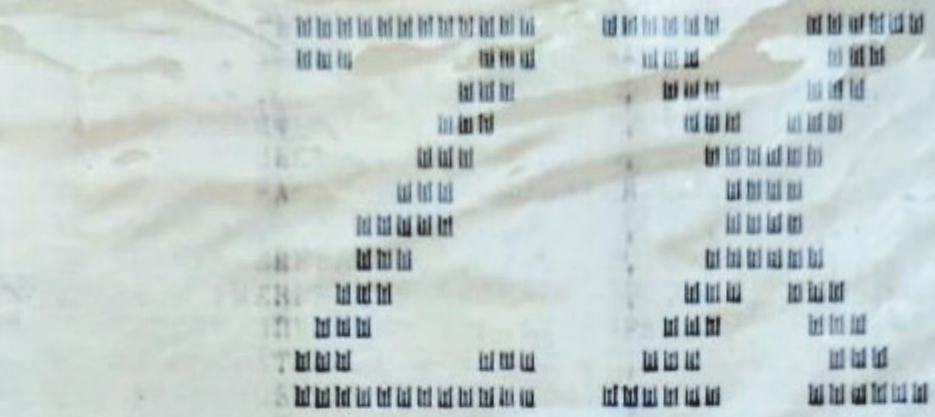
SINCLAIR RESEARCH LIMITED





ACCEMBAEP



SCANNED BY @ZCASPER

ESCAPE FROM FREEDOM

2022

О Г Л А В Л Е Н И Е

КАК ОРИЕНТИРОВАТЬСЯ В МАШИННОМ ЯЗЫКЕ 1

ВВЕДЕНИЕ 1
ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ 3
ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР 3
ВЫПОЛНЕНИЕ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ 6
ВИДЫ АДРЕСАЦИИ 8
ФЛАГИ И ИХ ПРИМЕНЕНИЕ 12
УВЕЛИЧЕНИЕ И УМЕНЬШЕНИЕ ЧИСЕЛ 15
АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ НАД 8-РАЗРЯДНЫМИ ЧИСЛАМИ 16
ЛОГИЧЕСКИЕ ОПЕРАТОРЫ 18
РАБОТА С 16-БИТОВЫМИ ЧИСЛАМИ 20
ОБРАБОТКА ЧИСЕЛ НА ДВУХ РЕГИСТРАХ 21
РАБОТА СО СТЕКОМ 24
АРИФМЕТИЧЕСКИЕ ДЕЙСТВИЯ 25
ЦИКЛЫ И ПЕРЕХОДЫ 27
ПРИМЕНЕНИЕ ПОДПРОГРАММ В ВАШИХ ПРОГРАММАХ
НА МАШИННОМ ЯЗЫКЕ 29
ОПЕРАЦИИ НАД БЛОКАМИ 31
БОЛЕЕ РЕДКО ИСПОЛЬЗУЕМЫЕ КОМАНДЫ Z80 34

ОБМЕН РЕГИСТРАМИ 34
УСТАНОВКА И СБРОС БИТОВ 34
СДВИГИ И ЦИКЛИЧЕСКИЕ СДВИГИ 35
ВВОД И ВЫВОД 36
ПРЕДСТАВЛЕНИЕ ДВОИЧНО-КОДИРОВАННЫХ ДЕСЯТИЧНЫХ ЧИСЕЛ . . 38
ПРЕРЫВАНИЯ 39
КОМАНДА РЕСТАРТА 39
НАПИСАНИЕ ПРОГРАММ ДЛЯ "ZX SPECTRUM" 40

ПЛАНИРОВАНИЕ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ 40
СРЕДСТВА "ZX SPECTRUM" 43
ВВЕДЕНИЕ В МОНИТОРНЫЕ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ . . . 48
МОНИТОРНАЯ ПРОГРАММА EZ CODE 48
МОНИТОРНАЯ ПРОГРАММА ЗАГРУЗКИ ТЕКСТА ПРОГРАММЫ НА
МАШИННОМ ЯЗЫКЕ В ШЕСТНАДЦАТЕРИЧНОМ ФОРМАТЕ HEXLOAD . . . 55

ПРИЛОЖЕНИЯ

ТАБЛИЦА КЛАВИШ "ZX SPECTRUM" 59
ПЛАН ЭКРАНА ДИСПЛЕЯ 60
ТАБЛИЦА НАБОРА ЛИТЕР "ZX SPECTRUM" 61
ТАБЛИЦА ПРЕОБРАЗОВАНИЯ ДЕСЯТИЧНЫХ ЧИСЕЛ В
ШЕСТНАДЦАТЕРИЧНЫЕ 62
ТАБЛИЦА ПРЕОБРАЗОВАНИЯ ДЕСЯТИЧНЫХ ЧИСЕЛ В ФОРМЕ
ДОПОЛНЕНИЯ ДО 2 В ШЕСТНАДЦАТЕРИЧНЫЙ ФОРМАТ 63
ШЕСТНАДЦАТЕРИЧНАЯ ТАБЛИЦА СЛОЖЕНИЯ 63
СВОДНАЯ ТАБЛИЦА ИЗМЕНЕНИЯ ФЛАГОВ 64
КОМАНДЫ ЦП Z80 В ПОРЯДКЕ КОДОВ ОПЕРАЦИЙ 66
КОМАНДЫ ЦП Z80 В ПОРЯДКЕ ВОЗРАСТАНИЯ МНЕМОНИЧЕСКИХ
ОБОЗНАЧЕНИЙ 69

0 1 1 1 0 1 1 1

МЕЖДУ ЯЗЫКОМ АССЕМБЛЕРА И МАШИННЫМ ЯЗЫКОМ ЕСТЬ ТОЛЬКО ОДНО РАЗЛИЧИЕ: ЯЗЫК АССЕМБЛЕРА НА ОДИН УРОВЕНЬ ВЫШЕ, ЧЕМ МАШИННЫЙ ЯЗЫК. ЕГО ЛЕГЧЕ ЧИТАТЬ ЧЕЛОВЕКУ, ЧЕМ МАШИННЫЙ ЯЗЫК, НО С ДРУГОЙ СТОРОНЫ, ЭВМ НЕ МОЖЕТ ЧИТАТЬ ЯЗЫК АССЕМБЛЕРА.

ОН НЕ ЯВЛЯЕТСЯ АДАПТАЦИЕЙ МАШИННОГО ЯЗЫКА, ПОДОБНО "БЕЙСИКУ", ДЛЯ КАЖДОЙ КОМАНДЫ ЯЗЫКА АССЕМБЛЕРА ИМЕЕТСЯ ИДЕНТИЧНАЯ (ПО ФУНКЦИИ) КОМАНДА МАШИННОГО ЯЗЫКА И НАОБОРОТ, ИНЫМИ СЛОВАМИ МЕЖДУ НИМИ ИМЕЕТСЯ ВЗАИМНО ОДНОЗНАЧНОЕ СООТВЕТСТВИЕ, ПОЭТОМУ МОЖНО СКАЗАТЬ, ЧТО ЯЗЫК АССЕМБЛЕРА ЭКВИВАЛЕНТЕН МАШИННОМУ ЯЗЫКУ.

ЯЗЫК АССЕМБЛЕРА МОЖЕТ БЫТЬ ПРЕОБРАЗОВАН НЕПОСРЕДСТВЕННО В МАШИННУЮ ПРОГРАММУ С ПОМОЩЬЮ ПРОГРАММЫ ИЛИ ВАМИ САМИМИ, ТАКАЯ ПРОГРАММА НАЗЫВАЕТСЯ "АССЕМБЛЕРОМ", ВЫ МОЖЕТЕ РАССМАТРИВАТЬ ЕЕ КАК ПРОГРАММУ, ВЫПОЛНЯЮЩУЮ ДОВОЛЬНО УТОМИТЕЛЬНУЮ ЗАДАЧУ ТРАНСЛЯЦИИ ВАШЕЙ НАПИСАННОЙ НА ЯЗЫКЕ АССЕМБЛЕРА ПРОГРАММЫ В ПОСЛЕДОВАТЕЛЬНОСТЬ КОМАНД МАШИННОГО ЯЗЫКА, ПОНЯТНЫХ "СПЕКТРУМ" И МЫ СЧИТАЕМ, ЧТО "АССЕМБЛЕР" ДЛЯ "ZX СПЕКТРУМ" УЖЕ ИМЕЕТСЯ.

ТЕМ НЕ МЕНЕЕ, ТАКИЕ АССЕМБЛЕРЫ ОБЫЧНО ТРЕБУЮТ 6К ПАМЯТИ И ИМЕЮТ ОГРАНИЧЕННОЕ ПРИМЕНЕНИЕ НА ЭВМ С ОБ"ЕМОМ ПАМЯТИ 16К, ДИСПЛЕЙ "СПЕКТРУМ" ОТНИМАЕТ 7К ПАМЯТИ, И ПОСЛЕ ЗАГРУЗКИ АССЕМБЛЕРА У ВАС МОЖЕТ ОСТАТЬСЯ ВСЕГО 4К ПАМЯТИ ДЛЯ ПРОГРАММЫ НА ЯЗ.АССЕМБЛЕРА.

АЛЬТЕРНАТИВНЫЙ СПОСОБ РАБОТЫ - ВМЕСТО ПРИМЕНЕНИЯ ПРОГРАММЫ АССЕМБЛЕРА САМОМУ ТРАНСЛИРОВАТЬ МНЕМОНИКУ ЯЗЫКА АССЕМБЛЕРА-МАШИННЫЙ ЯЗЫК ВРУЧНУЮ, ПРМЕНЯЯ ПРИВЕДЕННЫЕ В ЭТОЙ КНИГЕ ТАБЛИЦЫ.

ЭТО ТРУДНО, НЕУДОБНО, НО ЭТО ПРЕКРАСНАЯ ПРАКТИКА И ДАЕТ ВАМ ГЛУБОКОЕ ПОНИМАНИЕ ТОГО, КАК РАБОТАЕТ ЦП "СПЕКТРУМ".

В ZX SPECTRUM ПРИНЯТА 16-РИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ ПО СЛЕДУЮЩИМ ПРИЧИНАМ:

1. ИЗ ЭТОЙ ФОРМЫ ЧИСЛА ЛЕГКО ПЕРЕВЕСТИ В ДВОИЧНЫЙ ФОРМАТ, КОТОРЫЙ ГОВОРИТ НАМ, ЧТО ОЗНАЧАЕТ КАЖДЫЙ БИТ (ИЛИ ПАЛЕЦ).
2. ОН ДАЕТ НАМ ВОЗМОЖНОСТЬ ЛЕГКО ОПРЕДЕЛИТЬ, БУДЕТ ЛИ ЧИСЛО ВОСЬМИРАЗРЯДНОЕ ИЛИ ШЕСТНАДЦАТИРАЗРЯДНОЕ.
3. ОН ДАЕТ СТАНДАРТНОЕ ПРЕДСТАВЛЕНИЕ ВСЕХ ЧИСЕЛ В ВИДЕ ПОСЛЕДОВАТЕЛЬНОСТИ ДВУХРАЗРЯДНЫХ ЧИСЕЛ.
4. ЭТО ОБЩЕПРИНЯТАЯ СИСТЕМА ОБОЗНАЧЕНИЙ И ЗНАКОМСТВО С ШЕСТНАДЦАТЕРИЧНОЙ СИСТЕМОЙ ПОЗВОЛИТ ВАМ С БОЛЬШЕЙ ЛЕГКОСТЬЮ ЧИТАТЬ ДРУГИЕ КНИГИ И РУКОВОДСТВА.
5. ПОСКОЛЬКУ ЦП СКОНСТРУИРОВАН ДЛЯ ОБРАБОТКИ ИНФОРМАЦИИ, ПРЕДСТАВЛЕННОЙ ДВОИЧНЫМИ ЧИСЛАМИ, КОТОРЫЕ ЛЮДЯМ ДЛЯ ЧТЕНИЯ НЕУДОБНЫ, ТРЕБУЕТСЯ БОЛЕЕ УДОБОЧИТАЕМОЕ ПРЕДСТАВЛЕНИЕ.

ШЕСТНАДЦАТЕРИЧНАЯ СИСТЕМА, КАК ОТМЕЧАЛОСЬ ВЫШЕ, ПОЗВОЛЯЕТ ПРЕДСТАВЛЯТЬ ЧИСЛА ОТ 0 ДО 15 С ПОМОЩЬЮ ВСЕГО 4 БИТОВ, ЛЮБАЯ 8-РАЗРЯДНАЯ ЯЧЕЙКА ПАМЯТИ ИЛИ 8-РАЗРЯДНЫЙ РЕГИСТР МОЖЕТ ПОЭТОМУ БЫТЬ ОПИСАНА КАК ДВА НАБОРА ПО 4 БИТА.

НАС ИНТЕРЕСУЮТ 8-РАЗРЯДНЫЕ ЯЧЕЙКИ ПАМЯТИ И 8-РАЗРЯДНЫЕ РЕГИСТРЫ ПОТОМУ, ЧТО ТАКОВА СТРУКТУРА "ZX СПЕКТРУМ", ВСЕ ЯЧЕЙКИ ПАМЯТИ И ВСЕ ОДИНАРНЫЕ РЕГИСТРЫ ИМЕЮТ ПО 8 РАЗРЯДОВ.

СЛЕДУЮЩАЯ ПРОГРАММА НА ЯЗЫКЕ "БЕЙСИК" ПОЗВОЛИТ ВАМ ВВЕСТИ В ВАШ "СПЕКТРУМ" ДЕСЯТИЧНОЕ ЧИСЛО И ПРЕОБРАЗОВАТЬ ЕГО В ШЕСТНАДЦАТЕРИЧНОЕ ЗНАЧЕНИЕ.

```
 1 0 0 REM DECIMAL TO HEXADECIMAL CONVERSION
 1 1 0  PRINT "PLEASE INPUT DECIMAL VALUE"
 1 2 0  INPUT N : PRINT N
 1 3 0  LET S$ -"": LET N2 = INT(N/16)
 1 4 0  LET N1 - INT(N -N2*16)
 1 5 0  LET S$ - CHR$ ((N1<=9)*(N1+48)+(N1>9)*(55+N1))+S$
 1 6 0  IF N2 = 0 THEN PRINT : PRINT "HEXADECIMAL - 0" : S$
            :"H": FOR I = 1 TO 200: NEXT I: RUN
 1 7 0  LET N = N2 : GO TO 140
```

1 - ПРЕОБРАЗОВАНИЕ ДЕСЯТИЧНЫХ В ШЕСТНАДЦАТЕРИЧНЫЕ; 2 - ВВЕДИТЕ ПОЖАЛУЙСТА, ДЕСЯТИЧНОЕ ЗНАЧЕНИЕ; S - ЗНАК ЗАМЕНЕН НА ЗНАК - $

ДУЮЩИЕ ЧИСЛА И С ПОМОЩЬЮ ПРОГРАММЫ НА ЯЗЫКЕ "БЕЙСИК" ПРОВЕРЬТЕ СВОЙ ОТВЕТ,

1, 16484 АДРЕС НАЧАЛА ДИСПЛЕЙНОГО ФАЙЛА "СПЕКТРУМ":
2, 22528 АДРЕС НАЧАЛА ФАЙЛА АТРИБУТОВ "СПЕКТРУМ":
3, 15360 АДРЕС НАЧАЛА НАБОРА ЛИТЕР "СПЕКТРУМ":
4, 15616 АДРЕС НАЧАЛА ЛИТЕР В КОДЕ ASCII "СПЕКТРУМ",

К А К П Р Е Д С Т А В Л Я Е Т С Я И Н Ф О Р М А Ц И Я

В ПРЕДСТАВЛЕНИИ ИНФОРМАЦИИ ЧЕЛОВЕКОМ И ЭВМ ИМЕЕТСЯ СУЩЕСТВЕННАЯ РАЗНИЦА,У ЧЕЛОВЕКА ИНФОРМАЦИЯ В ОСНОВНОМ СОСТОИТ ИЗ ЧИСЕЛ И ЛИТЕР (АЛФАВИТНО-ЦИФРОВАЯ ИНФОРМАЦИЯ),ТОГДА КАК ВСЯ ИНФОРМАЦИЯ В ЭВМ ХРАНИТСЯ В ВИДЕ ГРУПП БИТОВ,

БИТ ОЗНАЧАЕТ ДВОИЧНЫЙ РАЗРЯД (BINARI DIG JT, "0" ИЛИ "1"); В МИКРОПОЦЕССОРЕ Z80A ЭТИ БИТЫ СГРУППИРОВАНЫ ПО 8, ГРУППА В ИЗ ВОСЬМИ БИТОВ НАЗЫВАЕТСЯ "БАЙТОМ",

ТАКОЙ СПОСОБ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ С ПОМОЩЬЮ ДВОИЧНЫХ РАЗРЯДОВ НАЗЫВАЕТСЯ "ДВОИЧНЫМ ФОРМАТОМ", ТАКОВА СТРУКТУРА ЯЗЫКА, НА КОТОРОМ РАЗГОВАРИВАЕТ Z80 И БОЛЬШИНСТВО ПП МИКРОЭВМ

В ОСНОВНОМ ИМЕЕТСЯ ДВА ТИПА ИНФОРМАЦИИ, ПРЕДСТАВЛЯЕМОЙ ВНУТРИ "СПЕКТРУМ", ПЕРВЫЙ - ЭТО ПРОГРАММА, ВТОРОЙ - ДАННЫЕ, НАД КОТОРЫМИ ПРОГРАММА БУДЕТ ДЕЙСТВОВАТЬ И КОТОРЫЕ МОГУТ ВКЛЮЧАТЬ ЧИСЛА ИЛИ АЛФАВИТНО-ЦИФРОВОЙ ТЕКСТ, МЫ ТАК И БУДЕМ НИЖЕ РАССМАТРИВАТЬ ЭТИ ТРИ ПРЕДСТАВЛЕНИЯ: ПРОГРАММА, ЧИСЛА, АЛФАВИТНО-ЦИФРОВОЙ ТЕКСТ,

ПРЕДСТАВЛЕНИЕ ПРОГРАММЫ - П Р О Г Р А М М А - ПОСЛЕДОВАТЕЛЬНОСТЬ КОМАНД ЦП ВЫПОЛНИТЬ КОНКРЕТНОЕ ЗАДАНИЕ, КОТОРОЕ МОЖНО РАЗБИТЬ НА НЕКОТОРОЕ ЧИСЛО "ПОДРАЗДЕЛЕНИЙ",

В Z80 ВСЕ КОМАНДЫ ИМЕЮТ ВНУТРЕННЕЕ ПРЕДСТАВЛЕНИЕ В ВИДЕ ОДНОГО ИЛИ НЕСКОЛЬКИХ БАЙТОВ, КОМАНДЫ, ПРЕДСТАВЛЕННЫЕ ОДНИМ БАЙТОМ, НАЗЫВАЮТСЯ "КОРОТКИМИ КОМАНДАМИ", БОЛЕЕ ДЛИННЫЕ КОМАНДЫ ПРЕДСТАВЛЯЮТСЯ ДВУМЯ ИЛИ НЕСКОЛЬКИМИ БАЙТАМИ,

ПОСКОЛЬКУ Z80 - ВОСЬМИРАЗРЯДНЫЙ МИКРОПРЦЕССОР, ОН МОЖЕТ ЗА ОДИН РАЗ ОБРАБАТЫВАТЬ ТОЛЬКО ОДИН БАЙТ, И ЕСЛИ ЕМУ ТРЕБУЕТСЯ БОЛЕЕ ОДНОГО, ОН ВЕДЕТ ПОСЛЕДОВАТЕЛЬНЫЙ ПОИСК БАЙТОВ В ПАМЯТИ, ПОЭТОМУ В ОБЩЕМ СЛУЧАЕ ОДНОБАЙТОВАЯ КОМАНДА БУДЕТ ВЫПОЛНЯТЬСЯ БЫСТРЕЕ, ЧЕМ ДВУХ- ИЛИ ТРЕХБАЙТОВАЯ, ТАКИМ ОБРАЗОМ, КАК ПРАВИЛО ВЫГОДНО ПИСАТЬ ПРОГРАММУ НА МАШИННОМ ЯЗЫКЕ, ПРИМЕНЯЯ, ГДЕ МОЖНО, ОДНОБАЙТОВЫЕ КОМАНДЫ,

ВЫ МОЖЕТЕ ПОСМОТРЕТЬ "КОРОТКИЕ И ДЛИННЫЕ" КОМАНДЫ В ТАБЛИЦЕ КОДОВ КОМАНД В ПРИЛОЖЕНИИ, ПУСТЬ ВАС НЕ БЕСПОКОИТ, ЧТО ОНИ НЕПОНЯТНЫ, ПОЗЖЕ МЫ РАССМОТРИМ КАЖДУЮ КОМАНДУ БОЛЕЕ ГЛУБОКО,

ЧИСЛА

КАЖДАЯ ЯЧЕЙКА ПАМЯТИ МОЖЕТ ИСПОЛЬЗОВАТЬСЯ ЛИБО ДЛЯ ХРАНЕНИЯ ПОЛОЖИТЕЛЬНЫХ ЦЕЛЫХ ЧИСЕЛ, ЛИБО ЦЕЛЫХ ЧИСЕЛ СО ЗНАКОМ (ЧИСЕЛ, КОТОРЫЕ МОГУТ БЫТЬ И ПОЛОЖИТЕЛЬНЫМИ И ОТРИЦАТЕЛЬНЫМИ) ПО ВАШЕМУ ВЫБОРУ, ДИАПАЗОН ЧИСЕЛ МОЖЕТ БЫТЬ ЛИБО ОТ 0 ДО 255 ЛИБО ОТ -128 ДО +127,

ОТРИЦАТЕЛЬНЫЕ ЧИСЛА

БЫЛО ПРИНЯТО СОГЛАШЕНИЕ, ЧТО КОГДА МЫ РЕШАЕМ, ЧТО В ПАМЯТИ БУДЕТ ХРАНИТЬСЯ ЧИСЛО СО ЗНАКОМ (+ ИЛИ -), ПРИМЕНЯЕТСЯ СЛЕДУЮЩЕЕ ПРАВИЛО:

ЕСЛИ БИТ 7 УСТАНОВЛЕН, ТО ЧИСЛО ОТРИЦАТЕЛЬНО

ЕСЛИ БИТ 7 НЕ УСТАНОВЛЕН, ЧИСЛО ПОЛОЖИТЕЛЬНО, ЧТОБЫ ПОЛУЧИТЬ ЧИСЛО, ПРОТИВОПОЛОЖНОЕ ЛЮБОМУ ЗАДАННОМУ ЧИСЛУ, НУЖНО ВЗЯТЬ "ДОПОЛНЕНИЕ ДО ДВУХ" И ПРИБАВИТЬ 1,

ДОПОЛНЕНИЕ ДО 2

ДОПОЛНЕНИЕ ЛЮБОГО ЧИСЛА ДО ДВУХ - ЭТО ЕГО ОТРИЦАНИЕ В ДВОИЧНОМ ФОРМАТЕ, ЛЮБОЙ УСТАНОВЛЕННЫЙ БИТ СБРАСЫВАЕТСЯ И НАОБОРОТ,

Ц Е Н Т Р А Л Ь Н Ы Й П Р О Ц Е С С О Р

МЫ ГОВОРИЛИ, ЧТО МОЗГ "СПЕКТРУМ" -ЦП, ПРОЦЕССОР Z80A, ЭТО - БОЛЕЕ БЫСТРОДЕЙСТВУЮЩИЙ ВАРИАНТ ПРОЦЕССОРА Z80, ПРОИЗВОДИМЫЙ ПО ЛИЦЕНЗИИ ФИРМОЙ "ЗАЙЛОГ"

ЕДИНСТВЕННОЕ ОТЛИЧИЕ МЕЖДУ ПРОЦЕССОРАМИ Z80 И Z80A СОСТОИТ В ТОМ, ЧТО ПЕРВЫЙ ПРОЦЕССОР РАБОТАЕТ ПРИ ЧАСТОТЕ ГЕНЕРАТОРА СИНХРОИМПУЛЬСОВ 2 МГЦ (МЕГАГЕРЦ), А ВТОРОЙ - 3,5 МГЦ 4 "ЧАС-

ТОТА ГЕНЕРАТОРА СИНХРОИМПУЛЬСОВ" — ЭТО ПРОСТО МЕРА СКОРОСТИ ВЫПОЛНЕНИЯ ЦП ВЫЧИСЛЕНИЙ, В "СПЕКТРУМ" ЗА СЕКУНДУ ГЕНЕРИРУЕТСЯ 3,5 МИЛЛИОНА СИНХРОСИГНАЛОВ, Т,Е, ОДИН СИНХРОИМПУЛЬС КАЖДЫЕ 0,000000286 СЕКУНДЫ,

САМАЯ БЫСТРАЯ КОМАНДА, ВЫПОЛНЯЕМАЯ ЦП, ЗАНИМАЕТ 4 СИНХРОИМПУЛЬСА, А САМАЯ ДОЛГАЯ — 21 СИНХРОИМУЛЬС, ЭТО ОЗНАЧАЕТ, ЧТО ДАЖЕ, ЕСЛИ БУДУТ ВЫПОЛНЯТЬСЯ ТОЛЬКО САМЫЕ МЕДЛЕННЫЕ КОМАНДЫ, ВСЕ РАВНО ЗА СЕКУНДУ БУДЕТ ВЫПОЛНЯТЬСЯ ОКОЛО 160 000 КОМАНД!

ФИЗИЧЕСКАЯ КАРТА ЦП

ПРОЦЕССОР "СПЕКТРУМ" — КРЕМНИЕВЫЙ ЧИП С 40 КОНТАКТАМИ, ПРОНУМЕРОВАННЫМИ ОТ 1 ДО 40, ЭТИ КОНТАКТЫ — ЛИНИИ СВЯЗИ МЕЖДУ ПРОЦЕССОРОМ И ОСТАЛЬНОЙ ЭВМ, НАПРИМЕР, ПРОЦЕССОР ПОЛУЧАЕТ ПИТАНИЕ ОТ ИСТОЧНИКА ПИТАНИЯ ЧЕРЕЗ КОНТАКТ 11,, ПОЛУЧАЕТ СИНХРОИМПУЛЬСЫ ЧЕРЕЗ КОНТАКТ 6, ПОЛУЧАЕТ И ПРИНИМАЕТ АДРЕСА ЧЕРЕЗ КОНТАКТЫ С 1 ПО 5 И С 30 ПО 40 И ПОСЫЛАЕТ И ПРИНИМАЕТ ДАННЫЕ ЧЕРЕЗ КОНТАКТЫ С 7 ПО 15 ЗА ИСКЛЮЧЕНИЕМ 11, ОСТАЛЬНЫЕ КОНТАКТЫ ПРЕДНАЗНАЧЕНЫ ДЛЯ ПЕРЕДАЧИ СИГНАЛОВ УПРАВЛЕНИЯ,

ЛОГИЧЕСКАЯ КАРТИНА ЦП

ЛОГИЧЕСКИ Z80 МОЖНО РАЗДЕЛИТЬ НА ПЯТЬ ФУНКЦИОНАЛЬНЫХ ЧАСТЕЙ, ЭТО: 1, УСТРОЙСТВО УПРАВЛЕНИЯ 2, РЕГИСТР КОМАНД, 3, СЧЕТЧИК КОМАНД 4, АРИФМЕТИКО-ЛОГИЧЕСКОЕ УСТРОЙСТВО, 5, 24 РЕГИСТРА ПОЛЬЗОВАТЕЛЯ,

УСТРОЙСТВО УПРАВЛЕНИЯ

МЫ МОЖЕМ РАССМАТРИВАТЬ УСТРОЙСТВО УПРАВЛЕНИЯ КАК НЕКИЙ СУПЕРВИЗОР, УПРАВЛЯЮЩИЙ РАБОТОЙ ЦП, ЕГО ЗАДАЧА СОСТОИТ В СИНХРОНИЗАЦИИ И КООРДИНАЦИИ ВВОДА, ОБРАБОТКИ И ВЫВОДА ДЛЯ КОНКРЕТНОГО ЗАДАНИЯ, ПОЛУЧЕННОГО ЦП ВНЕ ЗАВИСИМОСТИ ОТ ТОГО, ИСХОДЯТ ЛИ КОМАНДЫ ИЗ ПЗУ С ЗАПИСАННОЙ В НЕМ ПРОГРАММОЙ ИЛИ ИЗ ВАШЕЙ ПРОГРАММЫ,

РЕГИСТР КОМАНД

ЭТО — РЕГИСТР, ИСПОЛЬЗУЕМЫЙ ЦП ДЛЯ ХРАНЕНИЯ ТЕКУЩЕЙ КОМАНДЫ, КОТОРУЮ ОН СОБИРАЕТСЯ ИСПОЛНЯТЬ, ЗАДАНИЕ В ЦЕЛОМ, СОСТАВЛЯЮЩЕЕ ПРОГРАММУ, ДОЛЖНО НАХОДИТЬСЯ ГДЕ-ТО В ПАМЯТИ — ЛИБО В ПЗУ, ЛИБО В ПАМЯТИ С ПРОИЗВОЛЬНЫМ ДОСТУПОМ, ПРОГРАММА — ЭТО ПОСЛЕДОВАТЕЛЬНОСТЬ КОМАНД, ТАК, ЧТОБЫ ВЫПОЛНИТЬ ЗАДАНИЕ, УСТРОЙСТВО УПРАВЛЕНИЯ ДОЛЖНО ПО ОЧЕРЕДИ ОТЫСКИВАТЬ КАЖДУЮ КОМАНДУ В ПАМЯТИ (ЛИБО В ПЗУ, ЛИБО В ПАМЯТИ С ПРОИЗВОЛЬНЫМ ДОСТУПОМ) И ПОМЕШАТЬ ЕЕ В РЕГИСТР КОМАНД ,

СЧЕТЧИК КОМАНД

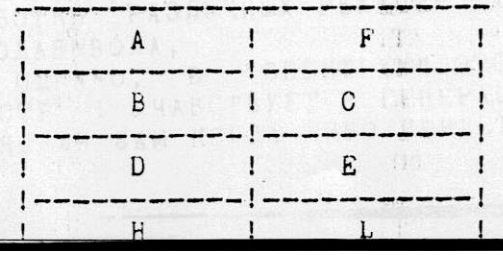
СООБЩАЮЩАЕТ ЦП, ГДЕ НАХОДИТСЯ СЛЕДУЮЩАЯ ЧАСТЬ ПРОГРАММЫ (АДРЕС СЛЕДУЮЩЕЙ ЯЧЕЙКИ ПАМЯТИ, В КОТОРОЙ УСТРОЙСТВО УПРАВЛЕНИЯ ДОЛЖНО НАЙТИ КОМАНДУ), ОН ПОДОБЕН УПРАВДОМУ ДЛЯ КОМАНД, СЛЕДЯЩЕМУ ЗА РАСПОЛОЖЕНИЕМ СЛЕДУЮЩЕЙ КОМАНДЫ,

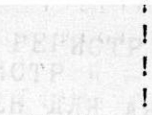
АРИФМЕТИКО-ЛОГИЧЕСКОЕ УСТРОЙСТВО (АЛУ)

ЭТО — КАЛЬКУЛЯТОР ВНУТРИ ЦП, ОН МОЖЕТ ВЫПОЛНЯТЬ КАК АРИФМЕТИЧЕСКИЕ, ТАК И ЛОГИЧЕСКИЕ ОПЕРАЦИИ, ИЗ ВСЕХ ОСНОВНЫХ АРИФМЕТИЧЕСКИХ ФУНКЦИЙ, ИЗВЕСТНЫХ НАМ С ВАМИ, ЭТО УСТРОЙСТВО МОЖЕТ ВЫПОЛНЯТЬ ТОЛЬКО ПРОСТЫЕ СЛОЖЕНИЕ И ВЫЧИТАНИЕ, УВЕЛИЧЕНИЕ (ДОБАВЛЕНИЕ 1) И УМЕНЬШЕНИЕ (ВЫЧИТАНИЕ 1), НО НЕ УМНОЖЕНИЕ И ДЕЛЕНИЕ, ЭТО УСТРОЙСТВО МОЖЕТ ТАКЖЕ СРАВНИВАТЬ ЧИСЛА ДЛЯ ОДНОГО РЕГИСТРА И ВЫПОЛНЯТЬ "ПОБИТОВЫЕ" ОПЕРАЦИИ, ТАКИЕ КАК ПЕРЕМЕЩЕНИЕ БИТОВ ПО КРУГУ, ВЫСТАВЛЕНИЕ И ПРИЖАТИЕ ОПРЕДЕЛЕННЫХ БИТОВ И Т,Д,

КАК ПОБОЧНЫЙ РЕЗУЛЬТАТ ВЫЧИСЛЕНИЙ, ПОРУЧАЕМЫХ АЛУ, ОБЫЧНО МЕНЯЕТСЯ СОСТОЯНИЕ РАЗЛИЧНЫХ "ФЛАГОВ" В РЕГИСТРЕ FLAG, НИЖЕ ЭТОТ ВОПРОС РАССМАТРИВАЕТСЯ БОЛЕЕ ПОДРОБНО,

НИЖЕ ПРИВОДИТСЯ ТРАДИЦИОННЫЙ СПОСОБ ПРЕДСТАВЛЕНИЯ РЕГИСТРОВ ПОЛЬЗОВАТЕЛЯ:





ОБРАТИТЕ ВНИМАНИЕ, ЧТО "F" СООТВЕТСТВУЕТ "A", ЗАТО ПОРЯДОК ОСТАЛЬНЫХ ДОСТАТОЧНО ЕСТЕСТВЕНЕН, ПРИЧИНА, ПО КОТОРОЙ РЕГИСТРЫ СГРУППИРОВАНЫ ТАКИМ ОБРАЗОМ, СОСТОИТ В ТОМ, ЧТО ИНОГДА МОЖНО ИЗ ДВУХ РЕГИТРОВ СОСТАВИТЬ ПАРУ,

