

А. В. Дрозд

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

**к контрольной работе
по теме:**

**Выполнение арифметико-логических операций
на однородной вычислительной среде**

в дисциплине

**“Арифметические основы вычислительной техники”
для студентов специальности 7.091501**

2012



1. Однородная вычислительная среда

Однородная вычислительная среда (ОВС) является мощным средством вычислительной техники, сочетающим достоинства двух противоположных подходов к решению вычислительных задач: **аппаратного** и **программного**. Первый подход заключается в построении устройства, специализированного на решение конкретной задачи. Это позволяет достигать высокого быстродействия при ее решении. Другой подход предполагает решение задачи на универсальной ЭВМ путем составления программы вычислений. В этом случае задача решается значительно медленнее, однако для решения следующей задачи нет необходимости разрабатывать новые специализированные устройства, а достаточно сменить программу. ОВС обладает быстродействием первого и гибкостью второго подходов.

Структура ОВС показана на рис. 1 и представляет собой матрицу ячеек - **операционных элементов (ОЭ)**.

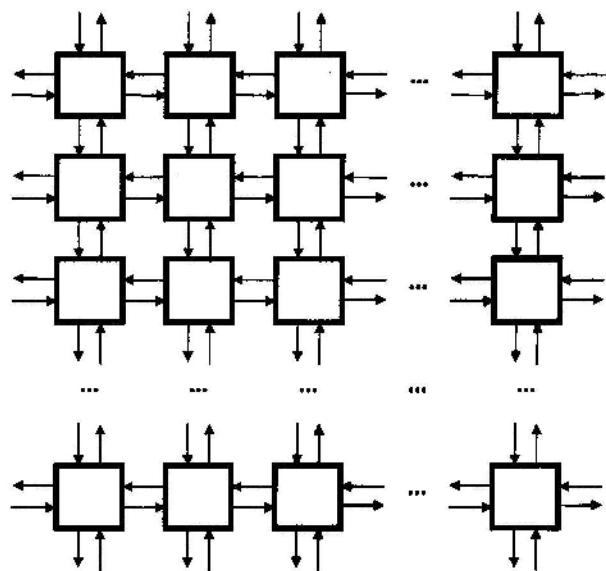


Рис. 1. Структура ОВС

Каждый ОЭ имеет:

- **информационные входы и выходы;**
- **синхровходы;**
- **настроечные входы и выходы.**

На рис. 1 показаны только информационные связи для каждого ОЭ – четыре входа и четыре выхода. Они связывают ОЭ с соседними ОЭ, расположенными по четырем сторонам света:

- **юг,**
- **север,**
- **запад,**
- **восток,**

и служат для приема и выдачи информации. Информационные входы и выходы ОЭ на границах матрицы ячеек служат для ввода операндов и вывода результатов. В ходе решения вычислительной задачи информация движется через ОЭ последовательно бит за битом.

Вычислительный процесс во времени представляется последовательностью одинаковых временных интервалов – **тактов**. Такты отмеряются с помощью тактовых импульсов, поступающих через синхровходы на все ОЭ. Каждый ОЭ вносит в распространение бита информации один, два или девять тактов задержки.

К настроечным входам и выходам относятся:

- **вход режима,**
- **вход и выход программирования.**

Входы режима всех ОЭ объединены между собой. По этим входам одновременно для всех ячеек матрицы задается режим работы или (предваряющий его) режим программирования ОВС.

Через вход программирования осуществляется (путем продвижения) запись в ОЭ **кода команды** длиной 16 бит. Через 16 тактов продвижения этот код достигает выхода программирования. Последовательное соединение выходов программирования одних ОЭ со входами программирования других ОЭ образует **цепь программирования**. Включение ОЭ в цепь программирования позволяет записывать коды команд этих ОЭ последовательно через один вход программирования.

Последовательное программирование ОЭ уменьшает количество входов программирования и упрощает управление ОВС, однако увеличивает время программирования.

2. Операционный элемент

Структура ОЭ показана на рис. 2. ОЭ содержит

- *регистр команды 1*,
- *мультиплексоры 2, 3, 4*,
- *арифметико-логическое устройство (АЛУ) 5*,
- *триггеры 6 ÷ 9*,
- *демультиплексоры 10, 11, 12*.

Перед началом работы ОВС ее ОЭ программируются на выполнение определенных операций с заданием направлений приема операндов и выдачи результатов, а также времени их задержки. Для этого в регистр команды *1* вдвигается код команды, содержащий:

- *КОП – код операции*,
- *Y* – бит, который для пяти операций, выполняемых в АЛУ, определяет задержку выдачи из ОЭ результатов – один или два такта соответственно, при $Y = 0$ и $Y = 1$, (задержки вносятся триггером 6 и триггером 7, включаемым в цепь выдачи результата при $Y = 1$),
- *A1, A2, A3* – коды управления мультиплексорами 2, 3, 4 по приему информации,
- *A4, A5, A6* – коды управления демультиплексорами 10, 11, 12 по выдаче информации.

