + (x2 \cdot \overline{x3} \cdot x4 + \overline{x3}) \cdot (x3 \cdot x4 \cdot x5 + x3 \cdot \overline{x4} + x3 \cdot x4 \cdot \overline{x5} + x3 \cdot \overline{x4}).$$

$$18. F = (x1 \cdot x2 + x1 \cdot \overline{x3} + \overline{x1} \cdot x2 \cdot \overline{x3} + \overline{x1} \cdot x2 \cdot x4) \cdot (\overline{x1} \cdot \overline{x3} \cdot \overline{x4} \cdot x1 \cdot x5 + x1 \cdot \overline{x3} + \overline{x1} \cdot x2 \cdot \overline{x4} + \overline{x1} \cdot \overline{x2} \cdot \overline{x3}) \cdot (x1 \cdot x2 \cdot x3 + x1 \cdot x2 \cdot \overline{x3} + x1 \cdot \overline{x2} + x1 \cdot \overline{x2}) \cdot (\overline{x1} + x3 \cdot x4 \cdot \overline{x1} \cdot x5).$$

$$19. F = \overline{x1} \cdot x2 + x1 \cdot \overline{x2} \cdot \overline{x4} \cdot x2 \cdot x5 + \overline{x1} \cdot \overline{x2} \cdot \overline{x5} + x2 \cdot \overline{x4} + \overline{x2} \cdot x3 \cdot \overline{x4} + \overline{x1} \cdot \overline{x2} \cdot x5 + x2 \cdot \overline{x4} + \overline{x2} \cdot \overline{x3} \cdot \overline{x4} + (\overline{x2} \cdot x1 \cdot x3 + x1) \cdot (\overline{x1} \cdot x3 + \overline{x1} \cdot \overline{x3} \cdot \overline{x4} + \overline{x1} \cdot \overline{x3} \cdot x4 + \overline{x1} \cdot x3).$$

$$20. F = \overline{\overline{x2} \cdot \overline{x3}} + x4 \cdot \overline{x5} + x1 \cdot \overline{x4} \cdot \overline{x5} + \overline{x2} \cdot \overline{x3} \cdot \overline{x5} + (x4 \cdot x3 \cdot \overline{x5} + x3) \cdot (\overline{x3} \cdot x4 + \overline{x2} \cdot \overline{x3} \cdot \overline{x4} + x2 \cdot \overline{x3} \cdot \overline{x4} + \overline{x3} \cdot x4) + x4 \cdot \overline{x5} + \overline{x1} \cdot \overline{x4} \cdot \overline{x5} + x2 \cdot x3 \cdot \overline{x3} \cdot \overline{x2} + \overline{x2} \cdot \overline{x3} \cdot x5 .$$

$$21. F = \overline{x1} \cdot x3 + x1 \cdot x2 + x1 \cdot \overline{x2} \cdot x3 + x1 \cdot x3 \cdot x4 + x2 \cdot x3 \cdot x4 \cdot \overline{x2} + x1 \cdot x2 + x1 \cdot \overline{x2} \cdot \overline{x3} + (\overline{x2} \cdot x3 \cdot x4 + \overline{x2} \cdot \overline{x3} + \overline{x2} \cdot x3 \cdot \overline{x4} + \overline{x2} \cdot \overline{x3}) \cdot (x1 \cdot x2 \cdot \overline{x3} + x2) + x1 \cdot x3 \cdot \overline{x4} .$$

$$22. F = \overline{x2} \cdot x4 + x2 \cdot x3 + (x3 + x1 \cdot \overline{x3} \cdot x5) \cdot (\overline{x3} \cdot \overline{x4} + \overline{x3} \cdot x4 \cdot x5 + \overline{x3} \cdot x4 \cdot \overline{x5} + \overline{x3} \cdot \overline{x4}) + x2 \cdot x3 + x2 \cdot \overline{x3} \cdot x4 + x2 \cdot x4 \cdot \overline{x5} + x1 \cdot x2 \cdot \overline{x1} \cdot x3 + x2 \cdot x4 \cdot x5 + x2 \cdot \overline{x3} \cdot \overline{x4} .$$

$$23. F = x1 \cdot x2 \cdot x3 \cdot \overline{x2} + x1 \cdot x2 \cdot x4 + x1 \cdot x3 + \overline{x1} \cdot \overline{x2} \cdot x3 + (x1 + x1 \cdot x3 \cdot x4) \cdot (\overline{x1} \cdot \overline{x2} + \overline{x1} \cdot x2 \cdot x3 + \overline{x1} \cdot x2 \cdot \overline{x3} + \overline{x1} \cdot \overline{x2}) + \overline{x1} \cdot \overline{x2} + x1 \cdot x3 + x1 \cdot x2 \cdot \overline{x4} + \overline{x1} \cdot x2 \cdot x3 .$$

$$24. F = \overline{x2} \cdot \overline{x3} \cdot x4 + x2 \cdot x4 + x1 \cdot x2 \cdot \overline{x5} + (x1 \cdot x2 \cdot x3 + x1) \cdot (\overline{x1} \cdot \overline{x3} + \overline{x1} \cdot x3 \cdot x4 + \overline{x1} \cdot x3 \cdot \overline{x4} + \overline{x1} \cdot x3) + \overline{x1} \cdot \overline{x2} + \overline{x2} \cdot x3 \cdot x4 + x1 \cdot x2 \cdot x5 + x3 \cdot x1 \cdot \overline{x4} \cdot \overline{x1} + x2 \cdot x4 .$$

$$25. F = x2 \cdot x3 \cdot x4 + x2 \cdot \overline{x3} + x4 \cdot x5 + (x1 + x1 \cdot x2 \cdot x5) \cdot (\overline{x1} \cdot x4 \cdot x5 + \overline{x1} \cdot x4 \cdot \overline{x5} + \overline{x1} \cdot \overline{x4} + \overline{x1} \cdot \overline{x4}) \cdot (x1 + x1 \cdot x2 \cdot x5) + x4 \cdot x5 + x2 \cdot \overline{x4} \cdot x5 + x2 \cdot x3 \cdot x4 + \overline{x1} \cdot x2 \cdot x1 \cdot \overline{x4} \cdot \overline{x5} + \overline{x2} \cdot \overline{x4} \cdot x5 + x2 \cdot x3 \cdot \overline{x4} .$$

2. В данных ФАЛ с помощью правила де Моргана заменить знак конъюнкции на знак дизъюнкции, а знак дизъюнкции - на знак конъюнкции:

$$1. F = x4 + \overline{x1} \cdot x2 + x1 \cdot \overline{x2} \cdot x3 + \overline{\overline{x1} \cdot \overline{x2} \cdot \overline{x3} \cdot x4} .$$

$$2. F = \overline{x1 + \overline{x1} \cdot x2 \cdot \overline{x3}} + x2 \cdot \overline{x3} + \overline{x1} \cdot \overline{x2} \cdot \overline{x3} \cdot x4 .$$

$$3. F = \overline{(x1 \cdot x2 \cdot \overline{x3} + x2 \cdot x4) \cdot (x1 + \overline{x1} \cdot x2 \cdot \overline{x3} \cdot x4)} .$$

$$4. F = \overline{\overline{x1} \cdot \overline{x2} \cdot \overline{x3} + \overline{x1} + x2 \cdot x3 \cdot x4 + \overline{x1} \cdot \overline{x2} \cdot \overline{x3} \cdot \overline{x4}} .$$

$$5. F = \overline{x1 + x1 \cdot x2 \cdot x3 \cdot x4 + \overline{x1 \cdot x3 + x2 \cdot x4}} .$$

$$6. F = \overline{x1 \cdot x4} + x2 \cdot \overline{x3} \cdot x4 + \overline{x1 + \overline{x1} \cdot x2 \cdot \overline{x3} \cdot x4} .$$

$$7. F = \overline{x1 \cdot \overline{x3} + x2} + \overline{x1 \cdot x2 \cdot \overline{x3} \cdot x4 + \overline{x1 \cdot \overline{x2} \cdot x4}} .$$

$$8. F = x2 \cdot x4 + \overline{x3 \cdot x4 + \overline{x1 \cdot x2 \cdot \overline{x3} + x1 \cdot \overline{x2} \cdot \overline{x3} \cdot x4}} .$$

$$9. F = x1 \cdot \overline{x2} + x3 + \overline{x2 \cdot \overline{x3} \cdot x4 + \overline{x1 \cdot \overline{x2} \cdot \overline{x3} \cdot x4}} .$$

10. $F = \overline{x_1 + \overline{x_1 \cdot x_2} + \overline{x_1 \cdot x_2 \cdot x_3} + \overline{x_1 \cdot x_2 \cdot x_3 \cdot x_4}}$.
11. $F = \overline{x_2 \cdot (x_1 \cdot \overline{x_3 \cdot x_2} + x_3 \cdot \overline{x_4})} + \overline{x_1 \cdot x_2 \cdot x_3 \cdot \overline{x_4}}$.
12. $F = x_1 \cdot \overline{x_2 \cdot x_4} + \overline{x_2 \cdot x_3 + \overline{x_4} + x_1 \cdot x_2 \cdot \overline{x_3 \cdot x_4}}$.
13. $F = x_1 \cdot \overline{x_3} + \overline{x_1 \cdot x_4 + x_2} + \overline{x_1 \cdot x_2 \cdot \overline{x_3 \cdot x_4}}$.
14. $F = \overline{x_1 \cdot x_2 \cdot \overline{x_3}} + \overline{x_4 + \overline{x_2 \cdot x_3} + x_1 \cdot \overline{x_2 \cdot x_3 \cdot x_4}}$.
15. $F = x_2 \cdot x_3 + \overline{x_1 \cdot x_2 \cdot \overline{x_4} + x_3} + \overline{x_1 \cdot x_2 \cdot x_3 \cdot \overline{x_4}}$.
16. $F = \overline{x_1 \cdot x_2 \cdot \overline{x_3 \cdot x_4} + x_1} + x_2 \cdot \overline{x_4} + \overline{x_1 \cdot \overline{x_3 \cdot x_4}}$.
17. $F = \overline{x_1 \cdot x_2 \cdot \overline{x_3} + x_2 \cdot x_3} + \overline{x_1 \cdot x_3 \cdot x_4 + x_1 \cdot \overline{x_2 \cdot x_3 \cdot x_4}}$.
18. $F = x_1 \cdot \overline{x_2 \cdot x_3} + \overline{x_2 + x_1 \cdot x_2 \cdot \overline{x_3 \cdot x_4} + x_2 \cdot \overline{x_4}}$.
19. $F = \overline{(x_2 + x_1 \cdot x_2 \cdot \overline{x_3 \cdot x_4}) \cdot (x_2 \cdot x_4 + x_1 \cdot \overline{x_2 \cdot x_3})}$.
20. $F = \overline{x_1 \cdot \overline{x_2} + x_1 \cdot \overline{x_2 \cdot x_3} + x_1 \cdot \overline{x_2 \cdot x_3 \cdot x_4} + x_3}$.
21. $F = \overline{x_1 \cdot \overline{x_3 \cdot x_4} + x_2 \cdot x_4 + x_3} + \overline{x_1 \cdot x_2 \cdot \overline{x_4}}$.
22. $F = \overline{x_1 \cdot \overline{x_3} + x_1 \cdot \overline{x_4} + x_2} + \overline{x_1 \cdot x_2 \cdot \overline{x_3 \cdot x_4}}$.
23. $F = \overline{x_1 \cdot \overline{x_2} + x_2 \cdot \overline{x_3 \cdot x_4} + x_3 \cdot \overline{x_4 \cdot x_5} + x_1 \cdot \overline{x_2 \cdot x_3 \cdot x_4}}$.
24. $F = x_1 + \overline{x_2 \cdot (x_1 \cdot \overline{x_3 + x_1 \cdot x_4 \cdot x_3})} + x_1 \cdot x_2 \cdot \overline{x_3 \cdot x_4}$.
25. $F = x_3 \cdot \overline{x_4 \cdot x_5} + \overline{x_1 \cdot x_2 + x_1 \cdot \overline{x_3 \cdot x_4} + x_1 \cdot \overline{x_2 \cdot x_3 \cdot x_4}}$.

3. Проконтролировать правильность выполненных преобразований с помощью программ, выводящих таблицу истинности исходной и преобразованной функций.

Содержание отчета:

- ~ Тема работы
- ~ Условие задания
- ~ Ход и порядок преобразований логических функций
- ~ Результаты работы программ

Пример выполнения:

Аксиомы и свойства алгебры логики

Задание:

- 1) Пользуясь аксиомами и свойствами алгебры логики, упростить выражение

$$F = x_1 \cdot \overline{x_2} \cdot x_5 + x_1 \cdot x_2 \cdot x_5 + x_5 \cdot (x_1 \cdot x_3 \cdot x_4 + x_1 \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} \cdot x_5 + x_2 x_3 + \overline{x_2} + x_2 \cdot \overline{x_3}) + x_1 \cdot x_2 \cdot x_5 .$$

- 2) В данной ФАЛ с помощью правила де Моргана заменить знак конъюнкции на знак дизъюнкции, а знак дизъюнкции - на знак конъюнкции

$$F = x_1 \cdot \overline{x_2} + \overline{x_1 \cdot x_3} + \overline{x_2 \cdot x_4 \cdot x_5} .$$

- 3) Проконтролировать правильность выполненных преобразований с помощью программ, выводящих таблицу истинности исходной и преобразованной функций.

- 1) Упрощение выражения

$$F = x_1 \cdot \overline{x_2} \cdot x_5 + x_1 \cdot x_2 \cdot x_5 + x_5 \cdot (x_1 \cdot x_3 \cdot x_4 + x_1 \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} \cdot x_5 + x_2 x_3 + \overline{x_2} + x_2 \cdot \overline{x_3}) + x_1 \cdot x_2 \cdot x_5 .$$

- 1) $x_1 \cdot x_2 \cdot x_5 + x_1 \cdot x_2 \cdot x_5 = x_1 \cdot x_2 \cdot x_5 ;$
- 2) $x_1 \cdot x_2 \cdot x_5 + x_1 \cdot \overline{x_2} \cdot x_5 = x_1 \cdot x_5 \cdot (x_2 + \overline{x_2}) = x_1 \cdot x_5 ;$
- 3) $x_2 x_3 + x_2 \cdot \overline{x_3} = x_2 \cdot (x_3 + \overline{x_3}) = x_2 ;$
- 4) $x_2 + \overline{x_2} = 1 ;$
- 5) $1 + x_1 \cdot x_3 \cdot x_4 + x_1 \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} \cdot x_5 = 1 ;$
- 6) $x_5 \cdot 1 = x_5 ;$
- 7) $x_1 \cdot x_5 + x_5 = x_5 .$

$$F = x_5 .$$

- 2) Преобразование знаков

$$x_1 \cdot \overline{x_2} + \overline{x_1 \cdot x_3} + \overline{x_2 \cdot x_4 \cdot x_5} =$$

$$\begin{aligned}
&= (\overline{\overline{x1} + x2}) + (\overline{\overline{x1} + x3}) + (\overline{\overline{x2} + x4 + x5}) = \\
&= (\overline{\overline{x1} + x2}) \cdot (\overline{\overline{x1} + x3}) \cdot (\overline{\overline{x2} + x4 + x5}) . \\
\mathbf{F} &= (\overline{\overline{x1} + x2}) \cdot (\overline{\overline{x1} + x3}) \cdot (\overline{\overline{x2} + x4 + x5}) .
\end{aligned}$$

3) Программы контроля правильности преобразований

Программа контроля правильности упрощения:

Program PRIM;

var X1, X2, X3, X4, X5, Y1, Y2 : boolean;

begin

writeln(' x1 x2 x3 x4 x5 y1 y2');

for X1:=false to true do

for X2:=false to true do

for X3:=false to true do

for X4:=false to true do

for X5:=false to true do

begin

Y1:=X1 and not X2 and X5 or X1 and X2 and X5 or

X5 and (X1 and X3 and X4 or

X1 and not X2 and not X3 and not X4 and X5 or

X2 and X3 or not X2 or X2 and not X3) or

X1 and X2 and X5;

Y2:=X5;

write(ord(X1):3, ord(X2):3, ord(X3):3, ord(X4):3,

ord(X5):3, ord(Y1):3, ord(Y2):3);

```

        if Y1=Y2
            then writeln(' -OK')
            else writeln(' -BAD')
        end;
    readln
end.

