**Лабораторная работа № 12**

**Тема:** Разработка и отладка программы с использованием команд переходов.

Исследование командного цикла МП при выполнении команд переходов.

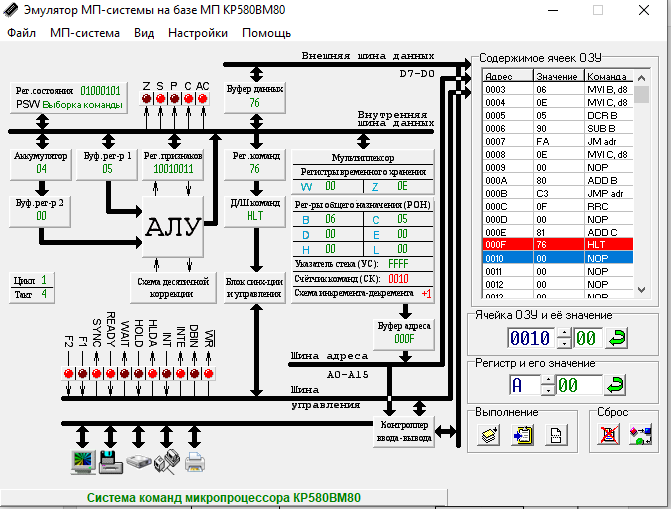
**Цель:** Научиться разрабатывать и осуществлять отладку программы с использованием команд переходов, исследовать командный цикл МП при выполнении команд переходов.

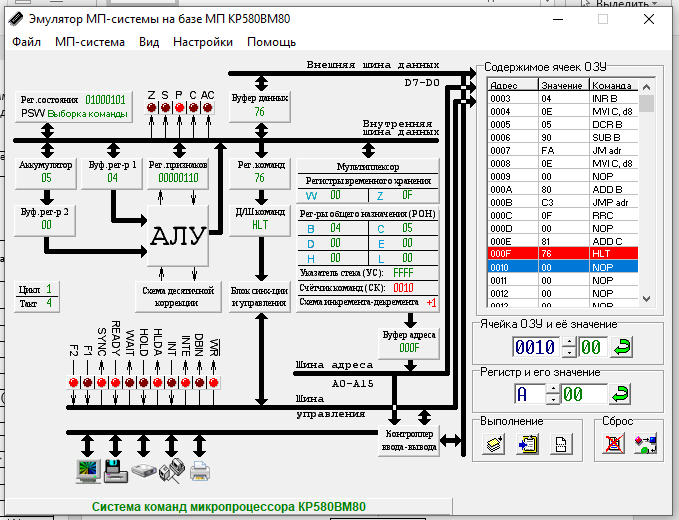
|  |  |
| --- | --- |
| 6 | Из числа **х** вычтите число **у**. Если результат отрицательный, то сложите этот результат с числом **х**, иначе - сложите его с числом **у**. |

X = 5 Y = 6 Ответ: 4

X = 5 Y = 4 Ответ: 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Адрес ОЗУ** | **Мнемоника** | **2-ый код** | **16-ый код** | **Комментарий** |
|
| 0000 | MVI A | 0011 1110 | 3E | <05> –> (A) |
| 0001 |  |  | 05 |  |
| 0002 | MVI B | 0000 0110 | 06 | <06> –> (B) |
| 0003 |  |  | 06 |  |
| 0004 | MVI C | 0000 1110 | 0E | <05> –> (C) |
| 0005 |  |  | 05 |  |
| 0006 | SUB B | 1001 0000 | 90 | (A) – (B) –> (A) |
| 0007 | JM | 1111 1010 | FA | / S == 1/ |
| 0008 |  |  | 0E |  |
| 0009 |  |  | 00 |  |
| 000A | ADD B | 1000 0000 | 80 | (A) + (B) –> (A) |
| 000B | JMP | 1100 0011 | C3 | Прыжок |
| 000C |  |  | 0F |  |
| 000D |  |  | 00 |  |
| 000E | ADD С | 1000 0001 | 81 | (A) + (С) –> (A) |
| 000F | HLT |  | 76 |  |





Блок-схема:

<y> --> (B)

(B) +1 -> (B)

(B) - 1 -> (B)

Начало

<x> --> (A)

<x> --> (С)

(A) – (B) --> (A)

S=1?

