

ЛАБОРАТОРНАЯ РАБОТА 5

УПАКОВКА БИТОВЫХ ГРУПП

ЦЕЛЬ РАБОТЫ

1. Изучение логических команд.
2. Изучение команд сдвига.
3. Изучение алгоритмов упаковки битовых групп.
4. Изучение обработки чисел различной длины.

ЗАДАНИЕ

В задании графически изображен формат 32-битового двоичного числа. В каждом поле представлено название этого поля, а под соответствующим полем - его размер в битах. Выполнить упаковку распакованных двоичных групп, для получения результирующего 32-битового числа такого формата учитывая следующие требования:

- название каждой переменной является одновременно названием соответствующего поля в упакованном 32-битовом коде, содержащем это поле;
- существенные биты каждого распакованного поля располагаются в младших разрядах соответствующей переменной, в то время как старшие разряды могут содержать вредную информацию, поэтому перед упаковкой битовых групп старшие разряды надо очищать;
- для размещения каждого поля использовать стандартную битовую группу минимальной длины (байт, слово или длинное слово).

ИСХОДНЫЕ ДАННЫЕ

Формат исходных данных для упаковки битовых групп определить самостоятельно на основании формата данных упакованных данных указанных в задании.

Var1	8	7	0	Var3	15	10	9	0	
	?	sssssss			?	nnnnnnnnnn			
Var2	8	3	2	0	Var4	15	12	11	0
	?	zzz				?	yyyyyyyyyyyyyy		

ТРЕБУЕМЫЙ РЕЗУЛЬТАТ

Формат упакованного числа указан в задании. Необходимо указать сверху каждого поля нумерацию битов. Название переменных определяет название соответствующего поля, а количество переменных - количество битовых полей в упакованном числе.

	31	25	24	22	21	12	11	0
Packing	var1			var2		var3		var4
	7			3		10		12

Тестовые примеры для упаковки приведены в таблице 1.

Таблица 1 – Тестовые примеры для упаковки

Номер	Var1	Var2	Var3	Var4	Packing
1					AAAA AAAA
2					5555 5555
3					1234 5678
4					FFFF FFFF

Внимание. При составлении тестовых примеров можно использовать тесты, используемые при распаковке битовых групп. В таком случае в качестве исходных значений использованы значения *существенных битов*.

Внимание. Помимо тестов, содержащих значения существенных битов, следует разработать тесты, содержащие единичные биты в старших несущественных разрядах. Например, рассмотрим поле X состоящее из трех битов (они располагаются в младших разрядах). Возможны такие тесты для проверки этих значений: **00000 101**, **00000 110**, **00000 100** и др. Возможны такие тесты, содержащие единичные биты в несущественных разрядах: **01101 101**, **11100 110**, **01011 100** и др. Полученные таким образом битовые группы следует записать в 16–м коде. Результат выполнения таких тестов результат должен быть таким же, как и в случае с нулевыми значениями в старших битах.

ХОД РАБОТЫ

Составить тестовые примеры для упаковки, которые следует оформить в виде таблицы (см. выше). Исходные данные и требуемый результат должны приводиться в 16-й системе счисления.

Ввод и вывод данных выполняется в 16-й системе счисления.

Программа должна быть зациклена.

Содержание отчета

1. Титульный лист.
2. Задание.
3. Исходные данные.
4. Требуемый результат.
5. Алгоритм решения задачи на псевдокоде. Схема алгоритма.
6. Текст программы с комментариями.
7. Тестовые примеры.

Варианты заданий

1

mod	kop	reg	scale
9	5	11	7

2

mod	kop	w_b	disp
6	8	1	17

3

kop	mod1	reg1	dst
5	14	6	7

4

mod	kop	w_b	reg1	reg2
7	12	2	7	4

5

kop	reg	mod2	reg2
12	6	5	9

6

kop	len	mod	reg
10	7	9	6

7

src	dst_t	dst_r	b_w	kop
11	6	5	1	9

8

src_t	src_r	dst	b_w	kop
2	7	10	1	12

9

dst_r	cnt	kop	b_w
6	6	18	2

10

kop	s	w	mod	reg	r_m
12	2	2	7	3	6

11

kop	w	reg	mod	kop1	r_m
9	2	4	6	6	5

12

kop	cond	mod	r_m	scale
10	7	4	6	5

13

mod	reg	r_m	kop	s	w
8	6	5	9	2	2

14

mod	reg1	r_m	kop	w	reg2
4	6	12	4	1	5

15	mod	r_m	kop	cond	range	
	2	6	12	10	2	
16	mod	reg	r_m	kop	s	w
	8	6	5	9	2	2
17	mod	reg1	r_m	kop	w	reg2
	4	6	12	4	1	5
18	ind	offset	trank	diff	scale	
	7	4	7	9	5	
19	time	year	weight	mod	send	whole
	10	7	6	7	1	1
20	paris	romul	city	viene	krakov	krit
	6	4	10	3	2	7
21	rand	sum	dif	cond	division	
	4	5	2	11	10	
22	monkey	gus	dog	cat	giraff	enymals
	3	5	11	7	5	1
23	src_t	src_r	dst	b_w	kop	
	2	10	7	3	9	
24	dst_r	cnt	kop	b_w		
	7	9	12	4		
25	kop	s	w	mod	reg	r_m
	10	7	5	3	4	3
26	kop	w	reg	mod	kop1	r_m
	5	2	7	9	3	6
27	kop	cond	mod	r_m	scale	
	7	11	6	3	5	