

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 17.12.2021 09:42:31
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра вычислительной техники



УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова

«30» декабря 2016г.

МОДЕЛИРОВАНИЕ МИКРОПРОГРАММ ВЫПОЛНЕНИЯ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ В ДВОИЧНО- ДЕСЯТИЧНОЙ СИСТЕМЕ СЧИСЛЕНИЯ

Методические рекомендации к лабораторным работам
для студентов направления 09.03.01

Курск 2016

УДК 004.04

Составители: И.Е. Чернецкая, Г.В. Петрухин

Рецензент

Кандидат технических наук, доцент *Ю.А. Халин*

Моделирование микропрограмм выполнения арифметических операций в двоично-десятичной системе счисления: методические рекомендации к лабораторным работам/ Юго-Зап. гос. ун-т; сост.; И.Е. Чернецкая, Г.В. Петрухин. – Курск, 2016. - 21 с.: - ил. 5 , табл. 1.– Библиогр.: с. 21

Содержат сведения по вопросам моделирования микропрограмм выполнения арифметических операций. Представлены теоретические материалы выполнения операции сложения и вычитания чисел в двоично-десятичной системе счисления, заданных в прямых, дополнительных и обратных кодах; операции умножения и деления. Излагаются методические указания по подготовке и выполнению лабораторных работ.

Методические рекомендации соответствуют рабочей программе дисциплины «Теория вычислительных процессов».

Предназначены для студентов направления 09.03.01 очной и заочной форм обучения.

Текст печатается в авторской редакции

Подписано в печать 31.12.16. Формат 60*84 1/16.
Усл. печ. л. . Уч.-изд. л. Тираж 50 экз. Заказ Бесплатно.
Юго-Западный государственный университет.
305040 Курск, ул. 50 лет Октября, 94.

Оглавление

Цель работы	4
1 Теоретические основы	4
2 Порядок выполнения лабораторной работы	16
3 Содержание отчета	17
4 Контрольные вопросы	18
5 Варианты задания на лабораторную работу	19
Список литературы	21

Цель работы

Цель данной лабораторной работы является моделирование микропрограммы выполнения операций арифметического сложения-вычитания и умножения чисел в двоично-десятичной системе счисления.

1 Теоретические основы

В двоично-кодированном представлении десятичного числа каждая десятичная цифра изображается тетрадой двоичных символов

$$A_{10} = \{\alpha_4^1 \alpha_3^1 \alpha_2^1 \alpha_1^1\}_1 \{\alpha_4^2 \alpha_3^2 \alpha_2^2 \alpha_1^2\}_2 \dots \{\alpha_4^n \alpha_3^n \alpha_2^n \alpha_1^n\}_n,$$

где α_i^j - двоичные разряды тетрады j ; n – количество десятичных разрядов.

Полученные таким образом десятичные коды называют *Д-кодом*.

При образовании *Д-кода* следует исходить из общих требований, предъявляемых к системам счисления:

- различным десятичным цифрам должны соответствовать различные тетрады;
- большая по значению десятичная цифра должна изображаться большей тетрадой (если разряды тетрады имеют вес по двоичной системе счисления).

Из всего множества *Д-кодов* наибольшее распространение получили код D_1 прямого замещения (система 8421) и код D_2 с избытком 3 (система 8421+3).

В таблице 1 представлено кодирование десятичных цифр в различных *Д-кодах*.

Таблица 1 – Кодирование десятичных цифр в различных *D*-кодах

Десятичные цифры	Эквиваленты в кодах	
	D_1 (система 8421)	D_2 (система 8421+3)
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

В таблице 1 для каждого кода указаны разрешенные комбинации, все другие комбинации являются запрещенными.

В коде D_1 разрешенные комбинации соответствуют двоичным эквивалентам десятичных цифр с весами разрядов, равных степеням основания 2.

В коде D_2 все цифры тетрады имеют значения на три единицы больше, чем тетрады кода D_1 .