```

Программа контроля правильности преобразования знаков:

Program PRIM;

var X1, X2, X3, X4, X5, Y1, Y2 : boolean;

begin

writeln(' x1 x2 x3 x4 x5 y1 y2');

for X1:=false to true do

for X2:=false to true do

for X3:=false to true do

for X4:=false to true do

for X5:=false to true do

begin

Y1:=X1 and not X2 or not(X1 and X3) or

not(X2 and not X4 and X5);

Y2:=not((not X1 or X2)and not(not X1 or not X3)and

not(not X2 or X4 or not X5));

write(ord(X1):3, ord(X2):3, ord(X3):3, ord(X4):3,

ord(X5):3, ord(Y1):3, ord(Y2):3);

if Y1=Y2

```

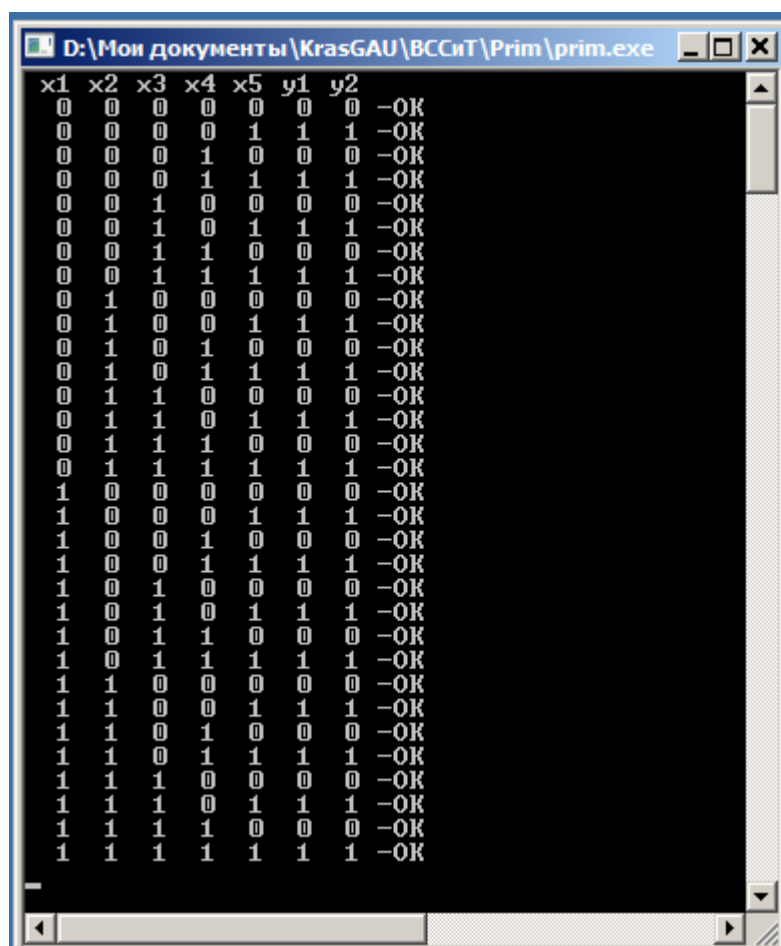
    then writeln(' -OK')
    else writeln(' -BAD')
end;

readln
end.

```

4) Результаты работы программ («скриншоты»)

Результаты контроля правильности упрощения выражения:



Результаты контроля правильности преобразования знаков:

```
D:\Мои документы\KrasGAU\BССТ\Prim\prim.exe
x1 x2 x3 x4 x5 y1 y2
0 0 0 0 0 1 1 -OK
0 0 0 0 1 1 1 -OK
0 0 0 1 0 1 1 -OK
0 0 0 1 1 1 1 -OK
0 0 1 0 0 1 1 -OK
0 0 1 0 1 1 1 -OK
0 0 1 1 0 1 1 -OK
0 0 1 1 1 1 1 -OK
0 1 0 0 0 1 1 -OK
0 1 0 0 1 1 1 -OK
0 1 0 1 0 1 1 -OK
0 1 0 1 1 1 1 -OK
0 1 1 0 0 1 1 -OK
0 1 1 0 1 1 1 -OK
0 1 1 1 0 1 1 -OK
0 1 1 1 1 1 1 -OK
1 0 0 0 0 1 1 -OK
1 0 0 0 1 1 1 -OK
1 0 0 1 0 1 1 -OK
1 0 0 1 1 1 1 -OK
1 0 1 0 0 1 1 -OK
1 0 1 0 1 1 1 -OK
1 0 1 1 0 1 1 -OK
1 0 1 1 1 1 1 -OK
1 1 0 0 0 1 1 -OK
1 1 0 0 1 1 1 -OK
1 1 0 1 0 1 1 -OK
1 1 0 1 1 1 1 -OK
1 1 1 0 0 1 1 -OK
1 1 1 0 1 0 0 -OK
1 1 1 1 0 1 1 -OK
1 1 1 1 1 1 1 -OK
```

ЛАБОРАТОРНАЯ РАБОТА № 4

Тема: Минимизация функций алгебры логики.

Цель: Приобретение навыков минимизации функций алгебры логики различными методами.

Задание:

ФАЛ четырёх переменных, заданную в числовой форме в табл.3, минимизировать:

- ~ аналитическим методом;
- ~ методом неопределенных коэффициентов;
- ~ методом карт Карно.

В базисе Буля построить функциональную схему, реализующую заданную минимизированную функцию.

Проконтролировать правильность работы полученной схемы с помощью программы моделирования логических схем.

Таблица 3.

Вариант	Функция алгебры логики
1	$f(x_1, x_2, x_3, x_4) = \bigvee_1(0, 1, 2, 4, 5, 6, 10, 12)$
2	$f(x_1, x_2, x_3, x_4) = \bigvee_1(0, 2, 3, 8, 9, 10, 11, 14, 15)$
3	$f(x_1, x_2, x_3, x_4) = \bigvee_1(0, 2, 4, 6, 7, 9, 12, 15)$
4	$f(x_1, x_2, x_3, x_4) = \bigvee_1(0, 1, 2, 3, 8, 10, 11, 12, 13, 15)$
5	$f(x_1, x_2, x_3, x_4) = \bigvee_1(1, 3, 5, 6, 7, 8, 10, 14, 15)$
6	$f(x_1, x_2, x_3, x_4) = \bigvee_1(3, 4, 5, 9, 10, 12, 14, 15)$
7	$f(x_1, x_2, x_3, x_4) = \bigvee_1(2, 4, 5, 6, 8, 9, 10, 14)$
8	$f(x_1, x_2, x_3, x_4) = \bigvee_1(1, 4, 8, 9, 10, 12, 13, 14, 15)$
9	$f(x_1, x_2, x_3, x_4) = \bigvee_1(1, 2, 4, 8, 9, 10, 12, 13, 15)$
10	$f(x_1, x_2, x_3, x_4) = \bigvee_1(0, 4, 5, 6, 7, 8, 9, 13, 15)$
11	$f(x_1, x_2, x_3, x_4) = \bigvee_1(4, 6, 7, 9, 10, 11, 12, 14, 15)$
12	$f(x_1, x_2, x_3, x_4) = \bigvee_1(3, 4, 5, 6, 9, 10, 12, 13, 14)$
13	$f(x_1, x_2, x_3, x_4) = \bigvee_1(2, 6, 7, 8, 11, 12, 13, 14, 15)$

14	$f(x_1, x_2, x_3, x_4) = \bigvee_1(1, 2, 3, 4, 7, 9, 10, 11, 14, 15)$
15	$f(x_1, x_2, x_3, x_4) = \bigvee_1(1, 4, 7, 8, 9, 10, 12, 13, 14, 15)$
16	$f(x_1, x_2, x_3, x_4) = \bigvee_1(3, 5, 6, 7, 8, 9, 10, 12, 13)$
17	$f(x_1, x_2, x_3, x_4) = \bigvee_1(4, 8, 9, 10, 11, 12, 14, 15)$
18	$f(x_1, x_2, x_3, x_4) = \bigvee_1(2, 3, 4, 6, 7, 8, 9, 11, 12)$
19	$f(x_1, x_2, x_3, x_4) = \bigvee_1(0, 6, 7, 8, 9, 10, 12, 13, 14, 15)$
20	$f(x_1, x_2, x_3, x_4) = \bigvee_1(1, 2, 4, 5, 6, 7, 11, 12, 15)$
21	$f(x_1, x_2, x_3, x_4) = \bigvee_1(1, 2, 4, 5, 7, 8, 10, 11, 12, 13)$
22	$f(x_1, x_2, x_3, x_4) = \bigvee_1(0, 1, 2, 5, 6, 7, 10, 11, 12, 15)$
23	$f(x_1, x_2, x_3, x_4) = \bigvee_1(0, 5, 6, 7, 8, 9, 10, 13, 14)$
24	$f(x_1, x_2, x_3, x_4) = \bigvee_1(3, 4, 7, 8, 9, 10, 13, 14, 15)$
25	$f(x_1, x_2, x_3, x_4) = \bigvee_1(1, 2, 3, 6, 7, 9, 10, 12, 13, 14)$

Содержание отчёта:

- ~ Тема работы
- ~ Условие задания
- ~ Таблица истинности заданной функции.
- ~ Ход минимизации аналитическим методом.
- ~ Ход минимизации методом неопределённых коэффициентов.
- ~ Ход минимизации с помощью карты Карно.
- ~ Функциональная схема, реализующая функцию.

Пример выполнения:

Минимизация функций алгебры логики

Задание:

ФАЛ трёх переменных $f(x_1, x_2, x_3) = \bigvee_1(0, 1, 3, 6, 7)$ минимизировать:

вать:

- ~ аналитическим методом;
- ~ методом неопределённых коэффициентов;

~ методом карт Карно.

В базисе Буля построить функциональную схему, реализующую заданную минимизированную функцию.

Проконтролировать правильность работы полученной схемы с помощью программы ModLogic.

Таблица истинности заданной функции

x1	x2	x3	f(x1,x2,x3)
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Минимизация аналитическим методом

СНДФ этой ФАЛ выглядит следующим образом:

$$F = \overline{x1} \cdot \overline{x2} \cdot \overline{x3} + \overline{x1} \cdot \overline{x2} \cdot x3 + \overline{x1} \cdot x2 \cdot x3 + x1 \cdot x2 \cdot \overline{x3} + x1 \cdot x2 \cdot x3 .$$

Для минимизации воспользуемся правилом "склеивания" и парно "склеим" следующие минтермы: 1 и 2, 2 и 3, 4 и 5.

$$1 \text{ и } 2: \overline{x1} \cdot \overline{x2} \cdot \overline{x3} + \overline{x1} \cdot \overline{x2} \cdot x3 = \overline{x1} \cdot \overline{x2} (\overline{x3} + x3) = \overline{x1} \cdot \overline{x2} .$$

$$2 \text{ и } 3: \overline{x1} \cdot \overline{x2} \cdot x3 + \overline{x1} \cdot x2 \cdot x3 = \overline{x1} \cdot x3 .$$

$$4 \text{ и } 5: x1 \cdot x2 \cdot \overline{x3} + x1 \cdot x2 \cdot x3 = x1 \cdot x2 .$$

Таким образом, из СНДФ получили НДФ, являющуюся минимальной формой записи заданной ФАЛ. Дальнейшее упрощение невозможно:

$$F = f(x1, x2, x3) = \overline{x1} \cdot \overline{x2} + \overline{x1} \cdot x3 + x1 \cdot x2 .$$

Минимизация методом неопределенных коэффициентов

x1	x2	x3	F	Коэффициенты						
0	0	0	1	$\neg K_1^0$	$\neg K_2^0$	$\neg K_3^0$	K_{12}^{00}	$\neg K_{13}^{00}$	$\neg K_{23}^{00}$	K_{123}^{000}
0	0	1	1	$\neg K_1^0$	$\neg K_2^0$	$\neg K_3^1$	K_{12}^{00}	$\neg K_{13}^{01}$	$\neg K_{23}^{01}$	K_{123}^{001}
0	1	0	0	$\neg K_1^0$	K_2^1	K_3^0	K_{12}^{01}	K_{13}^{00}	K_{23}^{10}	K_{123}^{010}
0	1	1	1	$\neg K_1^0$	$\neg K_2^1$	$\neg K_3^1$	$\neg K_{12}^{01}$	K_{13}^{01}	K_{23}^{11}	K_{123}^{011}
1	0	0	0	$\neg K_1^1$	K_2^0	K_3^0	K_{12}^{10}	K_{13}^{10}	K_{23}^{00}	K_{123}^{100}
1	0	1	0	$\neg K_1^1$	K_2^0	K_3^1	K_{12}^{10}	K_{13}^{11}	K_{23}^{01}	K_{123}^{101}
1	1	0	1	$\neg K_1^1$	$\neg K_2^1$	$\neg K_3^0$	K_{12}^{11}	K_{13}^{10}	$\neg K_{23}^{10}$	K_{123}^{110}
1	1	1	1	$\neg K_1^1$	$\neg K_2^1$	$\neg K_3^1$	K_{12}^{11}	$\neg K_{13}^{11}$	K_{23}^{11}	K_{123}^{111}

Получившаяся минимальная форма записи функции выглядит следующим образом:

$$F = f(x_1, x_2, x_3) = \overline{x_1} \cdot \overline{x_2} + \overline{x_1} \cdot x_3 + x_1 \cdot x_2.$$

Минимизация с помощью карты Карно

		x_2	$\overline{x_2}$	
x_1		$\boxed{1}$	0	$\overline{x_3}$
		$\boxed{1}$	0	
$\overline{x_1}$		$\boxed{1}$	$\boxed{1}$	x_3
		0	$\boxed{1}$	$\overline{x_3}$

Минимальной формой записи заданной функции согласно получившейся карте Карно является:

$$F = f(x_1, x_2, x_3) = \overline{x_1} \cdot \overline{x_2} + \overline{x_1} \cdot x_3 + x_1 \cdot x_2.$$

Функциональная схема

Функциональная схема, реализующая полученную ФАЛ в базе Буля, построенная с помощью программы ModLogic, выглядит следующим образом (рис.1):

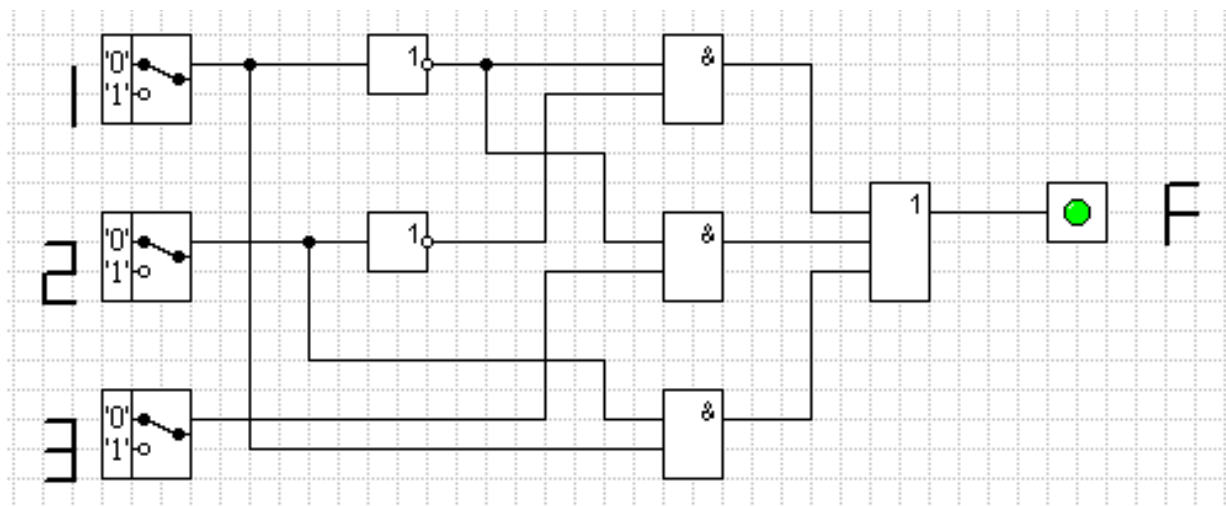


Рис.1. Функциональная схема в базисе Буля

ЛАБОРАТОРНАЯ РАБОТА № 5

Тема: Разработка дешифратора для семисегментного индикатора.