Коды управления A1 ÷ A6 определяют направления приема или передачи информации:

- 00_2 – юг,
- 10_2 – север,
- 01_2 – запад,
- 11_2 – восток.

Мультиплексоры 2, 3 выделяют первый *a* и второй *b* операнды, поступающие на входы АЛУ.

Демультиплексор 10 транслирует результат *c* с выхода АЛУ 5 и далее триггеров 6, 7 на заданное направление.

Одновременно с операцией в АЛУ выполняется также операция *транзита*: передача информации с заданного направления через мультиплексор 4 с задержкой на один такт (на триггере 8) или два такта (на триггерах 8, 9) и далее соответственно через демультиплексоры 12, 11.

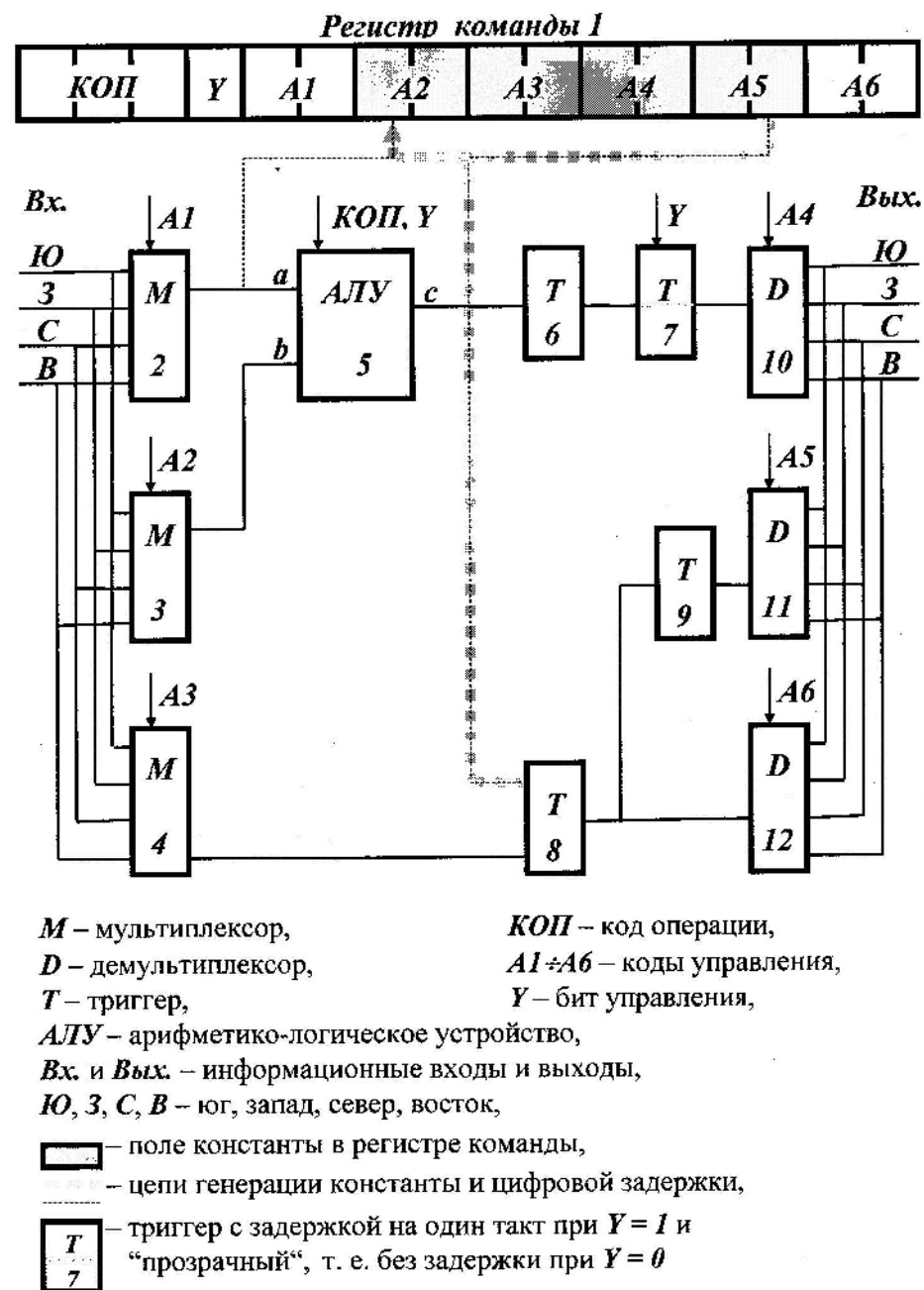


Рис. 2. ОЭ ОВС

Ниже приводятся коды, обозначение и содержание операций, выполняемых в ОЭ:

000_2 – логическое умножение (операция И), обозначается символом “&” и определяет результат $c = a \wedge b$.

001_2 – логическое умножение с инверсией (операция И-НЕ), обозначается символами “&→” и определяет результат $c = \neg(a \wedge b)$.