При сложении чисел в любом из *D*-кодов возникает необходимость в коррекции результата и трудности в формировании десятичного переноса в следующую тетраду.

При сложении двух двоично-кодированных десятичных числа $A = a_n a_{n-1} \dots a_0$ и $B = b_n b_{n-1} \dots b_0$ результатом является сумма

$$C = A + B,$$

причем

$$C_i = a_i + b_i + \Pi_{i-1} - \Pi_i p,$$

где C_i - i -й разряд суммы;

Π_{i-1} - перенос из младшей тетрады $\Pi_{i-1} = \{0,1\}$;

Π_i - перенос в старшую тетраду $\Pi_i = \{0,1\}$;

$p = 10$.

Так как наибольшее десятичное одноразрядное число равно 9, то с учетом переноса в данный разряд, значение результата разрядного суммирования лежит в пределах от 1 до 19. При этом единица во втором разряде представляет собой десятичный перенос в следующую тетраду, а сумма получается в двоичном коде, отличном от требуемого двоично-кодированного представления, т.е. требует коррекции.

Сложение чисел в D_1 коде.

При сложении чисел могут возникнуть следующие случаи.

1. Пусть $a_i + b_i + \Pi_{i-1} < 10$, где a_i, b_i - тетрады кода D_1 . При сложении в данном разряде числа образуется сумма меньше 10. Если действия над разрядами тетрады производят по правилам двоичной арифметики, то правильный результат получают без коррекции.

2. Пусть $a_i + b_i + \Pi_{i-1} \geq 10$, то возникает десятичный перенос и сумма должна быть равна $a_i + b_i + \Pi_{i-1} - \Pi_i \cdot 10$, где $\Pi_i = 1$. Свидетельством того, что результат неправильный, является появление запрещенной комбинации, если $15 \geq a_i + b_i + \Pi_{i-1} \geq 10$, либо появление потетрадного переноса $\Pi_i' = 16$, что превышает значение десятичного переноса на 6. Следовательно, требуется коррекция результата путем введения поправки, равной +0110.

При этом, если в нескольких тетрадах, начиная с $(i+1)$ -й, разрядная сумма равна 1001, то перенос приведет к формированию запрещенной комбинации в $(i+1)$ -й тетраде. В результате этого потребуются коррекция, которая приведет к запрещенной комбинации в $(i+2)$ -й тетраде и т.д. Следовательно, из-за последовательного распространения потетрадных переносов время сложения в коде D_1 составит в худшем случае n тактов. Для ускорения операции сложения в данном случае схемы строят таким образом, чтобы перенос, возникающий при добавлении потетрадной поправки, проходил сквозь тетрады, в которых предварительная сумма равна 1001 и сбрасывал их в 0000. При этом сумма формируется всегда за 2 такта.

Т.о. существуют следующие правила потетрадного сложения чисел в D_1 – коде:

- если при потетрадном сложении перенос в соседнюю старшую тетраду не возникает ($P_i=0$), то результата суммирования не требует коррекции;
- коррекция результата потетрадного сложения путем добавления поправки +0110 требуется в случае, если:
 - а) возникает потетрадный перенос в старшую тетраду ($P_i=1$);
 - б) возникает запрещенная комбинация.

Например.

Сложить два числа $A=456$, $B=398$ (представленных в прямом коде).

Решение:

$$\begin{array}{r}
 0100 \ 0101 \ 0110 \\
 + \\
 0011 \ 1001 \ 1000 \\
 \hline
 0111 \ 1110 \ 1110 \\
 + \\
 0000 \ 0110 \ 0110 \quad \text{поправки}
 \end{array}$$

$$\overline{1000 \quad 0101 \quad 0100}$$

Ответ: $C = 854$.

Сложение чисел в D_2 коде.

При сложении чисел в D_2 коде возможны следующие случаи с учетом того, что $a'_i = a_i + 3$ и $b'_i = b_i + 3$ - тетрады для кода D_2 .