(A) + (B) --> (A)

(A) + (C) --> (A)

Конец

**Такты команды безусловного перехода JMP:**

**Т1 Т2**

Счетчик команд

Буфер адреса

ША

ВнешнШД

Буфер данных

**Т3 Т4**

ВнутрШД

Счетчик команд

Рег.команд

Д/Ш команд

+1

**Т5 Т6**

ВнешнШД

Буфер адреса

ША

Буфер данных

**Т7 Т8**

ВнутрШД

Z

Счетчик команд

+1

Буфер адреса

ША

ВнутрШД

**Т9 Т10**

Буфер данных

Z

Счетчик команд

ВнешнШД

**Вывод:** Научиться разрабатывать и осуществлять отладку программы с использованием команд переходов, исследовать командный цикл МП при выполнении команд переходов.

Микропроцессор Intel 8080 выполняет условные переходы по любому значению признаков: 1 или 0. Все команды условных переходов используют 16-разрядный прямой адрес.  
Наиболее простыми командами условных переходов являются ПЕРЕХОД ПО НУЛЮ JZ (JUMP ON ZERO) и ПЕРЕХОД ПРИ ОТСУТСТВИИ НУЛЯ JNZ (JUMP ON NOT ZERO). Команда JNZ вызывает переход по программе, если признак НУЛЬ равен 0 (предыдущий результат не был равен 0). В большинстве программных циклов используется команда JZ или JNZ. Типичная последовательность команд заставляет ЦП выполнять программу (с начальным адресом LOOP), пока содержимое регистра С остается равным 0. На рис. 3.19 показан типичный цикл программы, в которой требуемое число итераций (COUNT) определено в регистре С. По ходу основной программы содержимое регистра С уменьшается на 1 (декрементируется), и команда JNZ используется для определения числа повторения основной программы.  
**Команды JZ и JNZ** можно также использовать для обращения к отдельным байтам данных. Последовательность вызывает переход по адресу F100, если содержимое аккумулятора равно 100.  
Другие простые команды условного перехода — это ПЕРЕХОД ПО ПРИЗНАКУ ПЕРЕНОСА JC (JUMP ON CARRY) и ПЕРЕХОД ПРИ ОТСУТСТВИИ ПРИЗНАКА ПЕРЕНОСА JNC (JUMP ON NOT CARRY). Эти команды часто используются вместе с командой СРАВНИТЬ (COMPARE). Команда СРАВНИТЬ Устанавливает признак ПЕРЕНОС в 1 в том случае, если 8-битное число без знака в аккумуляторе меньше числа, с которым оио сравнивается. При сравнении CARRY = 1 означает, что для выполнения вычитания нужен заем. Последовательность команд вызывает переход к ячейке с Bj$pec©Bi\*LSTEN, если я аккумуляторе содержится я и ело без знака, которое ние ныне it) (CARRY = 10, если содержимое аккумулятора равно 10).  
Таким же образом последовательность команд доказывает переход к ячейке с адресом LARGE, если в аккумуляторе содержится •число\*без знака, значение которого лежит между 64 и 255  
Команды JC и JNC можно использовать для проверки значения разряда, сдвигаемого на место разряда ПЕРЕНОС. Заметим, что в МП Intel 8080 команды сдвига не влияют ни на какие другие признаки. Таким образом, для того чтобы определить, четное ли число содержится в аккумуляторе (последний значащий разряд есть 0), используется последовательность  
**JNC EVEN**  
По команде RAR последний значащий разряд аккумулятора перемещается в разряд ПЕРЕНОС, и команда JNC вызывает переход к ячейке с адресом EVEN, если ПЕРЕНОС = 0.