010_2 – логическое сложение (операция ИЛИ), обозначается символом “|” и определяет результат $c = a \vee b$.

011_2 – сумма по модулю два (операция условного инвертирования, неравнозначности, ИСКЛЮЧАЮЩЕЕ ИЛИ), обозначается символом “⊕” и определяет результат $c = a \oplus b$.

100_2 – арифметическое сложение, обозначается символом “+” и определяет значение суммы $c = SUM(a, b, p)$ и значение переноса $p = CAR(a, b, p)$, которое запоминается в АЛУ для сложения с разрядами операндов в следующем такте (SUM , CAR – функции суммы и переноса полного двоичного сумматора).

101_2 – условное запоминание, обозначается символом “T” и определяет результат $c = b$, если первый операнд $a = 1$, и сохраняет предыдущее значение c , если $a = 0$.

110_2 – при $Y = 0$ – операция GC генерации константы. В процессе программирования в регистр команды 1 на место кодов $A2 \div A5$ записывается 8-битовая константа (в $A5$ – младшие разряды). В процессе работы ОВС константа циклически сдвигается (от $A2$ к $A5$) и через триггер 8 и демультиплексор 12 выдается на заданный кодом $A6$ выход ОЭ. Триггер 8 вносит один такт задержки. При выполнении операции демультиплексоры 10 и 11 блокируются.

При $Y = 1$ – операция DD задержки на 9 тактов. Код $A1$ задает направление приема операнда a , который через мультиплексор 2 продвигается через разряды $6 \div 13$ регистра команды 1 (где задерживается на 8 тактов), триггер 8 (вносящий еще один такт задержки) и демультиплексор 12 , выдающий под управлением кода $A6$ операнд на заданный выход ОЭ. Коды $A2 \div A5$ при программировании определяются произвольно, поскольку демультиплексоры 10 , 11 блокируются. При операциях генерации константы и цифровой задержки операция транзита не выполняется.

3. Выполнение на ОВС простейших операций

Обратимся к технике программирования ОВС. Рассмотрим пример сложения на ОЭ двух положительных двоичных чисел A и B , поступающих бит за битом, соответственно, с юга и востока. Результат C необходимо выдать на северное направление через один такт, а первый операнд транзитом на восток с задержкой на один такт и на запад с задержкой на два такта.

В соответствии с условиями примера, $KOP = 100_2$, $Y = 0$, $A1 = 00_2$, $A2 = 11_2$, $A3 = 00_2$, $A4 = 10_2$, $A5 = 01_2$, $A6 = 11_2$, а код команды ОЭ представлен в табл. 1.

Таблица 1

Коды команды ОЭ ОВС

KOP	Y	A1	A2	A3	A4	A5	A6
100	0	00	11	00	10	01	11

На рис. 3 показан ОЭ ОВС, выполняющий описанные действия.

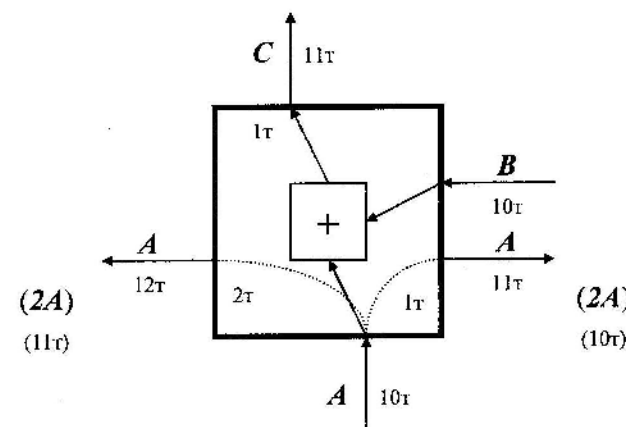


Рис. 3. Выполнение сложения чисел на ОЭ ОВС

Внутри ОЭ обозначаются:

- выполняемая операция сложения (показана в квадрате);
- направления поступления операндов и выдачи результата;
- цепи транзита (пунктиром);
- задержки "1т" и "2т" на 1 и 2 такта результата и транзита.

На входах и выходах ОЭ показываются:

- названия обрабатываемых и получаемых данных;
- **текущие такты**, определяемые по **суммарным задержкам** распространения данных с учетом предыдущих ОЭ ОВС.

Операнды должны поступать на ОЭ одновременно, т. е. с одинаковой суммарной задержкой.

Для *A* и *B* суммарная задержка составляет 9 тактов, а текущий такт является 10-м.

Результат *C* вычисляется за 1 такт на 11-м текущем такте с суммарной задержкой 10 тактов, а транзиты с задержками на 1 и 2 такта выдаются на 11-м и 12-м текущих тактах с суммарными задержками 10 и 11 тактов.

Суммарная задержка может отсчитываться от начала работы ОВС или от момента подачи на ее входы операндов. В выполняемой контрольной работе необходимо указать принятый вариант отсчета суммарной задержки.