Если $a'_i + b'_i + \Pi_{i-1} < 10$, то результат требует коррекции путем введения поправки -0011 .

Если $a'_i + b'_i + \Pi_{i-1} \geq 10$, то здесь возникает десятичный перенос, который «уносит» с собой шесть избыточных комбинаций. В данном случае требуется коррекция результата путем введения поправки $+0011$.

Здесь поправки вводятся при блокировке цепей потетрадного переноса. Правила введения поправок формулируется следующим образом:

если при сложении тетрад не возникает переноса ($\Pi_i = 0$), то поправка равна -0011 (или дополнение $+1101$);

если возникает потетрадный перенос ($\Pi_i = 1$), то поправка равна $+0011$.

Например.

Сложить два числа $A=456$, $B=398$ (представленных в прямом коде).

Решение:

$$\begin{array}{r}
 0111 \quad 1000 \quad 1001 \\
 + \\
 \overline{0110 \quad 1100 \quad 1011} \\
 1110 \leftarrow 0101 \leftarrow 0100 \\
 + \\
 1101 \quad 0011 \quad 0011 \quad \text{поправки} \\
 \overline{1011 \quad 1000 \quad 0111}
 \end{array}$$

Ответ: $C = 854$.

Отрицательные числа могут быть представлены в прямом, обратном или дополнительном кодах. Если A - отрицательное число и $A = -0, a_n a_{n-1} \dots a_0$, где a_i - тетрады, то

$$[A]_{np} = 1, a_n a_{n-1} \dots a_0;$$

$$[A]_{об} = 1, \overline{a_n} \overline{a_{n-1}} \dots \overline{a_0};$$

$$[A]_{\partial} = 1, \overline{a_n} \overline{a_{n-1}} \dots \overline{a_0},$$

где $\overline{a_i}$ - дополнение до $p-1=9$ во всех тетрадах

$\overline{a_i}$ - дополнение до $p-1=9$ во всех тетрадах за исключением младшей для которой это дополнение до $p=10$.

Обратный код D_2 получается простым инвертированием цифр тетрад. В коде D_1 для получения обратного кода во все тетрады вначале прибавляют +0110, после чего производится инвертирование цифр тетрад.

Например: представить отрицательное число $A=-786$ в коде D_1 и коде D_2 . Код D_1 :

$$\begin{array}{r} [A]_{np} = 1, 0111 \quad 1000 \quad 0110 \\ + \\ \hline \qquad \qquad 0110 \quad 0110 \quad 0110 \quad \text{поправки} \\ \qquad \qquad 1101 \quad 1110 \quad 0011 \quad \text{инвертировать} \end{array}$$

в результате получаем, представление отрицательного числа в коде D_1

$$[A]_{об} = 1, 0010 \quad 0001 \quad 0011.$$

В коде D_2 число A представляется в следующем виде:

$$[A]_{об} = 1, 1000 \quad 0111 \quad 1001.$$

Выполнение операций сложения и вычитания в D -кодах рассмотрим на следующих примерах.

Выполнить операцию суммирования двух чисел $A=456$, $B=-398$ в D_1 -коде, с использованием сумматора дополнительного кода.

Решение:

Представляем операнды в требуемых кодах.

$$[A]_{np} = 0,0100\ 0101\ 0110; [B]_o = 1,0110\ 0000\ 0010$$

Выполняем операцию суммирования.

$$\begin{array}{r} 0,0100\ 0101\ 0110 \\ + \\ 1,0110\ 0000\ 0010 \\ \hline 1,1010\ 0101\ 1000 \\ + \\ 0110\ 0000\ 0000 \quad \text{вводим поправки} \\ \hline 0,0000\ 0101\ 1000 \end{array}$$

Ответ: $C = 58$.