IX И IY ТАКЖЕ ИМЕЮТ СПЕЦИАЛЬНОЕ НАЗВАНИЕ: ОНИ НАЗЫВАЮТСЯ "ИНДЕКСНЫМИ РЕГИСТРАМИ",

НАКАПЛИВАЮЩИЙ РЕГИСТР (РЕГИСТР А)

ЭТО 8-БИТОВЫЙ РЕГИСТР (ОДНОБАЙТОВЫЙ) — САМЫЙ ВАЖНЫЙ В Z80, ЕГО НАЗВАНИЕ ВОСХОДИТ К ПЕРВЫМ ПОКОЛЕНИЯМ ЭВМ,, КОГДА БЫЛ ЛИШЬ ОДИН РЕГИСТР, ПРИМЕНЯВШИЙСЯ ДЛЯ "НАКОПЛЕНИЯ" РЕЗУЛЬТАТА,

И ХОТЯ МЫ УШЛИ ВПЕРЕД ОТ ПЕРВЫХ ПОКОЛЕНИИ ЭВМ, НКАПЛИВАЮЩИЙ РЕГИСТР ПО-ПРЕЖНЕМУ ШИРОКО ПРИМЕНЯЕТСЯ ДЛЯ ЛОГИЧЕСКИХ И АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ, НА САМОМ ДЕЛЕ МНОГИЕ ЭВМ ВСЕ ЕЩЕ КОНСТРУИРУЮТСЯ ТАКИМ ОБРАЗОМ, ЧТО МНОГИЕ ОПЕРАЦИИ МОЖНО ВЫПОЛНЯТЬ ТОЛЬКО С ПОИОШЬЮ РЕГИСТРА А,

ЭТО ИМЕЕТ МЕСТО И ДЛЯ ЧИПА Z80, И РЕГИСТР А — ПРИВИЛЕГИРОВАННЫЙ,

ПАРА РЕГИСТРОВ

ИЗ ТРЕХ ПАР РЕГИСТРОВ (BC, DE, HL) ПАРА HL, ВОЗМОЖНО, — НАИБОЛЕЕ ВАЖНАЯ, ПОМИМО ТОГО, ЧТО ПОЛЬЗОВАТЕЛЮ ПРЕДОСТАВЛЕН ВЫБОР, ПРИМЕНЯТЬ ЛИ ЕЕ В ВИДЕ ДВУХ ОТДЕЛЬНЫХ РЕГИСТРОВ ИЛИ КАК ПАРУ РЕГИСГРОВ, Z80 СКОНСТРУИРОВАН ТАКИМ ОБРАЗОМ, ЧТО ЕСТЬ ОПРЕДЕЛЕННЫЕ ОПЕРАЦИИ 16-БИТОВОЙ АРИФМЕТИКИ, КОТОРЫЕ МОЖНО ВЫПОЛНЯТЬ ГОЛЬКО С ПОМОЩЬЮ ПАРЫ РЕГИСТРОВ HL,

ИЗ-ЗА ТАКОГО ПРИВИЛЕГИРОВАННОГО ,С ТОЧКИ ЗРЕНИЯ АППАРАТУРЫ ПОЛОЖЕНИЯ, ОПЕРАЦИИ НАД ПАРОЙ ОБЩИХ РЕГИСТРОВ БУДУТ ОБЫЧНО БЫСТРЕЕ ВЫПОЛНЯТЬСЯ ДЛЯ РЕГИСТРОВ HL, ИЗ-ЗА ЭТОГО HL ОКАЗЫВАЕТСЯ ПРЕДПОЧТЕНИЕ В ПРОГРАММИРОВАНИИ НА МАШИННОМ ЯЗЫКЕ,

АЛЬТЕРНАТИВНЫЙ НАБОР РЕГИСТРОВ

ПРЕДСТАВЛЕНИЕ ВСЕХ РЕГИСТРОВ ИМЕЕТ ВИД:

```
A  -  F  (===)  A"  -  F"
B  -  C  (===)  B"  -  C"
D  -  E  (===)  D"  -  E"
H  -  L  (===)  H"  -  L"
```

IX

IY

ЗНАЧЕНИЯ,ХРАНИМЫЕ В ОСНОВНЫХ РЕГИСТРАХ,СОХРАНЯЮТСЯ ЦП, ПОКА ИСПОЛЬЗУЕТСЯ АЛЬТЕРНАТИВНЫЙ НАБОР РЕГИСТРОВ,НО К НИМ НЕЛЬЗЯ ОСУЩЕСТВИТЬ ДОСТУП,

Е Щ Е РЕГИСТРЫ

"УКАЗАТЕЛЬ СТЕКА" — 2-БАЙТОВЫЙ РЕГИСТР АДРЕСА),

ОН ВСЕГДА УКАЗЫВАЕТ, ДО КАКОГО РАЗМЕРА ВЫРОС СТЕК, С РОСТОМ СТЕКА ОН СМЕЩАЕТСЯ ВНИЗ К ЯЧЕЙКАМ ПАМЯТИ С МЕНЬШИМИ НОМЕРАМИ,

ВАМ ОБЫЧНО НЕ ПРИХОДИТСЯ ПРИ ПРОГРАММИРОВАНИИ НА МАШИННОМ ЯЗЫКЕ ДЕЛАТЬ ЧТО-НИБУДЬ С УКАЗАТЕЛЕМ СТЕКА, ЗА НИМ СЛЕДИТ ЦП И МОДИФИЦИРУЕТ ЕГО КАЖДЫЙ РАЗ, КОГДА ВЫ ВЫПОЛНЯЕТЕ PUSH ИЛИ POP,

ОБРАТИТЕ ВНИМАНИЕ, ЧТО ЧАСТО ВСТРЕЧАЕТСЯ ОШИБКА, СОСТОЯЩАЯ В ТОМ, ЧТО ЗАБЫВАЮТ ВЫТОЛКНУТЬ НАЗАД ЗНАЧЕНИЕ, КОТОРОЕ ВТОЛКНУЛИ В СТЕК, МОЖЕТЕ БЫТЬ УВЕРЕНЫ, ЧТО ЭТО ПРИВЕДЕТ К СИТУАЦИИ "CRASH" (СБОЙ) В ВАШЕЙ ПРОГРАММЕ,

РЕГИСТР I

ЭТО — РЕГИСТР ВЕКТОРА ПРЕРЫВАНИЯ, В СИСТЕМАХ, ОСНОВАННЫХ НА Z80, ОТЛИЧНЫХ ОТ "СПЕКТРУМ" ЭТОТ РЕГИСТР ОБЫЧНО ПРИМЕНЯЕТСЯ ДЛЯ ХРАНЕНИЯ БАЗОВОГО АДРЕСА ТАБЛИЦЫ АДРЕСОВ ДЛЯ ОБРАБОТКИ РАЗЛИЧНЫХ РЕАКЦИЙ НА ПРЕРЫВАНИЕ, НАПРИМЕР, ЗАПРОСОВ ВВОДАВЫВОДА,

ОДНАКО, В "СПЕКТРУМ" ЭТО СРЕДСТВО НЕ ПРИМЕНЯЕТСЯ, И РЕГИСТР I УЧАВСТВУЕТ В ГЕНЕРАЦИИ СИГНАЛОВ УПРАВЛЕНИЯ ЭКРАНОМ, ВРЯД ЛИ ВАМ КОГДА-ЛИБО ПРИДЕТСЯ ПОЛЬЗОВАТЬСЯ ЭТИМ РЕГИСТРОМ,

РЕГИСТР R

РЕГИСТР R – РЕГИСТР ВОССТАНОВЛЕНИЯ ПАМЯТИ, В Z80 ОН ПРЕДНАЗНАЧЕН ДЛЯ АВТОМАТИЧЕСКОГО ОБНОВЛЕНИЯ ДИНАМИЧЕСКОЙ ПАМЯТИ, В ПРОЦЕССЕ РАБОТЫ ПРОЦЕССОРА Z80 ИНФОРМАЦИЯ, ХРАНИМАЯ В ТЕХ ЧАСТЯХ ДИНАМИЧЕСКОЙ ПАМЯТИ, К КОТОРЫМ ДОЛГО НЕ БЫЛО ДОСТУПА, БУДЕТ "УТЕКАТЬ" ИЗ-ЗА ПАДЕНИЯ НАПРЯЖЕНИЯ СО ВРЕМЕНЕМ, ЕСЛИ НЕ ОБНОВЛЯТЬ (ПЕРЕЗАРЯЖАТЬ) ЭТИ ЯЧЕЙКИ ПАМЯТИ, ТО ХРАНИВШАЯСЯ ПЕВОНАЧАЛЬНО ИНФОРМАЦИЯ ИСЧЕЗАЕТ!

РЕГИСТР R СЛУЖИТ ПРОСТЫМ СЧЕТЧИКОМ, НАРАЩИВАЕМЫМ КАЖДЫЙ РАЗ, КОГДА ПРОИСХОДИТ "ЦИКЛ ПОИСКА ИНФОРМАЦИИ В ПАМЯТИ", ТАКИМ ОБРАЗОМ, ЗНАЧЕНИЕ В РЕГИСТРЕ R ВСЕ ВРЕМЯ ЦИКЛИЧЕСКИ МЕНЯЕТСЯ ОТ 0 ДО 255,

ЭТОТ ФАКТ МОЖЕТ ИСПОЛЬЗОВАТЬСЯ АППАРАТУРОЙ ДЛЯ ОБЕСПЕЧЕНИЯ "ОБНОВЛЕНИЯ" ВСЕХ ЧАСТЕЙ ПАМЯТИ, НЕ НУЖНО БЕСПОКОИТЬСЯ, ВАМ НИЧЕГО НЕ НУЖНО ЗНАТЬ ОБ ЭТОМ, ДЕЛО ГОСПОДИНА СИНКЛЕРА ПОЗАБОТИТЬСЯ ОБ ЭТОМ ПРИ КОНСТРУИРОВАНИИ "СПЕКТРУМ", МЫ МОЖЕМ ПРОСТО ПОЛЬЗОВАТЬСЯ ЕГО ЭВМ, НЕ ДУМАЯ ОБ ОБНОВЛЕНИИ И ТОМУ ПОДОБНЫХ ВЕЩАХ,

С ТОЧКИ ЗРЕНИЯ ПРОГРАММИРОВАНИЯ ВЫ МОЖЕТЕ СЧИТАТЬ РЕГИСТР R ОТНОСЯЩИМСЯ ЦЕЛИКОМ К АППАРАТУРЕ И СИСТЕМНЫМ ПРИМЕНЕНИЯМ, ОДНАКО ИНОГДА ВЫ МОЖЕТЕ ПОЛЬЗОВАТЬСЯ ИМ КАК СРЕДСТВОМ ПОЛУЧЕНИЯ СЛУЧАЙНОГО ЧИСЛА ОТ 0 ДО 255, ЭТО ПРИМЕНЕНИЕ МЫ ПРОДЕМОНСТРИРУЕМ НИЖЕ,

К А К В Ы П О Л Н И Т Ь П Р О Г Р А М М У Н А МАШИННОМ ЯЗЫКЕ

"ZX СПЕКТРУМ" НА САМОМ ДЕЛЕ ВСЕ ВРЕМЯ ВЫПОЛНЯЕТ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ! (КОГДА ОН ВКЛЮЧЕН), ПРОСТО ВЫ ОБ ЭТОМ НЕ ЗНАЕТЕ ДАЖЕ КОГДА ВЫ НИЧЕГО НЕ ДЕЛАЕТЕ, ПРОСТО СМОТРИТЕ НА ЭКРАН, ПЫТАЯСЬ ПРИДУМАТЬ, ЧТО ВВЕСТИ В КАЧЕСТВЕ ПЕРВОЙ СТРОКИ ВАШЕЙ ПОТЯСАЮЩЕЙ ОСНОВЫ ПРОГРАММЫ НА ЯЗЫКЕ "БЕЙСИК", ЭВМ "СПЕКТРУМ" ЗАНЯТА РАБОТОЙ ПОД УПРАВЛЕНИЕМ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ,

ЭТО ОДНА ИЗ ПРОГРАММ, ХРАНИМЫХ В ПЗУ, И ЕЕ НАЗЫВАЮТ "ОПЕРАЦИОННОЙ СИСТЕМОЙ", НАПРИМЕР, ЧАСТЬ ПРОГРАММЫ, ВЫПОЛНЯЮЩАЯСЯ, КОГДА ВЫ ПРОСТО СИДИТЕ И СМОТРИТЕ НА ЭКРАН, ВЫПОЛНЯЕТ СЛЕДУЮЩЕЕ: СКАНИРУЕТ КЛАВИАТУРУ В ПОИСКАХ ВВЕДЕННОЙ ИНФОРМАЦИИ;

ОТМЕЧАЕТ, ЧТО НИ ОДНА КЛАВИША НЕ НАЖАТА;

ВЫДАЕТ НА ДИСПЛЕЙ ТЕКУЩИЙ ЭКРАН (ПУСТОЙ),

ДАЖЕ КОГДА ВЫ ВЫПОЛНЯЕТЕ ПРОГРАММУ НА ЯЗЫКЕ "БЕЙСИК", ЦП УПРАВЛЯЕТСЯ ПРОГРАММОЙ НА МАШИННОМ ЯЗЫКЕ, КАК МЫ УЖЕ ОБ"ЯСНЯЛИ, ЭТА ПРОГРАММА ОТНОСИТСЯ К ТИПУ "ИНТЕРПРЕТАТОРОВ": ОНА БЕРЕТ ВАШУ СЛЕДУЮЩУЮ КОМАНДУ ЯЗЫКА "БЕЙСИК", ПРЕОБРАЗУЕТ ЕЕ НА МАШИННЫЙ ЯЗЫК, ВЫПОЛНЯЕТ ЭТУ ЧАСТЬ ПРОГРАММЫ И ЗАТЕМ ВОЗВРАЩАЕТСЯ К ИНТЕРПРЕТАЦИИ СЛЕДУЮЩЕЙ КОМАНДЫ,

ВСЕ ЭТО ОКАЗЫВАЕТСЯ НЕ ТАК, КОГДА ВЫ ВЫПОЛНЯЕЕТЕ СВОЮ СОБСТВЕННУЮ ПРОГРАММУ НА МАШИННОМ ЯЗЫКЕ!

ПОЛНАЯ СВОБОДА ОТ ОПЕРАЦИОННОЙ СИСТЕМЫ! ПРИМЕНЕНИЕ ФУНКЦИИ "USR" ПЕРЕДАЕТ УПРАВЛЕНИЕ ЦП ТЕМ КОМАНДАМ, КОТОРЫЕ ВЫ ПОМЕСТИЛИ ПО АДРЕСУ USR, КАКОВЫ БЫ ОНИ НИ БЫЛИ, ЧТО БЫ ТАМ НИ ОКАЗАЛОСЬ, ОН БУДЕТ ИНТЕРПРЕТИРОВАТЬ ЭТО КАК ДОПУСТИМЫЕ КОМАНДЫ МАШИННОГО ЯЗЫКА,

ПЕРСПЕКТИВА ДОВОЛЬНО ПУГАЮЩАЯ, ВЕДЬ ПРИ ПОТЕРЕ УПРАВЛЕНИЯ ВЫ МОЖЕТЕ ПОТЕРЯТЬ ВСЕ, ЧТО ХРАНИЛОСЬ В ПАМЯТИ, ОДНА ОШИБКА, ОДИН НЕВЕРНЫЙ СИМВОЛ, И ВАМ ПРИДЕТСЯ ВЫКЛЮЧАТЬ "СПЕКТРУМ" И ВНОВЬ НАЧИНАТЬ СНАЧАЛА,

НЕТ НИ СООБЩЕНИЙ ОБ ОШИБКАХ, ПОКАЗЫВАЮЩИХ, ГДЕ ВЫ ОШИБЛИСЬ, НИ СИНТАКСИЧЕСКОЙ ПРОВЕРКИ НЕВЕРНЫХ ПРЕДЛОЖЕНИЙ – ТАК ЧТО ЕСЛИ ВЫ СДЕЛАЕТЕ МАЛЕЙШУЮ ОШИБКУ, БУДУТ ПОТЕРЯНЫ ЧАСЫ РАБОТЫ, ЗАТРАЧЕННЫЕ НА ВВОД ВАШЕЙ ПРОГРАММЫ!

В КОНЦЕ ЭТОЙ КНИГИ МЫ ВКЛЮЧИЛИ ПРОГРАММУ НА ЯЗЫКЕ "БЕЙСИК", КОТОРАЯ ПОЗВОЛИТ ВАМ ВВОДИТЬ И РЕДАКТИРОВАТЬ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ, ВВЕДЯ ЭТУ ПРОГРАММУ В ВАШ "СПЕКТРУМ", ЗАПИШИТЕ ЕЕ НА ЛЕНТУ, ПОСКОЛЬКУ БОЛЕЕ ЧЕМ ВЕРОЯТНО, ЧТО ВЫ ПО КРАЙНЕЙ МЕРЕ ОДНАЖДЫ ПОТЕРЯЕТЕ УПРАВЛЕНИЕ ВАШЕЙ ПРОГРАММОЙ НА МАШИННОМ ЯЗЫКЕ,

С ДРУГОЙ СТОРОНЫ, НЕ БОЙТЕСЬ ЭКСПЕРИМЕНТИРОВАТЬ – ВЫ НЕ

СМОЖЕТЕ ИСПОРТИТЬ ЭВМ КАКОЙ БЫ ТО НИ БЫЛО ПРОГРАММОЙ НА МАШИННОМ ЯЗЫКЕ, ВВОДИМОЙ ВАМИ, САМОЕ ХУДШЕЕ, ЧТО МОЖЕТ ПРОИЗОЙТИ, - ВАМ ПРИДЕСТСЯ ВЫКЛЮЧИТЬ И ВНОВЬ ВКЛЮЧИТЬ ВАШ "СПЕКТРУМ",

СЕЙЧАС МЫ ВАМ ДЛЯ ВОЗБУЖДЕНИЯ АППЕТИТА ДАДИМ ОЧЕНЬ ПРОСТУЮ ПРОГРАММУ НА МАШИННОМ ЯЗЫКЕ, ЗАГРУЗИТЕ НАПИСАННУЮ НА ЯЗЫКЕ "БЕЙСИК" ПРОГРАММУ "РЕДАКТОР МАШИННОГО ЯЗЫКА В КОДЕ EZ", НАЙДЕННУЮ В КОНЦЕ ЭТОЙ КНИГИ, И ВЫПОНИТЕ ЕЕ,

ПРОГРАММА ЗАПРОСИТ У ВАС АДРЕС ЗАГРУЗКИ, ЭТО ЗНАЧИТ СПРОСИТ У ВАС, ГДЕ БЫ ВЫ ХОТЕЛИ РАЗМЕСТИТЬ МАШИННЫЕ КОМАНДЫ, В ЭТОЙ ПРОГРАММЕ КОДА ВЫ НЕ МОЖЕТЕ ПРИМЕНЯТЬ АДРЕСА МЕНЕЕ 3150 Ø, ТАК ЧТО ДАВАЙТЕ ВЫБЕРЕМ 32ØØØ, ВВЕДИТЕ ЧИСЛО 32ØØØ, ЗАТЕМ НАЖМИТЕ (ENTER),

НА ЭКРАНЕ ТЕПЕРЬ СВЕТИТСЯ: КОМАНДА ИЛИ СТРОКА (# # #):

ЭТО ОЗНАЧАЕТ,ЧТО ПРОГРАММА ЖДЕТ, ЧТОБЫ ВЫ ВВЕЛИ КОМАНДУ ИЛИ НОВУЮ СТРОКУ ТЕКСТА МАШИННОЙ ПРОГРАММЫ,

ДАВАЙТЕ ВВЕДЕМ "1", ЗАТЕМ ПРОБЕЛ, ЗАТЕМ "C" И ЗАТЕМ "9", ЭТО ПОХОЖЕ НА ВВОД СТРОКИ ЯЗЫКА "БЕЙСИК" С НОМЕРОМ СТРОКИ 1, НО ЭТО - СТРОКА МАШИННОГО ЯЗЫКА, ЕСЛИ ВСЕ В ПОРЯДКЕ, ТО НАЖМИТ (ENTER), НА ЭКРАНЕ ТЕПЕРЬ ДОЛЖНЫ БЫТЬ ПОКАЗАНЫ ВСЕ ВВЕДЕННЫЕ ВАМИ СТРОКИ:

1 C9 И ВНИЗУ ЭКРАНА - ПОДСКАЗКА "КОМАНДА ИЛИ СТРОКА (# # #):"

НА ЭТОМ ЭТАПЕ ВЫ НЕ ХОТИТЕ ДОБАВЛЯТЬ БОЛЬШЕ СТРОК, ТАК ЧТО ДАВАЙТЕ ВМЕСТО ЭТОГО ВВЕДЕМ КОМАНДУ,

ВВЕДИТЕ СЛОВО "DUMP" И ЗАТЕМ НАЖМИТЕ (ENTER), ДЕЙСТВИЕ ЭТОЙ КОМАНДЫ СОСТОИТ В ВЫДАЧЕ ТЕКСТА МАШИННОЙ ПРОГРАММЫ В ВИДЕ РАСПЕЧАТКИ ПО УКАЗАННОМУ ВАМИ АДРЕСУ,А ИМЕННО 32ØØØ,

ВАС МОЖНО ПОЗДРАВИТЬ ,ВЫ ТОЛЬКО ЧТО ВВЕЛИ КОМАНДУ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ! ВЫ МОЖЕТЕ ПРОВЕРИТЬ, ЧТО ОНА БЫЛА ВВЕДЕНА ПРАВИЛЬНО, ВВЕДЯ ТЕПЕРЬ КОМАНДУ "MEM" И ЗАТЕМ (ENTER). ЭТА КОМАНДА ПОЗВОЛЯЕТ ВАМ ПРОСМОТРЕТЬ СОДЕРЖИМОЕ ПАМЯТИ, И ОНА ЗАПРОСИТ У ВАС НАЧАЛЬНЫЙ АДРЕС, ВВЕДИТЕ ТОГДА 32ØØØ И ЗАТЕМ (ENTER),

ВЫ УВИДИТЕ СОДЕРЖИМОЕ ЯЧЕЕК ПАМЯТИ С 32ØØØ ПО 32Ø87, ВО ВСЕХ ДОЛЖНО БЫТЬ ØØ, ЗА ИСКЛЮЧЕНИЕМ 32ØØØ, В КОТОРОЙ ДОЛЖНО БЫТЬ C9, ЗАТЕМ НАЖМИТЕ КЛАВИШУ "M", ЧТОБЫ ВЕРНУТЬСЯ НА ЭТАП ВВОДА ОСНОВНОЙ КОМАНДЫ,

КОМАНДА "C9" ОЗНАЧАЕТ RETURN (ВОЗВРАТ)!

ВСЕ ЭТО НЕСКОЛЬКО НАПОМИНАЕТ ЕЗДУ НА ВЕЛОСИПЕДЕ В ПЕРВЫЙ РА РАЗ: ВЫ ДЕЙСТВИТЕЛЬНО ХОТИТЕ ПОЛУЧИТЬ САМОСТОЯТЕЛЬНОСТЬ, НО КАК ТОЛЬКО НЕМНОГО ПРОЕДЕТЕ, ХОЧЕТСЯ "ВЕРНУТЬСЯ" К НАДЕЖНОСТИ ТВЕРДОЙ ЗЕМЛИ ПОД НОГАМИ (ИНЫМИ СЛОВАМИ, ОПЕРАЦИОННОЙ СИСТЕМЫ)

ТЕПЕРЬ ПЕРЕХОДИМ К ВЫПОЛНЕНИЮ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ, ЧТОБЫ ВЫПОЛНИТЬ ЛЮБУЮ ПРОГРАММУ НА МАШИННОМ ЯЗЫКЕ, ЗАПИСАННУЮ ВАМИ В ПАМЯТЬ, ВВЕДИТЕ КОМАНДУ "RUN" И ЗАТЕМ (ENTER),

ЧТО ПРОИСХОДИТ? ПОЧЕМУ ВНИЗУ ЭКРАНА ПОЯВИЛОСЬ ЧИСЛО 32ØØØ? ЭТО - АДРЕС, ИСПОЬЗОВАННЫЙ ВАМИ В КАЧЕСТВЕ АДРЕСА ЗАГРУЗКИ В НАЧАЛЕ,

НЕ ЗАБЫВАЙТЕ, ЧТО ФУНКЦИЯ "USR" СОСТОИТ В ВЫПОЛНЕНИИ ПОДПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ, В ЧАСТНОСТИ ИЗ ЭТОЙ ФУНКЦИИ ВЫТЕКАЕТ, ЧТО ЗНАЧЕНИЕ USR ПРИ ВОЗВРАТЕ ИЗ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ, ПОМЕЩЕННОЙ ВАМИ В ПАМЯТЬ, БУДЕТ РАВНО ЗНАЧЕНИЮ ПАРЫ РЕГИСТРОВ BC,

ОТВЕТ НА ЭТОТ ВОПРОС ВЫТЕКАЕТ ИЗ СПОСОБА, КОТОРЫМ ОПЕРАЦИОН НАЯ СИСТЕМА "СПЕКТРУМ" (ДА, ВСЕ ТА ЖЕ САМАЯ) ОБРАЩАЕТСЯ С ФУНК ЦИЕЙ "USR",

КОГДА ОПЕРАЦИОННАЯ СИТЕМА ВСТРЕЧАЕТ ФУНКЦИЮ "USR", ОНА ЗАГРУЖАЕТ УКАЗАННЫЙ ПОЛЬЗОВАТЕЛЕМ АДРЕС В ПАРУ РЕГИСТРОВ BC - В ДАННОМ СЛУЧАЕ 32ØØØ,

ЗНАЧЕНИЕ "USR", КАК В ПРЕДЛОЖЕНИИ

LET A = USR 32ØØØ ЕСТЕСТВЕННО, ПРИВОДИТ К ОТВЕТУ 32ØØØ!

ЭТА ОСОБЕННОСТЬ ФУНКЦИИ "USR" БУДЕТ ОЧЕНЬ ПОЛЕЗНА, ПОСКОЛЬКУ ОНА ПОЗВОЛИТ НАМ СЛЕДИТЬ ЗА ТЕМ, ЧТО ПРОИСХОДИТ ВО ВРЕМЯ ВЫ ПОЛНЕНИЯ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ,

ДАВАЙТЕ ВВЕДЕМ СЛЕДУЮЩУЮ ПРОГРАММУ НА МАШИННОМ ЯЗЫКЕ:

ØB
C9

ВВЕСТИ ЭТУ КОРОТКУЮ, СОСТОЯЩУЮ ИЗ ДВУХ КОМАНД, ПРОГРАММУ МОЖНО ТАК,

ВВЕСТИ СТРОКУ 1 , ВВЕДЯ "1", ЗАТЕМ ПРОБЕЛ, ЗАТЕМ "Ø", ЗАТЕМ "B" И ЗАТЕМ НАЖАВ (ENTER), АНАЛОГИЧНЫМ ОБРАЗОМ ВВЕСТИ СТРОКУ 2 C9, РАСПЕЧАТКА ДОЛЖНА ПОКАЗАТЬ ВАМ, ЧТО ВЫ ВВЕЛИ СТРОКИ ПРАВИЛЬНО, ВВЕДИТЕ КОМАНДУ "DUMP" И ЗАТЕМ КОМАНДУ "RUN"

НА ЭТОТ РАЗ РЕЗУЛЬТАТ РАВЕН 31999! ТАК ПОЛУЧИЛОСЬ ПОТОМУ, ЧТО КОМАНДА "ØB" - ЭТО "DEC BC" (СОКРАЩЕННОЕ НАЗВАНИЕ ДЛЯ УМЕНЬШЕНИЯ ЗНАЧЕНИЯ BC НА 1),

В И Д Ы А Д Р Е С А Ц И И В ZX SPECTRUM:

НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ;
РЕГИСТРОВАЯ АДРЕСАЦИЯ;
КОСВЕННАЯ РЕГИСТРОВАЯ АДРЕСАЦИЯ;
РАСШИРЕННАЯ АДРЕСАЦИЯ;
ИНДЕКСНАЯ АДРЕСАЦИЯ.

ПРИВЕДЕННЫЙ ВЫШЕ СПИСОК НЕ ПОКРЫВАЕТ ВСЕ ВОЗМОЖНЫЕ КОМБИНАЦИИ - ТОЛЬКО ТЕ, КОТОРЫЕ ОТНОСЯТСЯ К ЧИСЛАМ ДЛЯ ОДНОГО РЕГИСТРА, ДАВАЙТЕ ПООЧЕРЕДНО РАССМОТРИМ КАЖДОЕ ИЗ ЭТИХ ВОЗМОЖНЫХ ВИДОИЗМЕНЕНИЙ.

НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ

ОБЩАЯ ФОРМА ДЛЯ НЕЕ ИМЕЕТ ВИД

LD R, N (ИЛИ ИНАЯ КОМАНДА - МЫ ИСПОЛЬЗУЕМ LD ТОЛЬКО В КАЧЕСТВЕ ПРИМЕРА

МЫ ИСПОЛЬЗУЕМ СОКРАЩЕНИЕ "R" ДЛЯ ОБОЗНАЧЕНИЯ ПРОИЗВОЛЬНОГО 8-БИТОВОГО РЕГИСТРА, А "N" - ДЛЯ 8-БИТОВОГО ЧИСЛА.

НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ - ЭТО МЕТОД, ИСПОЛЬЗУЮЩИЙ ТОЛЬКО ОДИН РЕГИСТР, РЕАЛЬНЫЕ ДАННЫЕ ВХОДЯТ СОСТАВНОЙ ЧАСТЬЮ В КОМАНДУ; ЭТО ОЗНАЧАЕТ, ЧТО ЦП МОЖЕТ ВЫПОЛНЯТЬ КОМАНДУ НЕПОСРЕДСТВЕННО ПОСЛЕ ЕЕ ПОЛУЧЕНИЯ, ЕМУ НЕ ПРИХОДИТСЯ ВЕСТИ ПОИСК ДОПОЛНИТЕЛЬНОЙ ИНФОРМАЦИИ В ПАМЯТИ, ЧТОБЫ ВЫПОЛНИТЬ ЭТУ КОМАНДУ.

ОБЫЧНО МЫ ПРИМЕНЯЕМ НЕПОСРЕДСТВЕННУЮ АДРЕСАЦИЮ ДЛЯ ИНИЦИАЛИЗАЦИИ СЧЕТЧИКОВ И ОПРЕДЕЛЕНИЯ КОНСТАНТ, НЕОБХОДИМЫХ ДЛЯ ВЫЧИСЛЕНИЙ.

НЕПОСРЕДСТВЕННУЮ АДРЕСАЦИЮ ЛЕГКО ПРИМЕНЯТЬ В ПРОГРАММИРОВАНИИ НА МАШИННОМ ЯЗЫКЕ, ТЕМ НЕ МЕНЕЕ, ЭТО - НАИМЕНЕЕ ГИБКИЙ ИЗ ВСЕХ МЕТОДОВ ПЕРЕДАЧИ ИНФОРМАЦИИ (СПОСОБОВ АДРЕСАЦИИ), ПОСКОЛЬ КУ И РЕГИСТР, И ДАННЫЕ ФИКСИРУЮТСЯ В МОМЕНТ НАПИСАНИЯ ПРОГРАММЫ, ЭКВИВАЛЕНТНОЕ ПРЕДЛОЖЕНИЕ НА ЯЗЫКЕ "БЕЙСИК" ВЫГЛЯДЕЛО БЫ ТАК: LET A=5 ОЧЕВИДНО, КОМАНДЫ ТАКОГО ТИПА НЕОБХОДИМЫ, НО МЫ НЕ МОГЛИ БЫ ПИСАТЬ ТАКИМ СПОСОБОМ ПРОГРАММЫ ЦЕЛИКОМ!

РЕГИСТРОВАЯ АДРЕСАЦИЯ

МЫ УЖЕ КРАТКО РАССМАТРИВАЛИ ЭТОТ СПОСОБ, ОБЩИЙ ФОРМА ИМЕЕТ ВИД

LD R,R

(ИЛИ ДРУГИЕ КОМАНДЫ) ПРИ ЭТОМ МЕТОДЕ УЧАСТВУЮТ ТОЛЬКО 2 РЕГИСТРА,

ЦП ДОПУСКАЕТ ПЕРЕДАЧУ ИНФОРМАЦИИ МЕЖДУ ЛЮБЫМИ ДВУМЯ РЕГИСТРАМИ, ЗА ИСКЛЮЧЕНИЕМ РУКИ "F",

ДЛЯ КОМАНД С РЕГИСТРОВОЙ АДРЕСАЦИЕЙ ТРЕБУЕТСЯ ВСЕГО ОДИН БАЙТ,

КОМАНДЫ ЭТОГО ТИПА НЕ ТОЛЬКО КОРОТКИ (ОДИН БАЙТ), ОНИ ТАКЖЕ И БЫСТРЕЕ ВЫПОЛНЯЮТСЯ, ТРЕБУЮЩЕЕСЯ ДЛЯ ИХ ВЫПОЛНЕНИЯ ВРЕМЯ СОСТАВЛЯЕТ 4 ИМПУЛЬСА ГЕНЕРАТОРА СИНХРОСИГНАЛОВ, ИЛИ МЕНЕЕ 1 МИКРОСЕКУНДЫ ДЛЯ "СПЕКТРУМ",

ДЛЯ НАПИСАНИЯ ПРОГРАММ НА МАШИННОМ ЯЗЫКЕ СУЩЕСТВУЕТ "ПРАВИЛО", ЧТО ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ПРОГРАММЫ В ПЛАНЕ ЗАТРАТ ВРЕМЕНИ И ПАМЯТИ ВСЕГДА, КОГДА ЕСТЬ ВОЗМОЖНОСТЬ, СЛЕДУТ ПРИМЕНЯТЬ ПЕРЕДАЧУ ИНФОРМАЦИИ РЕГИСТР - РЕГИСТР,

КОСВЕННАЯ РЕГИСТРОВАЯ АДРЕСАЦИЯ

LD (RR), A ИЛИ LD A,(RR)
LD (HL), N

ЭТОТ БОГАТЫЙ ВОЗМОЖНОСТЯМИ ТИП КОМАНДЫ ПРИВОДИТ К ПЕРЕДАЧЕ

ДАННЫХ МЕЖДУ ЦП И ЯЧЕЙКОЙ ПАМЯТИ, НА КОТОРУЮ УКАЗЫВАЕТ СОДЕРЖИМОЕ ОДНОЙ ИЗ 16-БИТОВЫХ ПАР.