Цель: Приобретение навыков минимизации систем неполностью определенных функций алгебры логики

Задание:

- ~ Для заданного варианта (табл.4) составить систему из семи ФАЛ ($f_a, f_b, f_c, f_d, f_e, f_f, f_g$). Обозначение сегментов индикатора показано на рис.2.
- ~ Минимизировать систему ФАЛ.
- ~ С помощью правила де Моргана перевести полученные выражения в базисы: "И-НЕ" и "ИЛИ-НЕ".
- ~ Составить функциональные логические схемы дешифратора в базисах: "И, ИЛИ, НЕ", "И-НЕ", "ИЛИ-НЕ".
- ~ Проконтролировать правильность работы полученных схем с помощью программы моделирования логических схем.

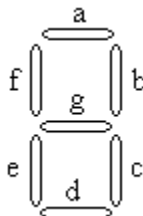


Рис.2. Расположение сегментов индикатора

Таблица 4

x1	x2	x3	x4	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	0	0	A	*	b	U	P	d	C	*	U	P	*	*	A
0	0	0	1	b	C	*	*	O	F	F	b	O	A	C	P	C
0	0	1	0	C	E	C	Y	U	E	A	d	*	d	Y	b	L
0	0	1	1	*	F	P	O	d	U	*	*	L	L	P	*	F
0	1	0	0	*	L	*	H	C	*	L	E	b	E	L	E	*
0	1	0	1	d	*	H	L	*	b	O	O	E	H	*	H	U
0	1	1	0	E	A	L	*	F	*	d	Y	*	*	O	*	P
0	1	1	1	*	d	F	*	Y	A	U	*	A	Y	H	Y	H
1	0	0	0	F	H	d	P	*	*	*	A	Y	*	d	C	*
1	0	0	1	L	O	*	F	A	C	P	C	H	U	F	F	b
1	0	1	0	H	*	E	E	*	Y	b	L	P	b	E	A	d
1	0	1	1	O	*	O	d	b	P	*	F	*	*	U	*	*
1	1	0	0	*	b	U	*	L	L	E	*	F	C	*	L	E
1	1	0	1	P	Y	*	C	*	*	H	U	d	F	b	O	O
1	1	1	0	U	P	Y	A	E	O	*	P	C	O	*	d	Y
1	1	1	1	Y	U	A	b	H	H	Y	H	*	*	A	U	*

Таблица 4 (продолжение)

x1	x2	x3	x4	14	15	16	17	18	19	20	21	22	23	24	25
0	0	0	0	Y	*	F	H	d	P	*	L	E	*	F	C
0	0	0	1	H	U	L	O	*	F	A	*	H	U	d	F
0	0	1	0	P	b	H	*	E	E	*	d	C	*	U	P
0	0	1	1	*	*	O	*	O	d	b	F	F	b	O	A
0	1	0	0	F	C	*	b	U	*	L	E	A	d	*	d
0	1	0	1	d	F	P	Y	*	C	*	U	*	*	L	L
0	1	1	0	C	O	U	P	Y	A	E	*	*	A	Y	*
0	1	1	1	*	*	Y	U	A	b	H	C	P	C	H	U
1	0	0	0	U	P	A	*	b	U	P	*	d	Y	*	*
1	0	0	1	O	A	b	C	*	*	O	A	U	*	A	Y
1	0	1	0	*	d	C	E	C	Y	U	Y	b	L	P	b
1	0	1	1	L	L	*	F	P	O	d	P	*	F	*	*
1	1	0	0	b	E	*	L	*	H	C	*	L	E	b	E
1	1	0	1	E	H	d	*	H	L	*	b	O	O	E	H
1	1	1	0	*	*	E	A	L	*	F	O	*	P	C	O
1	1	1	1	A	Y	*	d	F	*	Y	H	Y	H	*	*

Содержание отчета

- ~ Тема работы
- ~ Условие задания
- ~ Таблица истинности системы ФАЛ.
- ~ Карты Карно для семи функций системы ФАЛ.
- ~ Аналитическая запись минимизированных функций системы ФАЛ
 - в базисе "И, ИЛИ, НЕ";
 - в базисе "И-НЕ";
 - в базисе "ИЛИ-НЕ".
- ~ Три функциональные схемы дешифратора, построенные в указанных базисах.

Пример выполнения:

Разработка дешифратора для семисегментного индикатора

Задание:

- ~ Составить систему из семи ФАЛ ($f_a, f_b, f_c, f_d, f_e, f_f, f_g$) дешифратора для семисегментного индикатора, работающего согласно табл.5. При поступлении на вход дешифратора соответствующей комбинации индикатор должен высвечивать цифры 1, 2, 5, 7.

Таблица 5

x1	x2	x3	Инд
0	0	0	*
0	0	1	1
0	1	0	2
0	1	1	*
1	0	0	*
1	0	1	5
1	1	0	*
1	1	1	7

- ~ Минимизировать систему ФАЛ.
- ~ С помощью правила де Моргана перевести полученные выражения в базисы: "И-НЕ" и "ИЛИ-НЕ".

- ~ Составить функциональные логические схемы дешифратора в базисах: "И, ИЛИ, НЕ", "И-НЕ", "ИЛИ-НЕ".
- ~ Проконтролировать правильность работы полученной схемы с помощью программы ModLogic

Система функций дешифратора

Полностью система ФАЛ представлена в табл.6.

Таблица 6

x1	x2	x3	Инд	f _a	f _b	f _c	f _d	f _e	f _f	f _g
0	0	0	*	*	*	*	*	*	*	*
0	0	1	1	0	1	1	0	0	0	0
0	1	0	2	1	1	0	1	1	0	1
0	1	1	*	*	*	*	*	*	*	*
1	0	0	*	*	*	*	*	*	*	*
1	0	1	5	1	0	1	1	0	1	1
1	1	0	*	*	*	*	*	*	*	*
1	1	1	7	1	1	1	0	0	0	0

Минимизация системы функций дешифратора

Функция f_a

	x2	$\overline{x2}$	
x1	1	1	$\overline{x3}$
$\overline{x1}$	1	*	$\overline{x3}$

$$f_a = x1 + x2 ;$$

Функция f_b

	x2	$\overline{x2}$	
x1	1	0	$\overline{x3}$
$\overline{x1}$	1	*	$\overline{x3}$

$$f_b = \overline{x1} + x2 ;$$

Функция f_c

	x2	$\overline{x2}$	
x1	1	1	$\overline{x3}$
$\overline{x1}$	*	1	$\overline{x3}$

$$f_c = x3 ;$$

Функция f_d

	x_2	$\overline{x_2}$	
x_1	*	*	$\overline{x_3}$
	0	1	
$\overline{x_1}$	*	0	x_3
	1	*	$\overline{x_3}$

$$f_d = x_1 \cdot \overline{x_2} + \overline{x_3} ;$$

Функция f_e

	x_2	$\overline{x_2}$	
x_1	*	*	$\overline{x_3}$
	0	0	
$\overline{x_1}$	*	0	x_3
	1	*	$\overline{x_3}$

$$f_e = \overline{x_3} ;$$

Функция f_f

	x_2	$\overline{x_2}$	
x_1	*	*	$\overline{x_3}$
	0	1	
$\overline{x_1}$	*	0	x_3
	0	*	$\overline{x_3}$

$$f_f = x_1 \cdot \overline{x_2} ;$$

Функция f_g

	x_2	$\overline{x_2}$	
x_1	*	*	$\overline{x_3}$
	0	1	
$\overline{x_1}$	*	0	x_3
	1	*	$\overline{x_3}$

$$f_g = x_1 \cdot \overline{x_2} + \overline{x_3}$$

Системы функций в требуемых базисах

	Базис Буля	Базис "И-НЕ".	Базис "ИЛИ-НЕ".
f_a	$x_1 + x_2$	$x_1 \cdot x_2$	$x_1 + x_2$
f_b	$x_1 + x_2$	$x_1 \cdot x_2$	$x_1 + x_2$
f_c	x_3	x_3	x_3
f_d	$x_1 \cdot \overline{x_2} + x_3$	$\overline{x_1 \cdot x_2 \cdot x_3}$	$\overline{x_1 + x_2 + x_3}$
f_e	x_3	x_3	x_3
f_f	$x_1 \cdot x_2$	$x_1 \cdot x_2$	$x_1 + x_2$
f_g	$x_1 \cdot \overline{x_2} + x_3$	$\overline{x_1 \cdot x_2 \cdot x_3}$	$\overline{x_1 + x_2 + x_3}$

Функциональные схемы дешифратора

Анализируя эти ФАЛ, видим, что

$$f_d = f_g,$$

а также имеется ряд других соотношений, позволяющих уменьшить количество используемых логических элементов и, соответственно, упростить функциональные схемы:

~ В базисе Буля

$$f_d = f_f + \overline{x_3}$$

~ В базисе "И-НЕ"

$$f_f = \overline{f_b}$$
$$f_d = \overline{f_b \cdot x_3}$$

~ В базисе "ИЛИ-НЕ"

$$f_b = \overline{f_f}$$
$$f_d = f_f + \overline{x_3}$$

Функциональные схемы, реализующие полученные системы ФАЛ в базисах Буля, "И-НЕ" и "ИЛИ-НЕ", построенные с помощью программы ModLogic, выглядят следующим образом (рис.3):

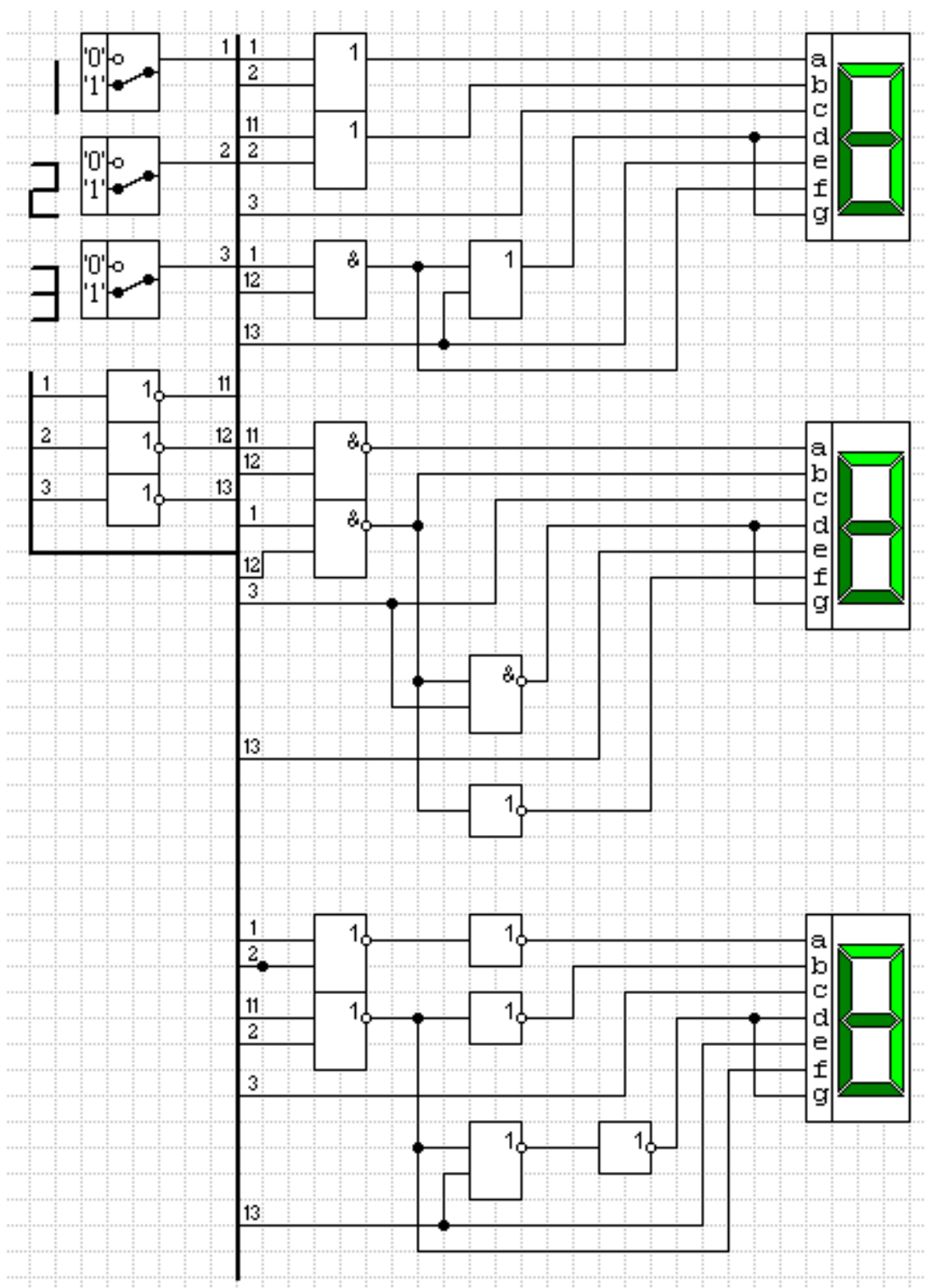


Рис. 3. Функциональная схема дешифратора

ЛАБОРАТОРНАЯ РАБОТА № 6

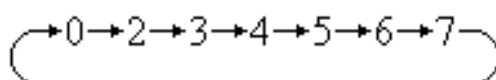
Тема: Проектирование реверсивного счетчика.

Цель: Приобретение навыков проектирования цифровых автоматов.

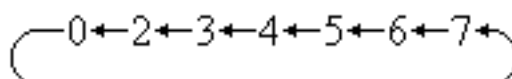
Задание:

В заданном базисе, на заданных триггерах, построить синхронный автомат Мура (синхронный реверсивный счетчик) с поступлением каждого синхроимпульса выдающий выходной сигнал $y(t)$ следующим образом:

если входной сигнал $x = 0$, то выходная последовательность сигналов выглядит следующим образом, например, для 1-го варианта:



если входной сигнал $x = 1$, то выходная последовательность сигналов для 1-го варианта выглядит так:



Варианты заданий приведены в табл.7.