Каждый такт задержки ОЭ может использоваться двумя способами:

- увеличивать суммарную задержку данного;
- удваивать данное при сохранении его исходной суммарной задержки.

В данном примере на рис. 3 в скобках показано использование тактов задержки для увеличения числа.

Транзит с задержкой на 1 такт удваивает число *A* при сохранении его суммарной задержки 9 тактов и соответственно текущего 10-го такта.

Транзит с задержкой на 2 такта использует один такт для удвоения числа, а другой – для увеличения суммарной задержки до 11 тактов. Оба такта могут также быть использованы для умножения числа на 4 при сохранении исходной суммарной задержки. Умножение на 8 выполняется уменьшением суммарной задержки на один такт.

Важную роль играет операция генерации константы. Константы используются не только в качестве операндов при вычислении выражений, но также как управляющие коды. Логическое умножение или логическое сложение операнда с константой позволяет в нем маскировать отдельные разряды.

Пусть, например, необходимо в 8-битовом коде *A*

$a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1$

выделить младшую тетраду, т. е. получить код *C*:

$0 0 0 0 a_4 a_3 a_2 a_1$.

Код *A* поступает на ОВС с севера, имея суммарную задержку 4 такта, а результат необходимо выдать через два такта на юг.

На рис. 4 показана ОВС, которая выполняет описанную операцию. Она содержит два ОЭ:

- ОЭ 1, обнуляющий старшую тетраду кода *A* путем выполнения операции логического умножения этого кода на константу;
- ОЭ 2, генерирующий эту константу $0 0 0 0 1 1 1 1$.

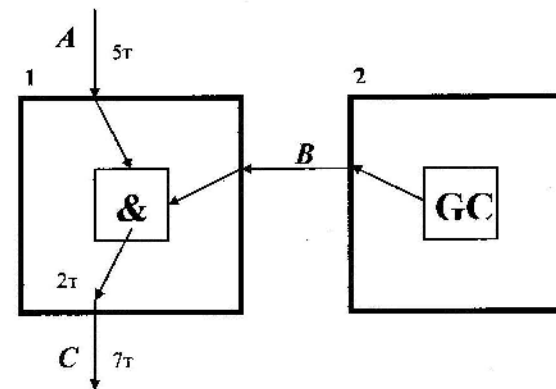


Рис. 4. Использование константы для выделения младшей тетрады числа

Тогда для ОЭ 1 $KOP = 000_2$, $Y = 1$, $A1 = 10_2$, $A2 = 11_2$, $A4 = 00_2$. Код *A3* можно определить произвольно, например, $A3 = 00_2$, а коды *A5* и *A6* не должны накладывать транзит на результат, т. е. *A5*, *A6* не равны *A4*, например, $A5 = 11_2$, $A6 = 11_2$.

Перейдем к программированию ОЭ 2. При записи константы в поля команды необходимо учитывать следующее:

- константа повторяется каждые 8 тактов;
- бит константы, поступивший на выход ОЭ, также записывается в левый разряд поля $A2$.

Поэтому для определения положения константы в команде ОЭ необходимо:

- привести константу к моменту начала работы ОВС;
- записать приведенную константу, начиная с младшего бита, в поля $A2 \div A5$ команды в порядке: левый разряд поля $A2$, правый разряд поля $A5$ и далее справа налево к правому разряду поля $A2$, т. е. в последовательности, показанной на рис. 5:

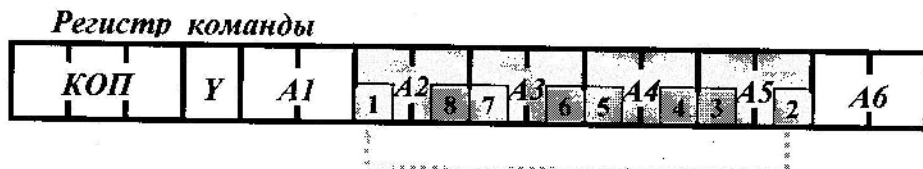


Рис. 5. Запись константы в регистр команды ОЭ

Приведение константы удобно выполнять по *временным диаграммам* работы ОЭ. Для данного примера временные диаграммы работы ОЭ показаны на рис. 6.

Временные диаграммы работы ОЭ 1 и 2

Имена диаграмм	Такты													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A					a1	a2	a3	a4	a5	a6	a7	a8		
B	0	0	0	0	1	1	1	1	0	0	0	0	1	1
C					←2т→		a1	a2	a3	a4	0	0	0	0

Рис. 6. Временные диаграммы работы ОЭ 1 и 2

Временные диаграммы показываются для входов ОВС и всех используемых выходов ОЭ и называются по именам или номерам этих точек схемы. В данном случае входом ОВС является северный вход ОЭ 1, названный по имени данного A , а используемыми выходами ОЭ являются западный выход ОЭ 2 и южный выход ОЭ 1, с именами константы B и результата C .