КОСВЕННАЯ РЕГИСТРОВАЯ АДРЕСАЦИЯ ДЕЙСТВУЕТ БЫСТРЕЕ, ЧЕМ ОБЫЧНАЯ КОСВЕННАЯ АДРЕСАЦИЯ, ПОСКОЛЬКУ ЦП НЕТ НЕОБХОДИМОСТИ ИЗВЛЕКАТЬ АДРЕС ИЗ ПАМЯТИ.

ТЕМ НЕ МЕНЕЕ, МЫ ДОЛЖНЫ СНАЧАЛА ЗАГРУЗИТЬ РЕГИСТР, ТАК ЧТО КОСВЕННАЯ РЕГИСТРОВАЯ АДРЕСАЦИЯ ДАЕТ ПРЕИМУЩЕСТВА ТОЛЬКО ТОГДА, КОГДА ПРОГРАММА ИСПОЛЬЗУЕТ ОДИН И ТОТ ЖЕ ИЛИ СОСЕДНИЕ АДРЕСА МНОГОКРАТНО.

НАПРИМЕР,
```
        LD HL,SHAPE  ;  ЗАГРУЗИТЬ НАЧАЛО БАЗЫ ДАННЫХ
                     ;  SHAPE В HL
LOOP    LD A,(HL)    ;  ПОИСК ДАННЫХ
        INC HL       ;  ДАТЬ ПРИРАЩЕНИЕ УКАЗАТЕЛЮ
        ПРОДОЛЖАТЬ LOOP ПОКА НЕ КОНЧИТСЯ
```
РАСШИРЕННАЯ АДРЕСАЦИЯ

LD A,(NN) ИЛИ LD (NN),A

ТЕПЕРЬ ПОСМОТРИМ, КАК ПРОИСХОДИТ ОБМЕН ИНФОРМАЦИЕЙ МЕЖДУ ПАМЯТЬЮ И РЕГИСТРАМИ.

ПРИ РАСШИРЕННОЙ АДРЕСАЦИИ КОМАНДА ПРОГРАММЫ ПРЕДОСТАВЛЯЕТ ЦП АДРЕС, ЗАДАННЫЙ ДВУМЯ БАЙТАМИ, ЕСЛИ ПЕРЕДАЧА ИНФОРМАЦИИ ОСУЩЕСТВЛЯЕТСЯ В ИЛИ ИЗ НАКАПЛИВАЮЩЕГО РЕГИСТРА, ТО ОНА ПОВЛИЯЕТ ТОЛЬКО НА СОДЕРЖИМОЕ ЯЧЕЙКИ ПАМЯТИ, ЗАДАННОЙ ДВУХБАЙТОВЫМ ЦЕЛЫМ ЧИСЛОМ.

ЕСЛИ ЖЕ ПЕРЕДАЧА ИНФОРМАЦИИ ОСУЩЕСТВЛЯЕТСЯ В ИЛИ ИЗ ПАРЫ РЕГИСТРОВ, ТО ОНА ПОВЛИЯЕТ И НА ЯЧЕЙКУ ПАМЯТИ, ЗАДАННУЮ ДВУХБАЙТОВЫМ ЦЕЛЫМ ЧИСЛОМ, И НА СЛЕДУЮЩУЮ ЯЧЕЙКУ.

КОМАНДА ТАКОГО ТИПА ИМЕЕТ СЛЕДУЮЩИЙ ФОРМАТ:

БАЙТ 1 КОД ОПЕРАЦИИ
БАЙТ 2 (ДОПУСТИМЫЙ ДОПОЛНИТЕЛЬНЫЙ КОД ОПЕРАЦИИ)
БАЙТ 3 МЛАДШИЕ РАЗРЯДЫ 16-БИТОВОГО ЦЕЛОГО ЧИСЛА
БАЙТ 4 СТАРШИЕ РАЗРЯДЫ ЭТОГО ЦЕЛОГО ЧИСЛА

ИМЕННО ТАКИМ СПОСОБОМ ПРОГРАММА МОЖЕТ СЧИТЫВАТЬ СОДЕРЖИМОЕ ПАМЯТИ В РЕГИСТРЫ ПОЛЬЗОВАТЕЛЯ, И ВНОВЬ ЗДЕСЬ ТРЕБУЕТСЯ АБСОЛЮТНЫЙ АДРЕС, ИНЫМИ СЛОВАМИ, В РЕЗУЛЬТАТЕ ПРИМЕНЕНИЯ ЭТОГО ТИПА АДРЕСАЦИИ ПРОГРАММА МОЖЕТ ОКАЗАТЬСЯ НЕПЕРЕМЕЩАЕМОЙ, ЗА ИСКЛЮЧЕНИЕМ СЛУЧАЕВ, КОГДА ПЕРЕМЕЩАЕМЫМ ОКАЗЫВАЕТСЯ АБСОЛЮТНЫЙ АДРЕС, НА КОТОРЫЙ ССЫЛАЕТСЯ КОМАНДА.
```
SHAPE: DB N,N,N,;,,,  ; БАЗА ДАННЫХ SHAPE
       ,
       ,
       LD A,(SHAPE)   ; ЗАГРУЗИТЬ ПЕРВЫЙ БАЙТ В
                      ; НАКАПЛИВАЮЩИЙ РЕГИСТР
```
ИНДЕКСНАЯ АДРЕСАЦИЯ

LD R,(IX/IY+Д) ИЛИ LD (IX/IY+Д),R

(ИЛИ ДРУГИЕ КОМАНДЫ)

В ПЕРЕДАЧЕ ИНФОРМАЦИИ ТАКОГО ТИПА УЧАСТВУЕТ ИНДЕКСНЫЙ РЕГИСТР IX ИЛИ IY.

ЦП СКЛАДЫВАЕТ СОДЕРЖИМОЕ ИНДЕКСНОГО РЕГИСТРА С АДРЕСОМ, ЗАДАННЫМ В КОМАНДЕ, ЧТОБЫ НАЙТИ ИСПОЛНИТЕЛЬНЫЙ АДРЕС.

ЭТО - ОДИН ИЗ ТИПОВ КОМАНД Z80, ИМЕЮЩИХ 16-БИТОВЫЙ КОД ОПЕРАЦИИ, ДРУГОЙ ШИРОКО ПРИМЕНЯЕМЫЙ 16-БИТОВЫЙ ТИП КОМАНДЫ - КОМАНДА ЗАГРУЗКИ БЛОКА, НАПРИМЕР, LDIR (LOAD INCREMENT AND REPEAT - ЗАГРУЗИТЬ, ДАТЬ ПРИРАЩЕНИЕ И ПОВТОРИТЬ)

ОДНО ИЗ ТИПИЧНЫХ ПРИМЕНЕНИЙ МЕТОДА АДРЕСАЦИИ ТАКОГО ТИПА - ВЫПОЛНЕНИЕ ТАБЛИЧНЫХ ОПЕРАЦИЙ.

ИНДЕКСНЫЙ РЕГИСТР МОЖЕТ ПРИМЕНЯТЬСЯ КАК УКАЗАТЕЛЬ НА НАЧАЛО ТАБЛИЦЫ ДАННЫХ, ЗНАЧЕНИЕ СМЕЩЕНИЯ ЗАДАЕТСЯ В КОМАНДЕ, ЧТОБЫ ОПРЕДЕЛИТЬ АДРЕС ТРЕБУЕМОГО ЭЛЕМЕНТА ТАБЛИЦЫ, К КОТОРОМУ НУЖНО ОБРАТИТЬСЯ ПРОГРАММЕ; НАПРИМЕР, LD IX,TABLESTART
```
; ИНИЦИАЛИЗИРОВАТЬ УКАЗАТЕЛЬ
                    ;  НА НАЧАЛО ТАБЛИЦЫ
        LD A,(IX+3) ;  ОБРАТИТЬСЯ К ТРЕТЬЕМУ БАЙТУ
                    ;  ОТ НАЧАЛА ТАБЛИЦЫ
```
КОМАНДА ЭТОГО ТИПА ИМЕЕТ СЛЕДУЮЩИЙ ФОРМАТ:

БАЙТ 1 (КОД ОПЕРАЦИИ)
БАЙТ 2 (КОД ОПЕРАЦИИ)

БАИТ 3 D ; ЦЕЛОЧИСЛЕННОЕ ЗНАЧЕНИЕ СМЕЩЕНИЯ

ЧИСЛО "D" – 8-БИТОВОЕ ЧИСЛО, КОТОРОЕ НЕОБХОДИМО ЗАДАВАТЬ ВМЕСТЕ С КОМАНДОЙ, И ОНО НЕ МОЖЕТ БЫТЬ ПЕРЕМЕННЫМ, ЭТО ОЗНАЧАЕТ, ЧТО ДИАПАЗОН АДРЕСАЦИИ ОГРАНИЧЕН ЗНАЧЕНИЯМИ ОТ -128 ДО 127 СЧИТАЯ ОТ АДРЕСА, ЗАДАННОГО ИНДЕКСНЫМ РЕГИСТРОМ,

ИНДЕКСНАЯ АДРЕСАЦИЯ ВЫПОЛНЯЕТСЯ МЕДЛЕННЕЕ, ПОСКОЛЬКУ ЦП ДЛЯ ПОЛУЧЕНИЯ ИСПОЛНИТЕЛЬНОГО АДРЕСА ДОЛЖЕН ВЫПОЛНИТЬ СЛОЖЕНИЕ, И ВСЕ-ТАКИ ИНДЕКСНАЯ АДРЕСАЦИЯ ГОРАЗДО ГИБЧЕ, ПОСКОЛЬКУ С ПОМОЩЬЮ ОДНОЙ И ТОЙ ЖЕ КОМАНДЫ МОЖНО ОБРАБАТЫВАТЬ ВСЕ ЭЛЕМЕНТЫ МАССИВА ИЛИ ТАБЛИЦЫ,

ТАБЛИЦА

КОМАНДЫ ОПЕРАЦИЙ ЗАГРУЗКИ ДЛЯ ОДНГО РЕГИСТРА

MNEMONIK	BYTES	TIME TAKEN	EFFECT ON FLAGE					
			C	Z	PV	S	N	H
LD REGISTER,REGISTER	1	4	-	-	-	X	-	-
LD REGISTER,NUMBER	2	7	-	-	-	8	-	-
LD A, (ADDRESS)	3	13	-	-	-	0	-	-
LD (ADDRESS),A	3	13	-	-	-	0	-	-
LD REGISTER,(HL)	1	7	-	-	-	H	-	-
LD A,(BC)	1	7	-	-	-	H	-	-
LD A,(DC)	1	7	-	-	-	N	-	-
LD (HL),REGISTER	1	7	-	-	-	M	-	-
LD (BC),A	1	7	-	-	-	Q	-	-
LD (DE), A	1	7	-	-	-	W	-	-
LD REGISTER,(IX+Д)	3	19	-	-	-	B	-	-
LD REGISTER,(IY+Д)	3	19	-	-	-	J	-	-
LD (IX+Д),REGISTER	3	19	-	-	-	D	-	-
LD (IY+Д),REGISTER	3	19	-	-	-	F	-	-
LD (HL),NUMBER	2	10	-	-	-	T	-	-
LD (IX+Д),NUMBER	4	19	-	-	-	Ж	-	-
LD (IY+Д),NUMBER	4	19	-	-	-	A	-	-

MNEMONIC-МНЕМОНИЧЕСКОЕ ОБОЗНАЧЕНИЕ; BYTES-КОЛИЧЕСТВО БАЙТОВ; TIME TAKEN-ЗАТРАЧИВАЕМОЕ ВРЕМЯ; EFFECT ON FLOGE-СОСТОЯНИЕ ФЛАГОВ ПОСЛЕ ВЫПОЛНЕНИЯ; REGISTER-РЕГИСТР; NUMBER-ЧИСЛО; ADDRESS-АДРЕС,

ОБОЗНАЧЕНИЯ ДЛЯ ФЛАГОВ;

\# - УКАЗЫВАЕТ, ЧТО ФЛАГ МЕНЯЕТСЯ В РЕЗУЛЬТАТЕ ОПЕРАЦИИ;
Ø - УКАЗЫВАЕТ, ЧТО ФЛАГ СБРОШЕН;
1 - УКАЗЫВАЕТ, ЧТО ФЛАГ УСТАНОВЛЕН;
\- - УКАЗЫВАЕТ, ЧТО ФЛАГ ОСТАЕТСЯ НЕИЗМЕННЫМ,

ОСТАНОВИМСЯ ПОДРОБНЕЕ НА НЕКОТОРЫХ СПОСОБАХ ПЕРЕДАЧИ ИНФОРМАЦИИ,

LD A,B
LD H,E
И Т.Д,

НАПОМНИМ ИСПОЛЬЗУЕМУЮ ТЕРМИНОЛОГИЮ: "LD" ОЗНАЧАЕТ "ЗАГРУЗИТЬ", "," ОЗНАЧАЕТ "В" И КОМАНДА В МНЕМОНИЧЕСКОМ ВИДЕ (СОКРАЩЕННОМ) ЧИТАЕТСЯ КАК РУССКОЕ ПРЕДЛОЖЕНИЕ, НО С ОБРАТНЫМ ПОРЯДКОМ ОПЕРАНДОВ, (В ОРИГИНАЛЕ, В СООТВЕТСТВИИ С НОРМАМИ АНГЛИЙСКОГО ЯЗЫКА ГОВОРИТСЯ О ПРЯМОМ ПОРЯДКЕ ОПЕРАНДОВ,

ТАКИМ ОБРАЗОМ, ВСЛУХ ТАКУЮ, НАПРИМЕР, КОМАНДУ, КАК

LD A,B МЫ ПРОЧИТАЛИ БЫ "ЗАГРУЗИТЬ В В А ", А СЛЕДУЮЩИЙ ПРИМЕР ЧИТАЛСЯ БЫ КАК "ЗАГРУЗИТЬ Е В Н",

ВТОРОЙ СПОСОБ РАССМОТРИМ НА ПРИМЕРЕ:

```
    LD D,D7
```

(D7 - ШЕСТНАДЦАТЕРИЧНОЕ ПРЕДСТАВЛЕНИЕ 215)
В ДАННОМ СЛУЧАЕ В РЕГИСТР D ЗАСЫЛАЕТСЯ ЧИСЛО 215,

ВОЗМОЖНО, ВЫ ПОМНИТЕ, ЧТО ЭТО НАЗЫВАЕТСЯ НЕПОСРЕДСТВЕННОЙ АДРЕСАЦЕЙ, (НАЗВАНИЕ ДОСТАТОЧНО ОЧЕВИДНОЕ, НЕ ПРАВДА ЛИ?

ОПЯТЬ-ТАКИ, ВЫ МОЖЕТЕ ПРОДЕЛАТЬ ЭТО С ЛЮБЫМИ РЕГИСТРАМИ И ЛЮБЫМИ ЧИСЛАМИ, ОГРАНИЧЕНИЕМ, КОНЕЧНО, СЛУЖИТ РАЗМЕР ЧИСЛА, КО ТОРОЕ МОЖНО ЗАДАТЬ С ПОМОЩЬЮ 8 БИТОВ: ОТ Ø ДО 256,

КРАТКАЯ ЗАПИСЬ ТАКОЙ СИНТАКСИС ЗАПИСИ
ОБОЗНАЧАЕТ "R" ПРОИЗВОЛЬНЫЙ РЕГИСТР, А "N" - ЛЮБОЕ ЧИСЛО, ЗДЕСЬ ПО-ПРЕЖНЕМУ ДЕЙСТВУЕТ РАНЕЕ ВВЕДЕННОЕ УСЛОВНОЕ ОБОЗНАЧЕНИЕ 8 БИТОВ ОДНОЙ БУКВОЙ,

ПРИМЕР "ВНЕШНЕЙ АДРЕСАЦИИ":

LD A,(NN) ЗАГРУЗИТЬ СОДЕРЖИМОЕ ЯЧЕЙКИ NN В А,

ДОПУСТИМА И ОБРАТНАЯ КОМАНДА, ЭТО - ОДНА ИЗ ОБРАЩАЮЩИХ НА СЕБЯ ВНИМАНИЕ ОСОБЕННОСТЕЙ Z80 - В НАБОРЕ КОМАНД ИМЕЕТСЯ СИММЕТРИЯ,ТА

LD (NN),A

ОБЯЗАТЕЛЬНО ОБРАТИТЕ ВНИМАНИЕ, ЧТО ЭТИ КОМАНДЫ ДОПУСТИМЫ ТОЛЬКО ДЛЯ РЕГИСТРА "А",

ДАВАЙТЕ И ЗАДЕРЖИМСЯ ЗДЕСЬ НА НАНОСЕКУНДУ И ПОСМОТРИМ, ЧТО ЭТИ ДВЕ КОМАНДЫ НА САМОМ ДЕЛЕ НАМ ДАЮТ,

ВО-ПЕРВЫХ ,ДИАПОЗОН ЗНАЧЕНИЙ ЧИСЕЛ, ЗАДАВАЕМЫХ ЧИСЛОМ ДЛЯ 2 РЕГИСТРОВ (NN), БУДЕТ ОТ 0 ДО 65 535, ЭТО 64К, ТО ЕСТЬ С ПОМОЩЬЮ ЭТОЙ КОМАНДЫ МОЖНО ПОЛУЧИТЬ ДОСТУП НЕ БОЛЕЕ ЧЕМ 64К ПАМЯТИ! ЭТО ЗНАЧИТ, ЧТО ВСЯ ПАМЯТЬ: ПЗУ, ПРОГРАММА, ДИСПЛЕЙ СВОБОДНАЯ ПАМЯТЬ - ДОЛЖНА УМЕСТИТЬСЯ В 64 К, В "16К СПЕКТРУМ" НА САМОМ ДЕЛЕ 16К ИСПОЛЬЗУЮТСЯ ПЗУ И 16К - ПАМЯТЬЮ С ПРОИЗВОЛЬНЫМ ДОСТУПОМ, ЧТО ВМЕСТЕ СОСТАВЛЯЕТ 32К, 16К ОТНОСИТСЯ ТОЛЬКО К ПАМЯТИ С ПРОИЗВОЛЬНЫМ ДОСТУПОМ, В "48К СПЕКТРУМ" ИМЕЮТСЯ ТЕЖЕ САМЫЕ 16К ПЗУ ПЛЮС 48К ПАМЯТИ С ПРОИЗВОЛЬНЫМ ДОСТУПОМ, ЧТО ВМЕСТЕ СОСТАВЛЯЕТ 64К, ПОЭТОМУ ДЛЯ Z80 НЕВОЗМОЖНО ПОЛУЧИТЬ ДОСТУП К БОЛЬШОМУ КОЛИЧЕСТВУ ПАМЯТИ, ЧЕМ "48К СПЕКТРУМ",

КОМАНДА LD A,(NN), ЧИТАЕМАЯ КАК "ЗАГРУЗИТЬ СОДЕРЖИМОЕ ЯЧЕЙКИ NN В А", - ОЧЕНЬ БОГАТА ВОЗМОЖНОСТЯМИ, ОНА ПОЗВОЛЯЕТ НАМ ЧИТАТЬ СОДЕРЖИМОЕ ЛЮБОЙ ЯЧЕЙКИ ПАМЯТИ, БУДЬ ТО ПЗУ, ИЛИ ПАМЯТЬ С ПРОИЗВОЛЬНЫМ ДОСТУПОМ,

ВЫ МОЖЕТЕ ИСПОЛЬЗОВАТЬ ЭТУ КОМАНДУ ДЛЯ ИССЛЕДОВАНИЙ ПО ВАШЕМУ ВКУСУ, ДАЖЕ ДЛЯ ТЕХ ЯЧЕЕК, ГДЕ ПАМЯТЬ ОТСУТСТВУЕТ, НАПРИМЕР, ПОПЫТАТЬСЯ ПОСМОТРЕТЬ ,ЧТО НАХОДИТСЯ ЗА ПРЕДЕЛАМИ 32К ПАМЯТИ, ДАЖЕ ЕСЛИ У ВАС НЕТ ДОПОЛНИТЕЛЬНОЙ ПАМЯТИ, ВАС ЖДЕТ СЮРПРИЗ -- ТАМ ВОВСЕ НЕ ВЕЗДЕ НУЛИ!

ОБРАТНАЯ КОМАНДА "LD (NN),A", КОТОРАЯ ЧИТАЕТСЯ КАК - "ЗАГРУЗИТЬ А В ЯЧЕЙКУ ПАМЯТИ NN", БУДЕТ ПЫТАТЬСЯ ВЕСТИ ЗАПИСЬ ТАКЖЕ В ЛЮБУЮ ЯЧЕКУ ПАМЯТИ, НО ДЛЯ НЕЕ СКАЗЫВАЮТСЯ ФИЗИЧЕСКИЕ ОГРАНИЧЕНИЯ,

ВЫ НЕ МОЖЕТЕ ВЕСТИ ЗАПИСЬ В ЯЧЕЙКИ , КОТОРЫЕ НЕ МОГУТ СОДЕРЖАТЬ ЭТУ ИНФОРМАЦИЮ, ПОДОБНО НЕСУЩЕСТВУЮЩЕЙ ПАМЯТИ ЗА ПРЕДЕЛАМИ РАЗМЕРА ВАШЕЙ СИСТЕМЫ,

ОДНО ИЗ СВОЙСТВЕННЫХ ЭТОЙ КОМАНДЕ ОГРАНИЧЕНИЙ СОСТОИТ В ТОМ, ЧТО ВАМ НЕОБХОДИМО ВО ВРЕМЯ НАПИСАНИЯ ПРОГРАММЫ ЗНАТЬ, КАКУЮ ЯЧЕЙКУ ПАМЯТИ МЫ ХОТИМ ПРОВЕРЯТЬ ИЛИ ЗАПИСАТЬ В НЕЕ,

СОКРАЩЕНИЕ "NN" ОЗНАЧАЕТ ОПРЕДЕЛЕННОЕ ЧИСЛО, НАПРИМЕР, 17 100, А НЕ ПЕРЕМЕННУЮ,

ВЫ НЕ МОЖЕТЕ ИСПОЛЬЗОВАТЬ ЭТУ КОМАНДУ В ЭКВИВАЛЕНТЕ ЦИКЛУ "FOR - NEXT" НА МАШИННОМ ЯЗЫКЕ, ПОЭТОМУ ОСНОВНОЕ ПРИМЕНЕНИЕ ЭТОЙ КОМАНДЫ - ОТВЕДЕНИЕ КОНКРЕТНЫХ ЯЧЕЕК ПАМЯТИ ПОД ХРАНЕНИЕ ПЕРЕМЕННЫХ;

А ЧТО ЕСЛИ МЫ НЕ ЗНАЕМ ТОЧНЫЙ АДРЕС ЯЧЕЙКИ ПАМЯТИ, ГДЕ МЫ ДОЛЖНЫ ИСКАТЬ ИФОРМАЦИЮ? ПРЕДПОЛОЖИМ, МЫ МОЖЕМ ТОЛЬКО ПОСЧИТАТЬ ,ГДЕ ЭТА ИНФОРМАЦИЯ ДОЛЖНА НАХОДИТЬСЯ ? ПОСКОЛЬКУ НАМ НЕОБХОДИМО ,16 БИТОВ, ЧТОБЫ ЗАДАТЬ АДРЕС ПРОИЗВОЛЬНОЙ ЯЧЕЙКИ ПАМЯТИ, НАМ ПРИДЕТСЯ ХРАНИТЬ ЕГО В 16-БИТОВОМ РЕГИСТРЕ ,Т,Е , В ОДНОЙ ИЗ ПАР РЕГИСТРОВ BC, DE ИЛИ HL, ИЛИ В ОДНОМ ИЗ ИНДЕКСНЫХ РЕГИСТРОВ IX ИЛИ IY,

ОДИН ИЗ СПОСОБОВ ДЛЯ ЭТОГО - СДЕЛАТЬ ТАК, ЧТОБЫ В ОДНОЙ ИЗ ПАР РЕГИСТРОВ СОДЕРЖАЛСЯ АДРЕС ЯЧЕЙКИ ПАМЯТИ, ПОСКОЛЬКУ ИНФОРМАЦИЯ СОДЕРЖИТСЯ В РЕГИСТРЕ И НАМ НЕ ДАН АДРЕС НЕПОСРЕДСТВЕННО, МЫ НАЗЫВАЕМ ЭТУ ФОРМУ АДРЕСАЦИИ КОСВЕННОЙ РЕГИСТРОВОЙ АДРЕСАЦИЕЙ,

МНЕМОНИЧЕСКОЕ СОКРАЩЕНИЕ ДЛЯ ТАКИХ КОМАНД ИМЕЕТ ВИД

LD R,(HL)

LD A,(BC)

LD A,(DE) ПО-РУССКИ ЭТИ КОМАНДЫ ЧИТАЮТСЯ ТАК:

"ЗАГРУЗИТЬ В РЕГИСТР СОДЕРЖИМОЕ ЯЧЕЙКИ ПАМЯТИ, НА КОТОРУЮ УКАЗЫВАЕТ HL"
"ЗАГРУЗИТЬ В А СОДЕРЖИМОЕ ЯЧЕЙКИ ПАМЯТИ, НА КОТОРУЮ УКАЗЫВАЕТ BC"
"ЗАГРУЗИТЬ В А СОДЕРЖИМОЕ ЯЧЕЙКИ ПАМЯТИ, НА КОТОРУЮ УКАЗЫВАЕТ DE"

ОБРАТИТЕ ВНИМАНИЕ, ЧТО, ИСПОЛЬЗУЯ "HL" В КАЧЕСТВЕ УКАЗАТЕЛЯ НА НАШУ ЯЧЕЙКУ ПАМЯТИ, МЫ МОЖЕМ ПРОВЕСТИ ЗАГРУЗКУ В ЛЮБОЙ РЕГИСТР, ДАЖЕ В H ИЛИ L, ХОТЯ ЭТО И ЗВУЧИТ СТРАННО, НО ЧТО, ПРИМЕНЯЯ BC ИЛИ DE, МЫ МОЖЕМ ЗАГРУЗИТЬ ТОЛЬКО В РЕГИСТР A, ТАК ПРОИСХОДИТ ПОТОМУ, ЧТО ПАРА РЕГИСТРОВ HL ПРИВИЛЕГИРОВАНА ТОЧНО ТАК ЖЕ, КАК ПРИВИЛЕГИРОВАН РЕГИСТР A,

И ЗДЕСЬ ТАКЖЕ ИМЕЕТСЯ СИММЕТРИЯ ПО ОТНОШЕНИЮ К ЭТИМ КОМАНДАМ, И МЫ МОЖЕМ АНАЛОГИЧНЫМ ОБРАЗОМ ЗАГРУЖАТЬ ИНФОРМАЦИЮ В ЯЧЕЙКИ ПАМЯТИ:

```
LD (HL),R
LD (BC),A
```

LD (DE),A ЭТО ТАКЖЕ НАЗЫВАЕТСЯ "КОСВЕННОЙ РЕГИСТРОВОЙ АДРЕСАЦИЕЙ" ВНЕ ЗАВИСИМОСТИ ОТ ТОГО, В КАКОМ НАПРАВЛЕНИИ ПЕРЕЩАЕТСЯ ИНФОРМАЦИЯ,

С ДРУГОЙ СТОРОНЫ, МЫ МОГЛИ БЫ ПРИМЕНИТЬ ИНДЕКСНЫЕ РЕГИСТРЫ IX И IY ДЛЯ УКАЗАНИЯ ЯЧЕЙКИ ПАМЯТИ, КРАТКАЯ ЗАПИСЬ ЭТИХ КОМАНД ИМЕЕТ ВИД:

```
LD R,(IX + Д)
```

LD R,(IY + Д) ГДЕ "R" - ОПЯТЬ ПРОИЗВОЛЬНЫЙ РЕГИСТР, А "Д" - "СМЕЩЕНИЕ" ОТНОСИТЕЛЬНО АДРЕСА, НА КОТОРЫЙ УКАЗЫВАЕТ IX ИЛИ IY, (НЕ СПУТАЙТЕ ОБОЗНАЧЕНИЯ "Д" - МЫ ИМЕЕМ ВВИДУ НЕ РЕГИСТР "D", А "Д" = СМЕЩЕНИЕ (DISPLACEMENT)),

ЧИСЛО "Д" - 8-БИТОВОЕ ЧИСЛО, КОТОРОЕ НУЖНО УКАЗЫВАТЬ ВО ВРЕМЯ ПРОГРАММИРОВАНИЯ И НЕЛЬЗЯ СДЕЛАТЬ ПЕРЕМЕННЫМ, В ЭТОМ - СЛАБОСТЬ ДАННОЙ КОНКРЕТНОЙ КОМАНДЫ, И ЭТО ОЗНАЧАЕТ, ЧТО ЕЕ ПРИМЕНЕНИЕ ОБЫЧНО ОГРАНИЧИВАЕТСЯ ЧТЕНИЕМ И ЗАПИСЬЮ ТАБЛИЦ, СОДЕРЖАЩИХ ДАННЫЕ,

ИМЕЕТСЯ ТАКЖЕ СИММЕТРИЧНАЯ КОМАНДА:

```
LD (IX + Д),R
LD (IY + Д),R
```

ЕСЛИ ЭТОТ КОНКРЕТНЫЙ СПОСОБ АДРЕСАЦИИ ВЫГЛЯДИТ НЕСКОЛЬКО УСЛОЖНЕННЫМ, НЕ БЕСПОКОИТЕСЬ, ВАМ ОН ВРЯД ЛИ ПОНАДОБИТСЯ В НЕСКОЛЬКИХ ПЕРВЫХ ВАШИХ ПРОГРАММАХ,

ЧИП Z80 В ЭВМ "СИНКЛЕР" В ВЫСШЕЙ СТЕПЕНИ УНИВЕРСАЛЕН, И ВЫ МОЖЕТЕ КОМБИНИРОВАТЬ РАЗНЫЕ СПОСОБЫ ЗАГРУЗКИ ЧИСЕЛ, ОПИСАННЫХ НАМИ ВЫШЕ,

НАПРИМЕР, ВЫ МОЖЕТЕ КОМБИНИРОВАТЬ НЕПОСРЕДСТВЕННУЮ АДРЕСАЦИЮ (Т,Е, ЗАДАНИЕ ЧИСЛА, КОТОРОЕ ВЫ ХОТИТЕ ЗАГРУЗИТЬ) С ВНЕШНЕЙ АДРЕСАЦЕЙ (Т,Е, ЗАДАНИЕ АДРЕСА ЗАГРУЗКИ С ПОМОЩЬЮ ПАРЫ РЕГИСТРОВ),

ЭТО НАЗЫВАЕТСЯ "НЕПОСРЕДСТВЕННОЙ ВНЕШНЕЙ АДРЕСАЦИЕЙ",

К СОЖАЛЕНИЮ ВЫ МОЖЕТЕ ИСПОЛЬЗОВАТЬ ТОЛЬКО ПАРУ РЕГИСТРОВ HL И ПОЭТОМУ СОКРАЩЕННАЯ ЗАПИСЬ ИМЕЕТ ВИД:

```
LD (HL),N
```

ЭТО УДОБНО, ПОСКОЛЬКУ ВЫ МОЖЕТЕ НЕПОСРЕДСТВЕННО ЗАПОЛНИТЬ ЯЧЕЙКУ ПАМЯТИ БЕЗ НЕОБХОДИМОСТИ СНАЧАЛА ЗАГРУЗИТЬ ЭТО ЗНАЧЕНИЕ В РЕГИСТР,

ВОЗМОЖНА АНАЛОГИЧНАЯ КОМБИНАЦИЯ С ИНДЕКСНЫМИ РЕГИСТРАМИ, ОНА НАЗЫВАЕТСЯ "НЕПОСРЕДСТВЕННАЯ ИНДЕКСНАЯ АДРЕСАЦИЯ", ЭТА КОМАНДА ИСПОЛЬЗУЕТСЯ БОЛЕЕ ОГРАНИЧЕННО, И СОКРАЩЕННАЯ ФОРМА ТАКИХ КОМАНД ИМЕЕТ ВИД:

```
LD (IX+ Д),N
LD (IY+ Д),N
```

ПРИМЕНЕНИЕ ЭТИХ КОМАНД В ПРОГРАММЕ НА МАШИННОМ ЯЗЫКЕ

ДАВАЙТЕ ПОПРОБУЕМ НА ПРАКТИКЕ ПРИМЕНИТЬ НЕКОТОРЫЕ ИЗ ЭТИХ КОМАН "LD",

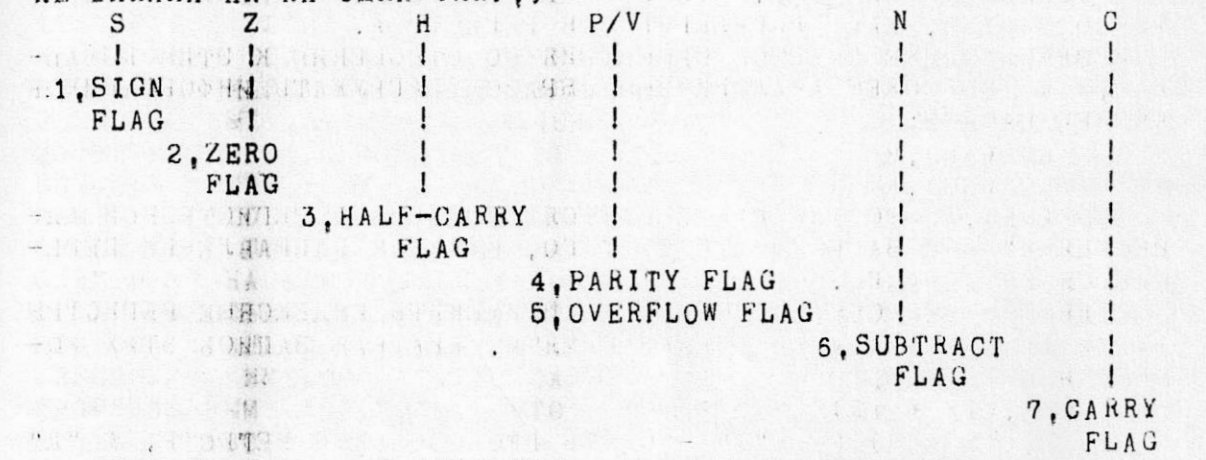
Ф Л А Г И И И Х П Р И М Е Н Е Н И Е

В МАШИННОМ ЯЗЫКЕ "ФЛАГ" ОЗНАЧАЕТ "ОТМЕТКА", ФЛАГ - ЭТО ТО, ЧТО ВЫ ПОДНИМАЕТЕ, ЕСЛИ ХОТИТЕ ПОМЕТИТЬ ДЛЯ КОГО-ТО ДРУГОГО, ЧТО ВЫПОЛНЕНО ОПРЕДЕЛЕННОЕ УСЛОВИЕ,

ПРИЧИНА ПО КОТОРОЙ КОНСТРУКТОРЫ Z80 (КАК И БОЛЬШИНСТВО

КОНСТРУКТОРЫ ЦП) ИСПОЛЬЗУЮТ ФЛАГИ В СВОИХ ИНТЕРЕСАХ, В ТОМ, ЧТОБЫ ДАТЬ ПРОГРАММИСТУ ИНФОРМАЦИЮ О ЧИСЛЕ ,НАХОДЯЩЕМСЯ В РЕГИСТРЕ "А", ИЛИ ИН ФОРМАЦИЮ О ТОЛЬКО ЧТО ВЫПОЛНЕННОМ ПОСЛЕДНЕМ ВЫЧИСЛЕНИИ,

НАПОМНИМ,ЧТО ОДИН ИЗ РЕГИСТРОВ ЦП ОТВЕДЕН ПОД ФЛАГИ – ЭТО РЕГИСТР "F", ВЫ МОГЛИ ТАКЖЕ ОБРАТИТЬ ВНИМАНИЕ НА ТО, ЧТО В НАЧАЛЕ ПОСЛЕДНЕЙ ГЛАВЫ В ТАБЛИЦЕ СВЕДЕНЫ РАЗЛИЧНЫЕ КОМАНДЫ, КОТОРЫЕ ДОЛЖНЫ БЫЛИ В ЭТОЙ ГЛАВЕ РАССМАТРИВАТЬСЯ, И ЧАСТЬ ТАБЛИЦЫ БЫЛА ОТВЕДЕНА НА ОПИСАНИЕ ВОЗДЕЙСТВИЯ КАЖДОЙ КОМАНДЫ НА ФЛАГИ, (К СЧАСТЬЮ, НИ ОДНА ИЗ КОМАНД ЭТОЙ ПОСЛЕДНЕЙ ГЛАВЫ НЕ ВЛИЯЛА НИ НА ОДИН ФЛАГ,)



1–ФЛАГ ЗНАКА; 2–ФЛАГ НУЛЯ; 3–ФЛАГ ПОЛОВИННОГО ПЕРЕНОСА
4–ФЛАГ ЧЕТНОСТИ; 5–ФЛАГ ПЕРЕПОЛНЕНИЯ; 6–ФЛАГ ВЫЧИТАНИЯ;
7–ФЛАГ ПЕРЕНОСА.

