

ЛАБОРАТОРНАЯ РАБОТА № 2

МИКРОПРОГРАММИРОВАНИЕ АЛГОРИТМОВ

Цель работы: овладение методами микропрограммирования, разработка и отладка микропрограмм.

1. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Составить, ввести и отладить микропрограмму для одной из задач раздела 4. Решая задачу в режиме МИКРОКОМАНДА, записать и проанализировать изменения состояний основных элементов процессора (протокол решения).

2. МИКРОПРОГРАММИРОВАНИЕ АЛГОРИТМОВ

Микропрограммирование начинается с разработки содержательной граф-схемы алгоритма. Затем распределяются оперативная и регистровая память для хранения и преобразования данных.

Рекомендуется такой порядок микропрограммирования:

- написать программу алгоритма в терминах алгоритмического языка (ассемблер, Паскаль);
- произвести интерпретацию команд программы микрокомандами.

При этом интерпретация одной команды может потребовать нескольких МК и, наоборот, одна МК может интерпретировать более чем одну команду.

Пример. Подсчитать количество единиц в двоичном представлении числа.

Задача решается с помощью маски и логической операции «И». Если просматривать число с младших разрядов, то маска равна 0001h и число сдвигается вправо. Если со старших разрядов, то маска равна 8000h, а число сдвигается влево. Количество сдвигов зависит от разрядности числа. В качестве счетчика можно использовать любой РОН или РАСТ.

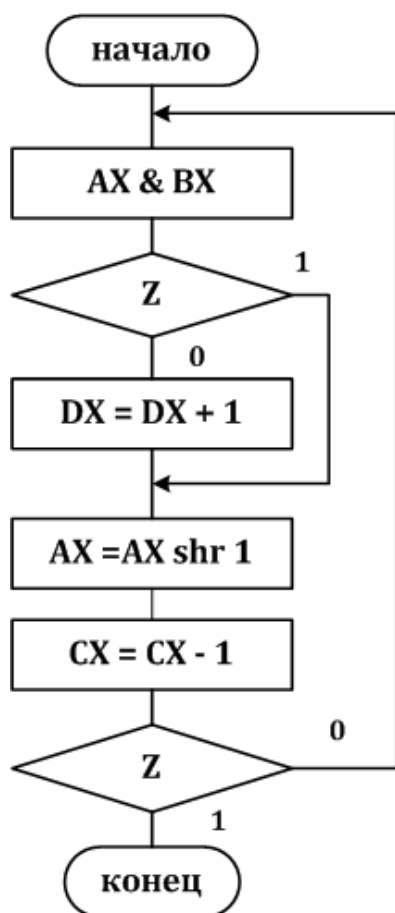


Рис. 2. Алгоритм

Распределение регистров:

BX – маска; BX := 0001;

CX – счетчик циклов; CX := 10h;

AX – исходное число;

DX – счетчик единиц; DX=0;

RW = 0 – вспомогательный регистр.

На рис. 2 приведена схема алгоритма.

Микропрограмма приведена в таблице 13. В таблице отображаются только поля, которые не совпадают со значением по умолчанию.

Таблица 13. Микропрограмма для подсчета единиц в числе

Адрес МК	Операция	Поле	Значение	Функция
00	M1: AX & BX if Z then goto M2	A	0	AX
		B	3	BX
		ALU	9	R & S
		CC	1	JZ
		CHA	3	CJP
		CONST	0002	Адрес перехода
01	DX = DX + 1	A	2	DX
		B	2	DX
		CCX	1	C0=1
		DST	4	PЗУ[B]=SDA
02	M2: AX = AX shr 1	SH	2	ЛС сдвиг вправо
		N	1	Сдвиг на 1 разряд
		DST	4	PЗУ[B]=SDA

03	CX = CX-1 if not Z then goto M1	A	F	RW
		B	1	CX
		ALU	1	S – R – 1 + C0
		DST	4	PЗУ[B]=SDA
		CC	1	JNZ
		JFI	1	I=1
		CHA	3	CJP
		CONST	0000	Адрес перехода
04	STOP	JFI	5	Останов

3. СОДЕРЖАНИЕ ОТЧЕТА

В отчет входят схема алгоритма, микропрограмма выполняемого алгоритма по форме таблицы 13, значения исходных данных и результатов, протокол изменения состояний основных элементов модели, выводимых на экран в режиме МИКРОКОМАНДА.