Столбцы диаграмм отмеряют такты работы ОВС. Первая диаграмма показывает, что в такте 5 в данную точку (северный вход ОЭ 1) поступил младший бит $a1$ кода A , в такте 6 – следующий бит $a2$ и т. д. до поступления в такте 12 старшего бита $a8$.

Константа B (в жирной рамке) записывается под разрядами данного A , что определяет их одновременный приход на входы ОЭ 1, причем в столбцах разрядов младшей тетрады указываются единичные значения, а напротив разрядов старшей тетрады – нулевые. Логическое умножение кода A на константу B (в ОЭ 1) определяет результат C с сохранением младшей и обнулением старшей тетрады кода A , а также сдвигом на два такта. Приведенная константа помечена серым цветом. Она в соответствии с рис. 5 определяет поля $A2 \div A5$ команды:

$$A2.A3.A4.A5 = 01111000_2;$$

Символ "." в записи поля константы обозначает операцию конкатенации, т. е. объединение отдельных полей в единое.

Остальные поля команды для ОЭ 2 принимают следующие значения $КОП = 110$; $Y = 0$; $A6 = 01_2$. Код $A1$ определяется произвольно, например, $A1 = 00_2$.

Коды команд для ОЭ 1 и 2 приведены в табл. 2:

Таблица 2

Коды команд ОЭ ОВС

ОЭ	КОП	Y	A1	A2	A3	A4	A5	A6
1	000	1	10	11	00	00	11	11
2	110	0	00	01	11	10	00	01

Рассмотрим умножение числа на константу.

Как было показано в первом примере умножение на константу вида 2^L выполняется с использованием для этого L тактов задержки при сохранении суммарной задержки данного.

Каждый такт удваивает число. Любая другая константа является суммой констант указанного вида. Например, $5 = 2^2 + 2^0$.

На рис. 7 показана ОВС, выполняющая умножение двоичного числа A на константу 5.

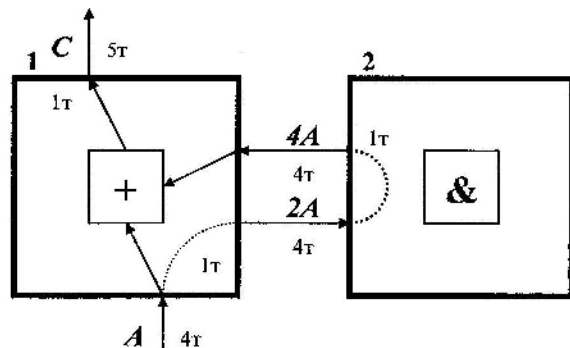


Рис. 7. Выполнение на ОБС умножения на константу

Коды команд для ОЭ 1 и 2 приведены в табл. 3, а временные диаграммы показаны на рис. 8

Таблица 3

Коды команд ОЭ ОБС

ОЭ	КОП	Y	A1	A2	A3	A4	A5	A6
1	100	0	00	11	00	10	00	11
2	000	0	00	00	01	00	00	01

Имена диаграмм	Такты													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A				a1	a2	a3	a4	a5	a6	a7	a8			
I_{1T}				0	a1	a2	a3	a4	a5	a6	a7	a8		
2_{1T}				0	0	a1	a2	a3	a4	a5	a6	a7	a8	
C				1	a1	a2	x	x	x	x	x	x	x	x

Рис. 8. Временные диаграммы работы ОЭ 1 и 2

Выходы транзитов названы номерами ОЭ с указанием их задержек: I_{1T} и 2_{1T} . Символом "x" обозначены разряды числа C, получаемые при сложении разрядов чисел A и $4A$.

Рассмотрим деление числа на константу вида 2^L . Эта операция выполняется как обратная к умножению на указанную константу, т. е. путем увеличения суммарной задержки на L тактов, но с предварительным обнулением L младших разрядов, что для $L=1$ показано на рис. 9.

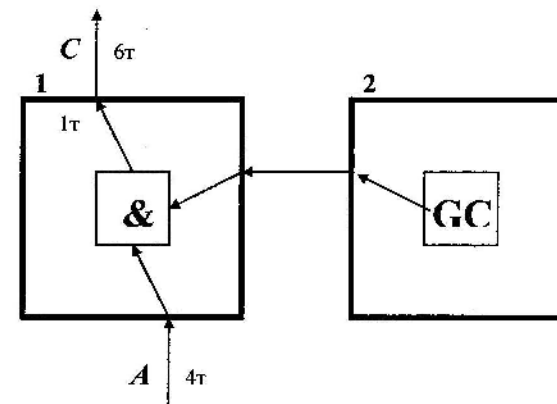


Рис. 9. Выполнение на ОБС деления числа на константу 2^L

Коды команд для ОЭ 1 и 2 приведены в табл. 4, а временные диаграммы показаны на рис. 10.