НА САМОМ ДЕЛЕ РАЗРАБОТЧИКИ ПРИДУМАЛИ СЕМЬ ФЛАГОВ, НО РЕШИЛИ, ЧТО ОДНИМ РЕГИСТРОМ (ТАК В ОРИГИНАЛЕ, ОЧЕВИДНО, ИМЕЕТСЯ В ВИДУ БИТ,8ПРИМЕЧ. ПЕР,))МОЖНО РЕАЛИЗОВАТЬ ДВА ФЛАГА: И ЧЕТНОСТИ И ПЕРЕПОЛНЕНИЯ,

ДАВАЙТЕ ТЕПЕРЬ ПОДРОБНЕЕ РАССМОТРИМ КАЖДЫЙ ИЗ ЭТИХ ФЛАГОВ,

ФЛАГ НУЛЯ

ЭТОТ ФЛАГ МЫ УЖЕ РАССМАТРИВАЛИ ВЫШЕ, ЕГО ПРИКЛАДНОЕ ЗНАЧЕНИЕ ОЧЕВИДНО, ОН ОБЫЧНО УСТАНАВЛИВАЕТСЯ ПОСЛЕ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ, ПОСКОЛЬКУ СЛУЖИТ ДЛЯ ОПИСАНИЯ СОДЕРЖИМОГО РЕГИСТРА "А",

ОБРАТИТЕ, ОДНАКО, ОСОБОЕ ВНИМАНИЕ НА ТО, ЧТО В РЕГИСТРЕ "А" МОЖЕТ СОДЕРЖАТЬСЯ Ø, А ФЛАГ НУЛЯ УСТАНОВЛЕН НЕ БУДЕТ, ТАК ЛЕГКО МОЖЕТ СЛУЧИТЬСЯ ПРИ ВЫПОЛНЕНИИ КОМАНДЫ

```
LD A,Ø
```

ФЛАГ НУЛЯ ТАКЖЕ УСТАНАВЛИВАЕТСЯ, ЕСЛИ НУЛЕВЫМ ОКАЗЫВАЕТСЯ РЕЗУЛЬТАТ КОМАНД ИЗ ГРУППЫ "ПЕРЕМЕЩЕНИЙ В КРУГОВОМ ПОРЯДКЕ И СДВИГОВ",

КРОМЕ ТОГО, ФЛАГ НУЛЯ ОКАЗЫВАЕТСЯ ЕДИНСТВЕННЫМ ЗРИМЫМ РЕЗУЛЬТАТОМ НЕКОТОРЫХ КОМАНД ПРОВЕРКИ, ТАКИХ КАК ГРУППА КОМАНД "ПРОВЕРКА БИТА", В ЭТИХ СЛУЧАЯХ ФЛАГ НУЛЯ УСТАНАВЛИВАЕТСЯ, ЕСЛИ ПРОВЕРЯЕМЫЙ БИТ РАВЕН НУЛЮ,

ФЛАГ ЗНАКА

ФЛАГ ЗНАКА ОЧЕНЬ НАПОМИНАЕТ ФЛАГ НУЛЯ И ДЕЙСТВУЕТ ПО ПРЕИМУЩЕСТВУ НА ТОМ ЖЕ САМОМ НАБОРЕ КОМАНД (РАСХОЖДЕНИЕ В ОСНОВНОМ КАСАЕТСЯ ГРУППЫ "ПРОВЕРИ БИТА", ГДЕ ПОНЯТИЕ ОТРИЦАТЕЛЬНОГО БИТА ПРИ ЛЮБОЙ ИНТЕРПРЕТАЦИИ ЛИШЕНО СМЫСЛА),

ФЛАГ ПЕРЕНОСА

ЭТО – ОДИН ИЗ НАИБОЛЕЕ ВАЖНЫХ ФЛАГОВ, ИМЮЩИХСЯ В ЯЗЫКЕ АССЕМБЛЕРА, ПОСКОЛЬКУ БЕЗ НЕГО РЕЗУЛЬТАТЫ АРИФМЕТИЧЕСКИХ ДЕЙСТВИЙ В АССЕМБЛЕРЕ БЫЛИ БЫ СОВЕРШЕННО БЕССМЫСЛЕНЫМИ,

ЧТО НУЖНО ПОМНИТЬ, ЭТО ТО, ЧТО КОМАНДЫ ЯЗЫКА АССЕМБЛЕРА ВСЕГДА ОТНОСЯТСЯ ЛИБО К 8–БИТОВЫМ ЧИСЛАМ ИЛИ К 16–БИТОВЫМ ЧИСЛАМ,

ЭТО ОЗНАЧАЕТ, ЧТО ЧИСЛА, С КОТОРЫМИ МЫ ИМЕЕМ ДЕЛО, МОГУТ БЫТЬ ЛИБО

```
 8-BIT **)   Ø-255
16-BIT **) Ø-65535 ЛИБО, ЕСЛИ ВЫ ВКЛЮЧАЕТЕ ПЕРЕНОС,
 8-BIT **)   Ø-256
16-BIT **) Ø-65536 РАССМОТРИМ СИТУАЦИЮ,В КОТОРОЙ МЫ ВЫПОЛ-
```

"... SPECTRUM" NOT COPY

НЯЕМ СЛЕДУЮЩЕЕ ВЫЧИТАНИЕ

```
   2 0 0
 -
   2 0 1
   -----
```

РЕЗУЛЬТАТ = 2 5 5 !!!

ЭТО - ПРЯМОЕ СЛЕДСТВИЕ ТОГО, ЧТО У НАС ИМЕЕТСЯ ТОЛЬКО ОГРАНИЧЕННЫЙ ДИАПОЗОН ЧИСЕЛ, И ТО ЖЕ САМОЕ МОЖЕТ ПРОИЗОИТИ, ОЧЕВИДНО, И С 16-БИТОВЫМИ ЧИСЛАМИ,

МЫ ВИДЕЛИ, ЧТО ФЛАГ ПЕРЕНОСА МОЖЕТ БЫТЬ УСТАНОВЛЕН В РЕЗУЛЬТАТЕ ВЫЧИТАНИЯ ПРИ НАЛИЧИИ "УХОДА ЗА НОЛЬ", ФЛАГ ПЕРЕНОСА МОЖЕТ БЫТЬ ТАКЖЕ УСТАНОВЛЕН В РЕЗУЛЬТАТЕ ОПЕРАЦИИ СЛОЖЕНИЯ ЕСЛИ ДОЛЖНО БЫТЬ "ПЕРЕПОЛНЕНИЕ",

ПОЭТОМУ УДОБНО ПРЕДСТАВЛЯТЬ СЕБЕ БИТ ПЕРЕНОСА КАК 9-Й БИТ РЕГИСТРА "A":

```
  NUMBER     CARRY BIT    NUIBER IN BIT FORM
   132          -          1 0 0 0  0 1 0 0
 + 135          -          1 0 0 0  0 1 1 1
   ----                    ---------------
```

267 1 0 0 0 0 1 0 1 1 NUMBER - ЧИСЛО; CARRY BIT - БИТ ПЕРЕНОСА; NUMBER IN BIT FROM ЧИСЛО В ВИДЕ БИТОВ

ОДНАКО ПОСКОЛЬКУ У НАС НЕТ 9 БИТОВ, А РЕГИСТРЕ "A" БУДЕТ СОДЕРЖАТЬСЯ ЧИСЛО 0BH (ДЕСЯТИЧНОЕ - 11) И ПЕРЕНОС БУДЕТ УСТАНОВЛЕН (T,E, РАВЕН 1),

МОЖНО ЗАМЕТИТЬ, ЧТО ПРИ ВЫЧИТАНИИ ЗАНИМАНИЕ ИЗ 9-ГО БИТА ТАКЖЕ ПРИВЕДЕТ К ПОЯВЛЕНИЮ В НЕМ "1" ,

ИСПОЛЬЗОВАНИЕ ФЛАГОВ В КОНСТРУКЦИИ НА МАШИННОМ ЯЗЫКЕ, ЭКВИВАЛЕНТНОИ ПРЕДЛОЖЕНИЮ "IF , , , THEN, , , ,"

В ЯЗЫКЕ "БЕЙСИК" У НАС ЕСТЬ ВОЗМОЖНОСТЬ КОНСТРУИРОВАТЬ СИТУАЦИЮ ТИПА "IF ,,, THEN", ТАКИЕ КАК

```
    IF A=0 THEN ,,,,
1, WHERE WHAT FOOLLOWS CAN BE "LET ,,,,"
                       2  OR "GOTO,,,,"
                       2  OR "GOSUB,,,"
```

1 - ГДЕ-ТО, ЧТО СЛЕДУЕТ, МОЖЕТ БЫТЬ; 2 - ИЛИ ПРИНЯТИЯ РЕШЕНИЯ ТОЧНО ТАКОГО ЖЕ ТИПА МОЖНО ЗАПРОГРАММИРОВАТЬ НА МАШИННОМ ЯЗЫКЕ (ЗА ИСКЛЮЧЕНИЕМ ПРЕДЛОЖЕНИЯ "LET ,,,") ВМЕСТО ТОГО ЧТОБЫ НАПИСАТЬ "IF A=0", МЫ ПРОСТО ПРОВЕРЯЕМ ФЛАГ НУЛЯ: ЕСЛИ ОН УСТАНОВЛЕН, ТО МЫ ЗНАЕМ, ЧТО A=0,

ТРИ РАССМОТРЕННЫХ НАМИ К НАСТОЯЩЕМУ МОМЕНТУ ФЛАГА - ЭТО В ОСНОВНОМ ВСЕ, ЧТО НАМ НУЖНО, ЧТОБЫ ВЫБРАТЬ СЛЕДУЮЩУЮ КОМАНДУ ДЛЯ ВЫПОЛНЕНИЯ,

ПОДОБНАЯ КОМАНДА ИМЕЕТ СЛЕДУЮЩИЙ ФОРМАТ: НАПРИМЕР; JP CC, END , ГДЕ "JP" - МНЕМОНИЧЕСКОЕ ОБОЗНАЧЕНИЕ КОМАНДЫ ПЕРЕХОДА (JUMP), А "END" - УДОБНАЯ МЕТКА,

ЭТА КОМАНДА ПО-РУССКИ ЧИТАЕТСЯ КАК "ПЕРЕЙТИ ПРИ УСЛОВИИ CC НА END"

УСЛОВИЕ "CC" МОЖЕТ БЫТЬ ЛЮБЫМ ИЗ СЛЕДУЮЩИХ:

```
    Z    (= НОЛЬ)
   NZ    (= НЕ НОЛЬ)
    P    (= ПОЛОЖИТЕЛЬНОЕ)
    M    (= МИНУС)
    C    (= ПЕРЕНОС УСТАНОВЛЕН)
   NC    (= НЕТ ПЕРЕНОСА
```

ФЛАГ ЧЕТНОСТИ/ПЕРЕПОЛНЕНИЯ

ЭТОТ ФЛАГ ДЛЯ НЕКОТОРЫХ КОМАНД ДЕИСТВУЕТ КАК ФЛАГ ЧЕТНОСТИ, А ДЛЯ ДРУГИХ - КАК ФЛАГ ПЕРЕПОЛНЕНИЯ, ОДНАКО ПУТАНИЦА ВОЗНИКАЕТ РЕДКО, ПОСКОЛЬКУ ЭТИ ДВА ТИПА КОМАНД ОБЫЧНО НЕ ВСТРЕЧАЮТСЯ ВМЕСТЕ,

РОЛЬ ФЛАГА КАК УКАЗАТЕЛЯ ЧЕТНОСТИ ВЫСТУПАЕТ В ЛОГИЧЕСКИХ ОПЕРАЦИЯХ, И ОН УСТАНАВЛИВАЕТСЯ В ТОМ СЛУЧАЕ, ЕСЛИ В РЕЗУЛЬТАТЕ УСТАНОВЛЕННЫМ ОКАЗЫВАЕТСЯ ЧЕТНОЕ КОЛИЧЕСТВО БИТОВ , БОЛЕЕ ПОДРОБНО ЭТОТ ВОПРОС МЫ РАССМАТРИВАЕМ В ГЛАВЕ ЛОГИЧЕСКИХ ОПЕРАЦИЯХ,

ПЕРЕПОЛНЕНИЕ ГОВОРИТ ВАМ О ТОМ, ЧТО ТОЛЬКО ЧТО ВЫПОЛНЕННАЯ ВАМИ АРИФМЕТИЧЕСКАЯ ОПЕРАЦИЯ МОЖЕТ ДАТЬ РЕЗУЛЬТАТ, НЕ УМЕЩАЮЩИИСЯ В 8 БИТОВ , ВМЕСТО ТОГО, ЧТОБЫ НА САМОМ ДЕЛЕ ОЗНА

ЧАТЬ, ЧТО ДЛЯ ЗАПИСИ РЕЗУЛЬТАТА ПОНАДОБИЛСЯ 9-Й БИТ, ЭТОТ ФЛАГ ГОВОРИТ О ТОМ, ЧТО В РЕЗУЛЬТАТЕ ОПЕРАЦИИ 8-Й БИТ ИЗМЕНИЛСЯ!

В ПРИВЕДЕННОМ ВЫШЕ ПРИМЕРЕ ДО СЛОЖЕНИЯ 132 И 135 8-Й БИТ БЫЛ РАВЕН "1", А ПОСЛЕ — "Ø", ТАК ЧТО ДОЛЖЕН БЫЛ БЫТЬ УСТАНОВЛЕН ФЛАГ ПЕРЕПОЛНЕНИЯ, НО ФЛАГ ПЕРЕПОЛНЕНИЯ БУДЕТ ТАКЖЕ УСТАНОВЛЕН И В РЕЗУЛЬТАТЕ ТАКОГО СЛОЖЕНИЯ,

```
  64          Ø 1 Ø Ø  Ø Ø Ø Ø
+ 65          Ø 1 Ø Ø  Ø Ø Ø 1
-----         ----------------
 129          1 Ø Ø Ø  Ø Ø Ø 1
```

ФЛАГ ВЫЧИТАНИЯ

ЭТОТ ФЛАГ УСТАНАВЛИВАЕТСЯ ,ЕСЛИ ПОСЛЕДНЕЙ ОПЕРАЦИЕЙ БЫЛО ВЫЧИТАНИЕ!

ФЛАГ ПОЛОВИННОГО ПЕРЕНОСА

ЭТОТ ФЛАГ УСТАНАВЛИВАЕТСЯ СПОСОБОМ, АНАЛОГИЧНЫМ ФЛАГУ ПЕРЕНОСА, НО ТОЛЬКО В ТОМ СЛУЧАЕ, КОГДА ИМЕЕТСЯ ПЕРЕПОЛНЕНИЕ ИЛИ ЗАНИМАНИЕ ИЗ 5-ГО БИТА, А НЕ ИЗ 9-ГО!

КАК ФЛАГ ВЫЧИТАНИЯ, ТАК И ФЛАГ ПОЛОВИННОГО ПЕРЕНОСА ПРИМЕНЯЮТСЯ ТОЛЬКО ДЛЯ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ НАД "ДВОИЧНО КОДИРОВАННЫМИ ДЕСЯТИЧНЫМИ ЧИСЛАМИ", МЫ РАССМАТРИВАЕМ ЭТИ ФЛАГИ В ГЛАВЕ "АРИФМЕТИЧЕСКИЕ ДЕЙСТВИЯ НАД ДВОИЧНО-КОДИРОВАННЫМИ ДЕСЯТИЧНЫМИ ЧИСЛАМИ",

У В Е Л И Ч Е И Н Е И У М Е Н Ь Ш Е Н И Е Ч И С Е Л

УВЕЛИЧЕНИЕ

INC R "INC" ПО-АНГЛИЙСКИ ЧИТАЕТСЯ КАК "INCREASE" (УВЕЛИЧИТЬ) И ГОВОРИТ САМО ЗА СЕБЯ,

УВЕЛИЧЕНИЕ ЧИСЛА НА ПАРАХ РЕГИСТРОВ ЗАПИСЫВАЮТСЯ ТАК:

```
INC RR
INC IX
```

INC IY ГДЕ "RR" ОБОЗНАЧАЕТ ПАРУ РЕГИСТРОВ, ТАКУЮ КАК "BC", "DE", "HL"

ОБРАТИТЕ ЕЩЕ РАЗ ВНИМАНИЕ НА ПРОСТОЙ СПОСОБ ОБОЗНАЧЕНИЯ ТОГО, КАКИЕ ОПЕРАЦИИ ИСПОЛЬЗУЮТ 8-БИТОВЫЕ ЧИСЛА, А КАКИЕ 16-БИТОВЫЕ

8-БИТОВЫЕ ЧИСЛА ОБОЗНАЧАЮТСЯ ОДНОЙ БУКВОЙ, А 16-БИТОВЫЕ — ДВУМЯ,

ОДНАКО КОМАНДА "УВЕЛИЧЕНИЯ" НА САМОМ ДЕЛЕ ИМЕЕТ ЕЩЕ БЛЬШИЕ ВОЗМОЖНОСТИ, ЧЕМ ВИДНО ИЗ ПРИВЕДЕННЫХ ПРИМЕРОВ, МОЖНО УВЕЛИЧИТЬ СОДЕРЖИМОЕ ЛЮБОЙ ЯЧЕЙКИ ПАМЯТИ, ЕСЛИ МЫ СМОЖЕМ ЗАДАТЬ ЕЕ АДРЕС С ПОМОЩЬЮ ИНДЕКСНЫХ РЕГИСТРОВ ИЛИ ПРИВИЛЕГИРОВАННОЙ ПАРЫ РЕГИСТРОВ, HL:

```
INC (IX + Д)
INC (IY + Д)
```

INC (HL) ГДЕ "Д " — СМЕЩЕНИЕ (DISPLACEMENT), А НЕ РЕГИСТР D!

УМЕНЬШЕНИЕ ЧИСЛА СИММЕТРИЧНОСТЬ НАБОРА КОМАНД Z8Ø ПОЧТИ НАВЕРНЯКА ДОЛЖНА ОБЕСПЕЧИВАТЬ, ЧТОБЫ ВСЕ, ЧТО ВЫ МОЖЕТЕ УМЕНЬШИТЬ, ВЫ МОГЛИ ТАКЖЕ И УВЕЛИЧИТЬ , И ИМЕННО ТАК ОБСТОИТ ДЕЛО

```
DEC R
DEC RR
DEC IX
DEC IY
DEC (HL)
DEC (IX+ Д)
DEC (IY+ Д)
```

МНЕМОНИЧЕСКОЕ СОКРАЩЕНИЕ "DEC" ПО-АНГЛИЙСКИ ЧИТАЕТСЯ КАК "DECREASE" (УМЕНЬШИТЬ,И ЗДЕСЬ НУЖНО ТОЖЕ БЫТЬ ВНИМАТЕЛЬНЫМ ПРИ ИСПОЛЬЗОВАНИИ СКОБОК,

ВЛИЯНИЕ НА ФЛАГИ

ПОСКОЛЬКУ КОМАНДЫ УМЕНЬШЕНИЯ И УВЕЛИЧЕНИЯ 8-БИТОВЫХ ЧИСЕЛ ВОЗДЕЙСТВУЮТ НА ВСЕ ФЛАГИ, ЗА ИСКЛЮЧЕНИЕМ ФЛАГА ПЕРЕНОСА, ИМЕННО ЗДЕСЬ УДОБНО ДАТЬ ОБЗОР РАБОТЫ С ФЛАГАМИ,

ВАЖНОЕ ЗАМЕЧАНИЕ: КОМАНДЫ УВЕЛИЧЕНИЯ И УМЕНЬШЕНИЯ ДЛЯ 16-БИТОВЫХ ЧИСЕЛ НЕ ВЛИЯЮТ НА КАКИЕ БЫ ТО НИ БЫЛО ФЛАГИ, ИЗМЕНЯЮТ ФЛАГИ ТОЛЬКО ОПЕРАЦИИ УВЕЛИЧЕНИЯ И УМЕНЬШЕНИЯ ДЛЯ 8-БИ-

ТОВЫХ ЧИСЕЛ,

ФЛАГ ЗНАКА: ЭТОТ ФЛАГ БУДЕТ УСТАНОВЛЕН В (=1) ,ЕСЛИ БИТ 7 8-БИТОВОГО РЕЗУЛЬТАТА РАВЕН 1,

ФЛАГ НУЛЯ: ЭТОТ ФЛАГ БУДЕТ УСТАНОВЛЕН (=1), ЕСЛИ 8-БИТОВЫЙ РЕЗУЛЬТАТ РАВЕН НУЛЮ,

ФЛАГ ПЕРЕПОЛНЕНИЯ: ЭТОТ ФЛАГ БУДЕТ УСТАНОВЛЕН (=1), ЕСЛИ СОДЕРЖИМОЕ БИТА 7 8-БИТОВОГО ЧИСЛА ИЗМЕНЯЕТСЯ В РЕЗУЛЬТАТЕ ОПЕРАЦИИ,

ФЛАГ ПОЛОВИННОГО ПЕРНОСА: ЭТОТ ФЛАГ БУДЕТ УСТАНОВЛЕН (=1), ЕСЛИ ИМЕЕТСЯ ПЕРЕНОС ИЛИ ЗАНИМАНИЕ ИЗ ЧЕТВЕРТОГО БИТА 8-БИТОВОГО ЧИСЛА,

ФЛАГ ОТРИЦАНИЯ: ЭТОТ ФЛАГ УСТАНАВЛИВАЕТСЯ, ЕСЛИ ПОСЛЕДНЕЙ ВЫПОЛНЕННОЙ КОМАНДОЙ БЫЛО ВЫЧИТАНИЕ, ТАК, ОН СБРАСЫВАЕТСЯ (=0)ДЛЯ "INC" И УСТАНАВЛИВАЕТСЯ (=1) ДЛЯ "DEC",

А Р И Ф М Е Т И Ч Е С К И Е О П Е Р А Ц И И Н А Д 8-РАЗРЯДНЫМИ ЧИСЛАМИ,

ПО ПОВОДУ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ ДЛЯ ОДНОГО РЕГИСТРА НУЖНО ТОЛЬКО ЗАМЕТИТЬ, ЧТО ВСЕ ОПЕРАЦИИ В ДАННОЙ ГЛАВЕ ОТНОСЯТСЯ ТОЛЬКО К 8-БИТОВЫМ ЧИСЛАМ И ВСЕ ОНИ ДОЛЖНЫ ВЫПОЛНЯТЬСЯ С ПОМОЩЬЮ ПРИВИЛЕГИРОВАННОГО РЕГИСТРА А,

ЭТОТ ФАКТ НАСТОЛЬКО ОРГАНИЧНО ВПЛЕТАЕТСЯ В МНЕМОНИКУ МАШИННОГО ЯЗЫКА Z80, ЧТО В НЕКОТОРЫХ МНЕМОНИЧЕСКИХ ОБОЗНАЧЕНИЯХ СОКРАЩЕНИЕ "A" ПРОСТО ОПУСКАЕТСЯ, НАПРИМЕР, ЕСЛИ НУЖНО ВЫЧЕСТЬ "B" ИЗ "A", ЕСТЕСТВЕННО БЫЛО БЫ ОЖИДАТЬ КОМАНДЫ, ПОДОБНОЙ

SUB A,B НА САМОМ ЖЕ ДЕЛЕ МНЕМОНИЧЕСКОЕ ОБОЗНАЧЕНИЕ ИМЕЕТ ВИД:

SUB B

КОМАНДЫ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ ДЛЯ 8-РАЗРЯДНЫХ ЧИСЕЛ

MNEMONIC	BYTES	TIME	EFFECT ON FLAGS					
		TAKEN	C	Z	PV	S	N	H
ADD A,REGISTER	1	4	#	#	#	#	Ø	#
ADD A,NUMBER	2	7	#	#	#	#	Ø	#
ADD A,(HL)	1	7	#	#	#	#	Ø	#
ADD A,(IX+ Д)	3	19	#	#	#	#	Ø	#
ADD A,(IY+ Д)	3	19	#	#	#	#	Ø	#
ADC A,REGISTR	1	4	#	#	#	#	Ø	#
ADC A,NUMBER	2	7	#	#	#	#	Ø	#
ADC A,(HL)	1	7	#	#	#	#	Ø	#
ADC A,(IX+ Д)	3	19	#	#	#	#	Ø	#
ADC A,(IY+ Д)	3	19	#	#	#	#	Ø	#
SUB A,REGISTR	1	4	#	#	#	#	1	#
SUB A,NUMBER	2	7	#	#	#	#	1	#
SUB A,(HL)	1	7	#	#	#	#	1	#
SUB A,(IX+ Д)	3	19	#	#	#	#	1	#
SUB A,(IY+ Д)	3	19	#	#	#	#	1	#
SBC A,REGISTER	1	4	#	#	#	#	1	#
SBC A,NUMBER	2	7	#	#	#	#	1	#
SBC A,(HL)	1	7	#	#	#	#	1	#
SBC A,(IX+ Д)	3	19	#	#	#	#	1	#
SBC A,(IY+ Д)	3	19	#	#	#	#	1	#
CP REGISTER	1	4	#	#	#	#	1	#
CP NUMBER	2	7	#	#	#	#	1	#
CP (HL)	1	7	#	#	#	#	1	#
CP (IX+ Д)	3	19	#	#	#	#	1	#
CP (IY+ Д)	3	19	#	#	#	#	1	#

MNEMONIC-МНЕМОНИЧЕСКОЕ ОБОЗНАЧЕНИЕ; BYTES-БАЙТЫ;TIME TAKEN-ВРЕМЯ ВЫПОЛНЕНИЯ; EFFECT ON FLAGS - ВОЗДЕЙСТВИЕ НА ФЛАГИ; REGISTER-РЕГИСТР; NUMBER - ЧИСЛО ,

ПОКАЗЫВАЕТ, ЧТО ФЛАГ ИЗМЕНЕН ОПЕРАЦИЕЙ;
Ø ПОКАЗЫВАЕТ, ЧТО ФЛАГ СБРАСЫВАЕТСЯ;
1 ПОКАЗЫВАЕТ, ЧТО ФЛАГ УСТАНАВЛИВАЕТСЯ;
- ПОКАЗЫВАЕТ, ЧТО ФЛАГ ОСТАЕТСЯ НЕИЗМЕННЫМ.

ВОПРЕКИ ЭТИМ ОГРАНИЧЕНИЯМ, НАКЛАДЫВАЕМЫМ НА АРИФМЕТИЧЕСКИЕ КОМАНДЫ (ОНИ ДОЛЖНЫ ОГРАНИЧИВАТЬСЯ РЕГИСТРОМ А), ЯЗЫК Z8Ø ОЧЕНЬ РАЗНОСТОРОНЕН В ПЛАНЕ ТОГО, ЧТО МЫ МОЖЕМ СЛОЖИТЬ С ПРОИЗВОЛЬНЫМ ЧИСЛОМ,ОТЛОЖЕННЫМ В РЕГИСТРЕ А,

```
ADD A,R          СЛОЖИТЬ С А ПРОИЗВОЛЬНЫЙ ОДИНАРНЫЙ РЕГИСТР;
ADD A,N          СЛОЖИТЬ С А ПРОИЗВОЛЬНОЕ 8-БИТОВОЕ ЧИСЛО;
ADD A,(HL)       ПРИБАВИТЬ 8-БИТОВОЕ ЧИСЛО,
                 АДРЕС КОТОРОГО ЗАДАЕТСЯ (HL);
ADD A,(IX+ Д)    ПРИБАВИТЬ 8-БИТОВОЕ ЧИСЛО,
                 АДРЕС КОТОРОГО ЗАДАЕТСЯ (IX+ Д);
ADD A,(IY+ Д)    ПРИБАВИТЬ 8-БИТОВОЕ ЧИСЛО,
                 АДРЕС КОТОРОГО ЗАДАЕТСЯ (IY+ Д),
```

КОНСТРУКТОРЫ Z8Ø ПРЕДОСТАВИЛИ НАМ ЕЩЕ ОДНУ КОМАНДУ, АНАЛОГИЧНУЮ ADD, НО ПРИНИМАЮЩУЮ ВО ВНИМАНИЕ ВОЗМОЖНОСТЬ ПЕРЕПОЛНЕНИЯ В ФЛАГЕ ПЕРЕНОСА.

ЭТО - ОЧЕНЬ ПОЛЕЗНАЯ КОМАНДА: "ADC", ЧИТАЮШАЯСЯ КАК "СЛОЖЕНИЕ С ПЕРЕНОСОМ" (ADD WITH CARRY).

ОНА СОВЕРШЕННО СОВПАДАЕТ С КОМАНДОЙ "ADD", ИМЕЕТ ТОТ ЖЕ ДИАПАЗОН ЧИСЕЛ, РЕГИСТРОВ,И Т.П., КОТОРЫЕ МОЖНО СКЛАДЫВАТЬ С РЕГИСТРОМ "А", ЗА ИСКЛЮЧЕНИЕМ ТОГО, ЧТО ПЕРЕНОС ТАКЖЕ СКЛАДЫВАЕТСЯ (ЕСЛИ ОН УСТАНОВЛЕН).