4. ВАРИАНТЫ ЗАДАНИЙ

1. Составить программы умножения двух положительных чисел по алгоритму с анализом старшего бита множителя со сдвигом множимого.

Указание. Множитель и множимое – байт, произведение (СЧП) – слово.

2. Составить программы умножения двух положительных чисел по алгоритму с анализом младшего бита множителя со сдвигом множимого.

Указание. Множитель и множимое – байт, произведение (СЧП) – слово.

3. Составить программу деления двух положительных чисел по алгоритму с восстановлением остатка.

Указание. Делимое – слово, делитель и частное – байт.

4. Составить программы умножения двух положительных чисел по алгоритму с анализом старшего бита множителя со сдвигом СЧП.

Указание. Множитель и множимое – байт, произведение (СЧП) – слово.

5. Составить программы умножения двух положительных чисел по алгоритму с анализом младшего бита множителя со сдвигом СЧП.

Указание. Множитель и множимое – слова, произведение (СЧП) – двойное слово.

6. Найти сумму квадратов натуральных чисел от 1 до K. При вычислении квадрата очередного числа использовать формулу $(X+1)^2=X^2+2X+1$

7. Преобразовать 4-разрядное положительное десятичное число в двоичное по формуле $B=((D1*10+D2)*10+D3)*10+D4$, где D_i - тетрады числа начиная со старшей.

Рекомендации по микропрограммированию:

- чтобы выделить тетраду, десятичное число необходимо поместить в RGQ, используя функцию сдвигателя $ALU \Rightarrow RGQ$. А затем выполняем двойной сдвиг ALU и RGQ влево на 4 бита с установкой $SRC=0$;
- умножение RgBin на 10 сводится к сложению и сдвигам по формуле: $(RgBin * 4 + RgBin) * 2$.

8. Определить остаток от деления двоичного числа на 15.

Указание. Остаток от деления числа на 15 равен остатку от деления на 15 суммы его 16-ричных цифр.

Чтобы выделить тетраду, число необходимо поместить в RGQ, используя функцию сдвигателя $ALU \Rightarrow RGQ$. А затем выполняем двойной сдвиг ALU и RGQ влево на 4 бита с установкой $SRC=0$. Второй способ с помощью маски и сдвигов.

9. Определить остаток от деления двоичного числа на 17.

Указание. Остаток от деления на 17 равен остатку от деления на 17 суммы цифр числа, взятых попеременно со знаками плюс и минус, начиная с младшей цифры.

Чтобы выделить тетраду, число необходимо поместить в RGQ, используя функцию сдвигателя $ALU \Rightarrow RGQ$. А затем выполняем двойной сдвиг ALU и RGQ влево на 4 бита с установкой $SRC=0$. Второй способ с помощью маски и сдвигов.

10. Определить остаток от деления десятичного числа на 9.

Указание. Остаток от деления числа на 9 равен остатку от деления на 9 суммы его десятичных цифр.

Чтобы выделить тетраду, число необходимо поместить в RGQ, используя функцию сдвигателя $ALU \Rightarrow RGQ$. А затем выполняем двойной сдвиг ALU и RGQ влево на 4 бита с установкой $SRC=0$. Второй способ с помощью маски и сдвигов.

11. Определить остаток от деления десятичного числа на 11.

Указание. Остаток от деления на 11 равен остатку от деления на 11 суммы цифр числа, взятых попеременно со знаками плюс и минус, начиная с младшей цифры.

Чтобы выделить тетраду, число необходимо поместить в RGQ, используя функцию сдвигателя $ALU \Rightarrow RGQ$. А затем выполняем двойной сдвиг ALU и RGQ влево на 4 бита с установкой $SRC=0$. Второй способ с помощью маски и сдвигов.

12.. Составить программу деления двух положительных чисел по алгоритму без восстановления остатка.

Указание. Делимое – слово, делитель и частное – байт.

13. Составить программу нормализации числа со знаком: арифметический сдвиг влево до появления единицы в старшем значимом разряде мантииссы, при модификации порядка. Мантииссу и порядок записать в соседние РОНы.

14. Найти наибольший общий делитель двух чисел по алгоритму Евклида.

Указание. Числа А и В сравниваются между собой. Из большего числа вычитается меньшее. Сравнения и вычитания повторяются до выполнения условия $A = B$.