Выполнить операцию суммирования двух чисел $A=456$, $B = -398$ в D_1 -коде, с использованием сумматора обратного кода.

$$[A]_{np} = 0,0100\ 0101\ 0110$$

Для определения обратного кода произведем коррекцию путем введения поправки +0110 и выполним инвертирование полученных тетрад.

$$\begin{array}{r} [B]_{np} = 1,0011\ 1001\ 1000 \\ \quad \quad \quad \underline{0110\ 0110\ 0110} \quad \text{поправка} \\ 1,1001\ 1111\ 1110 \quad \text{инвертировать} \end{array}$$

$$[B]_{ob} = 1,0110\ 0000\ 0001.$$

Выполняем операцию суммирования.

$$\begin{array}{r} 0,0100\ 0101\ 0110 \\ + \\ 1,0110\ 0000\ 0001 \\ \hline 1,1010\ 0101\ 0111 \\ + \\ 0110\ 0000\ 0000 \quad \text{вводим поправки} \\ \hline 0,0000\ 0101\ 0111 \end{array}$$

Ответ: $C = 58$.

Выполнить операцию суммирования двух чисел $A = -456$, $B = 398$ в D_2 -коде, с использованием сумматора дополнительного кода.

Решение:

Представляем операнды в требуемых кодах.

$$[A]_d = 1,1000\ 0111\ 0111; [B]_{np} = 0,0110\ 1100\ 1011$$

Выполняем операцию суммирования.

$$\begin{array}{r} 1,1000\ 0111\ 0111 \\ + \\ \hline 0,0110\ 1100\ 1011 \\ \hline 1,1111 \leftarrow 0100 \leftarrow 0010 \\ + \\ \hline 1101\ 0011\ 0011 \quad \text{вводим необходимые поправки} \\ \hline 1,1100\ 0111\ 0101 \quad \text{переводим в прямой код из} \end{array}$$

дополнительного кода и получаем результат

$$C = 1,0011\ 1000\ 1011$$

Ответ: $C = -58$.

Выполнить операцию суммирования двух чисел $A = -456$, $B = 398$ в D_2 -коде, с использованием сумматора обратного кода.

$$[A]_{np} = 1,0111\ 1000\ 1001; [B]_{np} = 0,0110\ 1100\ 1011$$

$$[A]_{ob} = 1,1000\ 0111\ 0110;$$

Выполняем операцию суммирования.

$$\begin{array}{r} 1,1000\ 0111\ 0110 \\ + \\ \hline 0,0110\ 1100\ 1011 \\ \hline 1,1111 \leftarrow 0011 \leftarrow 0001 \\ + \\ \hline 1101\ 0011\ 0011 \quad \text{вводим необходимые поправки} \\ \hline 1,1100\ 0111\ 0100 \quad \text{переводим в прямой код} \\ \hline C = 1,0011\ 1000\ 1011 \end{array}$$

Ответ: $C = -58$.

Умножение чисел в *D*-кодах сводится к последовательному суммированию частных произведений, получаемых при умножении множимого на очередную цифру множителя. При этом умножение сопровождается расшифровкой значения очередной *i*-й тетрады множителя, и сдвигом множителя на четыре разряда сразу. Самым простым приемом расшифровки тетрады является последовательное вычитание единицы из значения тетрады до получения нуля и соответственно прибавление множимого к СЧП на каждом такте. Так как при умножении множимого на тетраду возможно переполнение разрядной сетки СЧП, то в ней необходимо предусмотреть дополнительную тетраду для учета возникающих переносов. Применяется способ умножения младшими разрядами множителя со сдвигом СЧП вправо. Умножение обычно производится в прямом коде.