ЭТО ДАЕТ ВОЗМОЖНОСТЬ СКЛАДЫВАТЬ ЧИСЛА, ПРЕВЫШАЮЩИЕ 255, С ПОМОЩЬЮ ЦЕПОЧКИ ОПЕРАЦИЙ:

НАПРИМЕР, СЛОЖИТЬ 1ØØØ (Т,Е, Ø3Е8Н) С 2ØØØ (Т,Е, Ø7DØН) И ЗАПОМНИТЬ РЕЗУЛЬТАТ В ВС:

```
LD A,E8H      : МЛАДШИЕ РАЗРЯДЫ ПЕРВОГО ЧИСЛА
ADD A,DØH     : МЛАДШИЕ РАЗРЯДЫ ВТОРОГО ЧИСЛА
LD C,A        : ЗАПИСАТЬ РЕЗУЛЬТАТ В С
LD A,Ø3H      : СТАРШИЕ РАЗРЯДЫ ПЕРВОГО ЧИСЛА
ADC A,Ø7H     : СТАРШИЕ РАЗРЯДЫ ВТОРОГО ЧИСЛА
LD B,A        : ЗАПИСАТЬ РЕЗУЛЬТАТ В В
```

ПОСЛЕ ПЕРВОГО СЛОЖЕНИЯ (Е8 +DØ) У НАС ФЛАГ ПЕРЕНОСА БУДЕТ УСТАНОВЛЕН (ПОСКОЛЬКУ РЕЗУЛЬТАТ БЫЛ БОЛЬШЕ FF), А В РЕГИСТРЕ А БУДЕТ В8 (ПРОВЕРЬТЕ ЭТО САМОСТОЯТЕЛЬНО).

ВТОРОЕ СЛОЖЕНИЕ (3+7)-ДАСТ НЕ ØАН (= 1Ø ДЕСЯТИЧНОЕ), КАК МОЖЕТ ПОКАЗАТЬСЯ НА ПЕРВЫЙ ВЗГЛЯД, А ØВН (= 11 ДЕСЯТИЧНОЕ) ИЗЗА ПЕРЕНОСА.

ПОЭТОМУ КОНЕЧНЫЙ РЕЗУЛЬТАТ БУДЕТ РАВЕН ØВВ8Н=3ØØØ! ТАКУЮ ЦЕПОЧКУ МОЖНО ПРОДОЛЖИТЬ И ОБРАБОТАТЬ ЧИСЛО ПРОИЗВОЛЬНОГО РАЗМЕРА, А РЕЗУЛЬТАТ БУДЕТ ХРАНИТЬСЯ В ПАМЯТИ, А НЕ В ПАРЕ РЕГИСТРОВ.

8-БИТОВОЕ ВЫЧИТАНИЕ

ОНО СОВЕРШЕННО ТАКОЕ ЖЕ, КАК И 8-БИТОВОЕ СЛОЖЕНИЕ, ИМЕЕТСЯ ДВА НАБОРА КОМАНД, ОДИН ДЛЯ ОБЫЧНОГО ВЫЧИТАНИЯ, ДРУГОЙ - ДЛЯ ВЫЧИТАНИЯ С ПЕРЕНОСОМ:

```
SUB S   - ВЫЧЕСТЬ S
SBC S   - ВЫЧЕСТЬ S С ПЕРЕНОСОМ, ОБОЗНАЧЕНИЕ "S" ПРЕДНАЗНАЧЕНО
```

ДЛЯ ТОГО ЖЕ ДИАПАЗОНА ВОЗМОЖНЫХ ОПЕРАНДОВ, ЧТО И ДЛЯ КОМАНД СЛОЖЕНИЯ.

СРАВНЕНИЕ ДВУХ 8-БИТОВЫХ ЧИСЕЛ

ДАВАЙТЕ ОТВЛЕЧЕМСЯ НА МИНУТКУ ОТ МАШИННОГО ЯЗЫКА И ПОСМОТРИМ, ЧТО ЖЕ МЫ НА САМОМ ДЕЛЕ ПОНИМАЕМ ПОД СРАВНЕНИЕМ ДВУХ ЧИСЕЛ.

МЫ ЗНАЕМ, ЧТО ПРОИСХОДИТ, КОГДА ДВА СРАВНИВАЕМЫХ ЧИСЛА СОВПАДАЮТ - ОНИ "РАВНЫ", ОДИН ИЗ СПОСОБОВ ОБОЗНАЧЕНИЯ ЭТОГО ФАКТА В АРИФМЕТИЧЕСКОЙ ФОРМЕ - СКАЗАТЬ, ЧТО РАЗНОСТЬ МЕЖДУ ДВУМЯ ЧИСЛАМИ РАВНА НУЛЮ.

ЧТО ЕСЛИ СРАВНИВАЕМОЕ ЧИСЛО ПРЕВЫШАЕТ ПЕРВОЕ ЧИСЛО? ТОГДА РЕЗУЛЬТАТ ПОСЛЕ ВЫЧИТАНИЯ НОВОГО ЧИСЛА БУДЕТ ОТРИЦАТЕЛЬНЫМ.

АНАЛОГИЧНО, ЕСЛИ НОВОЕ ЧИСЛО БЫЛО МЕНЬШЕ, ТО РАЗНОСТЬ БУДЕТ ПОЛОЖИТЕЛЬНОЙ.

МЫ МОЖЕМ С ПОМОЩЬЮ ЭТИХ ПОНЯТИЙ РАЗРАБОТАТЬ СИСТЕМУ СРАВ-

НЕНИЯ НА МАШИННОМ ЯЗЫКЕ, ВСЕ, ЧТО НАМ НУЖНО, - ЭТО ФЛАГИ И ОПЕРАЦИЯ ВЫЧИТАНИЯ, ПРЕДПОЛОЖИМ, НАМ НУЖНО СРАВНИТЬ ЧИСЛА ИЗ НЕКОТОРОГО ДИАПОЗОНА С ЧИСЛОМ, СКАЖЕМ 5:

```
LD A,5          : ИМЕЮЩЕЕСЯ ЧИСЛО
SUB N           : СРАВНИВАЕМОЕ ЧИСЛО
```

ТОГДА У НАС ПОЛУЧИТСЯ СЛЕДУЮЩИЕ РЕЗУЛЬТАТЫ,
ЕСЛИ N = 5 ФЛАГ НУЛЯ УСТАНОВЛЕН, ФЛАГ ПЕРЕНОСА СБРОШЕН,
ЕСЛИ N < 5 ФЛАГ НУЛЯ СБРОШЕН, ФЛАГ ПЕРЕНОСА СБРОШЕН,
ЕСЛИ N > 5 ФЛАГ НУЛЯ СБРОШЕН, ФЛАГ ПЕРЕНОСА УСТАНОВЛЕН,

ПОЭТОМУ ЯСНО, ЧТО ПРОВЕРКУ НА РАВЕНСТВО ДАЕТ ФЛАГ НУЛЯ, А ПРОВЕРКУ НА "ПРЕВЫШЕНИЕ" - ФЛАГ ПЕРНОСА, (ПРОВЕРКОЙ НА "МЕНЬШЕ" БУДЕТ СБРОС ОБОИХ ФЛАГОВ),

ЕДИНСТВЕННОЕ НЕУДОБСТВО ЭТОГО МЕТОДА СОСТОИТ В ТОМ,ЧТО СОДЕРЖИМОЕ РЕГИСТРА "А" ИЗМЕНЯЕТСЯ В РЕЗУЛЬТАТЕ ЭТОИ ОПЕРАЦИИ,

К СЧАСТЬЮ У НАС ЕСТЬ ОПЕРАЦИЯ "CPS", ПО-АНГЛИЙСКИ ОНА ЧИТАЕТСЯ КАК "COMPARE"(СРАВНИТЬ), ОБРАТИТЕ ВНИМАНИЕ, ЧТО ОНА ПОЗВОЛЯЕТ СРАВНИВАТЬ ТОЛЬКО ТО, ЧТО У НАС УЖЕ ИМЕЕТСЯ В РЕГИСТРЕ "А": ДИАПОЗОН ВОЗМОЖНЫХ ЧИСЕЛ ДЛЯ СРАВНЕНИЯ ТАКОЙ ЖЕ, КАК И ДЛЯ СЛОЖЕНИЯ,

"СРАВНЕНИЕ" СОВЕРШЕННО СОВПАДАЕТ С "ВЫЧИТАНИЕМ" ЗА ИСКЛЮЧЕНИЕМ ТОГО, ЧТО СОДЕРЖИМОЕ РЕГИСТРА "А" НЕ ИЗМЕНЯЕТСЯ, РЕЗУЛЬТАТ СКАЗЫВАЕТСЯ ТАКИМ ОБРАЗОМ ТОЛЬКО НА ФЛАГАХ,

Л О Г И Ч Е С К И Е О П Е Р А Т О Р Ы

ЕСТЬ ТРИ ОПЕРАЦИИ, КОТОРЫЕ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ НА МАШИННОМ ЯЗЫКЕ (ИЛИ ЯЗЫКЕ АССЕМБЛЕРА) ИМЕЮТ ТАКОЕ ЖЕ ЗНАЧЕНИЕ, КАК И БОЛЕЕ ШИРОКО ИСПОЛЬЗУЕМЫЕ СЛОЖЕНИЕ, ВЫЧИТАНИЕ, УМНОЖЕНИЕ ИЛИ ДЕЛЕНИЕ В ОБЫЧНОЙ АРИФМЕТИКЕ,

ИХ ОБЫЧНО НАЗЫВАЮТ БУЛЕВСКИМИ ОПЕРАТОРАМИ В ЧЕСТЬ ЧЕЛОВЕКА, СФОРМУЛИРОВАВШЕГО ДЛЯ ЭТИХ ОПЕРАЦИИ ПРАВИЛА, ЭТО СЛЕДУЮЩИЕ ОПЕРАЦИИ:

AND
OR
XOR

МЫ УЖЕ ЗНАКОМЫ С ПОНЯТИЕМ ОПЕРАЦИИ, ПРИЛОЖИМЫХ КО ВСЕМУ 8-БИТОВОМУ ЧИСЛУ, НО ПРИЧИНА, ПО КОТОРОЙ ЭТИ ОПЕРАЦИИ ИМЕЮТ ТАКОЕ ЗНАЧЕНИЕ, СОСТОИТ В ТОМ, ЧТО ОНИ ЗАТРАГИВАЮТ ОТДЕЛЬНЫЕ БИТЫ ЧИСЛА,

ДАВАЙТЕ РАССМОТРИМ ОДНУ ИЗ ЭТИХ ОПЕРАЦИЙ "AND":

BIT A	BIT B	RESULT OF BIT A "AND" BIT B
0	0	0
1	0	0
0	1	0
1	1	1

BIT - БИТ, RESULT - РЕЗУЛЬТАТ

ОЧЕВИДНО, ЧТО РЕЗУЛЬТАТ ОПЕРАЦИИ "AND" СОСТОИТ В ТОМ, ЧТО "1" ПОЛУЧАЕТСЯ ТОЛЬКО В ТОМ СЛУЧАЕ, ЕСЛИ "А" И "В" ОБА СОДЕРЖАЛИ "1",

НА МАШИННОМ ЯЗЫКЕ ,ЕСЛИ ВЫ ВЫПОЛНЯЕТЕ AND ДЛЯ ДВУХ ЧИСЕЛ, ТО РЕЗУЛЬТАТ БУДЕТ ТАКИМ, КАК ЕСЛИ БЫ ВЫ ВЫПОЛНЯЛИ "AND" ДЛЯ КАЖДОГО ОТДЕЛЬНОГО БИТА ЭТИХ ДВУХ ЧИСЕЛ,

ОПЕРАЦИЯ "AND" ОЧЕНЬ ПОЛЕЗНА ТЕМ, ЧТО ПОЗВОЛЯЕТ НАМ МАСКИРОВАТЬ БАЙТ, ТАК ЧТО ОН ИЗМЕНЯЕТСЯ ТАК, ЧТОБЫ СОДЕРЖАТЬ ТОЛЬКО ОПРЕДЕЛЕННЫЕ БИТЫ,

ЕСЛИ, НАПРИМЕР, МЫ ХОТИМ ОГРАНИЧИТЬ КОНКРЕТНУЮ ПЕРЕМЕННУЮ ДИАПОЗОНОМ 0 - 7, НАМ ХОЧЕТСЯ ВПОЛНЕ ЯСНО УКАЗАТЬ, ЧТО ИНФОРМАЦИЯ ДОЛЖНА СОДЕРЖАТЬСЯ ТОЛЬКО В БИТАХ 0 - 2, (ЕСЛИ БЫ ИНФОРМАЦИЯ СОДЕРЖАЛАСЬ В БИТЕ 3, ТО ЧИСЛО БЫЛО БЫ РАВНО ПО КРАЙНЕЙ МЕРЕ 8)

```
НАПРИМЕР,   0 0 0 0  0 1 0 1    =  5
          (----------)
            ЭТИ БИТЫ ДОЛЖНЫ БЫТЬ "0"
```

ПОЭТОМУ ЕСЛИ МЫ БЕРЕМ НЕКОТОРОЕ ЧИСЛО, ЗНАЧЕНИЕ КОТОРОГО НАМ НЕИЗВЕСТНО, И СОВЕРШАЕМ НАД НИМ ОПЕРАЦИЮ "AND" С"7", ТО В РЕЗУЛЬТАТЕ ПОЛУЧИТСЯ ЧИСЛО В ДИАПАЗОНЕ 0 - 7,

```
НАПРИМЕР,           0 1 1 0  1 0 0 1  = 105
                    0 0 0 0  0 1 1 1  = 7=) МАСКА
```

РЕЗУЛЬТАТ ОПЕРАЦИИ "AND" Ø Ø Ø Ø Ø Ø Ø 1 = 1=)В ДИАПАЗОНЕ Ø - 7

КОМАНДЫ ЛОГИЧЕСКИХ ОПЕРАЦИЙ

MNEMONIC	BYTES	TIME TAKEN	EFFECT ON FLAGS C	Z	PV	S	N	H
AND REGISTER	1	4	Ø	#	#	#	Ø	1
AND NUMBER	2	7	Ø	#	#	#	Ø	1
AND (HL)	1	7	Ø	#	#	#	Ø	1
AND (IX+ Д)	3	19	Ø	#	#	#	Ø	1
AND (IY+ Д)	3	19	Ø	#	#	#	Ø	1
OR REGISTER	1	4	Ø	#	#	#	Ø	Ø
OR NUMBER	2	7	Ø	#	#	#	Ø	Ø
OR (HL)	1	7	Ø	#	#	#	Ø	Ø
OR (IX+ Д)	3	19	Ø	#	#	#	Ø	Ø
OR (IY+ Д)	3	19	Ø	#	#	#	Ø	Ø
XOR REGISTER	1	4	Ø	#	#	#	Ø	Ø
XOR NUMBER	2	7	Ø	#	#	#	Ø	Ø
XOR (HL)	1	7	Ø	#	#	#	Ø	Ø
XOR (IX+ Д)	3	19	Ø	#	#	#	Ø	Ø
XOR (IY+ Д)	3	19	Ø	#	#	#	Ø	Ø

MNEMONIC- МНЕМОНИЧЕСКОЕ ОБОЗНАЧЕНИЕ; BYTES - БАЙТЫ; TIME TAKEN- ВРЕМЯ ВЫПОЛНЕНИЯ; EFFECT ON FLAGS - ВОЗДЕЙСТВИЕ НА ФЛАГИ; REGISTER - РЕГИСТР; NUMBER - ЧИСЛО,

ОБОЗНАЧЕНИЯ ФЛАГОВ:

- ОЗНАЧАЕТ, ЧТО ФЛАГ ИЗМЕНЕН ОПЕРАЦИЕЙ;
Ø - ОЗНАЧАЕТ, ЧТО ФЛАГ СБРАСЫВАЕТСЯ;
1 - ОЗНАЧАЕТ, ЧТО ФЛАГ УСТАНАВЛИВАЕТСЯ;
- - ОЗНАЧАЕТ, ЧТО ФЛАГ НЕ ИЗМЕНИЛСЯ,

ОБРАТИТЕ ВНИМАНИЕ, ЧТО ЧИП Z80 ПОЗВОЛЯЕТ ВЫПОЛНЯТЬ ОПЕРАЦИЮ "AND" ТОЛЬКО С РЕГИСТРОМ "A", ДЛЯ РЕГИСТРА "A" МОЖНО ВЫПОЛНИТЬ ОПЕРАЦИЮ "AND" С 8-БИТОВЫМ ЧИСЛОМ, ЛЮБЫМ ИЗ ОСТАЛЬНЫХ 8-БИТОВЫХ РЕГИСТРОВ,(HL), (IX+) ИЛИ (IY+);

НАПРИМЕР, AND 7
AND E
AND (HL)

ОРАТИТЕ ВНИМАНИЕ, ЧТО ПОСКОЛЬКУ ДЕЙСТВИЕ ОПРЕДЕЛЕНО ТОЛЬКО ДЛЯ РЕГИСТРА "A", НЕТ НЕОБХОДИМОСТИ УПОМИНАТЬ ЕГО В КОМАНДЕ,

ДЛЯ ДРУГИХ БУЛЕВСКИХ ОПЕРАЦИЙ, "OR" И "XOR", ИМЕЕТ МЕСТО ТОТ ЖЕ ДИАПОЗОН ВОЗМОЖНОСТЕЙ И ОГРАНИЧЕНИЕ ИСПОЛЬЗОВАНИЯ РЕГИСТРОМ "A",

ОПЕРАЦИЯ "OR" ПО СМЫСЛУ АНАЛОГИЧНОА ОПЕРАЦИИ "AND":

BIT A	BIT B	BIT A OR BIT B
Ø	Ø	Ø
Ø	1	1
1	Ø	1
1	1	1

ОЧЕВИДНО, РЕЗУЛЬТАТ ОПЕРАЦИИ "OR" ДОЛЖЕН ДАВАТЬ НАМ "1", ЕСЛИ "A" ИЛИ "B" СОДЕРЖАЛО "1",

ОПЕРАЦИЯ "OR" ТКАЖЕ ВЕСЬМА ПОЛЕЗНА ТЕМ, ЧТО ОНА ПОЗВОЛЯЕТ НАМ УСТАНАВЛИВАТЬ ЛЮБОЙ БИТ ЧИСЛА, ЕСЛИ, НАПРИМЕР, НАМ ЖЕЛАТЕЛЬНО ГАРАНТИРОВАТЬ, ЧТО ЧИСЛО БУДЕТ НЕЧЕТНЫМ, ТО СОВЕРШЕННО ОЧЕВИДНО,ЧТО НАМ НУЖНО УСТАНОВИТ БИТ Ø, (ТОГО ЖЕ РЕЗУЛЬТАТА МОЖНО ДОБИТЬСЯ С ПОМОШЬЮ КОМАНДЫ "SET"),

```
LD A,NUMBER
OR 1 : MAKE NUMBER ODD
```

NUMBER - ЧИСЛО; MAKE NIMBER ODD - СДЕЛАТЬ ЧИСЛО НЕЧЕТНЫМ,

ПРИВЕДЕННЫЕ ВЫШЕ ДВЕ СТРОКИ - ТИПИЧНЫЙ ФРАГМЕНТ РАСПЕЧАТКИ НА АССЕМБЛЕРЕ

СМЫСЛ ОПЕРАЦИИ "XOR" (ПРОИЗНОСИТСЯ "ИСКЛЮЧАЮЩЕЕ ИЛИ") ТАКЖЕ ЛЕГКО ПОНЯТЬ, НО ЕЕ РЕАЛЬНОЕ ИСПОЛЬЗОВАНИЕ В ПРОГРАММИРОВАНИИ БОЛЕЕ ОГРАНИЧЕНО, РЕЗУЛЬТАТ ОПЕРАЦИИ "XOR" РАВЕН "1" ТОЛЬКО В ТОМ СЛУЧАЕ, ЕСЛИ "A" ИЛИ "B" ,НО НЕ ОБА СРАЗУ , СО-

ДЕРЖИТ "1",
ИНЫМИ СЛОВАМИ, РЕЗУЛЬТАТ ТАКОЙ ЖЕ, КАК ДЛЯ ОПЕРАЦИИ "OR" ВО ВСЕХ СЛУЧАЯХ, ЗА ИСКЛЮЧЕНИЕМ ТОГО, КОГДА И А И В СОДЕРЖАТ "1",

XOR =) OR - AND

BIT A	BIT B	BIT A "XOR" BIT B
Ø	Ø	Ø
1	Ø	1
Ø	1	1

BIT - БИТ; BIT A "XOR" BIT B - РЕЗУЛЬТАТ ВЫПОЛЕНИЯ "XOR ДЛЯ БИТОВ А И В,
ПОСЛЕДНЕЕ, ЧТО НАМ НЕОБХОДИМО РАССМОТРЕТЬ,ЭТО РЕЗУЛЬТАТ ВОЗДЕЙСТВИЯ ЭТИХ ОПЕРАЦИЙ НА ФЛАГИ,

ФЛАГ НУЛЯ — ЭТОТ ФЛАГ БУДЕТ УСТАНАВЛИВЛЕН (=1), ЕСЛИ РЕЗУЛЬТАТ РАВЕН НУЛЮ;

ФЛАГ ЗНАКА — ЭТОТ ФЛАГ БУДЕТ УСТАНОВЛЕН (=1), ЕСЛИ БИТ 7 РЕЗУЛЬТАТА УСТАНОВЛЕН

ФЛАГ ПЕРНОСА — ФЛАГ БУДЕТ СБРОШЕН (=Ø), ПОСЛЕ "AND", " OR", "XOR", Т,Е, ПЕРЕНОС БУДЕТ СБРОШЕН,

ФЛАГ ЧЕТНОСТИ — ЭТОТ ФЛАГ БУДЕТ УСТАНОВЛЕН (=1), ЕСЛИ В РЕЗУЛЬТАТЕ БУДЕТ ЧЕТНОЕ КОЛИЧЕСТВО ЕДИНИЧНЫХ БИТОВ:
Ø 1 1 Ø 1 1 1 Ø =) УСТАНОВЛЕН
Ø 1 1 Ø 1 Ø 1 Ø =) СБРОШЕН
ОБРАТИТЕ ВНИМАНИЕ, ЧТО ЭТОТ ФЛАГ ТАКЖЕ ДУБЛИРУЕТ ФЛАГ ПЕРЕПОЛНЕНИЯ

ФЛАГ ПОЛОВИННОГО ПЕРЕНОСА — ОБА ФЛАГА СБРОШЕНЫ (=Ø) ПОСЛЕ "AND","OR","XOR"

ФЛАГ ВЫЧИТАНИЯ — ЭТИ ФЛАГИ ПОЛЕЗНЫ, ЕСЛИ ИСПОЛЬЗУЕТСЯ АРИФМЕТИКА ДЛЯ ДВОИЧНО-КОДИРОВАННЫХ ДЕСЯТИЧНЫХ ЧИСЕЛ,

ПРИМЕНЕНИЕ БУЛЕВСКИХ ОПЕРАЦИЙ НАД ФЛАГАМИ
ЕСТЬ ОСОБЫЙ СЛУЧАЙ БУЛЕВСКИХ ОПРЕАТОРОВ, ОЧЕНЬ УДОБНЫЙ, СЛУЧАЙ, КОГДА РЕГИСТР А ДЕЙСТВУЕТ НА САМОГО СЕБЯ,

AND A — "A" НЕ МЕНЯЕТСЯ, ФЛАГ ПЕРЕНОСА СБРАСЫВАЕТСЯ;

OR A — "A" НЕ МЕНЯЕТСЯ, ФЛАГ ПЕРЕНОСА СБРАСЫВАЕТСЯ;

XOR A — "A" УСТАНАВЛИВАЕТСЯ РАВНЫМ НУЛЮ, ФЛАГ ПЕРЕНОСА СБРАСЫВАЕТСЯ,

ЭТИ КОМАНДЫ ШИРОКО ИСПОЛЬЗУЮТСЯ, ПОСКОЛЬКУ В НИХ НУЖЕН ВСЕГО ОДИН БАЙТ ДЛЯ ВЫПОЛНЕНИЯ ТОГО, ЧТО В ПРОТИВНОМ СЛУЧАЕ ПОТРЕБОВАЛО БЫ ДВУХ , НАПРИМЕР, LD A,Ø,
ФЛАГ ПЕРЕНОСА ЧАСТО ПРИХОДИТСЯ СБРАСЫВАТЬ, НАПРИМЕР,В РАБОЧЕМ ПОРЯДКЕ ПЕРЕД ПРИМЕНЕНИЕМ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ, ТАКИХ КАК

ADC — СЛОЖЕНИЕ С ПЕРЕНОСОМ

SBC ВЫЧИТАНИЕ С ПЕРЕНОСОМ И ЭТО МОЖНО ЛЕГКО СДЕЛАТЬ С ПОМОЩЬЮ КОМАНДЫ AND БЕЗ ИЗМЕНЕНИЯ КАКОГО БЫ ТО НИ БЫЛО РЕГИСТРА,

Р А Б О Т А С 16-БИТОВЫМИ Ч И С Л А М И

ДО СИХ ПОР МЫ РАБОТАЛИ ТОЛЬКО С 8-РАЗРЯДНЫМИ ЧИСЛАМИ, НО МЫ ГОВОРИЛИ О ТОМ, ЧТО ЦП МОЖЕТ ТАКЖЕ В НЕКОТОРЫХ СЛУЧАЯХ ОБРАБАТЫВАТЬ И 16-РАЗРЯДНЫЕ ЧИСЛА, ОДИН ИЗ ЭТИХ СЛУЧАЕВ МЫ УЖЕ УПОМИНАЛИ - ЭТО ИНДЕКСНЫЕ РЕГИСТРЫ,
ТЕПЕРЬ ДАВАЙТЕ РАССМОТРИМ РАЗЛИЧНЫЕ СПОСОБЫ АДРЕСАЦИИ, ИМЕЮЩИЕСЯ ДЛЯ ОБРАБОТКИ 16-БИТОВЫХ ЧИСЕЛ,

НЕПОСРЕДСТВЕННАЯ РАСШИРЕННАЯ АДРЕСАЦИЯ

LD RR,NN
(ИЛИ ДРУГИЕ КОМАНДЫ)

ЭТО - ЭКВИВАЛЕНТ НЕПОСРЕДСТВЕННОЙ АДРЕСАЦИИ ДЛЯ 8-БИТОВЫХ ЧИСЕЛ, ПРОСТО НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ РАСШИРЕНА ТАКИМ ОБРАЗОМ, ЧТОБЫ ПРИМЕНЯТЬСЯ ДЛЯ ПЕРЕДАЧИ 16-БИТОВЫХ ДАННЫХ,
В ОБЩЕМ СЛУЧАЕ КОМАНДЫ, ОБРАБАТЫВАЮЩИЕ 16-БИТОВЫЕ ЧИСЛА, ДЛИННЕЕ И МЕДЛЕННЕЕ, ЧЕМ ПРЕДНАЗНАЧЕННЫЕ ДЛЯ 8-БИТОВЫХ, НАПРИМЕР, ЕСЛИ 8-БИТОВЫЕ КОМАНДЫ С НЕПОСРЕДСТВЕННОЙ АДРЕСАЦИЕИ ИМЕЮТ ДЛИНУ 2 БАЙТА (ОДИН ДЛЯ КОМАНДЫ И ОДИН ДЛЯ ЧИСЛА), ТО РАСШИРЕННАЯ ВЕРСИЯ (Т,Е, ДЛЯ 16 БИТОВ) ТРЕБУЕТ ТРЕХ БАЙТОВ,
ФОРМАТ ДЛЯ РАСШИРЕННОЙ НЕПОСРЕДСТВЕННОЙ АДРЕСАЦИИ ИМЕЕТ ВИД:

БАЙТ 1 — КОМАНДА
БАЙТ 2 — N1 МЛАДШИЙ БАЙТ ЧИСЛА

МЫ ИСПОЛЬЗУЕМ ЭТОТ ТИП АДРЕСАЦИИ КОМАНДЫ ДЛЯ ОПРЕДЕЛЕНИЯ СОДЕРЖИМОГО ПАРЫ РЕГИСТРОВ, НАПРИМЕР, В КАЧЕСТВЕ УКАЗАТЕЛЯ НА ЯЧЕЙКУ ПАМЯТИ,

РЕГИСТРОВАЯ АДРЕСАЦИЯ

ВЫ, НАВЕРНО, ПОМНИТЕ, ЧТО РЕГИСТРОВОЙ АДРЕСАЦИЕЙ МЫ НАЗЫВАЕМ ТАКУЮ АДРЕСАЦИЮ КОМАНДЫ, ПРИ КОТОРОЙ ОБРАБАТЫВАЕМОЕ ЗНАЧЕНИЕ ХРАНИТСЯ В ОДНОМ ИЗ РЕГИСТРОВ,

ТО ЖЕ САМОЕ ОСТАЕТСЯ ВЕРНЫМ ДЛЯ 16-БИТОВЫХ КОМАНД, ЗА ИСКЛЮЧЕНИЕМ ТОГО, ЧТО В НАБОРЕ КОМАНД ЦП ТАКИХ КОМАНД НЕМНОГО, ОНИ В ОСНОВНОМ ОТНОСЯТСЯ К АРИФМЕТИЧЕСКИМ ОПЕРАЦИЯМ И ОЧЕНЬ ОГРАНИЧЕНЫ В ПЛАНЕ ДОПУСТИМЫХ КОМБИНАЦИЙ РЕГИСТРОВ,

НАПРИМЕР, ADD HL,BC,

МЫ ЗДЕСЬ ВНОВЬ ОТМЕТИМ ПРЕДПОЧТЕНИЕ, ОКАЗЫВАЕМОЕ ЦП ПАРЕ РЕГИСТРОВ HL,НЕКОТОРЫЕ КОМАНДЫ МОЖНО ВЫПОЛНИТЬ ТОЛЬКО С ПОМОЩЬЮ ЭТОЙ ПАРЫ РЕГИСТРОВ, ЭТО ВЕРНО ДЛЯ АРИФМЕТИЧЕСКИХ КОМАНД, ПОДРОБНЕЕ МЫ НА ЭТОМ ОСТАНОВИМСЯ В ОДНОЙ ИЗ СЛЕДУЮЩИХ ГЛАВ,

КОСВЕННАЯ РЕГИСТРОВАЯ АДРЕСАЦИЯ

КОСВЕННОЙ РЕГИСТРОВОЙ АДРЕСАЦИЕЙ МЫ НАЗЫВАЕМ ТАКУЮ АДРЕСАЦИЮ КОМАНДЫ, ПРИ КОТОРОЙ ТРЕБУЕМОЕ ЗНАЧЕНИЕ НАХОДИТСЯ В ПАМЯТИ, А АДРЕС ЯЧЕЙКИ ПАМЯТИ ХРАНИТСЯ В ПАРЕ РЕГИСТРОВ,

В Z80 ЭТОТ ТИП АДРЕСАЦИИ, ОПЯТЬ-ТАКИ, В ОСНОВНОМ ИСПОЛЬЗУЕТСЯ С ПОМОЩЬЮ ПАРЫ РЕГИСТРОВ HL,

НАПРИМЕР, JP (HL),

РАСШИРЕННАЯ АДРЕСАЦИЯ

РАСШИРЕННАЯ АДРЕСАЦИЯ ПО СМЫСЛУ БЛИЗКА РАСШИРЕННОЙ КОСВЕННОЙ РЕГИСТРОВОЙ АДРЕСАЦИИ, ЗА ИСКЛЮЧЕНИЕМ ТОГО, ЧТО НУЖНОЕ ЗНАЧЕНИЕ ХРАНИТСЯ НЕ В ПАРЕ РЕГИСТРОВ, А В ПАРЕ ЯЧЕЕК ПАМЯТИ,

НАПРИМЕР, LD HL,(NN), ГДЕ NN ДОЛЖНО БЫТЬ ЗАДАНО НА СТАДИИ ПРОГРАММИРОВАНИЯ,

О Б Р А Б О Т К А Ч И С Е Л Н А Д В У Х РЕГИСТРАХ

ЗАДАНИЕ АДРЕСОВ С ПОМОЩЬЮ 16-БИТОВЫХ ЧИСЕЛ

ОБРАТИТЕ, ПОЖАЛИСТА, ВНИМАНИЕ, ЧТО ВСЕ АДРЕСА ЗАДАЮТСЯ С ПОМОЩЬЮ 16-БИТОВЫХ ЧИСЕЛ,

ВЫ ПРОСТО НЕ СМОЖЕТЕ ЗАДАТЬ АДРЕС С ПОМОЩЬЮ ВСЕГО 8 БИТОВ, ДАЖЕ ЕСЛИ ЭТОТ АДРЕС ЛЕЖИТ В ДИАПОЗОНЕ ОТ 0 ДО 255, В СООТВЕТСТВИИ СО СПОСОБОМ РАБОТЫ ЦП ЭТО НЕ БУДЕТ АДРЕСОМ, ПОСКОЛЬКУ В НЕМ НЕТ ДВУХ БАЙТОВ ПО 8 БИТОВ КАЖДЫЙ,

МЫ ПОДРАЗУМЕВАЛИ ЭТО, КОГДА ИСПОЛЬЗОВАЛИ СОКРАЩЕННУЮ ЗАПСЬ:

LD A,(NN)

ТАК ЧТО ПОМНИТЕ ТАКЖЕ, ЧТО 16-БИТОВЫЕ ЧИСЛА ХРАНЯТСЯ В ПАРЕ РЕГИСТРОВ ТАК, ЧТО ПЕРВЫМ ИДЕТ СТАРШИЙ БАЙТ

Х Р А Н Е Н И Е 16-Б И Т О В Ы Х Ч И С Е Л В П А М Я Т И

ЕСТЬ ОДИН АСПЕКТ КОНСТРУКЦИИ Z80, КОТОРЫЙ ОЧЕНЬ ТРУДНО ОБ"ЯСНИТЬ ИЛИ ОПРАВДАТЬ: ПРИ ЗАГРУЗКЕ 16-БИТОВЫХ ЧИСЕЛ В ПАМЯТЬ ИСПОЛЬЗУЕТСЯ ОБРАТНЫЙ ПОРЯДОК ПО СРАВНЕНИЮ С ПАРАМИ РЕГИСТРОВ, МЛАДШИЙ БИТ В ПАМЯТИ ВСЕГДА ХРАНИТСЯ ПЕРВЫМ!