Таблица 7

Вариант	Выходной сигнал $y(t)$	Триггеры			Базис
		ст.	ср.	мл.	
1	0 2 3 4 5 6 7	RS	RS	D	И-НЕ
2	0 2 3 4 5 6 7	JK	RS	D	ИЛИ-НЕ
3	0 1 2 4 5 6 7	D	T	RS	И-НЕ
4	0 1 2 3 5 6 7	JK	D	T	ИЛИ-НЕ
5	0 1 2 3 4 6 7	RS	D	RS	И-НЕ
6	0 1 2 3 4 5 7	D	JK	T	ИЛИ-НЕ
7	0 1 2 3 4 5 7	RS	T	D	И-НЕ
8	1 2 3 4 5 6 7	T	RS	JK	ИЛИ-НЕ
9	0 1 2 3 4 5 6 7	JK	D	RS	И-НЕ
10	0 2 1 3 4 5 6 7	T	T	RS	ИЛИ-НЕ
11	0 1 3 2 4 5 6 7	T	RS	JK	И-НЕ
12	0 1 2 4 3 5 6 7	RS	T	JK	ИЛИ-НЕ

13	0 1 2 3 5 4 6 7	RS	JK	T	И-НЕ
14	0 1 2 3 4 6 5 7	RS	D	JK	ИЛИ-НЕ
15	0 1 2 3 4 5 7 6	T	JK	D	И-НЕ
16	0 1 2 3 4 5	JK	JK	T	ИЛИ-НЕ
17	1 2 3 4 5 6	T	RS	JK	И-НЕ
18	2 3 4 5 6 7	JK	T	RS	ИЛИ-НЕ
19	0 3 4 5 6 7	T	JK	RS	И-НЕ
20	0 1 4 5 6 7	JK	D	RS	ИЛИ-НЕ
21	0 1 2 5 6 7	RS	JK	D	И-НЕ
22	0 1 2 3 6 7	T	JK	D	ИЛИ-НЕ
23	0 1 2 3 4 7	T	RS	JK	И-НЕ
24	0 1 2 3 4 5	RS	JK	T	ИЛИ-НЕ
25	0 2 4 5 6 7	T	JK	D	И-НЕ

Содержание отчета

- ~ Тема работы
- ~ Условие задания
- ~ Временная диаграмма выдаваемых автоматом Мура выходных сигналов.
- ~ Таблица кодов состояний автомата.
- ~ Отмеченная таблица переходов автомата.
- ~ Отмеченный граф работы автомата.
- ~ Структурная таблица автомата.
- ~ Минимизированные функции возбуждения триггеров автомата.
- ~ Функциональная схема автомата, реализованная в заданном базисе и на заданных триггерах.

Пример выполнения:

Проектирование реверсивного счетчика

Задание:

Синтезировать синхронный двоичный реверсивный счётчик (автомат Мура), имеющий модуль счёта $M=3$, на RS-триггерах. При $x=0$ счётчик должен работать в режиме суммирования и выдавать на выход циклически повторяющиеся сигналы (0, 1, 2, 0, 1, 2, 0, 1, ...), изменяющиеся с приходом каждого синхроимпульса. Если $x=1$, счётчик должен считать в обратную сторону (2, 1, 0, 2, 1, 0, 2, 1, ...), т.е. должен работать в вычитающем режиме.

Временная диаграмма работы счетчика

Временная диаграмма работы реверсивного счётчика с модулем счёта $M=3$ приведена на рис.4.

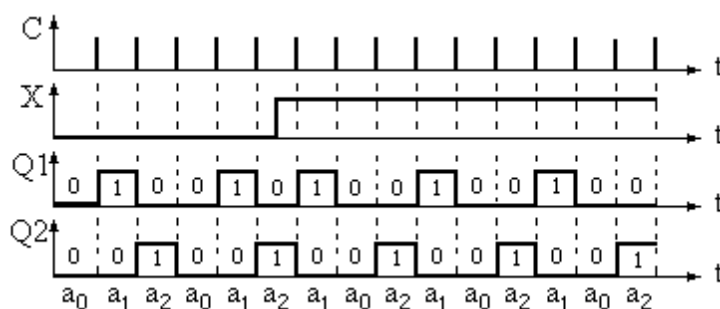


Рис.4. Временная диаграмма работы реверсивного счётчика с модулем счёта $M=3$

На рис.4 отмечены состояния автомата (a_0 , a_1 и a_2), соответствующие выходным сигналам (0, 1 и 2).

Таблица кодов состояний счетчика

Требуемое количество триггеров

$$n = \lceil \log_2 N \rceil = \lceil \log_2 3 \rceil = 2 .$$

Таким образом автомат будет иметь $2^n = 4$ состояния.

Кодировка состояний представлена в табл.8:

- ~ **Q2** - состояние старшего триггера,
- ~ **Q1** - состояние младшего триггера.

Таблица 8

a(t)	Q2	Q1
a ₀	0	0
a ₁	0	1
a ₂	1	0
a ₃	1	1

Выходные сигналы: $y_0 = a_0$; $y_1 = a_1$; $y_2 = a_2$.

Таблица переходов и граф работы счетчика

Отмеченная таблица переходов счётчика и граф переходов приведены соответственно в табл.9 и на рис.5.

Таблица 9

Вых. сигн.	y ₀	y ₁	y ₂	-
Исх. сост	a ₀	a ₁	a ₂	a ₃
$\bar{x}$	a ₁	a ₂	a ₀	a ₀
x	a ₂	a ₀	a ₁	a ₂

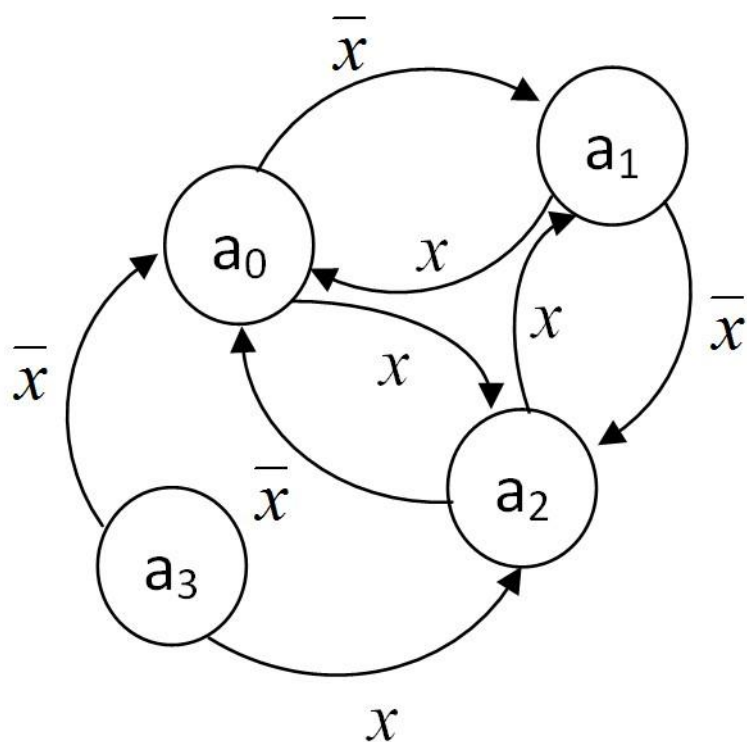


Рис.5. Граф переходов счётчика

Структурная таблица счетчика

Структурная таблица автомата Мура (табл.10):

Таблица 10

Переход	Исх. сост.			След. сост.			Входной сигнал	Сигналы возбуж- дения			
		Код			Код			старший		младший	
		Q2	Q1		Q2	Q1		R2	S2	R1	S1
1	a ₀	0	0	a ₁	0	1	$\bar{x}$	*	0	0	1
2	a ₁	0	1	a ₂	1	0	$\bar{x}$	0	1	1	0
3	a ₂	1	0	a ₀	0	0	$\bar{x}$	1	0	*	0
4	a ₃	1	1	a ₀	0	0	$\bar{x}$	1	0	1	0
5	a ₀	0	0	a ₂	1	0	x	0	1	*	0
6	a ₁	0	1	a ₀	0	0	x	*	0	1	0
7	a ₂	1	0	a ₁	0	1	x	1	0	0	1
8	a ₃	1	1	a ₂	1	0	x	0	*	1	0

Функции возбуждения счетчика

Функция возбуждения в общем виде:

$$F = [\text{исх. сост.1}] \cdot [\text{вх. сигн.1}] + [\text{исх. сост.2}] \cdot [\text{вх. сигн.2}] + \dots$$

Таким образом,

$$R2 = a_2 \cdot \bar{x} + a_3 \cdot \bar{x} + a_2 \cdot x;$$

$$S2 = a_1 \cdot \bar{x} + a_0 \cdot x;$$

$$R1 = a_1 \cdot \bar{x} + a_3 \cdot \bar{x} + a_1 \cdot x + a_3 \cdot x;$$

$$S1 = a_0 \cdot \bar{x} + a_2 \cdot x;$$

или

$$R2 = Q2 \cdot \bar{Q1} \cdot \bar{x} + Q2 \cdot Q1 \cdot \bar{x} + Q2 \cdot \bar{Q1} \cdot x;$$

$$S2 = \bar{Q2} \cdot Q1 \cdot \bar{x} + \bar{Q2} \cdot \bar{Q1} \cdot x;$$

$$R1 = \bar{Q2} \cdot Q1 \cdot \bar{x} + Q2 \cdot Q1 \cdot \bar{x} + \bar{Q2} \cdot Q1 \cdot x + Q2 \cdot Q1 \cdot x;$$

$$S1 = \bar{Q2} \cdot \bar{Q1} \cdot \bar{x} + Q2 \cdot \bar{Q1} \cdot x.$$

После минимизации с помощью карт Карно получены следующие соотношения:

Функция R2

	$Q2$	$\bar{Q2}$	
X	1	*	$\bar{Q1}$
$\bar{X}$	1	*	$Q1$
	1	*	$\bar{Q1}$

Функция S2

	$Q2$	$\bar{Q2}$	
X	*	1	$\bar{Q1}$
$\bar{X}$	*	1	$Q1$
			$\bar{Q1}$

Функция R1

	$Q2$	$\bar{Q2}$	
X	*	*	$\bar{Q1}$
$\bar{X}$	1	1	$Q1$
	1	1	$\bar{Q1}$

Функция S1

	$Q2$	$\bar{Q2}$	
X	1		$\bar{Q1}$
$\bar{X}$			$Q1$
		1	$\bar{Q1}$

$$R2 = Q2 \cdot \bar{Q1} + Q2 \cdot \bar{x};$$

$$S2 = \bar{Q2} \cdot Q1 \cdot \bar{x} + \bar{Q2} \cdot \bar{Q1} \cdot x;$$

$$R1 = Q1;$$

$$S1 = Q2 \cdot \bar{Q1} \cdot x + \bar{Q2} \cdot \bar{Q1} \cdot \bar{x}.$$

В приведенных функциях имеются одинаковые множители $\bar{Q2} \cdot \bar{Q1}$ и $Q2 \cdot \bar{Q1}$, которые можно реализовать однократно.

Функциональная схема счетчика

Схема двоичного реверсивного счётчика с модулем счёта $M=3$, построенная по полученным соотношениям с помощью программы ModLogic, приведена на рис.6. Для исключения возможных «гонок» («состязаний») в ней использованы двухтактные (двухступенчатые) JK – триггеры, включенные в режиме RS – триггеров. Изначально с помощью отдельного переключателя оба триггера приводятся в нулевое состояние.

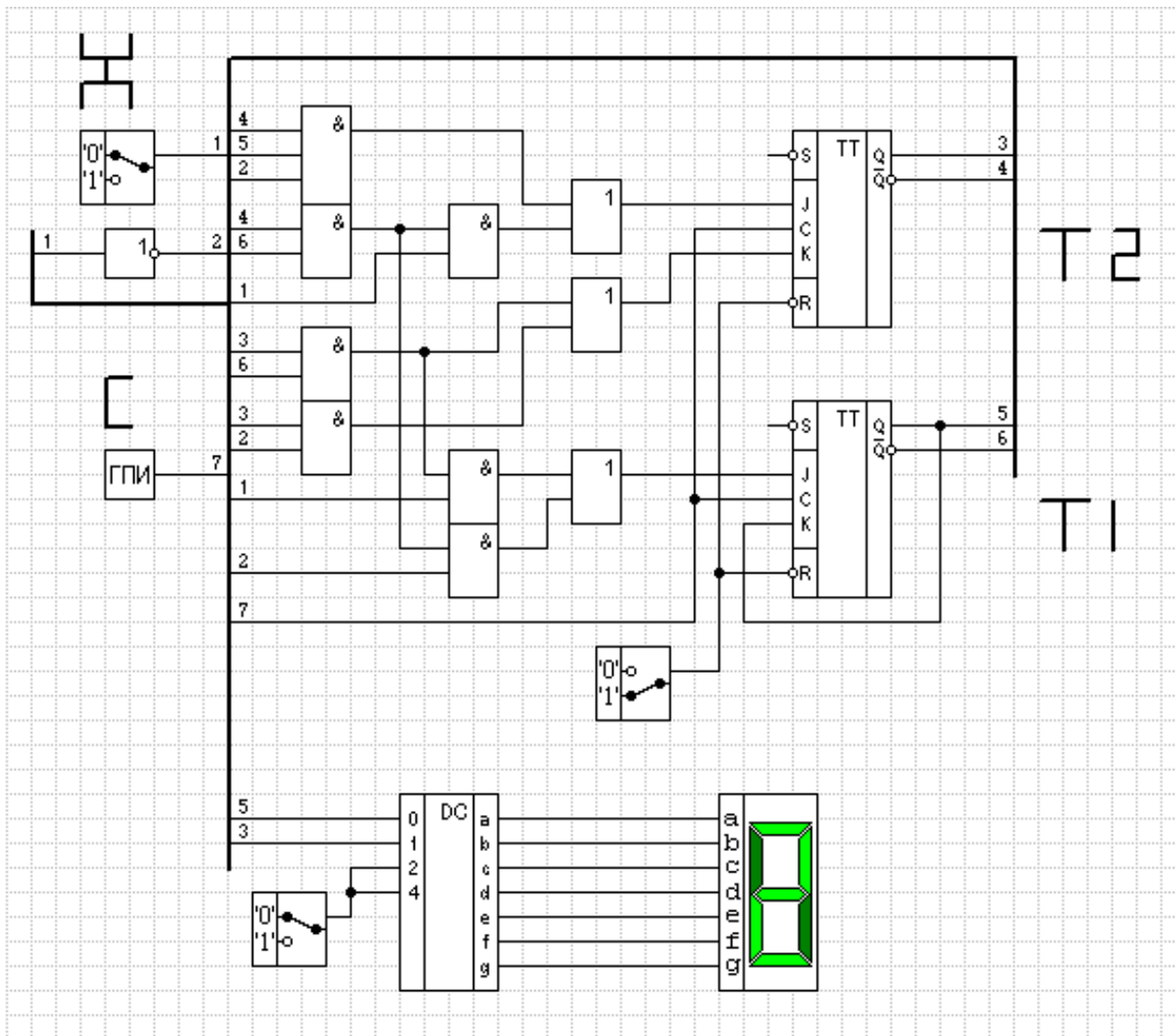


Рис.6. Функциональная схема счётчика

ЛАБОРАТОРНАЯ РАБОТА № 7

Тема: Система команд процессоров x86.

Цель: Приобретение навыков разработки программ с использованием команд процессора x86.

Задание: Пользуясь ассемблерными вставками в Pascal написать программу вычисления выражения. Варианты заданий приведены в таблице 11.