Таблица 4

Коды команд ОЭ ОБС

ОЭ	КОП	Y	A1	A2	A3	A4	A5	A6
1	000	0	00	11	00	10	00	00
2	110	0	00	11	11	10	11	01

Имена диаграмм	Такты													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A				a1	a2	a3	a4	a5	a6	a7	a8			
2	1	1	1	0	1	1	1	1	1	1	1	0	1	1
C				1	1	a2	a3	a4	a5	a6	a7	a8	0	

Рис. 10. Временные диаграммы работы ОЭ 1 и 2

4. Формирование ОВС и ее программирование

Выпускаемые промышленностью **микросхемы** ОВС содержат матрицу 3×3 ОЭ, из которых набираются ОВС заданного размера. Микросхемы имеют три входа и три выхода программирования, через которые последовательно программируются ОЭ строк. Строка ОЭ является **элементарным звеном** цепи программирования.

После составления программы для отдельных ОЭ определяются цепи программирования ОВС. Последовательное соединение регистров команды всех ОЭ ОВС позволяет программировать ОВС через один вход программирования, однако требует для этого максимальное количество тактов. Программирование одной строки ОЭ выполняется (в соответствии с разрядностью регистра команды) за $3 \cdot 16$ тактов, а последовательное программирование n микросхем – за $3 \cdot 3 \cdot 16 \cdot n$ тактов. При меньшем времени программирования выполняется несколько последовательных цепей, которые программируются параллельно через несколько входов программирования. Наименьшее **время программирования** достигается при равномерном распределении звеньев между цепями программирования.

На рис. 11 показан пример определения цепей программирования для ОВС, составленной из двух микросхем 1 и 2, а также времени программирования 128 тактов.

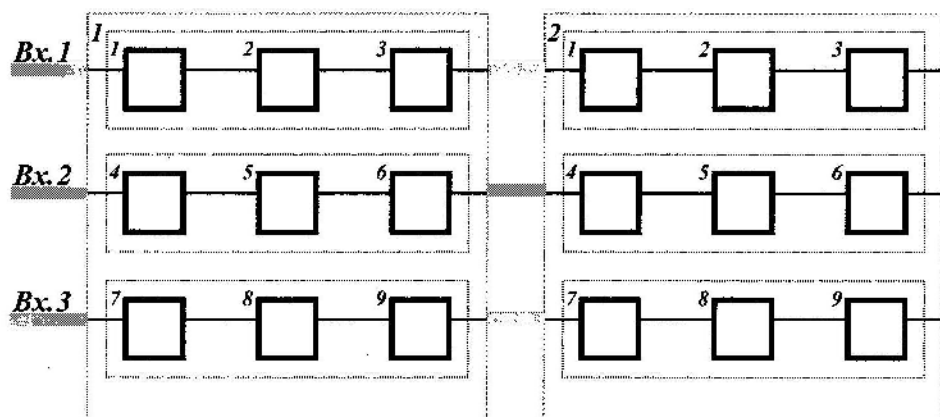


Рис. 11. Цепи программирования ОВС

Крупным и мелким пунктиром обозначены соответственно микросхемы и их строки, которые в качестве элементарных звеньев составляют три цепи программирования со входами **Вх. 1**, **Вх. 2**, **Вх. 3** и временем программирования 96 тактов. Линии соединения звеньев в цепи программирования показаны серым цветом.

Программа для ОВС представляется двумерным массивом бит с количеством строк, равным количеству входов программирования. В каждой строке записываются коды команд ОЭ в порядке их очередности в цепи программирования. Коды команд ОЭ удобно представлять в 16-ричной системе счисления Q_{16} . Для неиспользуемых ОЭ можно указывать нулевые коды команд.

Например, для ОВС, показанной на рис. 11, и кодов команд ее используемых ОЭ 13 ÷ 16, 24, 27, представленных в табл. 5, программа имеет вид трехмерного массива, записанного в табл. 6 (с указанием номеров ОЭ напротив их кодов команд). Номера ОЭ ОВС составлены слева направо из номера микросхемы и номера ОЭ внутри микросхемы.

Таблица 5

Коды команд ОЭ ОВС

ОЭ	КОП	У	А1	А2	А3	А4	А5	А6	Q_{16}
13	000	0	00	00	10	10	01	00	00A4
14	000	1	00	00	01	00	11	00	104C
15	100	1	01	01	00	11	00	0	9530
16	100	0	01	11	10	10	00	11	87A3
24	100	0	01	00	01	01	11	00	845C
27	001	1	10	10	00	10	00	00	3A20

Таблица 6

Программа для ОВС

Цепь	ОЭ	11	12	13	21	22	23
1	Команды	0000	0000	00A4	0000	0000	0000
Цепь	ОЭ	14	15	16	24	25	26
2	Команды	104C	9530	87A3	845C	0000	0000
Цепь	ОЭ	17	18	19	27	28	29
3	Команды	0000	0000	0000	3A20	0000	0000

4. Представление данных и алгоритмы их обработки

Входные данные для ОВС заданы в двоично-кодированных системах счисления, например, в (2-5)-й или (2-10)-й, т. е. в общем случае, (2- S)-й системе счисления, в которой цифры числа системы счисления с основанием S кодируются их двоичными эквивалентами. Входные данные являются положительными двухтетрадными двоично-кодированными числами, т. е. каждая цифра описывается тетрадой двоичных разрядов. Обработка чисел состоит в их переводе в двоичную систему счисления и выполнении над ними арифметических операций умножения и деления на константы, а также вычитания с использованием обратного или дополнительного кода.