ДАВАЙТЕ РАССМОТРИМ СИТУАЦИЮ, В КОТОРОЙ МЫ ПОМЕЩАЕМ В ПАМЯТЬ СОДЕРЖИМОЕ HL;

ДО ТОГО:

	ЯЧЕЙКА	СОДЕРЖИМОЕ
	32000	00
H L	32001	00
01 02	32002	00

ПРЕДПОЛОЖИМ, ЧТО В HL СОДЕРЖИТСЯ ДЕСЯТИЧНОЕ ЧИСЛО 258 = 0102H, ВСЕ ЯЧЕЙКИ ПАМЯТИ ПУСТЫ,

ПОСЛЕ

	ЯЧЕЙКА	СОДЕРЖИМОЕ
	32000	02
H L	32001	01
01 02	32002	00

ПОРЯДОК ХРАНЕНИЯ 16-БИТОВЫХ ЧИСЕЛ В ПАМЯТИ (И В РАСПЕЧАТКЕ ПРОГРАММЫ) ТАКОВ , ЧТО МЛАДШИЙ БИТ ВСЕГДА ХРАНИТСЯ В НАЧАЛЕ.

КОМАНДЫ ОПЕРАЦИИ ЗАГРУЗКИ ДЛЯ 16-РАЗРЯДНЫХ ЧИСЕЛ

!	!	! TIME! EFFECT ON FLOGS !

MNEMONIC	BYTES	TAKEN	C	Z	PV	S	N	H
LD REG PAIR,NUMBER	3/4	10	-	-	-	-	-	-
LD IX,NUMBER	4	14	-	-	-	-	-	-
LD IY,NUMBER	4	14	-	-	-	-	-	-
LD (ADDRESS),BS OR DE	4	20	-	-	-	-	-	-
LD (ADDRESS),HL	3	16	-	-	-	-	-	-
LD (ADDRESS),IX	4	20	-	-	-	-	-	-
LD (ADDRESS),IY	4	20	-	-	-	-	-	-
LD BC OR DE,(ADDRESS)	4	20	-	-	-	-	-	-
LD HL,(ADDRESS)	3	16	-	-	-	-	-	-
LD IX,(ADDRESS)	4	20	-	-	-	-	-	-
LD IY,(ADDRESS)	4	20	-	-	-	-	-	-

MNEMONIC -МНЕМОНИЧЕСКОЕ ОБОЗНАЧЕНИЕ; BYTES -КОЛИЧЕСТВЛ БАЙТОВ; TIME TAKEN - ВРЕМЯ ВЫПОЛНЕНИЯ; EFFECT ON FLGS -ВОЗДЕЙСТВИЕ НА ФЛАГИ; REG PAIR - ПАРА РЕГИСТРОВ; NUMBER - ЧИСЛО; ADDRESS - АДРЕС; OR - ИЛИ,

ОБОЗНАЧЕНИЯ ДЛЯ ФЛАГОВ:

= ОЗНАЧАЕТ, ЧТО ФЛАГ ИЗМЕНИЛСЯ В РЕЗУЛЬТАТЕ ОПЕРАЦИИ;

0 ОЗНАЧАЕТ, ЧТО ФЛАГ СБРАСЫВАЕТСЯ;

1 ОЗНАЧАЕТ, ЧТО ФЛАГ УСТАНАВЛИВАЕТСЯ;

- ОЗНАЧАЕТ, ЧТО ФЛАГ НЕ ИЗМЕНЯЕТСЯ,

ОЧЕНЬ ЖЕЛАТЕЛЬНО, ЧТОБЫ ВЫ ВНИМАТЕЛЬНО ЭТО ПРОЧЛИ И УДОСТОВЕРИЛИСЬ, ЧТО ОСВОИЛИ ЭТО ОБРАЩЕНИЕ ТРАДИЦИОННЫХ ПРАВИЛ, СКОРЕЕ ВСЕГО ИМЕННО ЭТО СТАНЕТ ЕДИНСТВЕННЫМ ВАЖНЕЙШИМ ИСТОЧНИКОМ ОШИБОК В ПРОГРАММАХ:

В РЕГИСТРАХ: СТАРШИЙ БАЙТ ХРАНИТСЯ ПЕРВЫМ;

В ПАМЯТИ И ПРОГРАММАХ: МЛАДШИЙ БАЙТ ХРАНИТСЯ ПЕРВЫМ,

ЭТОТ ФАКТ НЕЛЬЗЯ ПРОЧИТАТЬ И ЗАБЫТЬ, ПОСКОЛЬКУ КАЖДЫЙ РАЗ, КОГДА ВАМ ПРИДЕТСЯ ИМЕТЬ ДЕЛО С 16-БИТОВОЙ КОМАНДОЙ НА МАШИННОМ ЯЗЫКЕ НУЖНО БУДЕТ ТЩАТЕЛЬНО ПРОДУМАТЬ ПОРЯДОК МЛАДШИХ И СТАРШИХ БИТОВ,

ЗАГРУЗКА 16-БИТОВЫХ ЧИСЕЛ

ГРУППА КОМАНД ЗАГРУЗКИ 16-БИТОВЫХ ЧИСЕЛ, ГОВОРЯ УПРОЩЕННО, СВОДИТСЯ К ЗАГРУЗКЕ 16-БИТОВОГО ЧИСЛА В ПАРУ РЕГИСТРОВ, ОБЩИЙ ВИД МНЕМОНИЧЕСКОГО СОКРАЩЕНИЯ СЛЕДУЮЩИЙ

```
LD RR,NN
```

ДРУГИЕ КОМАНДЫ ЗАГРУЗКИ ДЛЯ 16 БИТОВ

ПОМИМО ВОЗМОЖНОСТИ ЗАГРУЗКИ 16-БИТОВЫХ ЧИСЕЛ НЕПОСРЕДСТВЕННО В ПАРЫ РЕГИСТРОВ У НАС ЕСТЬ ЕЩЕ ВОЗМОЖНОСТЬ ЗАГРУЗКИ 16БИТОВЫХ ЧИСЕЛ НЕПОСРЕДСТВЕННО В ИНДЕКСНЫЕ РЕГИСТРЫ,

```
LD IX,NN
LD IY,NN
```

МЫ МОЖЕМ ТАКЖЕ УПРАВЛЯТЬ ОБМЕНОМ ИНФОРМАЦИЕЙ МЕЖДУ ПАРОЙ РЕГИСТРОВ И ДВУМЯ ПОСЛЕДОВАТЕЛЬНЫМИ ЯЧЕЙКАМИ ПАМЯТИ, (ЭТО 16-БИТОВЫЙ ЭКВИВАЛЕНТ ЗАГРУЗКИ ИНФОРМАЦИИ ИЗ ОДНОГО РЕГИСТРА В ОДНУ ЯЧЕЙКУ ПАМЯТИ),

ОБЩИЙ ВИД КОМАНДЫ СЛЕДУЮШИЙ

```
LD (NN),RR
LD (NN),IX
LD (NN),IY
```

НАПОМНИМ, ЧТО СКОБКИ - ЭТО СОКРАЩЕННОЕ ОБОЗНАЧЕНИЕ "СОДЕРЖИМОГО", ТАК ЧТО ПОСЛЕДНЯЯ КОМАНДА ЧИТАЕТСЯ КАК "ЗАГРУЗИТЬ СОДЕРЖИМОЕ РЕГИСТРА IY В ЯЧЕЙКУ ПАМЯТИ NN",

ПОСКОЛЬКУ МЫ ИМЕЕМ ДЕЛО С 16-БИТОВЫМИ ЧИСЛАМИ, НА САМОМ ДЕЛЕ МЫ ЗАГРУЖАЕМ УКАЗАННУЮ ЯЧЕЙКУ ПАМЯТИ И СЛЕДУЮЩУЮ ЗА НЕЙ В ПАРУ РЕГИСТРОВ, НЕТ НЕОБХОДИМОСТИ ЗАДАВАТЬ ОБА АДРЕСА (ПОСКОЛЬКУ ЦП МОЖЕТ ВЫЧИСЛИТЬ АДРЕС ВТОРОЙ ЯЧЕЙКИ), НО НАДО ПРОЯВИТЬ ВНИМАНИЕ, ЧТОБЫ НЕ СПУТАТЬ 8-БИТОВЫЕ ОПЕРАЦИИ С 16-БИТОВЫМИ,

ЗДЕСЬ ТАКЖЕ ПРОЯВЛЯЕТСЯ ВЗАИМООБРАЗНЫЙ ХАРАКТЕР МНОГИХ КОМАНД, ТАК ЧТО МЫ МОЖЕМ ТАКЖЕ ЗАГРУЗИТЬ В ПАРУ РЕГИСТРОВ ИЛИ В ИНДЕКСНЫЙ РЕГИСТР ВСЕ, ЧТО НИ НАХОДИТСЯ В КОНКРЕТНОЙ ПАРЕ

ЯЧЕЕК ПАМЯТИ:

LD RR,(NN)
LD IX,(NN)
LD IY,(NN)

КОМАНДЫ ОПЕРАЦИИ СО СТЕКОМ

MNEMONIC	BYTES	TIME TAKEN	EFFEKT ON FLAGS					
			C	Z	PV	S	N	H
PUSH REG PAIR	1	11	-	-	-	-	-	-
PUSH IX OR IY	2	15	-	-	-	-	-	-
POP REG PATR	1	10	-	-	-	-	-	-
POP IX OR IY	2	14	-	-	-	-	-	-
LD SP, ADDRESS	3	10	-	-	-	-	-	-
LD SP,(ADDRESS)	3	20	-	-	-	-	-	-
LD SP, HL	1	6	-	-	-	-	-	-
LD SP IX OR IY	2	10	-	-	-	-	-	-

MNEMONIC - МНЕМОНИЧЕСКОЕ ОБОЗНАЧЕНИЕ; BYTES - БАЙТЫ; TIME TAKEN -ВРЕМЯ ВЫПОЛНЕНИЯ; EFFEST ON FLAGS -РЕЗУЛЬТАТ ВОЗДЕЙСТВИЯ НА; REG PAIR - ПАРА РЕГИСТРОВ; OR - ИЛИ; ADDRESS - АДРЕС,

ОБОЗНАЧЕНИЯ ДЛЯ ФЛАГОВ:

\# ОЗНАЧАЕТ, ЧТО ФЛАГ ИЗМЕНЯЕТСЯ В РЕЗУЛЬТАТЕ ОПЕРАЦИИ;
Ø ОЗНАЧАЕТ, ЧТО ФЛАГ СБРАСЫВАЕТСЯ;
1 ОЗНАЧАЕТ, ЧТО ФЛАГ УСТАНАВЛИВАЕТСЯ;
\- ОЗНАЧАЕТ, ЧТО ФЛАГ НЕ ИЗМЕНЯЕТСЯ В РЕЗУЛЬТАТЕ ОПЕРАЦИИ.

ОДНО ИЗ ПРЕИМУЩЕСТВ, ВОЗМОЖНО НЕ ОЧЕНЬ СУЩЕСТВЕННЫХ ,ОПЕРАЦИИ СО СТЕКОМ СОСТОИТ В ТОМ, ЧТО МЫ МОЖЕМ ВТАЛКИВАТЬ И ВЫТАЛКИВАТЬ ИНФОРМАЦИЮ ПОРЦИЯМИ ПО 16 БИТОВ,ЭТО ПРОИСХОДИТ ПОТОМУ, ЧТО СТЕК ПРЕЖДЕ ВСЕГО СКОНСТРУИРОВАН ДЛЯ ЗАПОМИНАНИЯ АДРЕСОВ, А НАМ НЕОБХОДИМО ЗАДАВАТЬ АДРЕСА В ВИДЕ 16-БИТОВЫХ ЧИСЕЛ,

ОБЩИЙ ВИД КОМАНД ВТАЛКИВАНИЯ ИНФОРМАЦИИ В СТЕК СЛЕДУЮЩИЙ:

PUSH RR
PUSH IX

PUSH IY А ОБЩИЙ ВИД КОМАНД ВЫТАЛКИВАНИЯ ИНФОРМАЦИИ НАЗАД ИЗ СТЕКА:

POP RR
POP IX

POP IY ЭТО - ОЧЕНЬ ПРОСТЫЕ КОМАНДЫ, И ВЫ, КОНЕЧНО, ЗАМЕТИТЕ, ЧТО НЕТ НЕОБХОДИМОСТИ ЗАДАВАТЬ АДРЕС,

ДЛЯ ОБЫЧНЫХ ПАР РЕГИСТРОВ (Т,Е, НЕ ИНДЕКСНЫХ РЕГИСТРОВ) ЭТИ КОМАНДЫ ИМЕЮТ ДЛИНУ ВСЕГО ОДИН БАЙТ И ПОЭТОМУ ЭКОНОМНЫ С ТОЧКИ ЗРЕНИЯ ПРАКТИКИ ПРОГРАММИРОВАНИЯ,

КОМАНДЫ PUSH, КРОМЕ ТОГО, НЕ ПОРТЯТ РЕГИСТРОВ, Т,Е, 16-БИТОВЫЙ РЕГИСТР ПОСЛЕ ВЫПОЛНЕНИЯ КОМАНДЫ PUSH ПРОДОЛЖАЕТ СОДЕРЖАТЬ ТУ ЖЕ ИНФОРМАЦИЮ, ЧТО И ДО НЕЕ,

ОБРАТИТЕ ВНИМАНИЕ, ЧТО ПОСКОЛЬКУ МЫ МОЖЕМ ВЫПОЛНЯТЬ ОПЕРАЦИИ PUSH И POP ДЛЯ ПРОИЗВОЛЬНЫХ ПАР РЕГИСТРОВ, РЕГИСТР, ДЛЯ КОТОРОГО ВЫПОЛНЯЕТСЯ КОМАНДА POP НЕ ОБЯЗАН БЫТЬ ТЕМ ЖЕ САМЫМ, ДЛЯ КОТОРОГО ВЫПОЛНЯЛАСЬ КОМАНДА PUSH!

НАПРИМЕР,

PUSH BC

POP HL РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ЭТИХ ДВУХ КОМАНД СОСТОИТ В ТОМ, ЧТО СОДЕРЖИМОЕ РЕГИСТРА (ТАК В ОРИГИНАЛЕ, ТОЧНЕЕ БЫЛО БЫ ГОВОРИТЬ О ПАРЕ РЕГИСТРОВ, ПРИМЕЧ, ПЕР,) BC НЕ ИЗМЕНЯЕТСЯ, А СОДЕРЖИМОЕ HL СТАНОВИТСЯ РАВНЫМ СОДЕРЖИМОМУ РЕГИСТРА BC ВО ВРЕМЯ ВЫПОЛНЕНИЯ КОМАНДЫ PUSH,

ЭТО ПО СУЩЕСТВУ ДОБАВЛЯЕТ КОМАНДУ ТИПА

LD RR,RR" К ГРУППЕ КОМАНД ЗАГРУЗКИ 16-БИТОВЫХ ЧИСЕЛ, КОТОРАЯ ЯВНО ОТСУТСТВОВАЛА,

ПОСКОЛЬКУ КАЖДАЯ ИЗ КОМАНД PUSH И POP ДЛЯ ПАР РЕГИСТРОВ ИМЕЕТ ДЛИНУ ВСЕГО 1 БАЙТ, ЗАТРАТЫ С ТОЧКИ ЗРЕНИЯ ПАМЯТИ НЕ

ОЧЕНЬ ВЕЛИКИ,

ДРУГИМ ДОПОЛНИТЕЛЬНЫМ ПРЕИМУЩЕСТВОМ ЯВЛЯЕТСЯ ТО, ЧТО МЫ МОЖЕМ ВЫПОЛНЯТЬ КОМАНДЫ PUSH И POP ДЛЯ ПАРЫ РЕГИСТРОВ AF! ЭТО - ОДНА ИЗ НЕМНОГИХ КОМАНД, В КОТОРЫХ AF РАССМАТРИВАЕТСЯ КАК ПАРА РЕГИСТРОВ, И ОЧЕВИДНО В ЭТОМ ЕСТЬ СМЫСЛ, ПОСКОЛЬКУ НАМ НЕ РАЗ ПРИДЕТСЯ СОХРАНЯТЬ СОДЕРЖИМОЕ ФЛАГОВ,

ЭТО ОЗНАЧАЕТ, ЧТО ВЫ МОЖЕТЕ ВЫПОЛНИТЬ PUSH AF (ПО СУЩЕСТВУ ЗАПИСАТЬ ,ЧЕМУ РАВНЫ A И F), ПРОВЕСТИ ВЫЧИСЛЕНИЯ, НЕЖЕЛАТЕЛЬНЫМ ПОБОЧНЫМ ЭФФЕКТОМ КОТОРЫХ МОЖЕТ БЫТЬ ИЗМЕНЕНИЕ ФЛАГОВ, И ЗАТЕМ ВЫПОЛНИТЬ POP ДЛЯ AF, ОСТАВЛЯЯ ФЛАГИ НЕИЗМЕННЫМИ,

РАБОТА СО СТЕКОМ

КАК ВЫ ЗНАЕТЕ, ДЕЙСТВИТЕЛЬНЫЕ ПРЕИМУЩЕСТВА КОМАНД PUSH И POP СОСТОЯТ В ТОМ, ЧТО НАМ НЕ ПРИХОДИТСЯ ДУМАТЬ ОБ АДРЕСАХ ЧИСЕЛ, НАД КОТОРЫМИ ЭТИ КОМАНДЫ ВЫПОЛНЯЮТСЯ,,

ВЫ, БЕЗУСЛОВНО, СОГЛАСИТЕСЬ, ЧТО НЕ ВСЕГДА ИМЕЕТ СМЫСЛ, ЧТОБЫ ОДНА И ТА ЖЕ ОБЛАСТЬ ПАМЯТИ СЛУЖИЛА СТЕКОМ НЕЗАВИСИМО ОТ ТОГО, БУДЕТ У ВАС 16К ИЛИ 48К

НА САМОМ ДЕЛЕ ЦП ОТСЛЕЖИВАЕТ АДРЕС СТЕКА С ПОМОЩЬЮ "УКАЗАТЕЛЯ СТЕКА", КОТОРЫЙ МОЖНО СЧИТАТЬ 16-БИТОВЫМ РЕГИСТРОМ,МЫ КРАТКО УПОМЯНУЛИ ОБ ЭТОМ, КОГДА РАССМАТРИВАЛИ РЕГИСТРЫ, НО НИЧЕГО НЕ ГОВОРИЛИ ОБ ЭТОМ ПРИ РАССМОТРЕНИИ КОМАНД LOAD И ТОМУ ПОДОБНЫХ, ПОСКОЛЬКУ ЭТО НЕ ТАКОЙ РЕГИСТР, КОТОРЫЙ МОЖНО ОБРАБАТЫВАТЬ ТАК ЖЕ, КАК ОСТАЛЬНЫЕ РЕГИСТРЫ,

ОСНОВНОЕ , ЧТО ЖЕЛАТЕЛЬНО УМЕТЬ ДЕЛАТЬ С УКАЗАТЕЛЕМ СТЕКА, ЭТО ОПРЕДЕЛЯТЬ ЕГО ПОЛОЖЕНИЕ В ПАМЯТИ, И ИМЕННО ТАКОЙ ТИП КОМАНДЫ ИМЕЕТСЯ:

```
LD SP,NN
LD SP,(NN)
LD SP,IX
```

LD SP,IY ВЫ МОЖЕТЕ ПРОВЕРИТЬ СТЕК "СПЕКТРУМ" С ПОМОЩЬЮ КОМАНДЫ "МЕМ" ПРОГРАММЫ "РЕДАКТОРА МАШИННОГО ЯЗЫКА В КОДЕ EZ", ПРОСМАТРИВАЯ ПОСЛЕДНИЕ 30 - 40 БАЙТОВ ПЕРЕД RAMTOP,

НЕ ИЗМЕНЯЙТЕ СОДЕРЖИМОЕ ЯЧЕЕК СТЕКА

ПОЧТИ ЛЮБОЕ ИЗМЕНЕНИЕ ПРИВЕДЕТ К ОСТАНОВКЕ "СПЕКТРУМ" ЭКРАН ПОГАСНЕТ И ВАМ ПРИДЕТСЯ ВНОВЬ ВКЛЮЧАТЬ ПИТАНИЕ, ТАК ПРОИСХОДИТ ПОТОМУ, ЧТО ОПЕРАЦИОННАЯ СИСТЕМА ПОМЕЩАЕТ МНОГО ИНФОРМАЦИИ, НЕОБХОДИМОЙ ЕИ, В СТЕК, И ИЗМЕНЕНИЯ ПРИВОДЯТ К ТОМУ, ЧТ ОНА ИДЕТ ВРАЗНОС,

ПО ЭТОЙ ЖЕ САМОЙ ПРИЧИНЕ НЕ ПЫТАЙТЕСЬ МЕНЯТЬ ПОЛОЖЕНИЕ УКАЗАТЕЛЯ СТЕКА, ЕСЛИ ТОЛЬКО У ВАС НЕТ ПОЛНОЙ УВЕРЕННОСТИ В ТОМ, ЧТО ВЫ ДЕЛАЕТЕ,

ЗАМЕЧАНИЕ:

В ПРАВИЛЬНО ОРГАНИЗОВАННОЙ ПРОГРАММЕ КОЛИЧЕСТВО КОМАНД POP И PUSH ДОЛЖНО СОВПАДАТЬ, НЕЗАВИСИМО ОТ ТОГО, ПО КАКОЙ ВЕТВИ ШЛА ПРОГРАММА, ЛЮБЫЕ ОШИБКИ В ПОДСЧЕТЕ МОГУТ ПРИВЕСТИ К СТРАННЫМ РЕЗУЛЬТАТАМ,

АРИФМЕТИЧЕСКИЕ КОМАНДЫ ДЛЯ 16-РАЗРЯДНЫХ ЧИСЕЛ

MNEMONIC	BYTES	TIME TAKEN	EFFECT ON FLAGS					
			C	Z	PV	S	N	H
ADD HL,REG PAIR	1	11	#	-	-	-	Ø	?
ADD HL,SP	2	11	#	-	-	-	Ø	?
ADC HL,REG PAIR	2	15	#	#	#	#	Ø	?
ADC IX,SP	2	15	#	#	#	#	Ø	?
ADD IX,BC OR DE	2	15	#	-	-	-	Ø	?
ADD IX,IX	2	15	#	-	-	-	Ø	?
ADD IX,SP	2	15	#	-	-	-	Ø	?
ADD IY,BC OR DE	2	15	#	-	-	-	Ø	?
ADD IY,IY	2	15	#	-	-	-	Ø	?
ADD IY,SP	2	15	#	-	-	-	Ø	?
SBC HL,REG PAIR	2	15	#	#	#	#	1	?
SBC HL,SP	2	15	#	#	#	#	1	?

MNEMONIC -МНЕМОНИЧЕСКОЕ ОБОЗНАЧЕНИЕ;BYTES -БАЙТЫ; TIME TAKEN -ВРЕМЯ ВЫПОЛНЕНИЯ; EFFECT ON FLAGS -ВОЗДЕЙСТВИЕ НА ФЛАГИ;REG PAIR - ПАРА РЕГИСТРОВ; OR - ИЛИ;

ОБОЗНАЧЕНИЕ ФЛАГОВ

ОЗНАЧАЕТ, ЧТО ФЛАГ ИЗМЕНЯЕТСЯ В РЕЗУЛЬТАТЕ ОПЕРАЦИИ;
0 ОЗНАЧАЕТ, ЧТО ФЛАГ СБРАСЫВАЕТСЯ;
1 ОЗНАЧАЕТ, ЧТО ФЛАГ УСТАНАВЛИВАЕТСЯ
- ОЗНАЧАЕТ, ЧТО ФЛАГ НЕ ИЗМЕНЯЕТСЯ;
? ОЗНАЧАЕТ, ЧТО РЕЗУЛЬТАТ НЕИЗВЕСТЕН,

А Р И Ф М Е Т И Ч Е С К И Е Д Е Й С Т В И Я

ОДНО ИЗ ПРЕИМУЩЕСТВ ВОЗМОЖНОСТИ 16-БИТОВОГО ПРЕДСТАВЛЕНИЯ ЧИСЕЛ В 8-БИТОВОМ ПРОЦЕССОРЕ СОСТОИТ В ТОМ, ЧТО МЫ МОЖЕМ ПРИМЕНЯТЬ 16 БИТОВ ДЛЯ ЗАДАНИЯ ОДРЕСА И ДЛЯ ВЫПОЛНЕНИ ВЫЧИСЛЕНИЙ НАД ЦЕЛЫМИ ЧИСЛАМИ ДО 65 355 (ИЛИ В ДИАПАЗОНЕ ОТ -3276 8 ДО 32767, ЕСЛИ ДОПУСКАЮТСЯ ОТРИЦАТЕЛЬНЫЕ ЧИСЛА),

В СВЕТЕ ВЫШЕСКАЗАННОГО ЛЕГКО ПОНЯТЬ ,ПОЧЕМУ В НЕКОТОРЫХ РАННИХ МОДЕЛЯХ МИКРО-ЭВМ, ТАКИХ КАК ПЕРВОНАЧАЛЬНАЯ МОДЕЛЬ "СИНКЛЕР-ZX80", ВСЕ АРИФМЕТИЧЕСКИЕ ДЕИСТВИЯ В ЯЗЫКЕ "БЕЙСИК" ОГРАНИЧИВАЛИСЬ ЦЕЛЫМИ ЧИСЛАМИ ИЗ ДИАПАЗОНА -32 000 ДО +32 000,

ПРИВЕЛИГИРОВАННАЯ ПАРА РЕГИСТРОВ

СОВЕРШЕННО ТАК ЖЕ, КАК РЕГИСТР "A" ЯВЛЯЕТСЯ ПРИВЕЛИГИРОВАННЫМ С ТОЧКИ ЗРЕНИЯ АРИФМЕТИЧЕСКИХ ДЕЙСТВИЙ ДЛЯ 8-РАЗР,ЧИСЕЛ,ЕСТЬ И ПРИВЕЛИГИРОВАННАЯ ПАРА РЕГИСТРОВ ДЛЯ АРИФМЕТИЧЕСКИХ ДЕЙСТВИЙ НАД ЧИСЛАМИ ДЛЯ 16-РАЗР,ЧИСЕЛ, И ЭТО - ПАРА РЕГИСТРОВ HL

ЭТА ПРИВИЛЕГИРОВАННОСТЬ НЕ СТОЛЬЯВНО ВЫРАЖЕНА, КАК В СЛУЧАЕ 8-БИТОВЫХ ЧИСЕЛ, ТАК ЧТО МЫ НЕ ОПУСКАЕМ ИМЯ ПАРЫ РЕГИСТРОВ,

СЛОЖЕНИЕ

ОПЕРАЦИИ СЛОЖЕНИЯ ВПОЛНЕ ПРЯМОЛИНЕЙНЫ:

```
ADD HL,BC
ADD HL,DE
ADD HL,HL
```

ADD HL,SP НО ИМЕННО ТАК ОНИ И ЗАПИСЫВАЮТСЯ!

ОБРАТИТЕ ВНИМАНИЕ, ЧТО НЕ УДАЕТСЯ СЛОЖИТЬ АБСОЛЮТНОЕ ЧИСЛО С HL, - НАПРИМЕР НЕ ДОПУСКАЕТСЯ "ADD HL,NN", ЧТОБЫ ВЫПОЛНИТЬ ВЫЧИСЛЕНИЯ ТАКОГО ТИПА,, НАМ НУЖНО:

```
LD DE,NN
ADD HL,DE
```

КОГДА ВЫ УЧТЕТЕ, ЧТО ТЕПЕРЬ У ВАС СВЯЗАННЫМИ ОКАЗАЛИСЬ ЧЕТЫРЕ ИЗ 8-БИТОВЫХ РЕГИСТРА, КОТОРЫХ ВСЕГО 7, ВЫ СРАЗУ ПОЙМЕТЕ, ЧТО ВЫ НЕ ЗАХОТИТЕ ДЕЛАТЬ ЭТО СЛИШКОМ ЧАСТО ,

ОБРАТИТЕ ТАКЖЕ ВНИМАНИЕ, ЧТО МЕЖДУ HL И ИНДЕКСНЫМИ РЕГИСТРАМИ НЕЛЬЗЯ ВЫПОЛНИТЬ СЛОЖЕНИЕ, ВЫ ТАКЖЕ ВПОМНИТЕ, ЧТО НЕТ КОМАНДЫ LOAD,КОТОРАЯ ПОЗВОЛЯЛА БЫ ВАМ ПЕРЕДАВАТЬ СОДЕРЖИМОЕ IX ИЛИ IY В BC ИЛИ DE, ТАК ЧТО ЕДИНСТВЕННЫМ СПОСОБОМ ВЫПОЛНЕНИЯ ТАКОГО СЛОЖЕНИЯ БЫЛО БЫ:

```
PUSH IX
POP DE
```

ADD HL,DE ОДИН ИЗ МОМЕНТОВ,О КОТОРОМ СЛЕДУЕТ УПОМЯНУТЬ -ЭТО РЕГИСТР "SP" - УКАЗАТЕЛЬ СТЕКА,ЭТО-ОДНА ИЗ ОЧЕНЬ НЕМНОГИХ ОПЕРАЦИЙ,В КОТОРЫХ "SP" ОБРАБАТЫВАЕТСЯ КАК НАСТОЯЩИЙ РЕГИСТР, НО,ОЧЕВИДНО,ВЫ НЕ МОЖЕТЕ ПРИМЕНЯТЬ ЕГО В КАЧЕСТВЕ ПЕРЕМЕННОЙ! ПОДУМАЙТЕ, ЧТО СЛУЧИЛОСЬ БЫ СО ВСЕМИ КОМАНДАМИ POP И PUSH, ЕСЛИ БЫ ВЫ ПРОИЗВОЛЬНЫМ ОБРАЗОМ МЕНЯЛИ СОДЕРЖИМОЕ УКАЗАТЕЛЯ СТЕКА!

ВОЗДЕЙСТВИЕ НА ФЛАГИ

16-БИТОВЫЕ АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ - ЭТО ИМЕННО ТА ОБЛАСТЬ, В КОТОРОЙ ФЛАГ ПЕРЕНОСА ИГРАЕТ ПРИСУШУЮ ЕМУ РОЛЬ,ПОСКОЛЬКУ,КАК МОЖНО ВИДЕТЬ ИЗ ТАБЛИЦЫ,ПОМЕЩЕННОЙ В НАЧАЛЕ ЭТОЙ ГЛАВЫ,ЕДИНСТВЕННЫЙ ДРУГОЙ ФЛАГ, НА КОТОРЫЙ ОКАЗЫВАЕТ ВЛИЯНИЕ КОМАНДА "ADD" (СЛОЖЕНИЕ) - ЭТО ФЛАГ "ВЫЧИТАНИЯ" (И ВСЕ ЧТО ОН В ЭТОМ СЛУЧАЕ ЗНАЧАЕТ - ЭТО ТО, ЧТО КОМАНДА СЛОЖЕНИЯ НЕ ЯВЛЯЕТСЯ ВЫЧИТАНИЕМ

ФЛАГ ПЕРЕНОСА БУДЕТ УСТАНОВЛЕН, ЕСЛИ ИМЕЕТСЯ ПЕРЕПОЛНЕНИЕ В СТАРШЕМ БИТЕ "H", - ЛЮБОЕ ПЕРЕПОЛНЕНИЕ В "L" АВТОМАТИЧЕСКИ

ПОМЕЩАЕТСЯ В РЕЗУЛЬТАТЕ ВЫЧИСЛЕНИЯ В "H",

СЛОЖЕНИЕ С ПЕРЕНОСОМ

ИЗ-ЗА ОГРАНИЧЕННОГО ХАРАКТЕРА РАБОТЫ С 16-БИТОВЫМИ ЧИСЛАМИ МЫ МОЖЕМ ВЫПОЛНЯТЬ СЛОЖЕНИЕ ЦЕПОЧКОЙ ТОЧНО ТАК ЖЕ ,КАК И В 8БИТОВОМ СЛУЧАЕ, КОМАДА "СЛОЖЕНИЕ С ПЕРЕНОСОМ" - ЕЕ МНЕМОНИЧЕСКОЕ ОБОЗНАЧЕНИЕ "ADC" (ADD WITH CARRY) - ДЕЙСТВУЕТ АНАЛОГИЧНО КОМАНДЕ "ADD" И ИМЕЕТ ТОТ ЖЕ ДИАПОЗОН ПАР РЕГИСТРОВ:

```
ADC HL,BC
ADC HL,DE
ADC HL,HL
ADC HL,SP
```

16-БИТОВОЕ ВЫЧИТАНИЕ

16-БИТОВОЕ ВЫЧИТАНИЕ - ТОЖЕ ДОСТАТОЧНО ПРОСТАЯ ОПЕРАЦИЯ, НО ВЫЧИТАНИЯ БЕЗ ПЕРЕНОСА НЕ БЫВАЕТ; ЕСЛИ ВЫ НЕ УВЕРЕНЫ В СОСТОЯНИИ ФЛАГА ПЕРЕНОСА, УДОСТОВЕРТЕСЬ, ЧТО В ВАШЕЙ ПРОГРАММЕ ИМЕЕТСЯ СТРОКА, ОЧИЩАЮЩАЯ ФЛАГ ПЕРНОСА ПЕРЕД КАЖДОЙ ОПЕРАЦИЕЙ ВЫЧИТАНИ

```
SBC HL,BC
SBC HL,DE
SBC HL,HL
SBC HL,SP
```

ВОЗДЕИСТВИЕ АРИФМЕТИЧЕСКИХ ОПЕРАЦИИ С ПЕРЕНОСОМ НА ФЛАГИ

ВЫ,ВОЗМОЖНО, ЗАМЕТИЛИ, ЧТО ТРИ ДРУГИХ ФЛАГА ИЗМЕНЯЮТСЯ В РЕЗУЛЬТАТЕ КОМАНД "СЛОЖЕНИЕ С ПЕРЕНОСОМ" И "ВЫЧИТАНИЕ С ПЕРЕНОСОМ", ХОТЯ ОНИ НЕ ИЗМЕНЯЛИСЬ В РЕЗУЛЬТАТЕ ПРОСТЫХ КОМАНД 16-БИТОВОГО СЛОЖЕНИЯ,

ЭТИ ФЛАГИ - ФЛАГ НУЛЯ, ФЛАГ ЗНАКА И ФЛАГ ПЕРЕПОЛНЕНИЯ, КАЖДЫЙ ИЗ НИХ УСТАНАВЛИВАЕТСЯ В СООТВЕТСТВИИ С РЕЗУЛЬТАТОМ ОПЕРАЦИИ,

АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ С ИНДЕКСНЫМ РЕГИСТРОМ

ИНДЕКСНЫЕ РЕГИСТРЫ ОГРАНИЧИВАЮТСЯ ТОЛЬКО ОПЕРАЦИЕЙ СЛОЖЕНИЯ БЕЗ ПЕРЕНОСА!

