**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)**

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | Информатика и системы управления |
| КАФЕДРА | Информационная безопасность (ИУ8) |

АППАРАТНЫЕ СРЕДСТВА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**Лабораторная работа №3 на тему:**

«**Микропрограммирование арифметических операций**»

**Выполнил:**

Евула А. С.

**Проверил:**

Рафиков А. Г.

**Группа:**

ИУ8-63

Москва, 2021

# **Цель работы:**

* Изучение способов представления чисел в микроЭВМ и алгоритмов арифметических операций;
* Микропрограммирование операций в системе микрофункций процессора К1804ВС1.

# **Представление чисел в микроЭВМ.**

Отрицательные числа обычно представляются в виде дополнений до основания системы счисления. При операциях над числами в микроЭВМ обычно полагают, что числа имеют следующий вид:

D = dn-1dn-2…d1d0

то есть точка находится справа и числа являются целыми.

В общем случае дополнение любого n – разрядного числа D до основания b системы счисления можно получить путем вычитания D из bn. Если D находится в пределах от 1 до bn –1, то при вычитании получается другое число в тех же пределах. Если D=0, то результат вычитания равен bn и имеет вид 100..0 при общем числе разрядов, равном (n+1). Отбросив цифру старшего разряда, получим 0. Следовательно, в системе представления чисел дополнением до основания системы счисления существует только одно представление 0.

В десятичной системе счисления дополнение до основания есть дополнение до десяти, которое можно получить путем вычитания n – разрядного числа из 10n.

/*Пример*. Десятичное число А = 1849. Дополнение до десяти [А]доп = 10000-А= 8151/

Для двоичных чисел дополнение до основания системы счисления называется дополнением до двух. В системе представления дополнением до двух, или в дополнительном коде, число является положительным, если значение старшего разряда dn-1 = 0, и отрицательным, если dn-1 = 1. Десятичный эквивалент двоичного числа, представленного дополнением до двух, вычисляется так же, как и для числа без знака, за исключением того, что вес старшего разряда равен – 2(n-1) , а не +2(n-1). Представляемые числа находятся в диапазоне от – 2(n-1) до +2(n-1)–1.

В системе представления чисел неполным дополнением до основания дополнение n – разрядного числа D получается путем его вычитания из bn-1. Для двоичных чисел неполное дополнение называется дополнение до единицы или обратным кодом. При вычислении десятичного эквивалента числа, записанного как дополнение до единицы, старшему разряду приписывается вес – (2(n-1) – 1), а не –2(n-1). Представляемые числа находятся в диапазоне от – (2(n-1) – 1) до + (2(n-1)– 1). Нуль имеет два представления - положительный нуль (00..00) и отрицательный нуль (11..11).Представления положительных чисел в системах с дополнением до единицы и до двух совпадают, тогда как представления отрицательных чисел отличаются на 1.

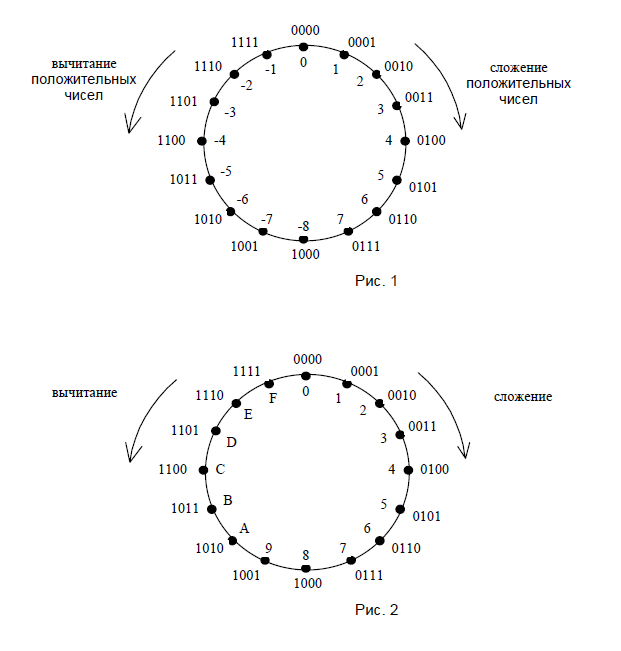
# **Сложение и вычитание чисел в дополнительном коде.**

Графическое представление 4–разрядных двоичных чисел в дополнительном коде приведено на рис.1. Сложение с положительными числами легко интерпретировать, перемещая указатель по часовой стрелке на + n позиций; вычитание (-***n***), перемещая указатель против часовой стрелки, или перемещая по часовой стрелке на (16–***n***) позиций, что равносильно замене вычитания сложением с дополнением числа до двух. Если при сложении получают результат, который выходит за пределы диапазона представляемых чисел, то имеет место переполнение.