Если сомножители имеют по *n* десятичных разрядов, то в регистре множимого должно быть *n* разрядов (тетрад) справа от запятой, а в регистре СЧП должно быть *n* основных разрядов, одна тетрада переполнений и один $(n+1)$ -й дополнительный десятичный разряд (ДДР), который предназначен для округления результата. При необходимости сохранить все $2n$ разрядов произведения справа от регистра СЧП должен быть еще сдвиговый регистр для младших разрядов произведения. Регистр множителя должен иметь *n* тетрад справа от запятой. Причем, если младшая тетрада этого регистра выполнена в виде реверсивного счетчика, т.е. счетчика, который может прибавлять или вычитать единицу, то операцию умножения можно ускорить в среднем почти в 2 раза. В этом случае необходимо иметь специальный разряд для записи 1, если при суммировании в счетчике - тетраде появится код 10_{10} .

Пример. Умножить на сумматоре прямого кода (код D_I) числа

$$[A]_{np} = 1, 1000 \quad 0110;$$

$$[B]_{np} = 1, 0011 \quad 0011.$$

Решение: При умножении используются сумматор прямого кода для кода D_I на три тетрады и регистры на две тетрады.

Последовательность выполнения операций представлена следующим образом.

СЧП			Регистр В		Примечание
0000	0000	0000	0011	0011	Анализ младшей тетрады множителя
	+			-	
	<u>1000</u>	0110		<u>1</u>	Результат на каждом шаге суммирования СЧП и множимого представлен после проведения коррекции
0000	1000	0110		0010	
	+			-	Конец анализа мл. тетрады. Сдвиг на четыре разряда
	<u>1000</u>	0110		<u>1</u>	
0001	0111	0010		0001	Анализ следующей за младшей тетрады множителя
	+			-	
	<u>1000</u>	0110		<u>1</u>	Конец анализа мл. тетрады. Сдвиг на четыре разряда
0010	0101	1000		0000	
0000	0010	0101	1000	0011	Конец анализа мл. тетрады. Сдвиг на четыре разряда
	+			-	
	<u>1000</u>	0110		<u>1</u>	Конец анализа мл. тетрады. Сдвиг на четыре разряда
0001	0001	0001		0010	
	+			-	Конец анализа мл. тетрады. Сдвиг на четыре разряда
	<u>1000</u>	0110		<u>1</u>	
0001	1001	0111		0001	Конец анализа мл. тетрады. Сдвиг на четыре разряда
	+			-	
	<u>1000</u>	0110		<u>1</u>	Конец анализа мл. тетрады. Сдвиг на четыре разряда
0010	1000	0011		0000	
0000	0010	1000	0011	1000	

Такой состав оборудования объясняется следующим образом. Если очередная цифра множителя, находящаяся в младшей тетраде регистра множителя, равна или меньше 5_{10} , то производится многократное прибавление множимого к СЧП. При каждом суммировании множимого вычитается 1 из счетчика. Такие действия выполняются до тех пор, пока на счетчике не появится код нуля. Если же очередная цифра множителя равна

или больше b_{10} , то производится многократное вычитание (сложение в дополнительном *Д-коде*) множимого из СЧП. При каждом вычитании множимого к содержимому счетчика прибавляется 1. Вычитания продолжаются до тех пор, пока в счетчике не появится код 10_{10} . При этом в специальный разряд записывается 1.

По окончании анализа текущей тетрады множителя производится сдвиг на одну тетраду вправо СЧП и множителя. При этом в счетчике регистра множителя сдвигается следующая тетрада множителя, к которой прибавляется содержимого специального разряда. Операция умножения заканчивается сдвигом множителя и СЧП. Всего должно быть выполнено n сдвигов. На последнем сдвиге в счетчик сдвигается нуль множителя. Если при этом в него добавляется 1 из специального разряда, то операция умножения заканчивается прибавлением множимого к СЧП. В результате получается $2n$ разрядное произведение. Таким образом, описанный алгоритм позволяет существенно сократить время умножения практически без изменения состава оборудования. Если необходим n разрядный результат, то произведение округляется путем прибавления к ДДР числа 5_{10} .