Таблица 11.

Вариант	Выражение	Вариант	Выражение
1	$\frac{2a^2b + 5ab^3}{2a^2 + 1}$	14	$\frac{2a^2b + 5ab^3}{2b^2 + 1}$
2	$\frac{3a^4b + 4ab^3}{3a^2 + 1}$	15	$\frac{3a^4b + 4ab^3}{3b^2 + 1}$
3	$\frac{4a^2b + 3ab^5}{4a^2 + 1}$	16	$\frac{4a^2b + 3ab^5}{4b^2 + 1}$
4	$\frac{5a^4b + 2ab^5}{5a^2 + 1}$	17	$\frac{5a^4b + 2ab^5}{5b^2 + 1}$
5	$\frac{2b^2a + 5ba^3}{2a^2 + 1}$	18	$\frac{2b^2a + 5ba^3}{2b^2 + 1}$
6	$\frac{3b^4a + 4ba^3}{3a^2 + 1}$	19	$\frac{3b^4a + 4ba^3}{3b^2 + 1}$
7	$\frac{4b^2a + 3ba^5}{4a^2 + 1}$	20	$\frac{4b^2a + 3ba^5}{4b^2 + 1}$
8	$\frac{5b^4a + 2ba^5}{5a^2 + 1}$	21	$\frac{5b^4a + 2ba^5}{5b^2 + 1}$
9	$\frac{2a^2b + 5ab^3}{2a^4 + 1}$	22	$\frac{2a^2b + 5ab^3}{2b^4 + 1}$
10	$\frac{3a^4b + 4ab^3}{3a^4 + 1}$	23	$\frac{3a^4b + 4ab^3}{3b^4 + 1}$
11	$\frac{4a^2b + 3ab^5}{4a^4 + 1}$	24	$\frac{4a^2b + 3ab^5}{4b^4 + 1}$
12	$\frac{5a^4b + 2ab^5}{5a^4 + 1}$	25	$\frac{5a^4b + 2ab^5}{5b^4 + 1}$
13	$\frac{3a^2b + 5ab^3}{2a^4 + 1}$		

Содержание отчета

- ~ Тема работы
- ~ Условие задания
- ~ Укрупненное словесное описание алгоритма
- ~ Текст программы с подробными комментариями
- ~ Результаты работы программы.

Пример выполнения:

Система команд процессоров x86

Задание:

Пользуясь ассемблерными вставками в Pascal написать программу вычисления выражения

$$y = \frac{2a^4b + 5b^3}{a^2 + 1}$$

Укрупненное описание алгоритма:

1. Ввод исходных данных (чисел A и B)
2. Вычисление первого слагаемого числителя $2a^4b$
3. Вычисление второго слагаемого числителя $5b^3$
4. Вычисление числителя
5. Вычисление знаменателя
6. Вычисление результата Y
7. Вывод значения выражения $\frac{2a^4b + 5b^3}{a^2 + 1}$ и числа из Y.

Текст программы:

```
Program Prim;  
var A, B, Y : integer; {переменные для исходных данных и  
                        результата}  
begin  
  write('A, B?:'); read(A, B);      {вводим исходные данные}
```

asm {начало ассемблерной вставки}

{вычисляем первое слагаемое числителя}

```
mov    ax, A      {помещаем в регистр ax число из A}
imul   ax          {возводим его в квадрат}
push   ax          {запоминаем квадрат A в стеке}
imul   ax          {получаем четвертую степень A}
imul   B           {четвертую степень A умножаем на B}
mov     bx, 2      {и еще на 2}
imul   bx
mov     si, ax      {полученное значение числителя сохраняем
                     в si}
```

{вычисляем второе слагаемое числителя}

```
mov     ax, 5      {помещаем в регистр ax число 5}
imul   B           {трижды умножаем его на B}
imul   B
imul   B
```

{вычисляем числитель}

```
add     ax, si      {к второму слагаемому добавляем первое}
```

{вычисляем знаменатель}

```
pop     bx          {квадрат A извлекаем из стека}
inc     bx          {увеличиваем его на 1}
```

{вычисляем результат}

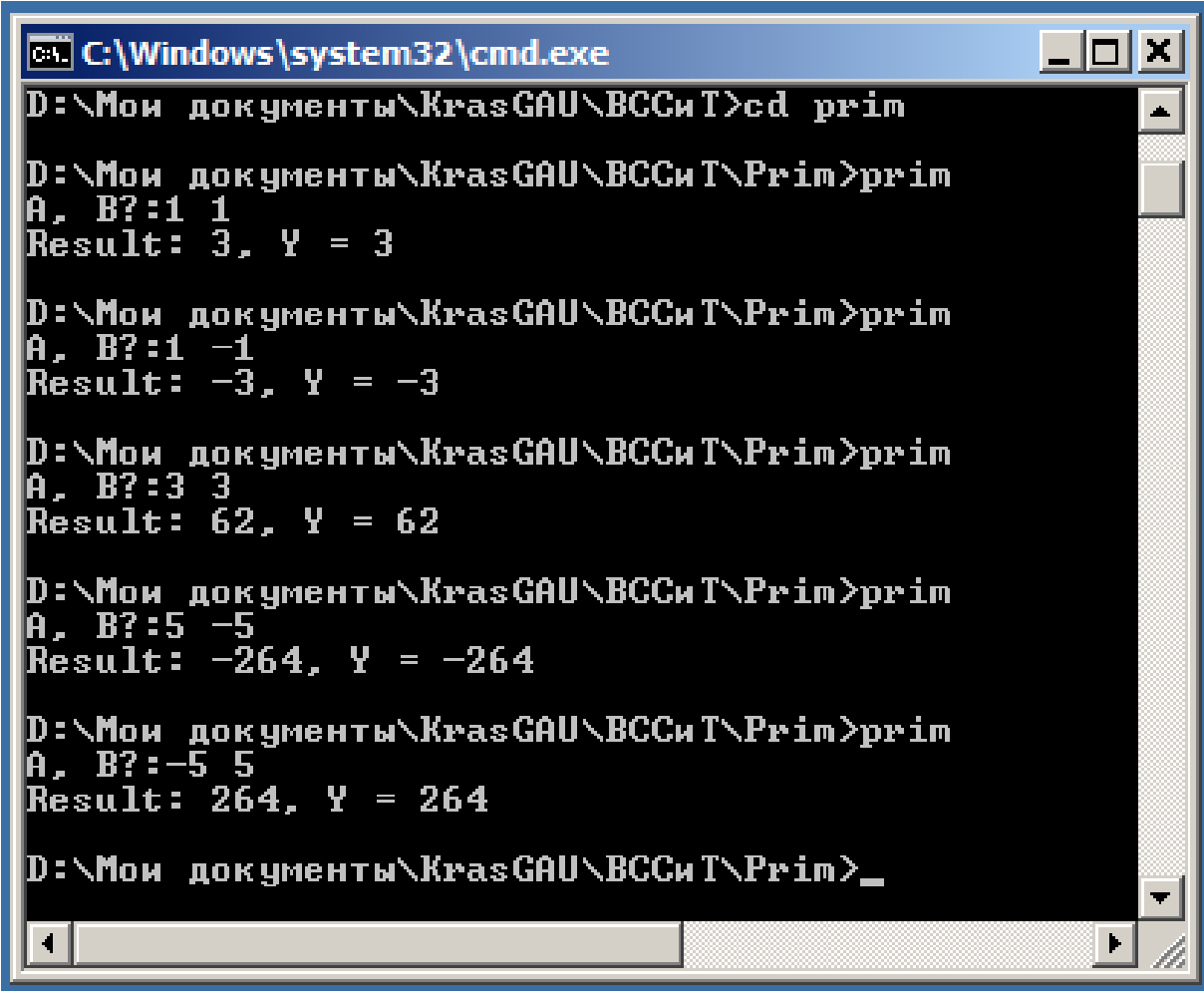
```
cwd                      {преобразуем 16 – разрядное значение
                           числителя в 32 - разрядное}
idiv    bx              {делим значение числителя на значение
                           знаменателя}
mov     Y, ax           {частное помещаем в переменную Y}
end;                    {конец ассемблерной вставки}
```

{выводим результаты по выражению и ассемблерной вставке}

```
writeln('Result: ', (2*sqr(sqr(A))*B+5*sqr(B)*B)
        div (sqr(A)+1), ', Y = ', Y)
```

end.

Результаты работы программы («скриншот»)



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The command prompt is currently in the directory 'D:\Мои документы\KrasGAU\BCCиТ\Prim'. The user has entered the command 'cd prim' to change to the subdirectory. Then, they have run the program 'prim' five times with different input values for 'A' and 'B'. Each run shows the input values and the resulting value of 'Y'.

```
C:\Windows\system32\cmd.exe
D:\Мои документы\KrasGAU\BCCиТ>cd prim
D:\Мои документы\KrasGAU\BCCиТ\Prim>prim
A, B?:1 1
Result: 3, Y = 3

D:\Мои документы\KrasGAU\BCCиТ\Prim>prim
A, B?:1 -1
Result: -3, Y = -3

D:\Мои документы\KrasGAU\BCCиТ\Prim>prim
A, B?:3 3
Result: 62, Y = 62

D:\Мои документы\KrasGAU\BCCиТ\Prim>prim
A, B?:5 -5
Result: -264, Y = -264

D:\Мои документы\KrasGAU\BCCиТ\Prim>prim
A, B?:-5 5
Result: 264, Y = 264

D:\Мои документы\KrasGAU\BCCиТ\Prim>_
```

ЛАБОРАТОРНАЯ РАБОТА № 8

Тема: Организация условных переходов в процессорах x86.

Цель: Приобретение навыков организации разветвлений в программах с использованием команд процессора x86.

Задание: Пользуясь ассемблерными вставками в Pascal написать программу вычисления выражения. Варианты заданий приведены в таблице 12.

Таблица 12.

Вариант	Выражение	Вариант	Выражение
1	$\frac{3ab-b^3}{a+3} + \frac{ab-2}{b}$	14	$\frac{3ab-b^3}{a+3} + \frac{ab-2}{b}$
2	$\frac{2ab-b^2}{a-3} + \frac{ab+2}{b}$	15	$\frac{2ab-b^2}{a-3} + \frac{ab+2}{b}$
3	$\frac{5a-b^3}{a} + \frac{ab-2}{b-3}$	16	$\frac{5a-b^3}{a} + \frac{ab-2}{b-3}$
4	$\frac{3ab+b^3}{a-3} + \frac{ab+2}{b+1}$	17	$\frac{3ab+b^3}{a-3} + \frac{ab+2}{b+1}$
5	$\frac{ab-3b^3}{3a+4} + \frac{ab-2}{b}$	18	$\frac{ab-3b^3}{3a+4} + \frac{ab-2}{b}$
6	$\frac{ab-b^3}{a+3} + \frac{3ab-2}{b}$	19	$\frac{ab-b^3}{a+3} + \frac{3ab-2}{b}$
7	$\frac{ab+2b^3}{a-3} + \frac{ab-2}{b}$	20	$\frac{ab+2b^3}{a-3} + \frac{ab-2}{b}$
8	$\frac{3ab-b^3}{a+b} + \frac{ab-2}{b-a}$	21	$\frac{3ab-b^3}{a+3} + \frac{ab-2}{b}$
9	$\frac{3ab-b^3}{a+3} + \frac{ab-2}{b}$	22	$\frac{3ab-b^3}{a+3} + \frac{ab-2}{b}$
10	$\frac{3ab+b^3}{a-b} + \frac{ab+2}{b-a}$	23	$\frac{3ab+b^3}{a-b} + \frac{ab+2}{b-a}$
11	$\frac{3ab+b^2}{a-b} + \frac{ab-2}{b+a}$	24	$\frac{3ab+b^2}{a-b} + \frac{ab-2}{b+a}$
12	$\frac{3ab-b^3}{a+b} + \frac{ab-2}{b-a}$	25	$\frac{3ab-b^3}{a+b} + \frac{ab-2}{b-a}$
13	$\frac{3ab+b^3}{a+b} + \frac{ab-2}{b-a}$		

Содержание отчета

- ~ Тема работы
- ~ Условие задания
- ~ Укрупненное словесное описание алгоритма
- ~ Текст программы с подробными комментариями
- ~ Результаты работы программы.

Пример выполнения:

Организация условных переходов в процессорах x86

Задание:

Пользуясь ассемблерными вставками в Pascal написать программу вычисления выражения

$$C = \frac{3a+b^4}{a+b} + \frac{|a-2|}{b-a}$$

Укрупненное описание алгоритма:

1. Ввод исходных данных и размещение их в переменных А и В
2. Вычисление первого знаменателя А+В и проверка: если он равен 0 – вывод сообщения об ошибке и завершение работы
3. Вычисление второго знаменателя В-А и проверка его на равенство 0; если он равен 0 – вывод сообщения об ошибке и завершение работы
4. Вычисление второго числителя $|A-2|$ и деление его на второй знаменатель В-А (получение значения второй дроби $\frac{|a-2|}{b-a}$)
5. Вычисление первого числителя $3A+B^4$ и деление его на первый знаменатель А+В (получение значения первой дроби $\frac{3a+b^4}{a+b}$)

6. Сложение значений первой и второй дробей (получение окончательного результата $\frac{3a+b^4}{a+b} + \frac{|a-2|}{b-a}$) и размещение его в переменной С
7. Вывод значения выражения $\frac{3a+b^4}{a+b} + \frac{|a-2|}{b-a}$ и числа из С.