Правило выявления переполнения. При сложении переполнение происходит только в том случае, если слагаемые имеют одинаковые знаки, а знак суммы отличается от знака слагаемых. Правило переполнения можно сформулировать иначе, используя понятие переносов, возникающих при сложении. Переполнение возникает, если значения переносов в знаковый разряд и из знакового разряда различны. Из анализа рис.1 следует, что переполнение возникает при сложении в случае, если указатель перейдет границу между позициями +7 и –8.

Числа в дополнительном коде складываются и вычитаются так же, как числа без знака той же длины. Поэтому для выполнения операций над числами обоих типов необходим всего один тип команды сложения или вычитания. Различие заключается лишь в том, что результаты операций интерпретируются по-разному в зависимости от того, какими числами оперирует ЭВМ: числами со знаком (то есть от –8 до +7) или без знака (от 0 до 15).

На рис.2 приведено графическое представление 4–разрядных двоичных чисел без знака. Из него видно, что двоичные кодовые комбинации занимают те же позиции, что и на рис.1, а сложение и вычитание можно осуществить, перемещая указатель на **n** позиций в том или ином направлении. При сложении чисел без знака результат выходит за пределы диапазона представления при переходе границы между 15 и 0. В этом случае говорят о возникновении переноса из старшего разряда. При вычитании чисел без знака результат выходит за пределы диапазона при переходе границы между 0 и 15. В этом случае возникает заем. Но так как вычитание ***n*** можно заменить сложением с дополнительным кодом числа ***n***, равным (16 – ***n***), то заем возникает при отсутствии переноса.

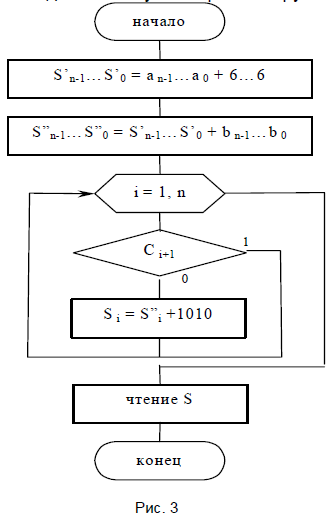


# **Двоично – десятичное сложение – вычитание.**

При сложении двух двоично – десятичных чисел A = an-1an-2..a1a0 и B = bn-1bb-2..b1b0 поступают следующим образом. Если оба операнда имеют одинаковые знаки, то выполняют сложение модулей этих чисел (|A| + |B|), а знаковый разряд сумм определяют по знаку одного из слагаемых. Если операнды имеют разные знаки, то предварительно знак суммы устанавливают по знаку первого операнда А. Затем производят вычитание модулей чисел (|A| - |B|). Если полученная разность больше 0, знак суммы сохраняется без изменений. Если разность меньше 0, следует найти дополнительный код разности и изменить знак суммы на противоположный.

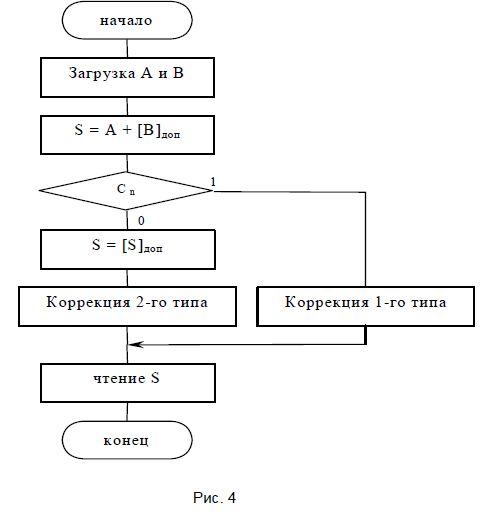
Операция сложения модулей (|A| + |B|) выполняется по алгоритму, схема которого приведена на рис. 3:

1. двоично – десятичный код первого операнда an-1an-2…a1a0 складывается с кодом 66..66, образуя первую промежуточную сумму S’n-1S’n-2…S’1S’0;
2. к полученной сумме прибавляется двоично – десятичный код 2-го операнда bn-1bn-2…b1b0, образуя вторую промежуточную сумму S”n-1S”n-2…S”1S”0;
3. выполняется потетрадно коррекция результата. Правило коррекции формулируется следующим образом: если в результате второго сложения перенос из i – ой тетрады отсутствует (ci+1 = 0), то к S”I прибавляется код 10102 или А16, что соответствует вычитанию 6. При возникновении переноса из i – ой тетрады коррекция не выполняется (или прибавляется код 0000), а полученный результат S”I является истинным. При выполнении коррекции потетрадные переносы в двоичном сумматоре блокируются.



Вычитание модулей (|A| - |B|) выполняют по алгоритму, схема которого представлена на рис. 4.

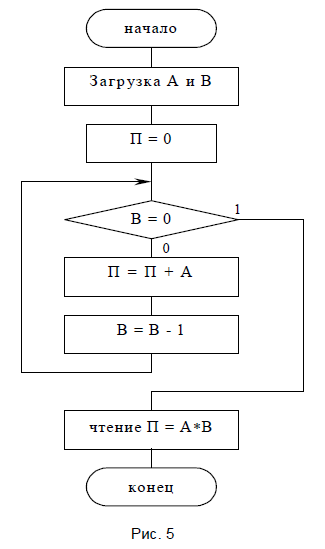
1. Операнд В представляют двоично - десятичным дополнением до десяти;
2. Двоично – десятичный код А складывают с дополнительным кодом В. Если в результате сложения образуется перенос из старшей тетрады (сn = 1), результат является положительным. При отсутствии переноса результат является отрицательным и его следует перевести в дополнительный код.
3. Коррекция положительного результата осуществляется по правилу, сформулированному для сложения модулей чисел. Коррекция отрицательного результата выполняется иначе: Если имел место перенос из i – ой тетрады при сложении A + [B]доп, то к i – ой тетраде прибавляется код 1010; если перенос отсутствует – прибавляется 0000 (см. приложение).

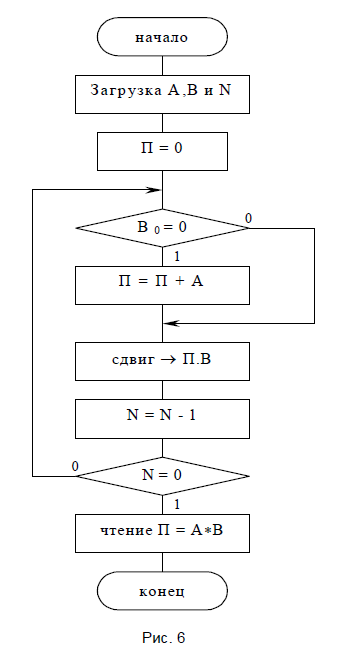


# **Умножение чисел без знака.**

Наиболее просто умножение можно выполнить по итерационной схеме алгоритма, изображенной на рис. 5. После загрузки множимого А и множителя В в регистры общего назначения и обнуления регистра произведения П производится анализ содержимого регистра множителя. Если В ≠ 0, то к сумме частичных произведений П прибавляется множимое А. Затем содержимое регистра множителя уменьшается на 1 и цикл умножения повторяется до тех пор, пока содержимое регистра множителя не окажется равным 0. При умножении n – разрядных сомножителей 2n – разрядное произведение размещают в двух регистрах. Данный метод умножения находит ограниченное применение в сравнительно несложных микропроцессорных системах.

На практике большое распространение имеют методы умножения путем сложения ряда сдвинутых относительно друг друга множимых, с учетом цифр множителя. Один из алгоритмов умножения, начиная с младших разрядов множителя, приведен на рис. 6. Этот алгоритм может быть использован для получения произведения двух двоичных чисел без знака. Количество итераций умножения N определяется числом разрядов множителя. Поскольку в процессе умножения на каждой итерации осуществляется сдвиг множителя В на 1 разряд вправо, на место освобождаемых разрядов можно записать выталкиваемые при сдвиге вправо разряды произведения П. При использовании n – разрядного сумматора или АЛУ исходные двоичные числа без знака не должны выходить за пределы диапазона от 1 до 2(n-1) -1.