При умножении в *Д-кодах* также можно использовать еще один способ ускорения операции умножения, основанный на следующих формулах:

$$\left. \begin{aligned} AB &= 2A \frac{B-1}{2} + A, & \text{если } B & - \text{нечетное число,} \\ AB &= 2A \frac{B}{2}, & \text{если } B & - \text{четное число.} \end{aligned} \right\} \quad (1)$$

При двоично-десятичном представлении чисел удвоение числа означает сдвиг влево, а деление на 2 – сдвиг вправо, причем т.к. сдвиги производятся над двоичными кодами, то требуется коррекция тетрад на каждом шаге. Корректирующие поправки определяются для каждого *Д-кода*. Коррекция выполняется тогда, когда происходит сдвиг единицы из

крайнего разряда данной тетрады в соседнюю тетраду. Для кода D_1 корректирующая поправка равна +0110 для тетрад множимого и -0011 (или +1101) для тетрад множителя.

Пример 2. Умножить ускоренным способом (1) на сумматоре прямого кода (код D_1) числа

$$[A]_{np} = 1, 1000 \ 0110;$$

$$[B]_{np} = 1, 0011 \ 0011.$$

Решение:

B	A	C	Примечание
0011 0011	0000 1000 0110	0000 0000 1000 0110	
0001 1001 +	0001 0000 1100		Сдвиг
<u> 1101</u>	0110 0110		поправки
0001 0110	0001 0111 0010		В -четное
0000 1011 +	0010 1110 0100		сдвиг
<u> 1101</u>	0110		поправки
0000 1000	0011 0100 0100		В -четное
0000 0100	0110 1000 1000		сдвиг
0000 0010	1101 0001 0000		Сдвиг
	0110 0110 0110		поправки
	0001 0011 0111 0110		В -четное
0000 0001 +	0010 0100 1110 1100		Сдвиг
	0110 0110		поправки
	0010 0111 0101 0010		В -нечетное
		+ 0010 0111 0101 0010	
		0010 0111 1101 1000	
		+ 0110	поправка
		0010 1000 0011 1000	

Ответ: $A \cdot B = 0010 \ 1000 \ 0011 \ 1000 (2838_{10})$.

2 Порядок выполнения лабораторной работы

Выбирается тип и число разрядов сумматора (таблицы 2 и 3), в зависимости от заданного варианта. Количество тетрад соответствует вводимому с клавиатуры десятичному числу (десятичное число 12345 представляется в двоично-десятичном виде с использованием 5-ти тетрад).

Описывается содержательный алгоритм (совокупная схема) выполнения арифметических операций сложения и вычитания в соответствующих *Д-кодах*.

Производится кодирование микроопераций и условий.

Проектируется операционный автомат арифметико-логического устройства. Производится описание микрокоманд и условий.

Затем выполняется программирование математической модели арифметических операций сложения и вычитания чисел в *Д-кодах*, с использованием заданного типа сумматора, на языке высокого уровня.

Математическая модель строится исходя из следующих ограничений.

1. Входные и выходные данные представляют собой десятичные векторы с заданной разрядностью.
2. Каждое устройство, включенное в схему операционного автомата, должно быть представлено в виде подпрограммы (функции, модуля).
3. Головная часть программы должна включать в себя последовательность микрокоманд и условных операторов и предусматривать возможности просмотра промежуточных результатов.
4. Каждое устройство, включенное в схему операционного автомата, должно быть представлено в виде подпрограммы (функции, модуля).

5. Головная часть программы должна включать в себя последовательность микрокоманд и условных операторов и предусматривать возможности просмотра промежуточных результатов.

3 Содержание отчета

1. Задание на лабораторную работу (см. таблицы 2 и 3).
2. Содержательный алгоритм выполнения арифметических операций в *Д-кодах*.
3. Структура операционного автомата выполнения арифметических операций сложения-вычитания чисел и умножения, заданных двоично-десятичными кодами.
4. Составленный список микрокоманд и условий.
5. Разработанная программа на алгоритмическом языке высокого уровня, моделирующая полученную микропрограмму.
6. Составленные тестовые примеры для различных соотношений знаков операндов.
7. Выводы по работе.