Для перевода двоично-кодированного числа в двоичную систему счисления необходимо выделить его младшую L и старшую H тетрады, а затем вычислить: $H \cdot S + L$.

Например, в двоично-кодированном числе

$$a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0,$$

выделяются тетрады L и H :

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ a_4 & a_3 & a_2 & a_1 \end{array} a_0,$$

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ a_8 & a_7 & a_6 & a_5 \end{array} a_4.$$

Для $S=5$ произведение $H \cdot S$ определяется выражением $H + 4H$, т. е. суммой:

$$0000a_8a_7a_6a_5 + 00a_8a_7a_6a_5000.$$

При нахождении разности положительных двоичных чисел вычитаемое рассматривается как отрицательное слагаемое, которое необходимо в соответствии с заданием перевести в обратный или дополнительный код. Такое преобразование обеспечивается, соответственно, инвертированием числа или инвертированием числа с предварительным вычитанием единицы, выполняемым путем сложения с константой 11111111_2 .

Перевод результата

$$3n \ c7 \ c6 \ c5 \ c4 \ c3 \ c2 \ c1 \ 2$$

из обратного кода в прямой код, взятый по абсолютной величине, т. е. с нулевым знаком, выполняется путем сложения по модулю два разрядов числа со значениями его знакового разряда $3n$. Последовательность знаков следует в ОВС за старшим разрядом числа. Для выполнения данной операции число дублируется и сдвигается не менее, чем на 7 позиций, относительно первого значения:

$$\begin{array}{cccccccccccccccc} \oplus & 3n & 3n & 3n & 3n & 3n & 3n & 3n & 3n & c7 & c6 & c5 & c4 & c3 & c2 & c1 & 2 \\ & 3n & c7 & c6 & c5 & c4 & c3 & c2 & c1 & 2 & & & & & & & \\ & 0 & x & x & x & x & x & x & x & x & 2 & & & & & & \end{array}$$

Сложение знака с самим собой определяет нулевое значение в позиции знака прямого кода.

В переводе результата из дополнительного кода в прямой (без знака) перед описанной операцией выполняется вычитание знака на месте младшего разряда числа путем арифметического сложения дополнительного кода с кодом, составленным из его знаков: $3n \ 3n \ 3n \ 3n \ 3n \ 3n \ 2$. Код знаков получается аналогично описанному выше. Следует отметить, что при арифметическом сложении и сложении по модулю два используются различные знаковые разряды: в первом случае – знак дополнительного кода, а во втором случае – знак результата арифметического сложения.

При выполнении всех операций считать, что все промежуточные и конечный результаты (с учетом знакового разряда) не выходят за рамки 8-битового кода.

Вычисления в ОВС выполняются последовательно-параллельно. Поэтому для их описания необходимо представить структуру вычислительного процесса и составляющие его алгоритмы. Для решаемой задачи удобно выделить три алгоритма, описывающие последовательности действий, связанные с обработкой числа A , числа B и их совместной обработкой до получения окончательного результата соответственно.

Например, для выполняемых в обратном коде вычислений $C = 2 | A - 3B |$, где $A=A_{2-5}$ и $B=B_{2-10}$, по алгоритмам вычисляются:

- $A^* = ((A_{2-5})_2)_{обр}$;
- $B^* = (- (3 (B_{2-10})_2))_{обр}$;
- $C = 2 | (A^* + B^*)_{пр} |$.

Индексы “2”, “обр” и “пр” определяют перевод чисел в двоичную систему счисления, обратный и прямой коды

Схема алгоритма вычисления A^* показана на рис. 12.

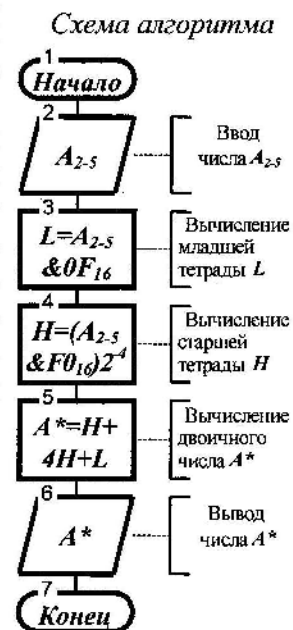


Рис. 12



На рис. 13 показана структура вычислительного процесса.