# **Деление чисел без знака.**

Для типичного алгоритма деления делимым является двойное слово, а делителем – одинарное; частное и остаток получаются в виде одинарных слов. Если при выполнении такого деления окажется, что делитель равен 0 , или для представления частного потребуется более одного слова, то происходит переполнение. Последнее имеет место в том случае, если делитель больше или равен старшего слова делимого.

В качестве примера рассмотрим метод деления A/B без восстановления остатка. В этом случае алгоритм деления представляет итерационную процедуру, на каждой итерации которой производится либо вычитание делителя В, представленного в дополнительном коде, либо прибавление В, в зависимости от знака остатка, полученного на предыдущей итерации деления. Если полученный остаток был больше 0, при очередной итерации деления производится вычитание В; если остаток был меньше 0, производится прибавление В. Перед каждым вычитанием (или сложением) производят удвоение остатка путем сдвига влево. На начальной итерации деления делимое сдвигается на 1 разряд влево.

# **Задание для самостоятельной подготовки**

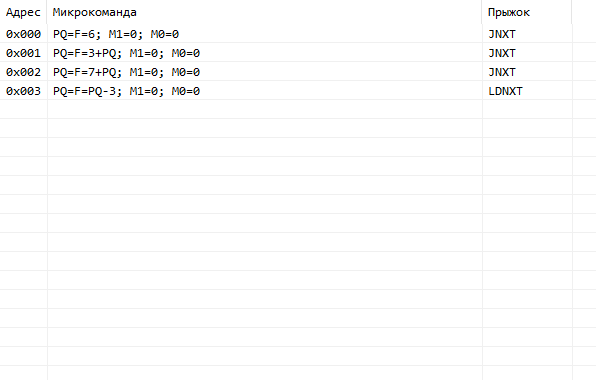
Составить схемы алгоритмов и подготовить микропрограммы по всем пунктам работы. Написать примеры для проверки работы микропрограмм.

# **Порядок выполнения работы**

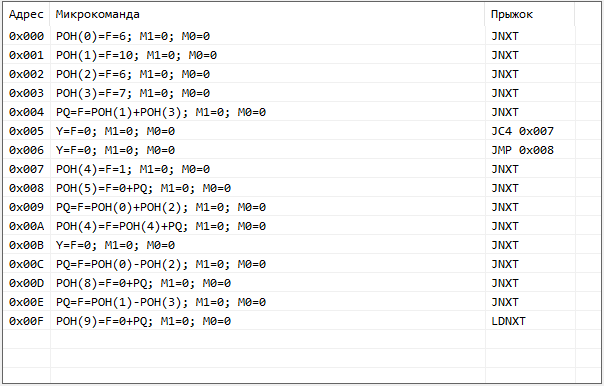
1. Выполнить операции сложения и вычитания двух 4 – разрядных чисел со знаком и без знака. Привести примеры образования признаков переноса, заема, переполнения и нуля
2. Разработать и выполнить микропрограммы сложения и вычитания 8 – разрядных чисел без знака.
3. Разработать и выполнить микропрограммы сложения и вычитания 8 – разрядных чисел со знаком. Отрицательные числа представлены в дополнительном коде.
4. Разработать и выполнить микропрограмму сложения модулей 2–разрядных двоично – десятичных чисел.
5. Разработать и выполнить в автоматическом режиме микропрограмму умножения 4-разрядных сомножителей без знака по схеме алгоритма на рис. 6.
6. Разработать и выполнить микропрограмму деления чисел без знака, полагая известным, что делитель всегда больше 0 и переполнение невозможно для заданных операндов.