БОЛЕЕ ТОГО, ДИАПОЗОН РЕГИСТРОВ, КОТОРЫЕ МОЖНО СКЛАДВАТЬ С ИНДЕКСНЫМИ РЕГИСТРАМИ ,ЧЕРЕЗВЫЧАЙНО ОГРАНИЧЕН:

СЛОЖЕНИЕ С ПАРОЙ РЕГИСТРОВ "BC" ИЛИ "DE";

СЛОЖЕНИЕ ИНДЕКСНОГО РЕГИСТРА С САМИМ СОБОЙ;

СЛОЖЕНИЕ С УКАЗАТЕЛЕМ СТЕКА,

УПРАЖНЕНИЕ НА РЕШЕНИЕ ЗАДАЧИ О СВОБОДНОЙ ПАМЯТИ

КОНЕЦ ПРОСТРАНСТВА ПАМЯТИ, ИСПОЛЬЗУЕМОГО ПРОГРАММОЙ ,ОПРЕДЕЛЯЕТСЯ СОДЕРЖИМЫМ ЯЧЕЙКИ ПАМЯТИ STKEND, В РУКОВОДСТВЕ ПО "СПЕКТРУМ" ОНА ОПРЕДЕЛЯЕТСЯ АДРЕСАМИ 23653 И 23564,

ОЧЕВИДНО,ЕСЛИ МЫ ЗАГРУЗИМ СОДЕРЖИМОЕ ЭТОЙ ЯЧЕЙКИ В HL, ТО ЗАДАЧА НАПОЛОВИНУ БУДЕТ РЕШЕНА!

LD HL,(STKEND) ЗАТЕМ НУЖНО ВЫЧЕСТЬ "УКАЗАТЕЛЬ СТЕКА" (SBC HL,SP?),

ИЗ-ЗА "ПЕРЕНОСА" НАМ НЕОБХОДИМО ОЧИСТИТЬ ФЛАГ ПЕРЕНОСА , НАИБОЛЕЕ ЛЕГКО ЭТО ДОСТИГАЕТСЯ С ПОМОЩЬЮ КОМАНДЫ "AND A", РАССМОТРЕННОЙ В ЭТОЙ КНИГЕ РАНЬШЕ (СТР,77),

```
AND A
SBC HL,SP
```

МОЖЕТЕ СЧИТАТЬ ЗАДАЧУ РЕШЕННОЙ НА ТРИ ЧЕТВЕРТИ, ЕСЛИ ВЫ ЗНАЛИ, ЧТО НУЖНО УЧЕСТЬ ПЕРЕНОС, НО НЕ ЗНАЛИ, КАК ЭТО СДЕЛАТЬ, ЕСЛИ ВЫ ВОВСЕ ЗАБЫЛИ О ПЕРЕНОСЕ, ЗАДАЧА РЕШЕНА НА ЧЕТВЕРТЬ,

ПОСКОЛЬКУ, УКАЗАТЕЛЬ СТЕКА ЗАДАЕТ ПОЛОЖЕНИЕ В ПАМЯТИ ВЫШЕ, ЧЕМ НАИБОЛЬШИЙ АДРЕС ВАШЕЙ ПРОГРАММЫ (ИНАЧЕ У ВАС ВОЗНИКЛИ БЫ СЕРЬЕЗНЫЕ ТРУДНОСТИ), РЕЗУЛЬТАТ БУДЕТ ОТРИЦАТЕЛЬНЫМ,

ТЕПЕРЬ ДАВАЙТЕ ПОЛУЧИМ КОЛИЧЕСТВО БАЙТОВ, ОСТАВШИХСЯ СВОБОДНЫМИ, В ВИДЕ ПОЛОЖИТЕЛЬНОГО ЧИСЛА С ПОМОЩЬЮ РЕГИСТРА "BC" (ТОЧНО ТАК ЖЕ ДЛЯ ЭТОГО ПОДОШЕЛ БЫ И РЕГИСТР "DE"), СНАЧАЛА МЫ ХОТИМ СДВИНУТЬ HL И BC, НО НЕТ КОМАНДЫ "ЗАГРУЗИТЬ" ДЛЯ ТАКОГО ДЕИСТВИЯ НАМ ПРИДЕТСЯ ИСПОЛЬЗОВАТЬ ПОСЛЕДОВАТЕЛЬНО КОМАНДЫ PUSH И POP:

```
PUSH HL
POP BC
```

В HL ПО-ПРЕЖНЕМУ НАХОДИТСЯ ТА ЖЕ ИНФОРМАЦИЯ,ТАК ЧТО HL=BC, ЧТОБЫ ПОЛУЧИТЬ HL = -BC ,НУЖНО ВЫЧЕСТЬ BC ИЗ HL ДВАЖДЫ (НО

МАНДОЙ ВЫЧИТАНИЯ, ТАК ЧТО ЕГО НЕОБХОДИМО ВНОВЬ ОЧИСТИТЬ):

```
AND A
SBC HL,BC
SBC HL,BC
```

В HL ТЕПЕРЬ СОДЕРЖИТСЯ ОТРИЦАТЕЛЬНОЕ ЗНАЧЕНИЕ, ПО МОДУЛЮ РАВНОЕ ПРЕЖНЕМУ,Т,Е, ПОЛОЖИТЕЛЬНОЕ ЧИСЛО ОСТАВШИХСЯ СВОБОДНЫМИ БАЙТОВ,

ТЕПЕРЬ НАМ НУЖНО ПОЛУЧИТЬ ЭТО ЧИСЛО СНОВА В ПАРЕ РЕГИСТРОВ BC, ЧТОБЫ ПОЛУЧИТЬ РЕЗУЛЬТАТ ИЗ ФУНКЦИИ "USR", ЧТОБЫ ПОЛУЧИТЬ HL НАЗАД В BC:

```
PUSH HL
POP  BS
```

И, НАКОНЕЦ , ВОЗВРАТ ИЗ ФУНКЦИИ USR:

```
RET
```

ПРАВИЛЬНО ЛИ ВЫ НАПИСАЛИ ПРОГРАММУ? ОБРАТИЛИ ЛИ ВЫ ВНИМАНИЕ, КАК УДОБНО ПОЛЬЗОВАТЬСЯ СТЕКОМ!

Ц И К Л Ы И П Е Р Е Х О Д Ы

В "БЕЙСИКЕ" ВЫ ЗНАЕТЕ КОМАНДУ "GO TO", ПЕРЕДАЮЩУЮ УПРАВЛЕНИ ВАШЕЙ ПРОГРАММОЙ КОМАНДАМ В ТОЙ СТРОЧКЕ, НА КОТОРУЮ УКАЗЫВАЕТ "GO TO",

НЕТ НИЧЕГО ПРОЩЕ, ЧЕМ РЕАЛИЗОВАТЬ ЭТО НА МАШИННОМ ЯЗЫКЕ, ПРОСТО ЗАДАЙТЕ ЯЧЕЙКУ ПАМЯТИ, В КОТОРОЙ ВЫ ХОТЕЛИ БЫ, ЧТОБЫ ЦП ОБНАРУЖИЛ СЛЕДУЮЩУЮ КОМАНДУ, И ЗАДАЧА НАПОЛОВИНУ РЕШЕНА, НАИБОЛЕЕ ПРОСТОЙ ВАРИАНТ - КОМАНДА "ПЕРЕЙТИ НА" (JUMP TO)

```
JP XX XX
JP (HL)
JP (IX)
JP (IY)
```

МОЖНО СДЕЛАТЬ ТАК, ЧТОБЫ ОДНА ИЗ ЭТИХ КОМАНД ЗАВИСЕЛА ОТ СОСТОЯНИЯ ОДНОГО ИЗ ФЛАГОВ, НАПРИМЕР ФЛАГА ПЕРНОСА, ЭТО КОМАНДА УСЛОВНОГО ПЕРЕХОДА ИМЕЕТ ВИД:

```
JP CC,NN
```

ГДЕ CC- УСЛОВИЕ, ВЫПОЛНЕНИЕ КОТОРОГО ПРОВЕРЯЕТСЯ, ЕСЛИ БЫ У НАС БЫЛО, НАПРИМЕР,

```
JP Z,0000
```

ТО ЭТО ЧИТАЛОСЬ БЫ "ПЕРЕХОД ПО АДРЕСУ 0000, ЕСЛИ ФЛАГ НУЛЯ УСТАНОВЛЕН", (ЭТО - АДРЕС, ПО КОТОРОМУ "СПЕКТРУМ" ОСУЩЕСТВЛЯЕТ ПЕРЕХОД ПРИ ВКЛЮЧЕНИИ ПИТАНИЯ, И В ТАКОМ ВИДЕ КОМАНДА "JP" НА НОЛЬ МОЖЕТ ПРИМЕНЯТЬСЯ В ПРОГРАММЕ НА МАШИННОМ ЯЗЫКЕ, ЕСЛИ ВЫ ЗАХОТЕЛИ ОЧИСТИТЬ ВСЮ ПАМЯТЬ И НАЧАТЬ ЗАНОВО С ПОМОЩЬЮ "K")

ТЕПЕРЬ ОБРАТИТЕ ВНИМАНИЕ,ЧТО ЦП НЕ ДОПУСКАЕТ НИКАКИХ ОШИБОК ЕСЛИ ВЫ ГОВОРИТЕ "JUMP", ОН СДЕЛАЕТ ПЕРЕХОД, ПОСКОЛЬКУ ПОЧТИ ЛЮБОЙ КОД МОЖЕТ БЫТЬ ВОСПРИНЯТ КАК КОМАНДА,ЦП НЕ ПРИНИМАЕТ ВО ВНИМАНИЕ, ЧТО ВЫ МОГЛИ СДЕЛАТЬ ПЕРЕХОД В СЕРЕДИНУ МАССИВА ДАННЫХ ИЛИ ВО ВТОРОЙ БАЙТ ДВУХБАЙТОВОЙ КОМАНДЫ: ОН БУДЕТ СЧИТЫВАТ БАЙТ ПО НАЙДЕННМУ АДРЕСУ И СЧИТАТЬ ,ЧТО ЭТО - НАЧАЛО СЛЕДУЮЩЕЙ КОМАНДЫ,

СПОСОБ, КОТОРЫМ ЦП ОБРАБАТЫВАЕТ КОМАНДУ ПЕРЕХОДА НА САМОМ ДЕЛЕ СОВЕРШЕННО ПРОСТ: У НЕГО ЕСТЬ НЕБОЛЬШОЙ СЧЕТЧИК, НАЗЫВАЕМЫЙ СЧЕТЧИКОМ КОМАНД, ГОВОРЯЩИЙ ЕМУ, ГДЕ ИСКАТЬ СЛЕДУЮЩУЮ ВЫПОЛНЯЕМУЮ КОМАНДУ, ПРИ НОРМАЛЬНОМ ХОДЕ ПРОГРАММИРОВАНИЯ (Т,Е, БЕЗ ПЕРЕХОДОВ) ЦП ПРОВЕРЯЕТ ПОДЛЕЖАЩУЮ ВЫПОЛНЕНИЮ КОМАНДУ И ДОБАВЛЯЕТ К СЧЕТЧИКУ КОМАНД СТОЛЬКО, СКОЛЬКО БАЙТОВ СОДЕРЖИТ КОМАНДА,

ТАК, ЕСЛИ ОН ВСТРЕЧАЕТ 2-БАЙТОВУЮ КОМАНДУ, ОН ДОБАВЛЯЕТ 2, ТОГДА КАК 4-БАЙТОВАЯ КОМАНДА ЗАСТАВИТ ЕГО ДОБАВИТЬ 4 К СЧЕТЧИКУ КОМАНД,

КОГДА ОН ВСТРЕЧАЕТ КОМАНДУ "ПЕРЕХОДА" ,ОН ПРОСТО ЗАМЕЩАЕТ СОДЕРЖИМОЕ СЧЕТЧИКА КОМАНД ТЕМ ЗНАЧЕНИЕМ, КОТОРОЕ ВЫ УКАЗАЛИ,

ИМЕННО ПОЭТОМУ ВЫ НЕ МОЖЕТЕ ДОПУСТИТЬ ПРОНИКНОВЕНИЯ В ПРОГРАММУ КАКИХ-ЛИБО ОШИБОК,

ДЛИННЫЕ ПЕРЕХОДЫ И КОРОТКИЕ ПЕРЕХОДЫ

МЫ МОЖЕМ СЧИТАТЬ ПРИВЕДЕННЫЕ ВЫШЕ КОМАНДЫ ЭКВИВАЛЕНТОМ ДЛИННОГО ПЕРЕХОДА В МАШИННОМ ЯЗЫКЕ, ПОСКОЛЬКУ 16-БИТОВЫЙ

...ЬЬЬ ЕХ SPECTRUM" NOT COPY

АДРЕС ПОЗ ВОЛЯЕТ НАМ ОСУЩЕСТВИТЬ ПЕРЕХОД В ЛЮБОЕ МЕСТО ,КУДА ПОЗВОЛЯЕТ ПЕРЕЙТИ ЧИП Z80,

НЕДОСТАТКИ ДЛИННОГО ПЕРЕХОДА:

А) ЧАСТО НАМ НЕТ НЕОБХОДИМОСТИ В ТАКОМ ДЛИННОМ ПЕРЕХОДЕ, НО НАМ ВСЕ-ТАКИ ПРИХОДИТСЯ ПРИМЕНЯТЬ 3-БАЙТОВУЮ КОМАНДУ,

В) МЫ НЕ МОЖЕМ БЕЗ ЗАТРУДНЕНИЙ ПЕРЕМЕСТИТЬ ПРОГРАММУ В ДРУГУЮ ЧАСТЬ ПАМЯТИ, ПОТОМУ ЧТО МЫ ЗАДАЕМ АБСОЛЮТНЫЙ АДРЕС,

ИМЕННО ДЛЯ ПРЕОДОЛЕНИЯ ЭТИХ ДВУХ НЕДОСТАТКОВ БЫЛ ВВЕДЕН "КОРОТКИЙ ПЕРЕХОД", ОН НАЗЫВАЕТСЯ "ОТНОСИТЕЛЬНЫМ ПЕРЕХОДОМ" И ПОЗВОЛЯЕТ НАМ СОВЕРШАТЬ ПЕРЕХОД ОТ ТЕКУЩЕЙ ПОЗИЦИИ НА +127 ИЛИ -128 БАЙТОВ, Т,Е, РАССТОЯНИЕ ПЕРЕХОДА МОЖНО ЗАДАТЬ В ОДНОМ БАЙТЕ!

КОМАНДА ОТНОСИТЕЛЬНОГО ПЕРЕХОДА

JP Д, ГДЕ Д - ОТНОСИТЕЛЬНОЕ СМЕЩЕНИЕ,

МЫ МОЖЕМ ТАКЖЕ СДЕЛАТЬ ОТНОСИТЕЛЬНЫЙ ПЕРЕХОД В ЗАВИСИМОСТИ ОТ НЕКОТОРОГО УСЛОВИЯ, НАПРИМЕР, ОТ ТОГО, УСТАНОВЛЕН ЛИ ФЛАГ ПЕРЕНОСА ИЛИ ФЛАГ НУЛЯ, ЭТИ УСЛОВНЫЕ ПЕРЕХОДЫ ЗАПИСЫВАЮТСЯ СЛЕДУЮЩИМ ОБРАЗОМ:

JP СС,Д , ГДЕ СС - ПРОВЕРЯЕМОЕ УСЛОВИЕ,

ЗНАЧЕНИЕ СМЕЩЕНИЯ Д ДОБАВЛЯЕТСЯ К СЧЕТЧИКУ КОМАНД,

ЭТО ОЗНАЧАЕТ, ЧТО БЕРЕТСЯ ТЕКУЩЕЕ ЗНАЧЕНИЕ СЧЕТЧИКА КОМАНД И К НЕМУ ПРИБАЛЯЕТСЯ ЗАДАННОЕ ВАМИ ОТНОСИТЕЛЬНОЕ ЗНАЧЕНИЕ, ВЫ МОЖЕТЕ ЗАДАВАТЬ ЛИБО ПОЛОЖИТЕЛЬНОЕ (ПЕРЕХОД ВПЕРЕД), ЛИБО ОТРИЦАТЕЛЬНОЕ (ПЕРЕХОД НАЗАД) ЗНАЧЕНИЕ, ЕСЛИ ВЫ ПОСМОТРИТЕ, В ПРЕДШЕСТВУЮЩЕЙ ГЛАВЕ ОБ ОТРИЦАТЕЛЬНЫХ ЧИСЛАХ, ТО ВЫ ПОЙМЕТЕ, ЧТО ЭТО ОЗНАЧАЕТ ОГРАНИЧЕНИЕ ОТНОСИТЕЛЬНЫХ ПЕРЕХОДОВ, ДИАПАЗОНОМ ОТ -128 ДО +127,

ОБРАТИТЕ ВНИМАНИЕ, ЧТО КОГДА ЦП ВЫПОЛНЯЕТ КОМАНДУ ОТНОСИТЕЛЬНОГО ПЕРЕХОДА, СЧЕТЧИК КОМАНД УЖЕ УКАЗЫВАЕТ НА СЛЕДУЮЩУЮ КОМАНДУ, КОТОРАЯ ВЫПОЛНЯЛАСЬ БЫ ,ЕСЛИ УСЛОВИЕ НЕ УДОВЛЕТВОРЯЛОСЬ,

ТАК ПРОИСХОДИТ ПОТОМУ, ЧТО КОГДА ЦП ВСТРЕЧАЕТ "JP", ОН ЗНАЕТ, ЧТО ИМЕЕТ ДЕЛО С 2-БАЙТОВОЙ КОМАНДОЙ ,И ДОБАВЛЯЕТ 2 К СЧЕТЧИКУ КОМАНД - ПОЭТОМУ СЧЕТЧИК КОМАНД УКАЗЫВАЕТ НА КОМАНДУ, СЛЕДУЮЩУЮ ЗА ОТНОСИТЕЛЬНЫМ ПЕРЕХОДОМ!

ПРИМЕР, В ТАКОЙ ПРОГРАММЕ ,КАК

```
ЯЧЕЙКА              ТЕКСТ
 32000              ADD A,B
 32001               JP Z,02H
 32003               LD B,0
 32005    NEXT       LD HL,4000H
```

НИЖЕ ПОКАЗАНО, КАК ЦП РАБОТЕТ С ЭТОЙ ПРОГРАММОЙ, ЕСЛИ ИГНОРИРУЕТСЯ КОМАНДА ПЕРЕХОДА, РАСПОЛОЖЕННАЯ ПО АДРЕСУ 32001 (Т, Е, ПРИ СБРОШЕННОМ ФЛАГЕ НУЛЯ):

ЗАГРУЗИТЬ БАЙТ ПО АДРЕСУ 32000 ПОСКОЛЬКУ В ЭТОМ БАЙТЕ СОДЕРЖИТСЯ КОМАНДА, СОСТОЯЩАЯ ТОЛЬКО ИЗ ОДНОГО БАЙТА,СЧЕТЧИК КОМАНД НУЖНО ЗАДАТЬ РАВНЫМ 32001,

ВЫПОЛНИТЬ КОМАНДУ,

ЗАГРУЗИТЬ БАЙТ, ЗАДАННЫЙ СЧЕТЧИКОМ КОМАНД (32001), ЭТОТ БАЙТ БУДЕТ ЯВЛЯТЬСЯ ЧАСТЬЮ 2-БАЙТОВОЙ КОМАНДЫ, ТАК ЧТО НУЖНО ПРИБАВИТЬ 2 К СЧЕТЧИКУ КОМАНД И СДЕЛАТЬ ЕГО РАВНЫМ 32003,

ПОЛУЧИТЬ СЛЕДУЮЩИЙ БАЙТ, ЧТОБЫ СДЕЛАТЬ КОМАНДУ ПОЛНОЙ,

ВЫПОЛНИТЬ КОМАНДУ,

ЗАГРУЗИТЬ БАЙТ, ЗАДАННЫЙ СЧЕТЧИКОМ КОМАНД (32003), ЭТОТ БАЙТ ЯВЛЯЕТСЯ ЧАСТЬЮ 2-БАЙТОВОЙ КОМАНДЫ,ТАК ЧТО НУЖНО ПРИБАВИТЬ К СЧЕТЧИКУ КОМАНД 2 (ТЕПЕРЬ ОН СТАНОВИТСЯ РАВНЫМ 32005)

ПОЛУЧИТЬ СЛЕДУЮЩИЙ БАЙТ, ЧТОБЫ СДЕЛАТЬ КОМАНДУ ПОЛНОЙ,

ВЫПОЛНИТЬ КОМАНДУ,

В ЯЧЕЙКЕ 32001 ПРОГРАММА ВСТРЕЧАЕТ КОМАНДУ ОТНОСИТЕЛЬНОГО ПЕРЕХОДА, ЕСЛИ ФЛАГ НУЛЯ НЕ УСТАНОВЛЕН, КАК В ПРИВЕДЕННОМ ВЫШЕ ПРИМЕРЕ, ТО ЦП НИЧЕГО НЕ ВЫПОЛНЯЕТ, В ОБЩЕМ СЛУЧАЕ ЦП ВЫПОЛНЯЕТ КОМАНДУ ПЕРЕХОДА СЛЕДУЮЩИМ ОБРАЗОМ:

ЕСЛИ ФЛАГ НУЛЯ УСТАНОВЛЕН, ПРИБАВИТЬ 2 К СЧЕТЧИКУ КОМАНД (ЭТО СОСТАВИТ 32005),

ЕСЛИ ФЛАГ НУЛЯ СБРОШЕН, НЕ НУЖНО ДЕЛАТЬ НИЧЕГО (СЧЕТЧИК КОМАНД ОСТАЕТСЯ РАВНЫМ 32003),

ИНЫМИ СЛОВАМИ,ОТНОСИТЕЛЬНЫЙ ПЕРЕХОД ПОЗВОЛЯЕТ НАМ В ОПРЕДЕ-

ЛЕННЫХ СЛУЧАЯХ ПЕРЕСКАКИВАТЬ ЧЕРЕЗ КОМАНДУ "LD B,Ø",

ЭТО ТАКЖЕ ОБ"ЯСНЯЕТ, ПОЧЕМУ ДЛЯ ЭТОЙ КОМАНДЫ УКАЗАНЫ ДВА ВРЕМЕНИ ВЫПОЛНЕНИЯ. ПОДСЧЕТ НОВОГО ЗНАЧЕНИЯ СЧЕТЧИКА КОМАНД ЗА НИМАЕТ БОЛЬШЕ ВРЕМЕНИ, ЧЕМ НЕВЫПОЛНЕНИЕ НИКАКИХ ДЕЙСТВИЙ,

ПОЭТОМУ ЦП ВЫПОЛНИТ ЛИБО КОМАНДУ, РАСПОЛОЖЕННУЮ ПО АДРЕСУ 32ØØ3, ЛИБО - ПО АДРЕСУ 32ØØ5, В ЗАВИСИМОСТИ ОТ ЗНАЧЕНИЯ ФЛАГА НУЛЯ,

КАК МЫ УЖЕ ГОВОРИЛИ, МОЖНО ТАКЖЕ СДЕЛАТЬ ОТНОСИТЕЛЬНЫЙ ПЕРЕХОД С ОТРИЦАТЕЛЬНЫМ ЗНАЧЕНИЕМ,

ЦИКЛЫ ОЖИДАНИЯ

В ПРОГРАММАХ НА МАШИННОМ ЯЗЫКЕ БЫВАЮТ СИТУАЦИИ, КОГДА СОБЫТИЯ ПРОИСХОДЯТ С ТАКОЙ БОЛЬШОЙ СКОРОСТЬЮ, ЧТО НЕОБХОДИМО ВВЕСТИ НЕКОТОРОЕ ОЖИДАНИЕ,

ПРИМЕРЫ, СРАЗУ ПРИХОДЯЩИЕ НА УМ - ПОСЫЛКА ИНФОРМАЦИИ НА КАССЕТУ МАГНИТОФОНА (СИГНАЛЫ ДОЛЖНЫ БЫТЬ РАСПОЛОЖЕНЫ ДОСТАТОЧНО ДАЛЕКО ДРУГ ОТ ДРУГА, ЧТОБЫ ИХ ПОТОМ МОЖНО БЫЛО ПРОЧИТАТЬ) ИЛИ ПОСЫЛКА ИНФОРМАЦИИ НА ПЕЧАТАЮЩУЮ МАШИНКУ (ПРЕДСТАВЬТЕ СЕБЕ ТОЛЬКО ПЕЧАТЬ СО СКОРОСТЬЮ НЕСКОЛЬКИХ ТЫСЯЧ ЗНАКОВ В СЕКУНДУ),

ПОЭТОМУ ПОЛЕЗНО УСТАНОВИТЬ ЦИКЛЫ ОЖИДАНИЯ С ПОМОЩЬЮ КОМАНДЫ DJNZ:

```
        LD B, COUNT
WAIT    DJNZ WAIT
```

COUNT - СЧЕТЧИК; WAIT - ОЖИДАНИЕ

КОМАНДА "DJNZ WAIT" ЗАСТАВИТ ЦП ВОЗВРАТИТЬСЯ К КОМАНДЕ DJNZ СТОЛЬКО РАЗ, СКОЛЬКО НУЖНО, ЧТОБЫ РЕГИСТР "B" ВНОВЬ ОБНУЛИЛСЯ, ПРЕЖДЕ ЧЕМ ОБРАБОТКА ПОЙДЕТ ДАЛЬШЕ;

ЭТО ДОЛЖНО ВАМ ДАТЬ ОТВЕТ НА НАШЕ УПРАЖНЕНИЕ, В КОТОРОМ СПРАШИВАЛОСЬ, ЧТО ПРОИЗОЙДЕТ, ЕСЛИ ВЫ НАПИШЕТЕ

```
WAIT JR WAIT
```

ВАМ ПРИДЕТСЯ ДОВОЛЬНО ДОЛГО ЖДАТЬ ,ПОКА ЦП ВЫЙДЕТ ИЗ ЭТОГО ЦИКЛА!

КОМАНДЫ ГРУППЫ ВЫЗОВА И ВОЗВРАТА

MNEMONIC	BYTES	TIME TAKEN	EFFEKT ON FLAGS C	Z	PV	S	N	H
CALL ADDRES	3	17	-	-	-	-	-	-
CALL CC,ADDRESS	3	1Ø/17	-	-	-	-	-	-
RET	1	1Ø	-	-	-	-	-	-
RET CC	1	5/11	-	-	-	-	-	-

MNEMONIC -МНЕМОНИЧЕСКОЕ ОБОЗНАЧЕНИЕ; BYTES -КОЛИЧЕСТВО БАЙТОВ; TIME TAKEN - ВРЕМЯ ВЫПОЛНЕНИЯ; EFFEST ON FLAGS - ВОЗДЕЙСТВИЕ НА ФЛАГИ; ADDRESS - АДРЕС ,

ЗАМЕЧАНИЕ: CC - УСЛОВИЕ, КОТОРОЕ ДОЛЖНО ВЫПОЛНЯТЬСЯ ДЛЯ ИСПОЛНЕНИЯ КОМАНДЫ, НИЖЕ ПРИВОДЯТСЯ ИСПОЛЬЗУЕМЫЕ УСЛОВИЯ:

ФЛАГ	СОКРАЩЕНИЕ	ЗНАЧЕНИЕ
ПЕРЕНОС	C	УСТАНОВКА ФЛАГА ПЕРЕНОСА(=1)
	NC	СБРОС ФЛАГА ПЕРЕНОСА (=Ø)
НОЛЬ	Z	УСТАНОВКА ФЛАГА НУЛЯ (=1)
	NZ	СБРОС ФЛАГА НУЛЯ (=Ø)
ЧЕТНОСТЬ	PE	ФЛАГ ЧЕТНОСТИ ЧЕТНЫЙ (=1)
	PO	ФЛАГ ЧЕТНОСТИ НЕЧЕТНЫЙ (=Ø)
ЗНАК	M	ЗНАК МИНУС (=1)
	P	ЗНАК ПОЛОЖИТЕЛЬНЫЙ (=Ø)

ВОЗДЕЙСТВИЕ НА ФЛАГИ: ОБРАТИТЕ ВНИМАНИЕ, ЧТО НИ ОДИН ИЗ ФЛАГОВ НЕ ИЗМЕНЯЕТСЯ ПРИ КОМАНДАХ ВЫЗОВА И ВОЗВРАТА,

ВРЕМЯ ВЫПОЛНЕНИЯ: ЕСЛИ УКАЗАНЫ ДВА ЗНАЧЕНИЯ ВРЕМЕНИ, ТО БОЛЕЕ КОРОТКОЕ ОТНОСИТСЯ К СЛУЧАЮ, КОГДА УСЛОВИЕ НЕ ВЫПОЛНЯЕТСЯ,

П Р И М Е Н Е Н И Е П О Д П Р О Г Р А М М В В А Ш И Х П Р О Г Р А М М А Х Н А М А Ш И Н Н О М Я З Ы К Е

ПРИМЕНЕНИЕ ПОДПРОГРАММ В МАШИННОМ ЯЗЫКЕ СТОЛЬ ЖЕ ПРОСТО, КАК В ОБЫЧНЫХ ПРОГРАММАХ НА ЯЗЫКЕ "БЕЙСИК", ЕСЛИ НЕ ПРОЩЕ,

НА САМОМ ДЕЛЕ, ВЫ ПОМНИТЕ, ЧТО ПРИМЕНЯЯ ФУНКЦИЮ "USR" В

"ZX SPECTRUM" NOT COPY

СВОЕЙ ПРОГРАММЕ НА ЯЗЫКЕ "БЕЙСИК" ,ВЫ В ДЕЙСТВИТЕЛЬНОСТИ ВЫЗЫВАЕТЕ ПОДПРОГРАММУ , И, КАК ВЫ ПОМНИТЕ, ДЛЯ ЗАВЕРШЕНИЯ НАМ НУЖНА КОМАНДА "RETURN"!

ПОЭТОМУ ВАМ ОЧЕНЬ ПРОСТО ПРОВЕРИТЬ ОПРЕДЕЛЕННЫЕ ПОДПРОГРАММЫ НЕЗАВИСИМО ОТ ВАШЕЙ ОСНОВНОЙ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ,

ОСНОВНОЕ РАЗЛИЧИЕ, С КОТОРЫМ ВАМ ПРИДЕТСЯ СТОЛКНУТЬСЯ ПРИ РЕАЛИЗАЦИИ ПОДПРОГРАММ В ВАШИХ ПРОГРАММАХ НА МАШИННОМ ЯЗЫКЕ, СОСТОИТ В ТОМ , ЧТО ВАМ НУЖНО ЗНАТЬ АДРЕС НАЧАЛА ПОДПРОГРАММЫ,

ЭТО МОЖЕТ ВЫЗВАТЬ ТРУДНОСТИ, ЕСЛИ ВЫ ХРАНИТЕ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ В ПЕРЕМЕННОМ МАССИВЕ, ПОСКОЛЬКУ АДРЕС ЭТОЙ ПЕРЕМЕННОЙ НЕ ОБЯЗАТЕЛЬНО ФИКСИРОВАН, ЭТО ТАКЖЕ ОЗНАЧАЕТ, ЧТО ПРОГРАММА НА МАШИННОМ ЯЗЫКЕ НЕ МОЖЕТ БЫТЬ ЛЕГКО ПЕРЕМЕЩЕНА В НОВУЮ ПОЗИЦИЮ ПАМЯТИ, ЕСЛИ В НЕЙ ИСПОЛЬЗУЮТСЯ ПОДПРОГРАММЫ,

ПОДПРОГРАММЫ ТАКЖЕ МОЖНО ВЫЗЫВАТЬ В ЗАВИСИМОСТИ ОТ УСЛОВИЙ, ЭТО - ЭКВИВАЛЕНТ НА МАШИННОМ ЯЗЫКЕ СЛЕДУЮЩЕГО ПРЕДЛОЖЕНИЯ ЯЗЫКА "БЕЙСИК":

IF (CONDITION) THEN GOSUB (LINE) CONDITION - УСЛОВИЕ; LINE - СТРОКА,

НАХОДЯСЬ В ПОДПРОГРАММЕ, НУЖНО ПРОЯВЛЯТЬ ОСТОРОЖНОСТЬ, ЧТОБЫ НЕ ИЗМЕНИТЬ КАКИЕ-ЛИБО ФЛАГИ ИЛИ РЕГИСТРЫ, ТРЕБУЮЩИЕСЯ ДЛЯ ДАЛЬНЕЙШИХ СРАВНЕНИЙ, ЭТО НЕОБХОДИМО ДЛЯ ТОГО, ЧТОБЫ ВЫ НЕ УШЛИ НА ВЕТВЬ АЛГОРИТМА СНОВА ПО СЛЕДУЮЩЕМУ ПРЕДЛОЖЕНИЮ "CALL" "ПОСЛЕ ВОЗВРАТА ТУДА,ОТКУДА ВЫШЛИ",

РАЗЛИЧИЕ СОСТОИТ В ТОМ, ЧТО ЕДИНСТВЕННЫМИ ДОПУСТИМЫМИ УСЛОВИЯМИ ЯВЛЯЮТСЯ СОСТОЯНИЯ ЧЕТЫРЕХ ФЛАГОВ:

ФЛАГ ПЕРЕНОСА;
ФЛАГ НУЛЯ;
ФЛАГ ЧЕТНОСТИ (А ТАКЖЕ ФЛАГ ПЕРЕПОЛНЕНИЯ)
ФЛАГ ЗНАКА,

НАПОМНИМ, ЧТО ВСЕ ЭТИ ФЛАГИ УСТАНАВЛИВАЮТСЯ В СООТВЕТСТВИИ С ПОСЛЕДНЕЙ КОМАНДОЙ, ПОВЛИЯВШЕЙ НА ЭТОТ КОНКРЕТНЫЙ ФЛАГ,

ПОЭТОМУ ХОРОШИЙ СТИЛЬ ПРОГРАММИРОВАНИЯ ПРЕДПОЛАГАЕТ ПОМЕЩЕНИЕ КОМАНДЫ "CALL" ИЛИ "RETURN" НЕПОСРЕДСТВЕННО ПОСЛЕ КОМАНДЫ ,УСТАНАВЛИВАЮЩЕЙ ФЛАГ,

НАПРИМЕР,

```
          LD A,(NUMBER)
          CP 1
        CALL Z,ONE
          CP 2
        CALL Z,TWO
          CP 3
        CALL Z,THREE
```

NUMBER - ЧИСЛО; ONE - ОДИН; TWO - ДВА; THREE - ТРИ,

ПРИВЕДЕННАЯ ВЫШЕ ПРОГРАММА ПОЗВОЛЯЕТ ВАМ ПЕРЕХОДИТЬ К РАЗЛИЧНЫМ ПРОГРАММАМ, В ЗАВИСИМОСТИ ОТ ЗНАЧЕНИЯ , ХРАНИМОГО В ЯЧЕЙКЕ "NUMBER" (ЧИСЛО), НО ОБРАТИТЕ ВНИМАНИЕ, ЧТО ОНА ПРЕДПОЛАГАЕТ ,ЧТО ПОДПРОГРАММЫ НЕ ИЗМЕНЯЮТ ЗНАЧЕНИЕ В РЕГИСТРЕ А!!