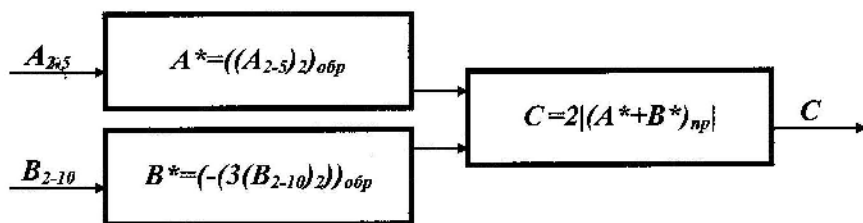


Рис. 13. Структура вычислительного процесса

5. Содержание контрольной работы

Темой контрольной работы является выполнение арифметико-логических операций на ОВС.

В процессе выполнения контрольной работы студент изучает функциональное описание ОВС, определяет вариант задания на контрольную работу, разрабатывает машинные алгоритмы побитовой обработки данных, переносит их на плоскость ОВС, определяет ее структуру, цепи программирования и программу, являющиеся результатом выполнения работы.

Вариант задания определяется по номеру студента на потоке.

$$K = (n - 1) \cdot 30 + m,$$

где m – номер студента в группе (по журналу);

n – порядковый номер группы.

Задания описаны в таблицах 7÷9 и выбираются из них по параметрам X , Y и Z , вычисленным по формулам:

$$X = K \bmod 11,$$

$$Y = K \bmod 3,$$

$$Z = K \bmod 7.$$

По параметру X выбирается формула вычисления результата C , реализуемая на ОВС.

Таблица 7
Определение параметра X

X	Формулы вычисления C
0	$3 A/2 - B/4 $
1	$ 3A/4 - B $
2	$ A - 5B/4 $
3	$ 3A - B / 4$
4	$ 3A - 4B $
5	$ 5A - 3B / 2$
6	$5 A - 2B / 4$
7	$ 5A/2 - 2B $
8	$ 7A/4 - 2B $
9	$7 A - B / 4$
10	$ 6A - 5B $

По параметрам Y и Z определяются системы счисления для операндов A и B , используемых в формуле вычислений.

Для нечетных K вычисления в формуле осуществляются в обратном коде, а для четных K – в дополнительном коде.

Направления поступления операндов A и B , а также выдачи результата C определяются указанными выше двухразрядными кодами, составляющими вместе (справа налево) шестизначный код V , вычисляемый в двоичной системе счисления по формуле

$$V = K \bmod 64.$$

ОВС составляется из описанных выше микросхем. После задания команд для отдельных ОЭ определяются цепи программирования, исходя из требования ко времени программирования в тактах

$$T_{np} \leq (X + Y + Z + 6) \cdot 16.$$

Например, студент второй группы, 8-й по списку журнала группы имеет $n = 2$, $m = 8$, $K = 38$, $X = 5$, $Y = 2$, $Z = 3$, K – четное, $V = 38 = 10\ 01\ 10_2$, $T_{np} = 16 \cdot 16 = 256$.

Таким образом, ставится задача выполнения вычислений по формуле 5 (табл. 7)

$$C = | 5A - 3B | / 2,$$

в которой числа A и B представляются в $(2-7)$ -й (табл. 8) и $(2-12)$ -й (табл. 9) системах счисления, т. е. $A = A_{2-7}$, $B = B_{2-12}$, обрабатываются в дополнительном коде и поступают с севера и запада соответственно, а результат C выдается на север. Программирование ОВС необходимо выполнить за время, не превышающее 256 тактов.

При выполнении контрольной работы необходимо стремиться к получению ОВС с минимальным количеством микросхем и наименьшим временем программирования, которое возможно при минимальном количестве входов программирования.

Таблица 8
Определение параметра Y

Y	Система счисления
0	2-5
1	2-6
2	2-7

Таблица 9
Определение параметра Z

Z	Система счисления
0	2-9
1	2-10
2	2-11
3	2-12
4	2-13
5	2-14
6	2-15

Контрольная работа должна содержать следующие разделы

- выбор варианта задания с определением его параметров и постановкой задачи на выполнение контрольной работы;
- структуру вычислительного процесса и схемы его алгоритмов, а также их описание;
- схемы ОВС для разработанных алгоритмов, временные диаграммы работы ОЭ ОВС и их описание;
- полную схему ОВС, разбитую на микросхемы, коды команд ОЭ, схему цепей программирования ОВС и их описание;
- программу ОВС;
- заключение о выполненной работе;

Дополнительно об организации и работе ОВС можно прочитать в [1]. Представление чисел в ЭВМ и машинные способы их обработки изложены в [2 ÷ 4].

Литература

1. Евреинов Э.В. Однородные вычислительные системы, структуры и среды. – М.: Радио и связь, 1981.
2. Рабинович З.Л., Раманаускас В.А. Типовые операции в вычислительных машинах. – Киев: Техника, 1980.
3. Савельев А.Я. Прикладная теория цифровых автоматов. – М.: Высш. шк. 1987.
4. Самофалов К.Г., Романкевич А.М., Валуцкий В.Н., Каневский Ю.С., Пиневиц М.М. Прикладная теория цифровых автоматов. – Киев: Вища школа, 1987.