4 Контрольные вопросы

1. Написать изображения чисел $-831, -305, -118, 445$ в прямом, обратном и дополнительном кодах в двоично-десятичной системе счисления.
2. Вычесть из числа 896 число 1009, предварительно представив их в *Д-кодах*.
3. Сложить на сумматоре дополнительного кода числа -328 и 545 , предварительно представив их в двоично-десятичной системе счисления.
4. Написать изображения чисел $-831, -305, -118, 445$ в прямом, обратном и дополнительном кодах в двоично-десятичной системе счисления.
5. Умножить два числа 34 и -56 последовательным (без ускорения) способом, предварительно представив их в *Д-кодах*.
6. Умножить два числа -45 и -34 любым из ускоренных способом, предварительно представив их в *Д-кодах*.

5 Варианты задания на лабораторные работы

Таблица 2 – Варианты заданий для моделирования операций сложения-вычитания в *Д-кодах*

Номер варианта	Тип сумматора	Код числового представления <i>Д-код</i>	Разрядность сумматора
1.	ДК	D_1	2
2.	ОК	D_2	2
3.	ДК	D_1	2
4.	ОК	D_2	4
5.	ДК	D_1	4
6.	ОК	D_2	4
7.	ДК	D_1	8
8.	ОК	D_2	8
9.	ДК	D_1	8
10.	ОК	D_2	2
11.	ДК	D_1	2
12.	ОК	D_2	2
13.	ДК	D_1	4
14.	ОК	D_2	4
15.	ДК	D_1	4
16.	ОК	D_2	8
17.	ДК	D_1	8
18.	ОК	D_2	8
19.	ДК	D_1	2
20.	ОК	D_2	2
21.	ДК	D_1	2
22.	ОК	D_2	4
23.	ДК	D_1	4
24.	ОК	D_2	4
25.	ДК	D_1	8
26.	ОК	D_2	8
27.	32	D_1	8
28.	ДК	D_2	2
29.	ОК	D_1	2
30.	ДК	D_2	2
31.	ОК	D_1	4
32.	ДК	D_2	4
33.	ОК	D_1	4

Таблица 3 – Варианты заданий для моделирования операции умножения в *Д-кодах*

Номер варианта	Способ умножения	Разрядность сумматора
1.	последовательный	2
2.	ускоренный	2
3.	последовательный	2
4.	ускоренный	4
5.	последовательный	4
6.	ускоренный	4
7.	последовательный	8
8.	ускоренный	8
9.	последовательный	8
10.	ускоренный	2
11.	последовательный	2
12.	ускоренный	2
13.	последовательный	4
14.	ускоренный	4
15.	последовательный	4
16.	ускоренный	8
17.	последовательный	8
18.	ускоренный	8
19.	последовательный	2
20.	ускоренный	2
21.	последовательный	2
22.	ускоренный	4
23.	последовательный	4
24.	ускоренный	4
25.	последовательный	8
26.	ускоренный	8
27.	последовательный	8
28.	ускоренный	2
29.	последовательный	2
30.	ускоренный	2
31.	последовательный	4
32.	ускоренный	4
33.	последовательный	4

Список литературы

1. Информатика. Базовый курс [Текст]: учебное пособие / под ред. С. В. Симоновича. - 3-е изд. - СПб.: Питер, 2012. - 640 с.
2. Громов, Ю.Ю. Архитектура ЭВМ и систем [Электронный ресурс]: учебное пособие / Ю.Ю. Громов, О. Г. Иванова, М. Ю. Серегин. - Тамбов: Издательство ФГБОУ ВПО «ТГТУ», 2012. - 200 с.
3. Процедурно-модульное программирование на Delphi [Текст]: учебное пособие / С. Г. Емельянов [и др.]. - Москва: Аргатак-Медиа, 2014. - 352 с. - ISBN 978-5-00024-0 20-5