Текст программы:

```

Program Prim;
label m, m1, m2, ex;
var A, B, C : integer; {переменные для исходных данных и
                        {результата}

begin
  write(A, B?:'); read(A, B); {вводим исходные
                              {данные}
  asm                          {начало ассемблерной вставки}

  {вычисляем первый знаменатель и проверяем его}
    mov si, A      {получаем сумму A+B в регистре si}
    add si, B
    jne m          {если она не равна 0 уходим на метку m}
end;               {конец ассемблерной вставки}
writeln('A+B=0'); {иначе выводим сообщение A+B=0}
asm               {начало ассемблерной вставки}
  jmp ex         {завершаем выполнение программы}

  {вычисляем второй знаменатель и проверяем его}
m:  mov di, B      {получаем разность B-A в регистре di}
    sub di, A
    jnz m1         {если она не равна 0 уходим на метку m1}
end;               {конец ассемблерной вставки}
writeln('B-A=0'); {иначе выводим сообщение B-A=0}
asm               {начало ассемблерной вставки}
  jmp ex         {завершаем выполнение программы}

  {вычисляем второй числитель}

```

```

m1: mov ax, A      {получаем разность A-2 в регистре ax}
    sub ax, 2
    {получаем абсолютную величину этой разности}
    jge m2         {если она больше или равна 0 уходим на
                    метку m2}
    neg ax         {иначе меняем ее знак на
                    противоположный}

{получаем значение второй дроби}
m2: cwd           {преобразуем числитель в 32 – разрядное
                    число}
    idiv di        {делим на знаменатель}
    mov bp, ax     {частное пересылаем в регистр bp
                    (освобождаем регистр ax) }

{вычисляем первый числитель}
    {в регистре ax получаем четвертую степень B}
    mov ax, B      {помещаем B в ax}
    imul ax        {возводим в квадрат}
    imul ax        {полученный квадрат B еще раз возводим в
                    квадрат}
    {к полученной четвертой степени B добавляем 3A}
    add ax, A
    add ax, A
    add ax, A

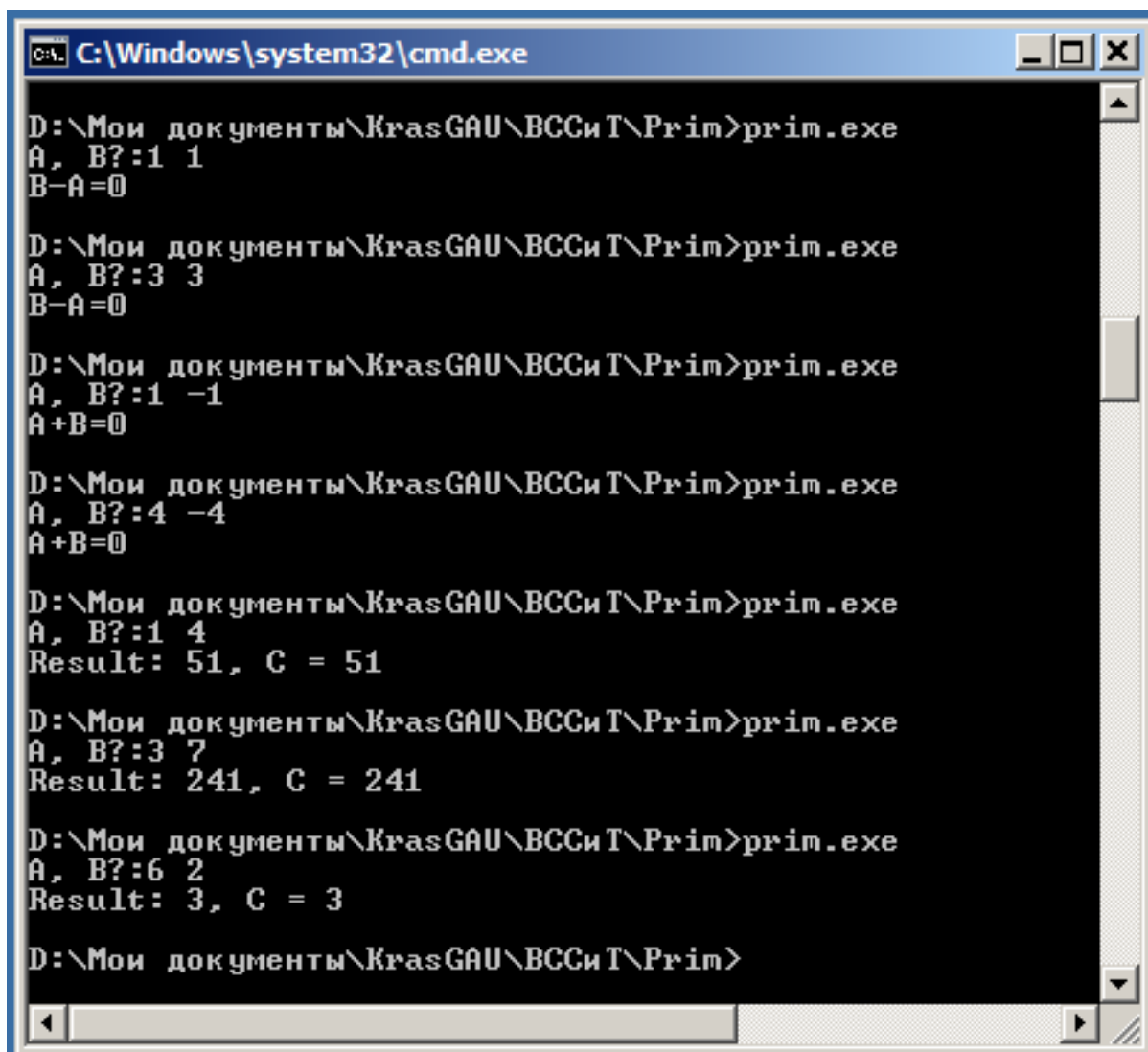
{получаем значение первой дроби}
    cwd           {преобразуем числитель в 32 – разрядное
                    число}
    idiv si        {делим на знаменатель}

{получаем окончательный результат}
    add ax, bp     {складываем значения первой и второй
                    дробей}
    mov C, ax      {помещаем полученную сумму в
                    переменную C}
end;              {конец ассемблерной вставки}

```

{выводим результаты по выражению и ассемблерной вставке}
writeln('Result: ', (3*A+sqr(sqr(B))) div (A+B)+abs(A-2) div (B-A),
 ', C = ', C)
ex:
end.

Результаты работы программы («скриншот»)



```
C:\Windows\system32\cmd.exe

D:\Мои документы\KrasGAU\BCCиТ\Prim>prim.exe
A, B?:1 1
B-A=0

D:\Мои документы\KrasGAU\BCCиТ\Prim>prim.exe
A, B?:3 3
B-A=0

D:\Мои документы\KrasGAU\BCCиТ\Prim>prim.exe
A, B?:1 -1
A+B=0

D:\Мои документы\KrasGAU\BCCиТ\Prim>prim.exe
A, B?:4 -4
A+B=0

D:\Мои документы\KrasGAU\BCCиТ\Prim>prim.exe
A, B?:1 4
Result: 51, C = 51

D:\Мои документы\KrasGAU\BCCиТ\Prim>prim.exe
A, B?:3 7
Result: 241, C = 241

D:\Мои документы\KrasGAU\BCCиТ\Prim>prim.exe
A, B?:6 2
Result: 3, C = 3

D:\Мои документы\KrasGAU\BCCиТ\Prim>
```

ЛАБОРАТОРНАЯ РАБОТА № 9

Тема: Организация циклов в процессорах x86.

Цель: Приобретение навыков организации циклов в программах с использованием команд процессора x86.

Задание: Пользуясь ассемблерными вставками в Pascal написать программу обработки целочисленного массива

Варианты заданий:

1. Написать программу расчета среднего арифметического (СА) значения положительных элементов в одномерном массиве, имеющих четные индексы.
2. Написать программу вычисления суммы отрицательных, произведения положительных и количества нулевых значений в одномерном массиве.
3. Написать программу расчета суммы положительных элементов одномерного массива, имеющих нечетные индексы.
4. Упорядочить элементы одномерного массива по неубыванию.
5. Написать программу расчета СА отрицательных элементов в одномерном массиве. Заменить минимальный элемент в одномерном массиве на СА.
6. Упорядочить элементы одномерного массива по невозрастанию.
7. В одномерном массиве поменять местами максимальный и минимальный элементы.
8. Написать программу расчета произведения положительных элементов в одномерном массиве.
9. Произвести попарные перестановки элементов одномерного массива: первый элемент поменять местами с последним, второй элемент – с предпоследним и т.д.
10. Отыскать последний положительный элемент в одномерном массиве и заменить его на СА элементов массива.
11. Написать программу расчета произведения отрицательных элементов в одномерном массиве.
12. Из одномерного массива $[A_i]$ сформировать одномерный массив $[B_i]$, записав в него сначала элементы массива A, имеющие четные индексы, потом – элементы с нечетными индексами.

13. Отыскать последний отрицательный элемент в одномерном массиве и заменить его на произведение элементов массива.
14. Заменить в одномерном массиве нулевые элементы на значение минимального элемента.
15. Заменить в одномерном массиве четные элементы на значение минимального элемента
16. Сформировать массив $[Xi]$, элементы которого равны полусумме двух соседних элементов одномерного массива $[Yi]$.
17. Сформировать массив $[Ai]$ из элементов одномерного массива $[Bi]$ по закону $A_i = (B_i + B_{N-i+1})/4, i = \overline{1, N}$
18. Сформировать массив $[Ai]$ из элементов одномерного массива $[Bi]$ $j = \overline{1, N}$ по закону $A_i = B_i + B_{N/2+i}; i = \overline{1, \frac{N}{2}}$
19. Заменить в одномерном массиве нечетные элементы на значение максимального элемента
20. Из одномерного массива $[Bi]$ сформировать массив $[Xi]$ по следующему закону:
$$Xi = \begin{cases} 1 & B_i > Y, \\ 0 & B_i = Y, \\ -1 & B_i < Y, \end{cases}$$
 где y – некоторая константа.
21. В одномерном массиве переставить местами соседние элементы с четными и нечетными индексами.
22. В одномерном массиве вычислить сумму четных элементов.
23. В одномерном массиве подсчитать количество нечетных элементов.
24. Сформировать массив $[Bi]$, содержащий последовательность чисел Фибоначчи:
 $B_i = B_{i-1} + B_{i-2}; i = \overline{3, N}; B_1 = X_1, B_2 = X_2$, где X_1, X_2 - некоторые числа.
25. Вычислить сумму правых разностей элементов одномерного массива $[Bi]$

$$S = \sum_{i=1}^{N-1} (B_i - B_{i+1})$$

Содержание отчета

~ Тема работы

- ~ Условие задания
- ~ Текст программы с подробными комментариями.
- ~ Результаты работы программы.

Пример выполнения:

Организация циклов в процессорах x86

Задание: Пользуясь ассемблерными вставками в Pascal написать программу вычисления произведения левых разностей элементов одномерного массива $S = \prod_{i=1}^{N-1} (A_{i+1} - A_i)$, замены четных элементов массива на полученное произведение и подсчета произведенных замен.

Текст программы:

```

Program PRIM;
label m, m1, m2, m3, m4;
var A : array [1..100] of integer;  {обрабатываемый массив}
    N, T : integer;                 {размер массива и вспомогательная
                                   переменная}
begin
    write('N?:'); read(N); {вводим размер
                           массива}
    writeln('Array?:');    {выводим приглашение к вводу массива}
    asm

{цикл ввода массива}
        mov  cx, N        {кладем в cx количество повторений цикла}
        lea  si, A        {заносим в si смещение до начала массива}
m:      push cx           {сохраняем в стеке значения cx и si на время
                           выполнения процедуры read}
        push si
    end;
    read(T); {вызываем процедуру ввода числа в переменную T}
    asm
        pop  si           {восстанавливаем из стека значения si и cx}

```

```

pop cx
mov ax, T      {введенное число из T через ax помещаем в
                элемент массива, адресуемый si }
mov [si], ax
add si, 2      {формируем в si смещение до следующего
                элемента массива}
loop m        {уменьшаем счетчик повторений цикла и,
                если он не равен 0, уходим на метку m}

{цикл подсчета произведения левых разностей элементов массива}
mov cx, N      {формируем в cx количество повторений
                цикла}
dec cx
lea si, A+2    {записываем в si смещение до второго элемента
                массива}
mov ax, 1      {записываем в ax начальное значение
                произведения}
m1: mov bx, [si] {помещаем в bx значение очередного
                элемента массива}
sub bx, -2[si] {вычитаем из него значение предыдущего
                элемента}
imul bx        {умножаем накопленное произведение на
                полученную левую разность}
add si, 2      {формируем в si смещение до следующего
                элемента массива}
loop m1        {уменьшаем счетчик повторений цикла и,
                если он не равен 0, уходим на метку m1}

{выводим полученное произведение левых разностей}
mov T, ax      {помещаем произведение в переменную T}
end;
writeln('Production: ', T); {выводим число
                             из переменной T}

asm
    mov ax, T    {восстанавливаем произведение в ax}

{цикл замены четных элементов массива на произведение}
xor bx, bx      {обнуляем счетчик произведенных замен}

```

```

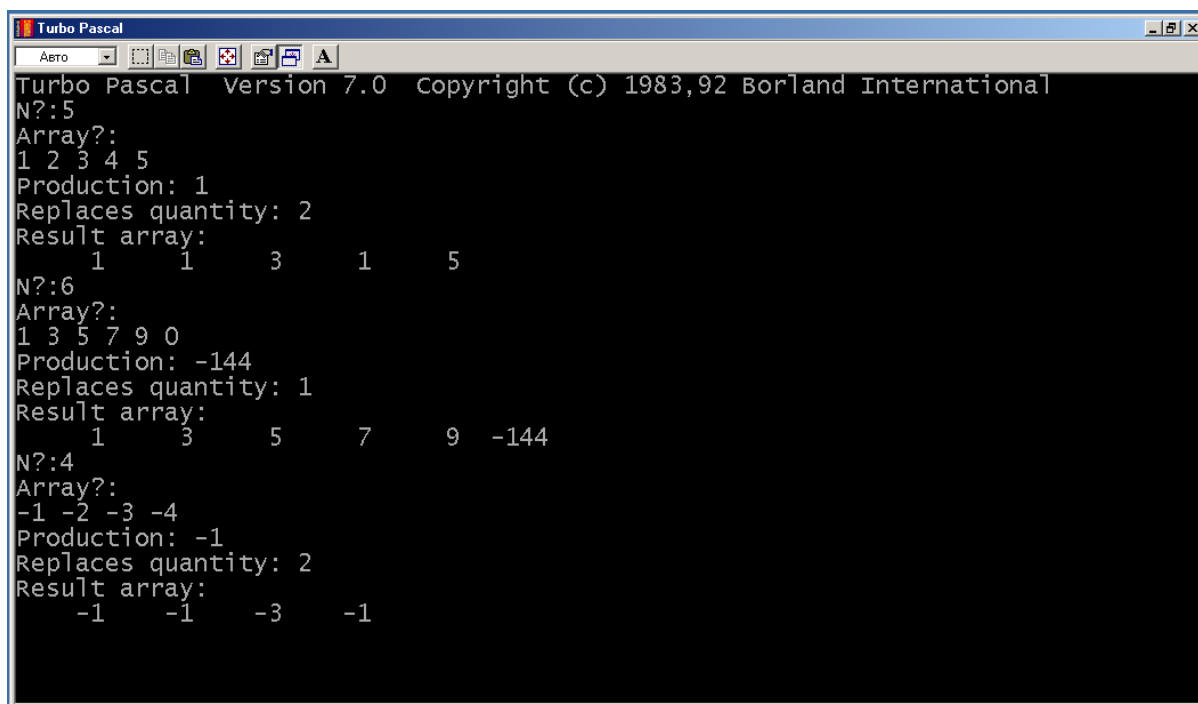
    mov cx, N      {кладем в cx количество повторений цикла}
    lea si, A      {заносим в si смещение до начала массива}
m3: test word ptr [si], 1 {проверяем элемент массива на
                        четность}
    jnz m2         {если он нечетный - уходим на метку m2}
    mov [si], ax   {иначе - заменяем на произведение}
    inc bx        {увеличиваем счетчик замен}
m2: add si, 2      {формируем в si смещение до следующего
                        элемента массива}
    loop m3        {уменьшаем счетчик повторений цикла и,
                        если он не равен 0, уходим на метку m3}

{выводим количество замен и получившийся массив}
    mov T, bx      {помещаем значение счетчика замен в
                        переменную T}
end;
writeln('Replaces quantity: ', T);    {выводим число из
                                        переменной T}
writeln('Result array:');
asm
    {цикл вывода массива}
    mov cx, N      {кладем в cx количество повторений цикла}
    lea si, A      {заносим в si смещение до начала массива}
m4: mov ax, [si]   {помещаем значение очередного элемента
                        массива в переменную T}
    mov T, ax
    push si        {сохраняем в стеке значения si и cx на время
                        выполнения процедуры write}
    push cx
end;
write(T:6);        {выводим число из переменной T}
asm
    pop cx         {восстанавливаем из стека значения cx и si}
    pop si
    add si, 2      {формируем в si смещение до следующего
                        элемента массива}
    loop m4        {уменьшаем счетчик повторений цикла и,
                        если он не равен 0, уходим на метку m4}

```

```
end;  
writeln  
end.
```

Результаты работы программы («скриншот»)



The screenshot shows the Turbo Pascal 7.0 IDE with a black command window. The program has been executed three times with different inputs. The output for each run shows the input array, the production value, the quantity of replacements, and the resulting array.

```
Turbo Pascal  Version 7.0  Copyright (c) 1983,92 Borland International  
N?:5  
Array?:  
1 2 3 4 5  
Production: 1  
Replaces quantity: 2  
Result array:  
1 1 3 1 5  
N?:6  
Array?:  
1 3 5 7 9 0  
Production: -144  
Replaces quantity: 1  
Result array:  
1 3 5 7 9 -144  
N?:4  
Array?:  
-1 -2 -3 -4  
Production: -1  
Replaces quantity: 2  
Result array:  
-1 -1 -3 -1
```

ЛАБОРАТОРНАЯ РАБОТА № 10

Тема: Программирование на ассемблере x86.

