

Цель работы

Исследовать работу с массивом и арифметические операции процессора CPU580.

Описание работы

Задание 1. Составить программу вычисления выражения:

$$\sum_{i=1}^n a_i = a_1 + a_2 + \dots + a_n \quad (1)$$

где a_i — число натурального ряда, начиная с 1;

n — количество чисел, при $n = 90$

Решение. Описание используемых регистров (см. таблицу 1).

Таблица 1 — Назначения регистров

Регистр	Назначение
А	Аккумулятор, где происходят все действия и результат
В	Число для сравнения
С	Прибавляемое число, выполняющее роль X_i в исходной формуле

Описание алгоритма: Записываем в аккумулятор и в переменную, которую будем прибавлять, число 1 (1-2). В регистр для сравнения, по заданию, записываем число 90 (3) Увеличиваем X_i (4) и прибавляем к аккумулятору (5). Если $X_i = 90$, конец программы, иначе перейти на шаг 4. Подробное описание шагов приведено в таблице 2.

Таблица 2 — Код программы

№	Команда	Описание
0	JMP 1	Прыжок по адресу 1
1	MVI A,01	Занесение значения 1 в регистр А (аккумулятор)
2	MVI C,01	Занесение значения 1 в регистр С
3	MVI B,5A	Занесение значения 90 в регистр В
4	INR C	Инкремент регистра С
5	ADD C	Сложение значения регистра С с аккумулятором(А)
6	CMP B	Сравнение аккумулятора с В (через вычитание А из В)
7	JNZ 4	Если флаг нуля после сравнения не обнулился - прыгаем по адресу 4 в начало цикла
8	HLT	Задержка

Задание 2. Составить программу деления однобайтных двоичных чисел.

Решение.

Описание используемых регистров приведены в таблице 3.

Таблица 3 — Назначение регистров

Регистр	Назначение
В	Делитель
С	Делимое
Д	Остаток
Е	Счетчик цикла
Н	Результат

Описание алгоритма: В данном алгоритме деление происходит практически также как при делении в столбик. Мы берем число с разрядностью вдвое больше исходного делимого и начинаем вычитать делитель, начиная со старшего разряда и каждый раз сдвигаясь к младшим. При вычитании, если мы получаем отрицательный результат, значит частичный остаток все ещё больше делителя, поэтому необходимо восстановить его до прежнего значения и продолжить выполнение (шаг 8-9). Подробное описание шагов приведено в таблице 4.

Таблица 4 — Код программы

0	MVI E 07	Счетчик цикла
1	LXI B, N1, N2	Загружаем из памяти делимое - число N1, делитель - число N2
2	MOV A, C	Сдвиг делимого
3	RAL	
4	MOV C, A	
5	MOV A, D	Сдвигаем значение частичного остатка
6	RAL	–
7	SUB B	Вычитаем делитель
8	JNC 10	Если происходит переполнение - восстанавливаем значение частичного остатка
9	ADD B	
10	MOV D, A	Возвращаем ЧО в регистр
11	CMC	Инвертируем перенос, так как если он произошел, то произошло переполнение, а значит вычитание делителя из ЧО нельзя производить
12	MOV A, H	Запоминаем перенос
13	RAL	
14	MOV H, A	
15	DCR E	Уменьшаем счетчик циклов
16	JNZ 2	Цикл - пока счетчик не равен 0
17	HLT	Иначе, конец программы

Вывод: в ходе данной практической работы мы научились реализовывать простые алгоритмы при помощи языка ассемблера CPU580. Алгоритм деления не является оптимальным, но при этом является более наглядным и простым для понимания, что важно, учитывая ознакомительных характер работы.