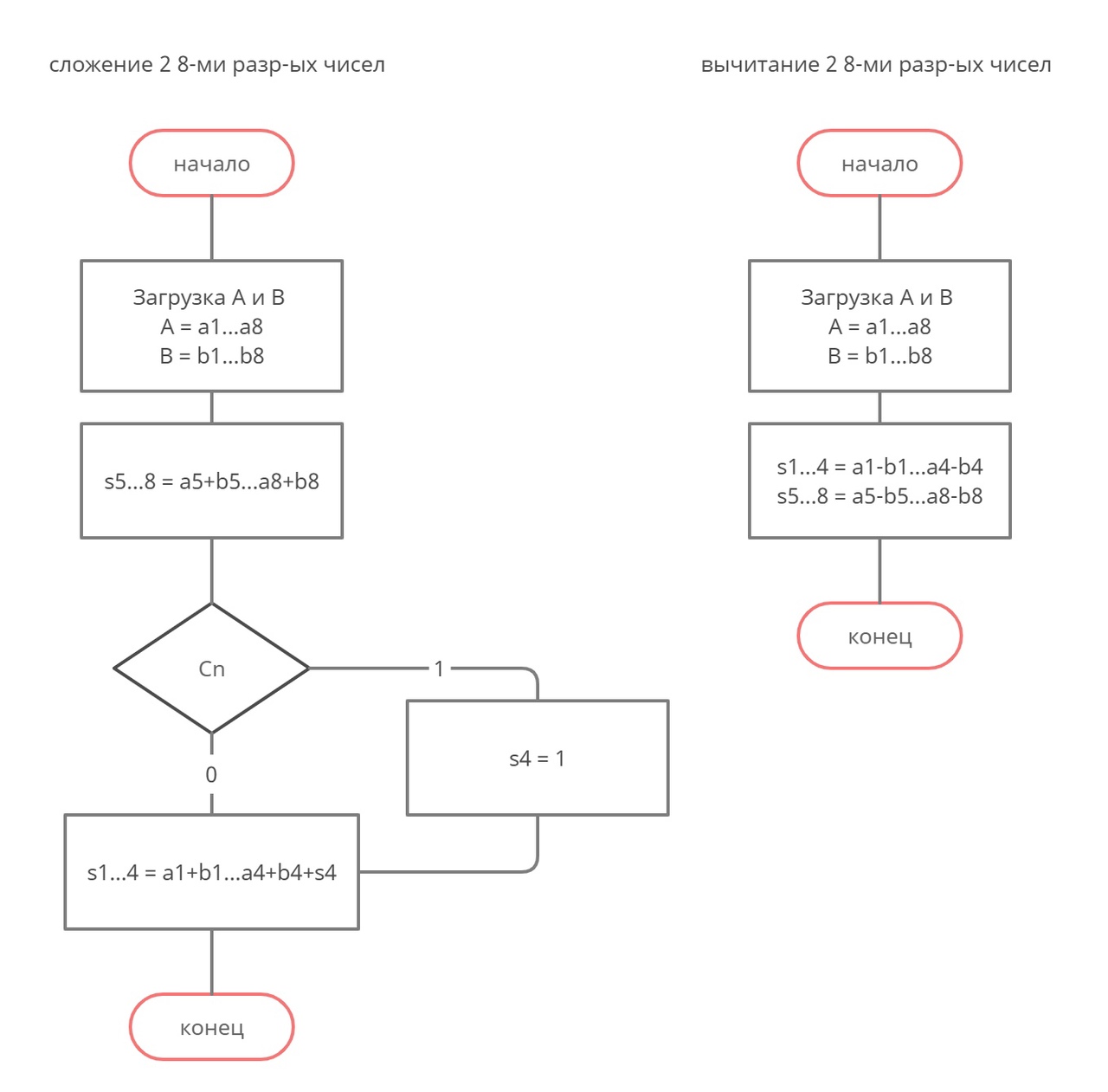
# **Практическая часть**

**Задание 1:**

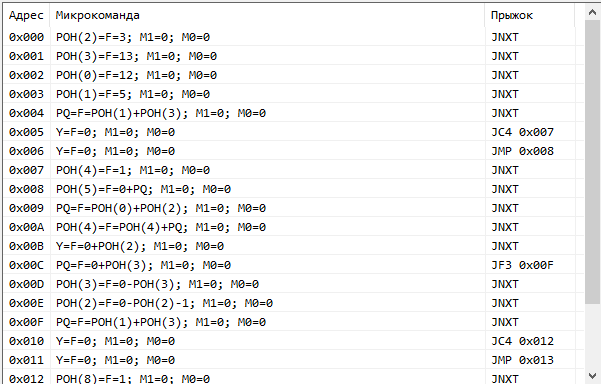
****

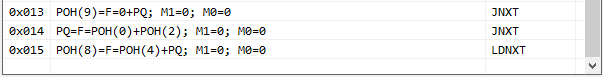
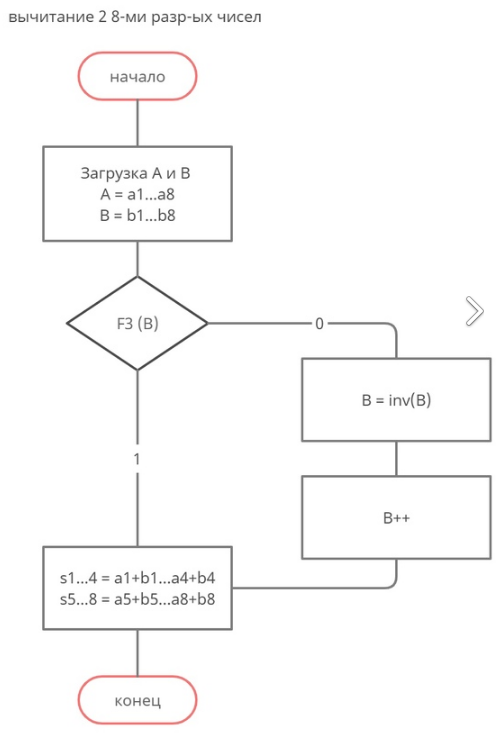
**Задание 2:**