Цель: Приобретение навыков разработки программ на ассемблере процессоров x86.

Задание: На ассемблере процессоров x86 написать две программы:

- 1) программу вычисления значения выражения
- 2) программу обработки целочисленного одномерного массива.

Варианты заданий приведены в табл. 13 и табл. 14 соответственно.

Таблица 13

Вариант	Выражение	Вариант	Выражение
1	$\frac{ 5a-b^3 }{a} + \frac{ ab-2 }{b-3}$	14	$\frac{ 5a-b^3 }{a} + \frac{ ab-2 }{b-3}$
2	$\frac{ 3ab+b^3 }{a-3} + \frac{ ab+2 }{b+1}$	15	$\frac{ 3ab+b^3 }{a-3} + \frac{ ab+2 }{b+1}$
3	$\frac{ 3ab-b^3 }{a+3} + \frac{ ab-2 }{b}$	16	$\frac{ 3ab-b^3 }{a+3} + \frac{ ab-2 }{b}$
4	$\frac{ 2ab-b^2 }{a-3} + \frac{ ab+2 }{b}$	17	$\frac{ 2ab-b^2 }{a-3} + \frac{ ab+2 }{b}$
5	$\frac{ 3ab+b^3 }{a-b} + \frac{ ab+2 }{b-a}$	18	$\frac{ 3ab+b^3 }{a-b} + \frac{ ab+2 }{b-a}$
6	$\frac{ ab-3b^3 }{3a+4} + \frac{ ab-2 }{b}$	19	$\frac{ ab-3b^3 }{3a+4} + \frac{ ab-2 }{b}$
7	$\frac{ ab-b^3 }{a+3} + \frac{ 3ab-2 }{b}$	20	$\frac{ ab-b^3 }{a+3} + \frac{ 3ab-2 }{b}$
8	$\frac{ ab+2b^3 }{a-3} + \frac{ ab-2 }{b}$	21	$\frac{ ab+2b^3 }{a-3} + \frac{ ab-2 }{b}$
9	$\frac{ 3ab+b^2 }{a-b} + \frac{ ab-2 }{b+a}$	22	$\frac{ 3ab+b^2 }{a-b} + \frac{ ab-2 }{b+a}$
10	$\frac{ 3ab-b^3 }{a+b} + \frac{ ab-2 }{b-a}$	23	$\frac{ 3ab-b^3 }{a+b} + \frac{ ab-2 }{b-a}$
11	$\frac{ 3ab-b^3 }{a+3} + \frac{ ab-2 }{b}$	24	$\frac{ 3ab-b^3 }{a+b} + \frac{ ab-2 }{b-a}$
12	$\frac{ 3ab-b^3 }{a+3} + \frac{ ab-2 }{b}$	25	$\frac{ 3ab-b^3 }{a+3} + \frac{ ab-2 }{b}$

13	$\frac{ 3ab + b^3 }{a + b} + \frac{ ab - 2 }{b - a}$		
----	--	--	--

Таблица 14

Вариант	Задание
1	Упорядочить элементы одномерного массива по неубыванию.
2	В одномерном массиве поменять местами максимальный и минимальный элементы.
3	Написать программу расчета среднего арифметического (СА) значения положительных элементов в одномерном массиве, имеющих четные индексы.
4	Произвести попарные перестановки элементов одномерного массива: первый элемент поменять местами с последним, второй элемент – с предпоследним и т.д.
5	Отыскать последний положительный элемент в одномерном массиве и заменить его на СА элементов массива.
6	Написать программу вычисления суммы отрицательных, произведения положительных и количества нулевых значений в одномерном массиве.
7	Написать программу расчета произведения отрицательных элементов в одномерном массиве.
8	Из одномерного массива $[A_i]$ сформировать одномерный массив $[B_i]$, записав в него сначала элементы массива А, имеющие четные индексы, потом – элементы с нечетными индексами.
9	Написать программу расчета СА отрицательных элементов в одномерном массиве. Заменить минимальный элемент в одномерном массиве на СА.
10	Упорядочить элементы одномерного массива по невозрастанию.
11	Отыскать последний отрицательный элемент в одномерном массиве и заменить его на произведение элементов массива.

12	Написать программу расчета суммы положительных элементов одномерного массива, имеющих нечетные индексы.
13	Написать программу расчета произведения положительных элементов в одномерном массиве.
14	Сформировать массив $[A_i]$ из элементов одномерного массива $[B_i]$ по закону $A_i = (B_i + B_{N-i+1})/4$, $i = 1, N$
15	Сформировать массив $[B_i]$, содержащий последовательность чисел Фибоначчи: $B_i = B_{i-1} + B_{i-2}$; $i = 3, N$; $B_1 = X_1$, $B_2 = X_2$, где X_1, X_2 - некоторые числа.
16	Заменить в одномерном массиве нулевые элементы на значение минимального элемента.
17	Заменить в одномерном массиве четные элементы на значение минимального элемента
18	Сформировать массив $[X_i]$, элементы которого равны полусумме двух соседних элементов одномерного массива $[Y_i]$.
19	Вычислить сумму правых разностей элементов одномерного массива $[B_i]$ $S = \sum_{i=1}^{N-1} (B_i - B_{i+1})$
20	Сформировать массив $[A_i]$ из элементов одномерного массива $[B_i]$ $j = 1, N$ по закону $A_i = B_i + B_{N/2+i}$; $i = 1, \frac{N}{2}$
21	В одномерном массиве вычислить сумму четных элементов.
22	В одномерном массиве подсчитать количество нечетных элементов.
23	Заменить в одномерном массиве нечетные элементы на значение максимального элемента
24	Из одномерного массива $[B_i]$ сформировать массив $[X_i]$ по следующему закону: $X_i = \begin{cases} 1 & B_i > Y, \\ 0 & B_i = Y, \\ -1 & B_i < Y, \end{cases}$ где Y – некоторая константа.
25	В одномерном массиве переставить местами соседние элементы с четными и нечетными индексами.

Содержание отчета

- ~ Тема работы
- ~ Условие задания
- ~ Тексты программ с подробными комментариями.
- ~ Результаты работы программы.

Пример выполнения:

Программирование на ассемблере x86.

Задание: На ассемблере процессоров x86 написать две программы:

- 1) программу вычисления выражения

$$Y = \frac{3a+b^4}{a+b} + \frac{|a-2|}{b-a}$$

- 2) программу вычисления произведения левых разностей элементов одномерного массива $S = \prod_{i=1}^{N-1} (A_{i+1} - A_i)$, замены четных элементов массива на полученное произведение и подсчета произведенных замен.

Текст программы 1:

```
title      Prim
assume     cs:C, ds:D, ss:S
```

;описываем сегмент стека

```
S          segment  stack
            dw  128 dup (?)      ;резервируем 128 слов с
                                   ;неопределенным содержимым
S          ends      ;закончили описание сегмента стека
```

;определяем символы возврата каретки и перевода строки

```
cr = 0dh
lf = 0ah
```

;описываем сегмент данных

```
D          segment
;резервируем место под переменные
A          dw  ?
B          dw  ?
```

;размещаем тексты сообщений программы

```
MSG1       db  'Введите А и В:$'
MSG2       db  'А+В=0', cr, lf, '$'
MSG3       db  'В-А=0', cr, lf, '$'
MSG4       db  'У=$'
CRLF       db  cr,lf,'$'      ;последовательность символов для
                               ;перевода курсора в начало следующей
                               ;строки
```

;описываем данные процедуры ввода числа

```
string     db  255, 0, 255 dup (?)
errmsg     db  'Недопустимый символ, можно использовать'
            db  'только цифры, первый символ может быть'
            db  'знаком + или -, cr, lf,$'
negflag     dw  ?
D          ends      ;закончили описание сегмента данных
```

;описываем макрокоманду вывода текстовой строки

```
PRINT    macro    STR
          push ax
          push dx
          mov ah, 9
          lea  dx, STR
          int  21h
          pop  dx
          pop  ax
          endm
```

;описываем сегмент кодов команд

```
C        segment
```

;описываем процедуру ввода целого числа в регистр ax

```
IntegerIn proc
im:      push bx
          push dx
          push si
          mov ah, 0ah
          lea  dx, string
          int  21h
          xor  ax, ax
          lea  si, string+2
          mov  negflag, ax
          cmp  byte ptr [si], '-'
          jne  im2
          not  negflag
          inc  si
          jmp  im1
im2:     cmp  byte ptr [si],
          '+' jne  im1
          inc  si
im1:     cmp  byte ptr [si], cr
          je   iex1
          cmp  byte ptr [si], '0'
          jb   ierr
          cmp  byte ptr [si], '9'
          ja   ierr
```

```

        mov bx, 10
        mul bx
        sub byte ptr [si], '0'
        add al, [si]
        adc ah, 0
        inc si
        jmp im1
ierr:   print errmsg
        jmp im
iex1:   cmp negflag, 0
        je iex
        neg ax
iex:    pop si
        pop dx
        pop bx
        ret
IntegerIn endp

```

;описываем процедуру вывода целого числа из регистра ax

```

IntegerOut proc
        push ax
        push bx
        push cx
        push dx
        xor cx, cx
        mov bx, 10
        cmp ax, 0
        jge om
        neg ax
        push ax
        mov ah, 2
        mov dl, '-'
        int 21h
        pop ax
om:     inc cx
        xor dx, dx
        div bx
        push dx

```

```

        or    ax, ax
        jnz   om
om1:    pop    dx
        add   dx, '0'
        mov   ah, 2
        int   21h
        loop  om1
        pop    dx
        pop    cx
        pop    bx
        pop    ax
        ret
IntegerOut    endp

```

;основная программа

```

start:                                     ;точка входа в программу
        mov   ax, D                       ;в ds заносим адрес сегмента данных
        mov   ds, ax
        print MSG1                        ;выводим приглашение к вводу
                                           ;исходных данных
        call  IntegerIn                   ;вызываем процедуру ввода числа
        mov   A, ax                       ;помещаем его в переменную A
        print CRLF                        ;переводим курсор в начало следующей
                                           ;строки
        call  IntegerIn                   ;вызываем процедуру ввода числа
        mov   B, ax                       ;помещаем его в переменную B
        print CRLF                        ;переводим курсор в начало следующей
                                           ;строки

```

;вычисляем первый знаменатель и проверяем его

```

        mov   si, A                       ;получаем сумму A+B в регистре si
        add   si, B
        jne   m                           ;если она не равна 0 уходим на метку
                                           ;m и продолжаем вычисления
        print MSG2                        ;иначе выводим сообщение A+B=0
        jmp   ex                          ;и завершаем выполнение программы

```

;вычисляем второй знаменатель и проверяем его

```

m:      mov di, B      ;получаем разность B-A в регистре di
        sub di, A
        jnz m1         ;если она не равна 0 уходим на метку
                        ;m1 и продолжаем вычисления
        print MSG3     ;иначе выводим сообщение B-A=0
        jmp ex         ;и завершаем выполнение программы

```

;вычисляем второй числитель

```

m1:      mov ax, A      ;получаем разность A-2 в регистре ax
        sub ax, 2
        ;получаем абсолютную величину этой разности
        jge m2         ;если она больше или равна 0 уходим на
                        ;метку m2
        neg ax          ;иначе меняем ее знак на
                        ;противоположный

```

;получаем значение второй дроби

```

m2:      cwd           ;преобразуем числитель в 32 –
                        ;разрядное число
        idiv di        ;делим на знаменатель
        mov bp, ax     ;частное пересылаем в регистр bp
                        ;(освобождаем регистр ax)

```

;вычисляем первый числитель

;в регистре ax получаем четвертую степень B

```

        mov ax, B      ;помещаем B в ax
        mul ax         ;возводим в квадрат
        mul ax         ;полученный квадрат B еще раз
                        ;возводим в квадрат

```

;к полученной четвертой степени B добавляем 3A

```

        add ax, A
        add ax, A
        add ax, A

```

;получаем значение первой дроби

```

        cwd           ;преобразуем числитель в 32 –
                        ;разрядное число
        idiv si        ;делим на знаменатель

```

```

;получаем и выводим окончательный результат
    add ax, bp      ;складываем значения первой и второй
                    ;дробей
    print MSG4      ;оформляем вывод результата
    call IntegerOut ;вызываем процедуру вывода целого
                    ;числа из ax

;завершаем выполнение программы
ex:    mov ah, 4ch
        int 21h
C      ends ;закончили описание сегмента кодов команд

        end start ;закончили программу с указанием точки входа в
                ;нее

```

Текст программы 2:

```

        title      Prim
        assume     cs:C, ds:D, ss:S

;описываем сегмент стека
S        segment    stack
        dw 128 dup (?) ;резервируем 128 слов с
                        ;неопределенным содержимым
S        ends ;закончили описание сегмента стека

;определяем символы возврата каретки и перевода строки
        cr = 0dh
        lf = 0ah

;описываем сегмент данных
D        segment
;резервируем место под переменные
N        dw ?      ;размер массива
T        dw ?      ;вспомогательная переменная
A        dw 100 dup (?) ;обрабатываемый массив

```

;размещаем тексты сообщений программы

CRLF db cr,lf,'\$' ;последовательность символов для
;перевода курсора в начало следующей
;строки

SPACE db ' '\$' ;последовательность пробелов для
;разделения чисел в строке

MSG1 db 'Размер массива?: '\$'

MSG2 db 'Массив?:', cr, lf, '\$'

MSG3 db 'Произведение левых разностей: '\$'

MSG4 db 'Произведено замен: '\$'

MSG5 db 'Итоговый массив:', cr, lf, '\$'

;описываем данные процедуры ввода числа

string db 255, 0, 255 dup (?)

errmsg db 'Недопустимый символ, можно использовать'
db 'только цифры, первый символ может быть'
db 'знаком + или -', cr, lf,'\$'

negflag dw ?

D ends ;закончили описание сегмента данных

;описываем макрокоманду вывода текстовой строки

PRINT macro STR
push ax
push dx
mov ah, 9
lea dx, STR
int 21h
pop dx
pop ax
endm

;описываем сегмент кодов команд

C segment

;описываем процедуру ввода целого числа в регистр ax

IntegerIn proc
im: push bx
push dx
push si

```

        mov ah, 0ah
        lea dx, string
        int 21h
        xor ax, ax
        lea si, string+2
        mov negflag, ax
        cmp byte ptr [si], '-'
        jne im2
        not negflag
        inc si
        jmp im1
im2:    cmp byte ptr [si],
        '+' jne im1
        inc si
im1:    cmp byte ptr [si], cr
        je iex1
        cmp byte ptr [si], '0'
        jb ierr
        cmp byte ptr [si], '9'
        ja ierr
        mov bx, 10
        mul bx
        sub byte ptr [si], '0'
        add al, [si]
        adc ah, 0
        inc si
        jmp im1
ierr:   print errmsg
        jmp im
iex1:   cmp negflag, 0
        je iex
        neg ax
iex:    pop si
        pop dx
        pop bx
        ret
IntegerIn endp

```

;описываем процедуру вывода целого числа из регистра ax

```
IntegerOut    proc
    push ax
    push bx
    push cx
    push dx
    xor  cx, cx
    mov  bx, 10
    cmp  ax, 0
    jge  om
    neg  ax
    push ax
    mov  ah, 2
    mov  dl, '-'
    int  21h
om:         pop  ax
    inc  cx
    xor  dx, dx
    div  bx
    push dx
    or   ax, ax
    jnz  om
om1:        pop  dx
    add  dx, '0'
    mov  ah, 2
    int  21h
    loop om1
    pop  dx
    pop  cx
    pop  bx
    pop  ax
    ret
IntegerOut    endp
```

;основная программа

```
start:                                ;точка входа в программу
      mov ax, D                       ;в ds заносим адрес сегмента данных
      mov ds, ax
```

;вводим размер массива

```
      print MSG1                      ;выводим приглашение к вводу
                                      ;размера массива
      call IntegerIn                  ;вызываем процедуру ввода числа
      mov N, ax                      ;помещаем его в переменную N
      print CRLF                     ;переводим курсор в начало следующей
                                      ;строки
```

;цикл ввода массива

```
      print MSG2                      ;выводим приглашение к вводу
                                      ;массива
      mov cx, N                      ;кладем в cx количество повторений
                                      ;цикла
      lea si, A                      ;заносим в si смещение до начала
                                      ;массива
m:     call IntegerIn                  ;вызываем процедуру ввода числа
      mov [si], ax                   ;введенное число из ax помещаем в
                                      ;элемент массива, адресуемый si
      print CRLF                     ;переводим курсор в начало следующей
                                      ;строки
      add si, 2                      ;формируем в si смещение до
                                      ;следующего элемента массива
      loop m                          ;уменьшаем cx на 1 и, если он не равен
                                      ;0, уходим на метку m
```

;цикл подсчета произведения левых разностей элементов ;массива

```
      mov cx, N                      ;формируем в cx количество
                                      ;повторений цикла
      dec cx
      lea si, A+2                    ;заносим в si смещение до второго
                                      ;элемента массива
      mov ax, 1                      ;заносим в ax начальное значение
                                      ;произведения
```

```

m1:    mov  bx, [si]    ;помещаем в bx значение очередного
                        ;элемента массива
      sub  bx, -2[si]   ;вычитаем из него значение
                        ;предыдущего элемента
      imul bx          ;домножаем накопленное произведение
                        ;на полученную левую разность
      add  si, 2        ;формируем в si смещение до
                        ;следующего элемента массива
      loop m1          ;уменьшаем cx на 1 и, если он не равен
                        ;0, уходим на метку m1

;выводим полученное произведение левых разностей
      print MSG3       ;оформляем вывод
      call IntegerOut ;вызываем процедуру вывода целого
                        ;числа из ax
      print CRLF       ;переводим курсор в начало следующей
                        ;строки

;цикл замены четных элементов массива на произведение
      xor  bx, bx       ;обнуляем счетчик произведенных
                        ;замен
      mov  cx, N        ;кладем в cx количество повторений
                        ;цикла
      lea  si, A        ;заносим в si смещение до начала
                        ;массива
m3:    test word ptr [si], 1 ;проверяем элемент массива на
                        ;четность
      jnz  m2          ;если он нечетный - уходим на метку
                        ;m2
      mov  [si], ax     ;иначе - заменяем на произведение
      inc  bx          ;увеличиваем счетчик замен
m2:    add  si, 2        ;формируем в si смещение до
                        ;следующего элемента массива
      loop m3          ;уменьшаем cx на 1 и, если он не равен
                        ;0, уходим на метку m3

```

```

;выводим количество замен и получившийся массив
    mov ax, bx      ;помещаем счетчик замен в ax
    print MSG4      ;оформляем вывод
    call IntegerOut ;вызываем процедуру вывода целого
                    ;числа из ax
    print CRLF      ;переводим курсор в начало следующей
                    ;строки

;цикл вывода массива
    print MSG5      ;оформляем вывод
    mov cx, N       ;кладем в cx количество повторений
                    ;цикла
    lea si, A       ;занося в si смещение до начала
                    ;массива
m4:   mov ax, [si]   ;помещаем значение очередного
                    ;элемента массива в ax
    call IntegerOut ;вызываем процедуру вывода целого
                    ;числа из ax
    print SPACE     ;выводим пробелы между числами
    add si, 2       ;формируем в si смещение до
                    ;следующего элемента массива
    loop m4         ;уменьшаем cx на 1 и, если он не равен
                    ;0, уходим на метку m4

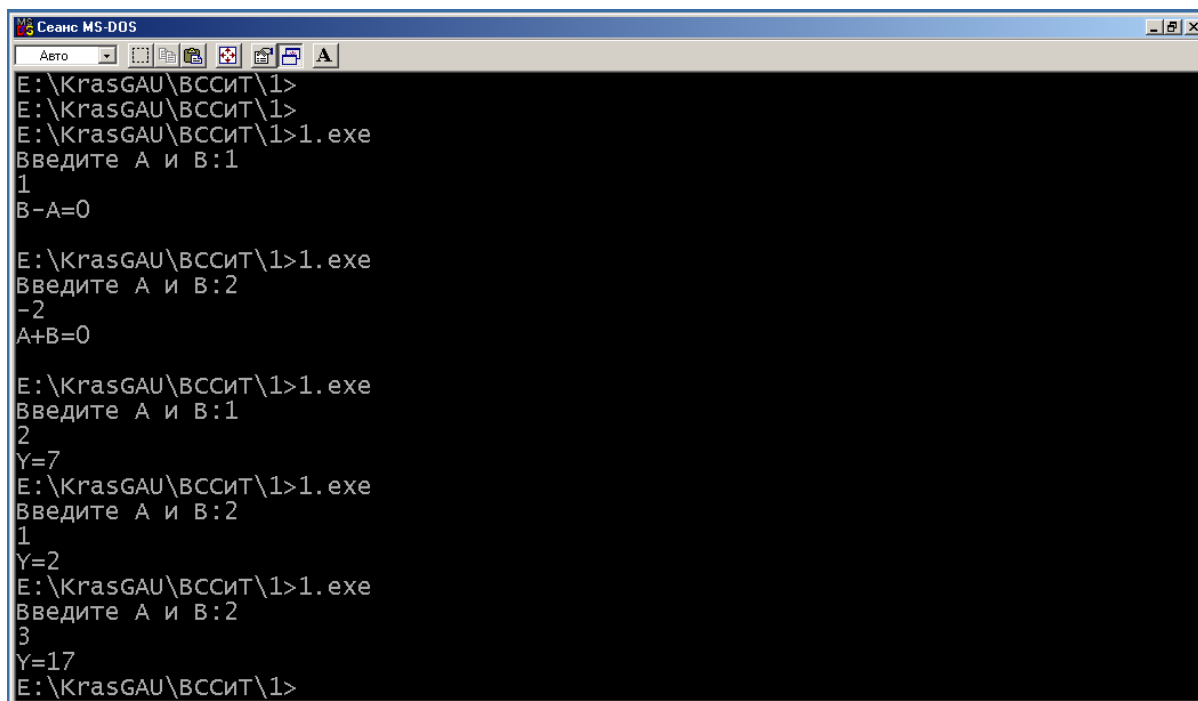
;завершаем выполнение программы
    mov ah, 4ch
    int  21h
C     ends ;закончили описание сегмента кодов команд

    end start ;закончили программу с указанием точки входа в
            ;нее

```

Результаты работы программ («скриншоты»)

Результаты работы программы вычисления выражения

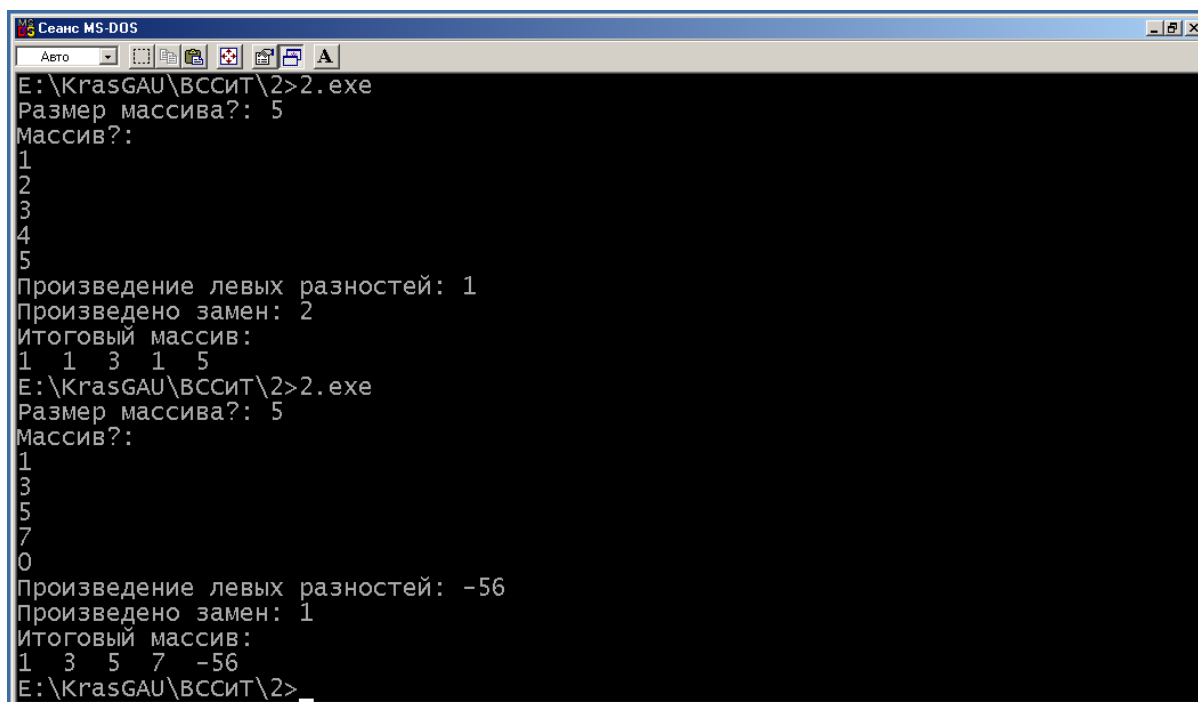


```
Сеанс MS-DOS
Автом
E:\KrasGAU\BCCIT\1>
E:\KrasGAU\BCCIT\1>
E:\KrasGAU\BCCIT\1>1.exe
Введите A и B:1
1
B-A=0

E:\KrasGAU\BCCIT\1>1.exe
Введите A и B:2
-2
A+B=0

E:\KrasGAU\BCCIT\1>1.exe
Введите A и B:1
2
Y=7
E:\KrasGAU\BCCIT\1>1.exe
Введите A и B:2
1
Y=2
E:\KrasGAU\BCCIT\1>1.exe
Введите A и B:2
3
Y=17
E:\KrasGAU\BCCIT\1>
```

Результаты работы программы обработки массива



```
Сеанс MS-DOS
Автом
E:\KrasGAU\BCCIT\2>2.exe
Размер массива?: 5
Массив?:
1
2
3
4
5
Произведение левых разностей: 1
Произведено замен: 2
Итоговый массив:
1 1 3 1 5
E:\KrasGAU\BCCIT\2>2.exe
Размер массива?: 5
Массив?:
1
3
5
7
0
Произведение левых разностей: -56
Произведено замен: 1
Итоговый массив:
1 3 5 7 -56
E:\KrasGAU\BCCIT\2>
```

ЗАКЛЮЧЕНИЕ

Основной тенденцией развития вычислительной техники в настоящее время является дальнейшее расширение сфер применения компьютеров и, как следствие, переход от отдельных машин к их системам – вычислительным системам и комплексам разнообразных конфигураций с широким диапазоном функциональных возможностей и характеристик.

Наиболее перспективные, создаваемые на основе персональных компьютеров, территориально распределенные информационно-вычислительные сети ориентируются не столько на вычислительную обработку информации, сколько на коммуникационные информационные услуги: электронную почту, системы телеконференций и информационно-справочные системы. Уже сегодня пользователям глобальной информационной сети Интернет стала доступной практически любая находящаяся в хранилищах знаний этой сети информация.

При разработке и создании собственно компьютеров существенный и устойчивый приоритет в последние годы имеют сверхмощные компьютеры – суперкомпьютеры, миниатюрные и сверхминиатюрные ПК. Ведутся, как уже указывалось, поисковые работы по созданию компьютеров 6-го поколения, базирующихся на распределенной «нейронной» архитектуре – нейрокомпьютеров. В частности, в нейрокомпьютерах могут использоваться уже имеющиеся специализированные сетевые МП – транспьютеры. Транспьютер – микропроцессор сети со встроенными средствами связи. Повсеместное использование мультимедийных широкополосных радио, волоконно-оптических и оптических каналов обмена информацией между компьютерами обеспечат практически неограниченную пропускную способность.

Широкое внедрение средств мультимедиа, в первую очередь аудио- и видеосредств ввода и вывода информации, позволят общаться с компьютером естественным для человека образом.

Информационная революция затронет все стороны жизнедеятельности. Компьютерные системы: при работе на компьютере с «дружественным интерфейсом» человек будет воочию видеть виртуального собеседника, активно общаться с ним на естественном речевом уровне с аудио- и видеоразъяснениями, советами, подсказками.

«Компьютерное одиночество», так вредно влияющее на психику активных пользователей, исчезнет.

Системы автоматизированного обучения: при наличии обратной видеосвязи ученик будет общаться с персональным виртуальным наставником, учитывающим психологию, подготовленность, восприимчивость подопечного.

Торговля: любой товар будет сопровождаться не штрих-кодом, нанесенным на торговый ярлык, а активной компьютерной табличкой, дистанционно общающейся с потенциальным покупателем и сообщаящей всю необходимую ему информацию – что, где, когда, как, сколько и почему. И так далее, и тому подобное.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Новожилов, О. П. Архитектура ЭВМ и систем в 2 ч. Часть 1 : учебное пособие для академического бакалавриата / О. П. Новожилов. — Москва : Издательство Юрайт, 2019. — 276 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-07717-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/442223>
2. Новожилов, О. П. Архитектура ЭВМ и систем в 2 ч. Часть 2 : учебное пособие для академического бакалавриата / О. П. Новожилов. — Москва : Издательство Юрайт, 2025. — 246 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-07718-6. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/444138>
3. Дибров, М. В. Сети и телекоммуникации. Маршрутизация в IP-сетях в 2 ч. Часть 1 : учебник и практикум для академического бакалавриата / М. В. Дибров. — Москва : Издательство Юрайт, 2024. — 333 с. — (Бакалавр. Академический курс). — ISBN 978-5-9916-9956-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <http://biblio-online.ru/bcode/437226>
4. Дибров, М. В. Сети и телекоммуникации. Маршрутизация в IP-сетях в 2 ч. Часть 2 : учебник и практикум для академического бакалавриата / М. В. Дибров. — Москва : Издательство Юрайт, 2019. — 351 с. — (Бакалавр. Академический курс). — ISBN 978-5-9916-9958-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <http://biblio-online.ru/bcode/437865>
5. Олифер, Виктор Григорьевич. Компьютерные сети : принципы, технологии, протоколы : [учебное пособие] / В. Г. Олифер, Н. А. Олифер. - 3-е издание. - СПб. : Питер, 2008. - 957 с.
6. Титовский С. Н. Вычислительные системы, сети и телекоммуникации: электронный ресурс <http://e.kgau.ru/course/view.php?id=1073>