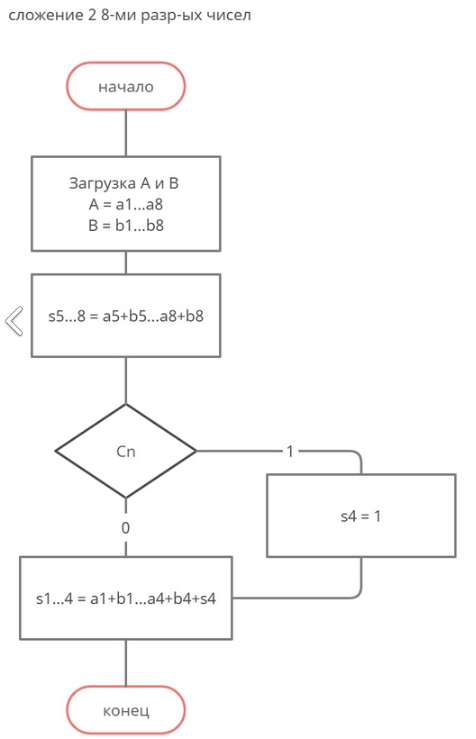
****

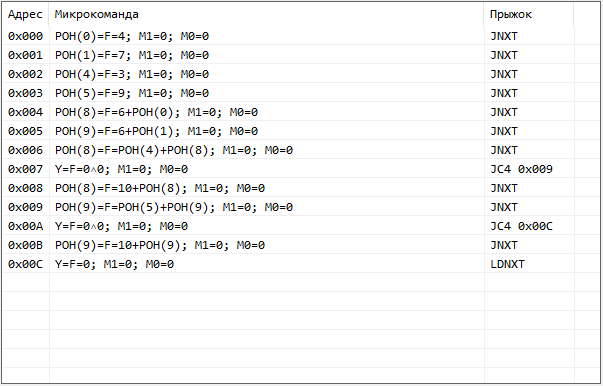
****

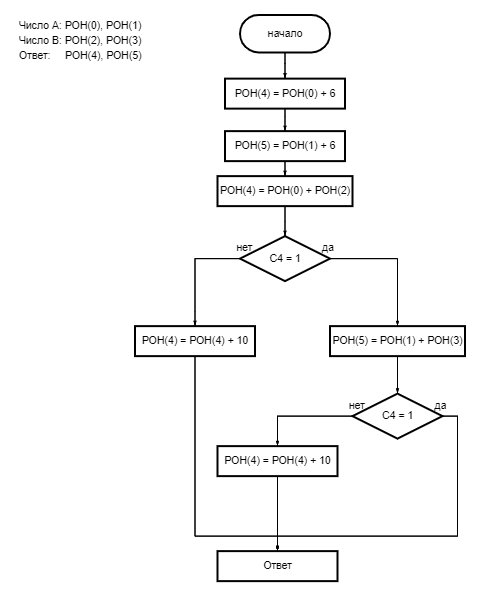
**Задание 3:**

****

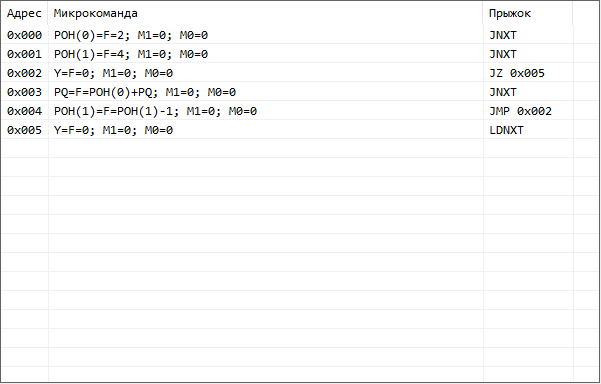
****

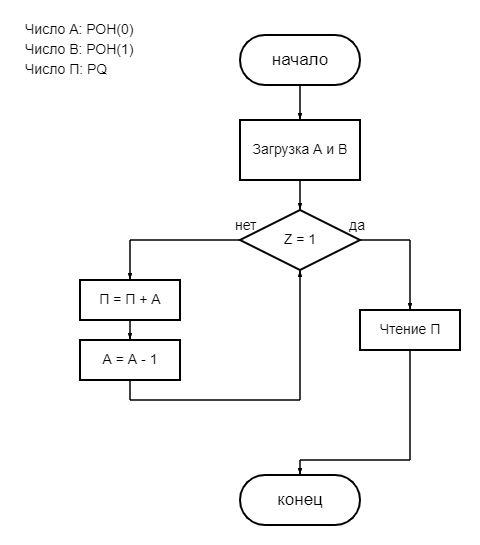
****

**Задание 4:** 

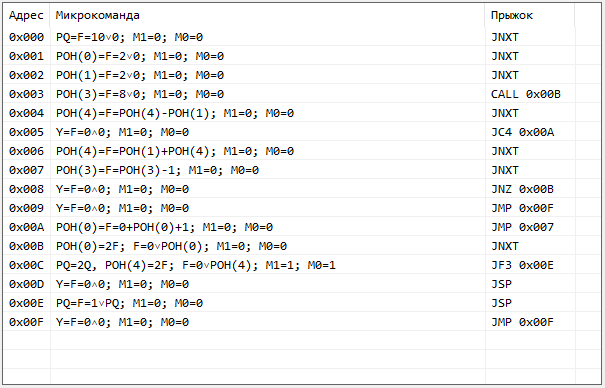
****

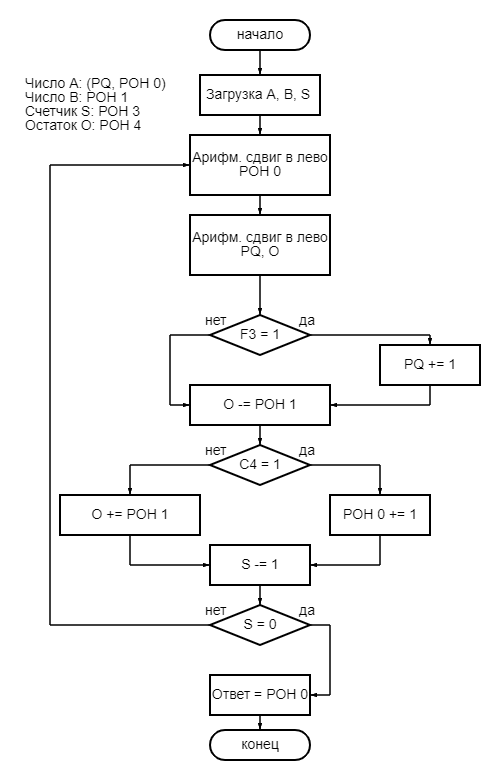
**Задание 5:**





**Задание 6:**



****