МОЖНО ПРИМЕНИТЬ И БОЛЕЕ КОРОТКУЮ ПРОГРАММУ ,ЕСЛИ ВАМ ИЗВЕСТНО, ЧТО ДЛЯ ХРАНИМОГО В "NUMBER" ЗНАЧЕНИЯ ИМЕЮТСЯ ТОЛЬКО ТРИ ЗАДАННЫЕ ВЫШЕ ВОЗМОЖНОСТИ:

```
          LD A,(NUMBER)
          CP 2
        CALL Z,TWO      : A = 2
        CALL C,ONE      : A (2 =) A =1
        CALL THREE      : A )2 =) A =3
```

NUMBER -ЧИСЛО; TWO - ДВА; ONE - ОДИН; THREE - ТРИ,

ТАК ПРОИСХОДИТ ПОТОМУ, ЧТО КОМАНДА "CP 2" УСТАНАВЛИВАЕТ И ФЛАГ НУЛЯ ,И ФЛАГ ПЕРНОСА, А КОМАНДА ВЫЗОВА НЕ ВЛИЯЮТ НА КАКИЕ БЫ ТО НИ БЫЛО ФЛАГИ,

АНАЛОГИЧНЫМ ОБРАЗОМ, ОЧЕНЬ ПОЛЕЗНЫМ ОКАЗЫВАЕТСЯ УСЛОВНЫЙ ВОЗВРАТ ИЗ ПОДПРОГРАММЫ ,(НО ЭТО НЕ СЧИТАЕТСЯ ХОРОШИМ СТИЛЕМ ПРОГРАММИРОВАНИЯ),

КОМАНДЫ ГРУППЫ СРАВНЕНИЯ И ПЕРЕМЕЩЕНИЯ БЛОКОВ

MNEMONIC	BYTES	TIME TAKEN	EFFECT ON FLAGS C	Z	PV	S	N	H
LDI	2	16	-	-	#	-	Ø	Ø
LDD	2	16	-	-	#	-	Ø	Ø
LDIR	2	21/16	-	-	Ø	-	Ø	Ø
LDDR	2	21/16	-	-	Ø	-	Ø	Ø
CPI	2	16	-	#	#	#	1	#
CPD	2	16	-	#	#	#	1	#
CPIR	2	21/16	-	#	#	#	1	#
CPDR	2	21/16	-	#	#	#	1	#

MNEMONIC -МНЕМОНИЧЕСКОЕ ОБОЗНАЧЕНИЕ;BYTES -КОЛИЧЕСТВО БАЙТОВ; TIME TAKEN -ВРЕМЯ ВЫПОЛНЕНИЯ; EFFECT ON FLAGS-ВОЗДЕЙСТВИЕ НА ФЛАГИ,

ОБОЗНАЧЕНИЕ ФЛАГОВ:

\# ПОКАЗЫВАЕТ, ЧТО ФЛАГ ИЗМЕНЯЕТСЯ В РЕЗУЛЬТАТЕ ОПЕРАЦИИ;

Ø ПОКАЗЫВАЕТ ,ЧТО ФЛАГ СБРАСЫВАЕТСЯ;

1 ПОКАЗЫВАЕТ, ЧТО ФЛАГ УСТАНАВЛИВАЕТСЯ;

\- ПОКАЗЫВАЕТ, ЧТО ФЛАГ НЕ МЕНЯЕТСЯ,

ВРЕМЯ ВЫПОЛНЕНИЯ;

ДЛЯ КОМАНД ПОВТОРЕНИЯ УКАЗАННОЕ ВРЕМЯ ОТНОСИТСЯ К КАЖДОМУ ЦИКЛУ, БОЛЕЕ КОРОТКОЕ ЗНАЧЕНИЕ ОТНОСИТСЯ К ПРЕРЫВАНИЮ КОМАНДЫ, НАПРИМЕР, ДЛЯ CPIR ЛИБО BC=Ø, ЛИБО A=(HL),

ОПЕРАЦИИ НАД БЛОКАМИ

В ЭТОЙ ГЛАВЕ РАССМАТРИВАЕТСЯ ПОСЛЕДНИЙ НАБОР ОЧЕНЬ ПОЛЕЗНЫХ ПРОГРАММ, В НЕСКОЛЬКИХ ПОСЛЕДУЮЩИХ ГЛАВАХ МЫ БУДЕМ ИМЕТЬ ДЕЛО С КОМАНДАМИ, КОТОРЫЕ ПРИЯТНО ИМЕТЬ ПОД РУКОЙ И КОТОРЫЕ ПРИ ОПРЕДЕЛЕННЫХ ОБСТОЯТЕЛЬСТВАХ ИГРАЮТ СВОЮ РОЛЬ, НО В ОБЩЕМ ВЫ ДОЛЖНЫ УМЕТЬ ПИСАТЬ МАШИННЫЕ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ С ПОМОЩЬЮ УЖЕ ИЗВЕСТНОГО ВАМ МАТЕРИАЛА,

ТЕМ НЕ МЕНЕЕ ,ОБЯЗАТЕЛЬНО ПРОЧТИТЕ ГЛАВУ О ПЛАНИРОВАНИИ ВАШЕЙ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ!

РАССМАТРИВАЕМЫЕ В ЭТОЙ ГЛАВЕ КОМАНДЫ ПО САМОЙ СВОЕЙ ПРИРОДЕ СПОСОБНЫ ОДНИМ МАХОМ ОХВАТЫВАТЬ ДЛИННЫЕ КОНСТРУКЦИИ БЫСТРЕЕ ЛЕТЯЩЕЙ ПУЛИ, ИНЫМИ СЛОВАМИ, КОМАНДЫ, ОБРАБАТЫВАЮЩИЕ ЦЕЛЫЕ БЛОКИ ПАМЯТИ, А НЕ ОТДЕЛЬНЫЕ 8-БИТОВЫЕ БАЙТЫ,

ДАВАЙТЕ НАЧНЕМ С ПРОСТЕЙШЕЙ ИЗ НИХ:

CPI ПРИ ВАШЕМ ЗНАНИИ ЯЗЫКА Z8Ø, ВЫ ДОЛЖНЫ СРАЗУ РАСПОЗНАТЬ ОДНОГО ИЗ ПРЕДСТАВИТЕЛЕЙ СЕМЕЙСТВА "СРАВНЕНИЙ", И НА САМОМ ДЕЛЕ ЭТО И ЕСТЬ РАСШИРЕННОЕ СРАВНЕНИЕ,

ПО-РУССКИ ОНА ЧИТАЕТСЯ "СРАВНИТЬ И ДАТЬ ПРИРАЩЕНИЕ", (ВЫ ПОМНИТЕ, ЧТО СРАВНИВАТЬ ЧТО БЫ ТО НИ БЫЛО МОЖНО ТОЛЬКО С СОДЕРЖИМЫМ РЕГИСТРА "A",И ОБ ЭТОМ В КОМАНДЕ НЕ ТРЕБУЕТСЯ УПОМИНАТЬ),

КОМАНДА "CPI" СРАВНИВАЕТ "A" С (HL) И АВТОМАТИЧЕСКИ УВЕЛИЧИВАЕТ HL, ЭТО ОЗНАЧАЕТ, ЧТО ПОСЛЕ ОПЕРАЦИИ CPI HL УЖЕ УКАЗЫВАЕТ НА СЛЕДУЮЩУЮ ЯЧЕЙКУ И ГОТОВ К ПОВТОРЕНИЮ ЕЕ,

С ПОМОЩЬЮ ЭТОЙ КОМАНДЫ МЫ МОГЛИ БЫ НАПИСАТЬ ПРОГРАММУ ПРОСМОТРА ВСЕЙ ПАМЯТИ В ПОИСКАХ КОНКРЕТНОГО СОВПАДЕНИЯ СЛЕДУЮЩИМ ОБРАЗОМ:

```
SEARCH    CPI
JR N2,SEARCH SEARCH - ПОИСК
```

ПРИ ТАКОМ СПОСОБЕ ,ПОКА НЕ БУДЕТ ОБНАРУЖЕНО СОВПАДЕНИЕ (БУДЕТ УСТАНОВЛЕН ФЛАГ НУЛЯ, КАК ВО ВСЕХ ОПЕРАЦИЯХ СРАВНЕНИЯ) ПРОГРАММА БУДЕТ ПРОДОЛЖАТЬ ПРОСМОТР,

К СОЖАЛЕНИЮ, ЭТО - НЕ ТАКАЯ ХОРОШАЯ ИДЕЯ, ПОСКОЛЬКУ, ЕСЛИ СОВПАДЕНИЕ НЕ БУДЕТ ОБНАРУЖЕНО, ПРОГРАММА НИКОГДА НЕ ЗАКОНЧИТСЯ! К СЧАСТЬЮ КОНСТРУКТОРЫ ЯЗЫКА Z8Ø ПОЗАБОТИЛИСЬ ОБ ЭТОМ, И КОМАНДА CPI ТАКЖЕ АВТОМАТИЧЕСКИ УМЕНЬШАЕТ BC!

ПОЭТОМУ МЫ МОЖЕМ ПО ЖЕЛАНИЮ ВЫБИРАТЬ ДЛИНУ БЛОКА, КОТОРЫЙ МЫ ХОТИМ ПРОСМОТРЕТЬ, И ТАКИМ ОБРАЗОМ ЗАДАТЬ КОНЕЦ ПРОСМОТ-

РА,

ДАВАЙТЕ ПРЕДПОЛОЖИМ, ЧТО ДЛИНА ПРОСМАТРИВАЕМОГО НАМИ БЛОКА НЕ ПРЕВЫШАЕТ 255 БАЙТОВ, ТАК ЧТО СЧЕТЧИК ВС БУДЕТ ХРАНИТЬСЯ ТОЛЬКО В РЕГИСТРЕ Ц, ТОГДА МЫ МОГЛИ БЫ НАПИСАТЬ:

```
SEARCH      CPI
             JR Z,FOUND
            INC C
            DEC C
             JR NZ,SEARCH
NOT FOUND   ,,,,,,,,,,,
            !!!!
FOUND       ,,,,,,,,,,,
```

SEARCH – ПОИСК; FOUND – НАЙДЕНО; NOT FOUND – НЕ НАЙДЕНО,

ОЧЕВИДНО, ЕСЛИ БЫ ДЛИНА БЛОКА ПРЕВШАЛА 255 БАЙТОВ, НУЖНА БЫЛА БЫ ДРУГАЯ ПРОГРАММА, ОБРАТИТЕ ВНИМАНИЕ НА ПРИМЕНЕНИЕ КОМАНД INC И DEC ДЛЯ ПРОВЕРКИ УСЛОВИЯ С= Ø, ЭТИ ДВЕ КОМАНДЫ ТРЕБУЮТ ПО ОДНОМУ ТОЛЬКО БАЙТУ КАЖДАЯ, И ПОСКОЛЬКУ ОБЕ ОНИ ДЕИСТВУЮТ НА ФЛАГ НУЛЯ, РЕЗУЛЬТАТ ИХ ДЕИСТВИЯ СОСТОИТ В ТОМ, ЧТО ФЛАГ УСТАНАВЛИВАЕТСЯ ТОЛЬКО ЕСЛИ "С" БЫЛ ПЕРВОНАЧАЛЬНО РАВЕН НУЛЮ, ЕЩЕ ОДНО ПРЕИМУЩЕСТВО СОСТОИТ В ТОМ, ЧТО ПРИ ТАКОМ НАПИСАНИИ ПРОГРАММЫ НЕ ИЗМЕНЯЮТСЯ НИКАКИЕ ДРУГИЕ РЕГИСТРЫ,

ТЕПЕРЬ МЫ МОГЛИ БЫ ТАКЖЕ ЗАХОТЕТЬ ПРОСМОТРЕТЬ БЛОКИ ПАМЯТИ НАЧИНАЯ С ВЕРШИНЫ, А НЕ С КОНЦА, И ПОЭТОМУ У НАС ИМЕЕТСЯ КОМАНДА:

```
CPD
```

КОТОРАЯ ПО-РУССКИ ЧИТАЕТСЯ "СРАВНИТЬ И УМЕНЬШИТЬ", УМЕНЬШЕНИЕ, КОНЕЧНО, ОТНОСИТСЯ К HL, А РЕЗУЛЬТАТ ДЛЯ ВС – ТОТ ЖЕ САМЫЙ!

ЕЩЕ БОЛЬШИМИ ВОЗМОЖНОСТЯМИ, ЧЕМ ЭТИ ДВЕ КОМАНДЫ, ОБЛАДАЮТ СЛЕДУЮЩИЕ ИСТИННЫЕ ТИТАНЫ В МИРЕ КОМАНД:

```
CPIR
CPDR
```

ОНИ ЧИТАЮТСЯ: "СРАВНИТЬ, УВЕЛИЧИТЬ И ПОВТОРИТЬ"
И "СРАВНИТЬ, УМЕНЬШИТЬ И ПОВТОРИТЬ",

ЭТИ ДВУХБАЙТОВЫЕ КОМАНДЫ ОБЛАДАЮТ НЕВЕРОЯТНЫМИ ВОЗМОЖНОСТЯМИ: ОНИ ПОЗВОЛЯЮТ ЦП АВТОМАТИЧЕСКИ ПРОДОЛЖИТЬ ПРОСМОТР БЛОКА ПАМЯТИ ДО ТЕХ ПОР, ПОКА ЛИБО НЕ БУДЕТ НАЙДЕНО СОВПАДЕНИЕ, ЛИБО НЕ БУДЕТ ДОСТИГНУТ КОНЕЦ БЛОКА, (ЕСТЕСТВЕННО, НАМ НУЖНО УКАЗАТЬ A,HL И BC ПЕРЕД НАЧАЛОМ, НО ДАЖЕ ПРИ ЭТИХ УСЛОВИЯХ ЭТО – НЕВЕРОЯТНО ЭКОНОМИЧЕНЫЙ СПОСОБ ЗАПИСИ)

ПОСКОЛЬКУ ЗАВЕРШЕНИЕ РАБОТЫ ЭТОИ КОМАНДЫ ПРОИСХОДИТ В ДВУХ ВОЗМОЖНЫХ СЛУЧАЯХ (А ИМЕННО, ПРИ НАЙДЕННОМ СОВПАДЕНИИ В СЕРЕДИНЕ БЛОКА И ПРИ ОТСУТСТВИИ СОВПАДЕНИЯ ВООБЩЕ), НАМ НЕОБХОДИМО ОБЕСПЕЧИТЬ НАЛИЧИЕ НЕКОТОРОГО ТЕКСТА В КОНЦЕ, ЧТОБЫ ПРОВОДИТЬ РАЗЛИЧИЕ МЕЖДУ ЭТИМИ ДВУМЯ ВОЗМОЖНОСТЯМИ,

ВАМ, ОДНАКО, СЛЕДУЕТ ЗНАТЬ ,ЧТО ВНЕ ЗАВИСИМОСТИ ОТ СКОРОСТИ ВЫПОЛНЕНИЯ ПРОГРАММ НА МАШИННОМ ЯЗЫКЕ CPIR И ДРУГИЕ АНАЛОГИЧНЫЕ КОМАНДЫ МОГУТ ОТНИМАТЬ ОЧЕНЬ МНОГО ВРЕМЕНИ,

КОМАНДА CPIR, НАПРИМЕР, ЗАТРАЧИВАЕТ 21 ЦИКЛ НА ПОИСК КАЖДОГО БАЙТА,КОНЕЧНО, КАЖДУЮ СЕКУНДУ ВЫПОЛНЯЕТСЯ 3 5ØØ ØØØ ЦИКЛОВ,1 НО ДАЖЕ С УЧЕТОМ ЭТОГО ПОИСК СРЕДИ 3 5ØØ БАЙТОВ ТРЕБУЕТ 1/5Ø СЕКУНДЫ,

ЭТО МОЖЕТ ПОКАЗАТЬСЯ ВАМ НЕ ОЧЕНЬ ДЛИТЕЛЬНЫМ ВРЕМЕНЕМ, НО ЕСЛИ УЧЕСТЬ, ЧТО ВЫДАЧА ИЗОБРАЖЕНИЯ НА ДИСПЛЕИ ОСУЩЕСТВЛЯЕТСЯ КАЖДЫЕ 1/5Ø СЕКУНДЫ ИЛИ ОКОЛО ТОГО, ВЫ ПОЙМЕТЕ, ЧТО ЭТО МОЖЕТ ОКАЗАТЬСЯ БАЖНЫМ,

ОСТАЛЬНЫЕ БЛОЧНЫЕ ОПЕРАЦИИ СТРОЯТСЯ ВОКРУГ ИДЕИ ПЕРЕМЕЩЕНИЯ ИНФОРМАЦИИ,

ВОТ ОНИ:

```
LDI  LDIR
```

LDD LDDR ОЧЕВИДНО, ОНИ ОТНОСЯТСЯ К СЕМЕЙСТВУ "ЗАГРУЗОК" И ЧИТАЮТСЯ КАК:

ЗАГРУЗИТЬ И ДАТЬ ПРИРАЩЕНИЕ;
ЗАГРУЗИТЬ, ДАТЬ ПРИРАЩЕНИЕ И ПОВТОРИТЬ;
ЗАГРУЗИТЬ И УМЕНЬШИТЬ;
ЗАГРУЗИТЬ ,УМЕНЬШИТЬ И ПОВТОРИТЬ;

РАЗБЕРЕМ СНАЧАЛА САМУЮ ПРОСТУЮ ИЗ НИХ, "LDI" - ЭТО НА САМОМ ДЕЛЕ КОМБИНАЦИЯ СЛЕДУЮЩЕГО НАБОРА ДЕЙСТВИЙ:

ЗАГРУЗИТЬ (HL) В (DE),
ДАТЬ ПРИРАЩЕНИЕ DE,HL,
УМЕНЬШИТЬ BC,

ОБРАТИТЕ ВНИМАНИЕ, ЧТО ЭТО - ЕДИНСТВЕННАЯ КОМАНДА, ЗАГРУЖАЮЩАЯ ИЗ ОДНОЙ ЯЧЕЙКИ ПАМЯТИ В ДРУГУЮ БЕЗ НЕОБХОДИМОСТИ ЗАГРУЗКИ СНАЧАЛА В РЕГИСТР,

ПРИМЕНЕНИЕ РЕГИСТРА "DE" В КАЧЕСТВЕ АДРЕСА-ЦЕЛИ ОЧЕНЬ РАЗУМНО, ТАК ВЫ НИКОГДА НЕ ЗАБУДЕТЕ, В КАКОМ РЕГИСТРЕ СОДЕРЖИТСЯ АДРЕС-ЦЕЛЬ! (В ОРИГИНАЛЕ ИСПОЛЬЗУЕТСЯ МНЕМОНИКА: "DE" DE-STINATION (АДРЕС-ЦЕЛЬ)(ПРИМЕЧ.ПЕР.))

СИММЕТРИЧНАЯ ЭТОЙ КОМАНДЕ LDD СОВЕРШЕННО СОВПАДАЕТ С ЭТОЙ, ЗА ИСКЛЮЧЕНИЕМ ТОГО, ЧТО В ПРОЦЕССЕ ЗАГРУЗКИ HL И DE УМЕНЬШАЮТСЯ, РАЗНИЦА МЕЖДУ LDI И LDD ОКАЗЫВАЕТСЯ ВАЖНА В ТОМ СЛУЧАЕ, КОГДА ДВА БЛОКА (ТОТ, ГДЕ ИНФОРМАЦИЯ НАХОДИТСЯ, И ТОТ, В КОТОРЫЙ ОНА НАПРАВЛЯЕТСЯ) ПЕРЕСЕКАЮТСЯ,

ПРЕДПОЛОЖИМ, МЫ ПРИМЕНЯЕМ ЭТУ КОМАНДУ В ПРИКЛАДНОЙ ЗАДАЧЕ ТЕКСТОВОЙ ОБРАБОТКИ И ХОТИМ УДАЛИТЬ СЛОВО ИЗ ПРЕДЛОЖЕНИЯ:

THE BIG BROWN DOG JUMPED OVER THE FOX,
(БОЛЬШАЯ КОРИЧНЕВАЯ СОБАКА ПЕРЕПРЫГНУЛА ЧЕРЕЗ ЛИСУ),
1 3 5 7 9 1 3 5 7 9 1 3 5 7 9 1 3 5 7 9

ЕСЛИ МЫ ХОТИМ УДАЛИТЬ СЛОВО "BROWN" (КОРИЧНЕВАЯ),ТО НАМ НУЖНО ТОЛЬКО СДВИНУТЬ ОСТАЛЬНУЮ ЧАСТЬ ПРЕДЛОЖЕНИЯ ВЛЕВО НА 6 ЛИТЕР,

DE = ЦЕЛЬ = ЛИТЕРА 9
HL = ИСТОЧНИК = ЛИТЕРА 15
BC = СЧЕТЧИК = 24 ЛИТЕРЫ,

ДАВАЙТЕ НАЧНЕМ С LDI: ПОСЛЕ ПЕРВОЙ КОМАНДЫ У НАС БУДЕТ
ПЕРВОНАЧАЛЬНО = THE BIG BROWN DOG JUMPED OVER THE FOX,
СДВИГ НА 1 ЛИТЕРУ; D(----D
НОВОЕ ПРЕДЛОЖЕНИЕ= THE BIG DROWN DOG JUMPED OVER THE FOX,
В HL = 10, DE = 16, BC = 23,
ПОСЛЕ ДВУХ СЛЕДУЮЩИХ КОМАНД:
THE BIG DOGWN DOG JUMPED OVER THE FOX,
И ПОСЛЕ ТОГО КАК ВСЕ КОМАНДЫ ВЫПОЛНЕНЫ:
THE BIG DOG JUMPED OVER THE FOX,E FOX,

(ЕСЛИ БЫ МЫ ХОТЕЛИ, ЧТОБЫ ЧАСТЬ ПРЕДЛОЖЕНИЯ ПОСЛЕ ТОЧКИ БЫЛА ЗАМЕНЕНА ПРОБЕЛАМИ, ТО ЭТОГО МОЖНО БЫЛО БЫ ДОСТИГНУТЬ ЗА СЧЕТ ДОБАВЛЕНИЯ ПРОБЕЛОВ В КОНЦЕ ПЕРВОНАЧАЛЬНОГО ПРЕДЛОЖЕНИЯ И УВЕЛИЧЕНИЯ BC, СКАЖЕМ ,ДО 30),

ЕСЛИ МЫ ТЕПЕРЬ РЕШИЛИ ОБРАТИТЬ ЭТОТ ПРОЦЕСС И ВОЗВРАТИТЬ СЛОВО "BROWN" В ПЕРДЛОЖЕНИЕ, ТО НАМ НУЖНО ПРОСТО СНОВА ПРИМЕНИТЬ КОМАНДУ "LDI", ПОСКОЛЬКУ МЫ ЗАБЬЕМ ИНФОРМАЦИЮ, КОТОРУЮ МЫ ХОТИМ СДВИНУТЬ:

НАПРИМЕР, HL = ИСТОЧНИК = ЛИТЕРА 9
DE = ЦЕЛЬ = ЛИТЕРА 15
BC = СЧЕТЧИК = 24 ЛИТЕРЫ

ПОСЛЕ ВЫПОЛНЕНИЯ ОДНОЙ КОМАНДЫ У НАС БУДЕТ:
ПЕРВОНАЧАЛЬНО = THE BIG DOG JUMPED OVER THE FOX,E FOX,
СДВИГ НА ЛИТЕРУ D----)D
НОВОЕ ПРЕДЛОЖЕНИЕ = THE BIG DOG JUDPED OVER THE FOX,E FOX,
ПОСЛЕ ВЫПОЛНЕНИЯ 6 КОМАНД МЫ ПОЛУЧИМ:
THE BIG DOG JUDOG JUVER THE FOX,E FOX,

ПОКА ВСЕ ХОРОШО, НО ПОСЛЕ ВЫПОЛНЕНИЯ СЛЕДУЮЩИХ ТРЕХ КОМАНД ПОЛУЧИМ:

THE BIG DOG JUDOG JUD OG THE FOX,E FOX,

ТРУДНОСТЬ СОСТОИТ В ТОМ ,ЧТО МЫ ЗАБИЛИ ИНФОРМАЦИЮ, КОТОРУЮ МЫ ХОТЕЛИ ПЕРЕМЕЩАТЬ, ВЫ МОЖЕТЕ ПРОВЕРИТЬ ЭТО, ПЫТАЯСЬ ПЕРЕМЕСТИТЬ ПО ОДНОМУ СИМВОЛУ ВРУЧНУЮ,

ПОЭТОМУ ЛУЧШЕ ИСПОЛЬЗОВАТЬ КОМАНДУ "LDI" С РЕГИСТРОМ DE, УКАЗЫВАЮЩИМ НА КОНЕЦ ПРЕДЛОЖЕНИЯ, ЭТО ГАРАНТИРУЕТ, ЧТО ИНФОРМАЦИЯ НЕ БУДЕТ ЗАБИТА ПРИ ПЕРЕМЕЩЕНИИ, (В ОРИГИНАЛЕ, ОЧЕВИДНО, ОПЕЧАТКА,WOLL ВМЕСТО WILL, (ПРИМЕЧ. ПЕР.))

КОМАНДЫ "LDIR" И "LDDR" ОБЛАДАЮТ ЕЩЕ БОЛЬШИМИ ВОЗМОЖНОСТЯМИ, ОНИ МОГУТ БЫСТРО ПЕРЕМЕЩАТЬ ТЫСЯЧИ БАЙТОВ,

Б О Л Е Е Р Е Д К О И С П О Л Ь З У Е М Ы Е К О М А Н Д Ы Z 8 Ø

О Б М Е Н Р Е Г И С Т Р А М И

ПЕРВАЯ КОМАНДА ТАКОВА:

```
EX AF,AF"
```

ОНА ДЕЛАЕТ В ТОЧНОСТИ ТО, ЧТО ПОДСКАЗЫВАЕТ ЕЕ НАЗВАНИЕ: " ОБМЕНЯТЬ ПАРЫ РЕГИСТРОВ AF И AF" ",

СЛЕДУЮЩАЯ ОБЩАЯ КОМАНДА ОБМЕНА ТАКОВА:

```
EXX
```

ЭТО КОМАНДА "ОБМЕНИВАЕТ" ВСЕ ОСТАЛЬНЫЕ 8-БИТОВЫЕРЕГИСТРЫ СЛЕДУЮЩИМ ОБРАЗОМ:

```
B C          B" C"
D E   (-)    D" E"
H L          H" L"
```

ПОЭТОМУ ЭТО - ОЧЕНЬ МОЩНАЯ КОМАНДА, НО ИМЕННО ЕЕ ВЗМОЖНОСТИ ДЕЛАЮТ ЕЕ ОГРАНИЧЕННОЙ В ПРИМЕНЕНИИ, ТАК ПРОИСХОДИТ ПОТОМУ, ЧТО ОНА ДЕЙСТВУЕТ НА ВСЕ РЕГИСТРЫ СРАЗУ, И НЕВОЗМОЖНО ОСТАВИТЬ НИ ОДНО ЗНАЧЕНИЕ.

(ЗА ИСКЛЮЧЕНИЕМ РЕГИСТРА "A", НА КОТОРЫЙ "EXX" НЕ ДЕЙСТВУЕТ),

ЕДИНСТВЕННЫЙ СПОСОБ ПРЕОДОЛЕТЬ ЭТУ ТРУДНОСТЬ - НАПИСАТЬ КОРОТКУЮ ПРОГРАММУ ТАКОГО ТИПА:

```
PUSH HL
EXX
POP HL
```

ЭТО ОЗНАЧАЕТ, ЧТО ВЫ ЗАПОМНИЛИ ЗНАЧЕНИЯ BC, DE И HL В НАБОРЕ АЛЬТЕРНАТИВНЫХ РЕГИСТРОВ, НО ПО-ПРЕЖНЕМУ ОСТАВЛЯЕМ ДЛЯ РАБОТЫ HL,

ПОСЛЕДНЯЯ КОМАНДА ИЗ ЭТОЙ ГРУППЫ НА САМОМ ДЕЛЕ НЕ ОТНОСИТСЯ К ТИПУ "ПЕРЕОДЕВАНИЯ ПЕРЧАТОК":

EX DE,HL В ЭТОЙ КОМАНДЕ ЗАДАЕТСЯ ПЕРЕКРЕСТНЫЙ ОБМЕН СОДЕРЖИМЫМ МЕЖДУ РЕГИСТРАМИ HL И DE,

ЭТА КОМАНДА НА САМОМ ДЕЛЕ ОЧЕНЬ ПОЛЕЗНА, ПОСКОЛЬКУ, КАК МЫ ВИДЕЛИ, HL - ПРИВИЛЕГИРОВАННАЯ ПАРА РЕГИСТРОВ ВО МНОГИХ ПРИКЛАДНЫХ ЗАДАЧАХ, И ЕСТЬ СИТУАЦИИ, КОГДА НУЖНОЕ НАМ ЗНАЧЕНИЕ ДЛЯ ОБРАБОТКИ НАХОДИТСЯ В DE,

У С Т А Н О В К А И С Б Р О С Б И Т О В

ДО СИХ ПОР ВСЕ КОМАНДЫ, С КОТОРЫМИ МЫ РАБОТАЛИ, ВКЛЮЧАЛИ ОБРАБОТКУ 8-БИТОВЫХ ИЛИ 16-БИТОВЫХ ЧИСЕЛ,

ГРУППА "УСТАНОВКА И СБРОС БИТОВ" ПОЗВОЛЯЕТ НАМ ОБРАБАТЫВАТЬ ОТДЕЛЬНЫЕ БИТЫ РЕГИСТРОВ И/ИЛИ СОДЕРЖИМОЕ ЯЧЕЕК ПАМЯТИ, ИЗ-ЗА ОЧЕНЬ ТРУДОЕМКОГО ХАРАКТЕРА РАБОТЫ С ОТДЕЛЬНЫМИ БИТАМИ ЭТО - НЕ ОЧЕНЬ ШИРОКО ИСПОЛЬЗУЕМАЯ ГРУППА КОМАНД,

БОЛЕЕ ТОГО, КАК ПРАВИЛО, УСТАНОВКА ОТДЕЛЬНОГО БИТА В РЕГИСТРЕ ИЛИ ЯЧЕЙКЕ-ПАМЯТИ ЗАНИМАЕТ ДАЖЕ БОЛЬШЕ, ЧЕМ ИЗМЕНЕНИЕ ИЛИ ЧТЕНИЕ ВСЕХ 8 БИТОВ ЭТОЙ ЯЧЕЙКИ ПАМЯТИ ИЛИ РЕГИСТРА,

ТЕМ НЕ МЕНЕЕ, ЕСТЬ СИТУАЦИИ, КОГДА ВАМ НЕОБХОДИМО ЗНАТЬ УСТАНОВЛЕН ИЛИ СБРОШЕН БИТ В СЕРЕДИНЕ БАЙТА, ИЛИ ДАЖЕ УСТАНОВИТЬ БИТ, ОБРАТИТЕ, ТЕМ НЕ МЕНЕ, ВНИМАНИЕ, ЧТО УСТАНОВКА И СБРОС БИТОВ МОЖЕТ ВЫПОЛНЯТЬСЯ С ПОМОЩЬЮ ЛОГИЧЕСКИХ ОПЕРАТОРОВ,

ГРУППА КОМАНД "УСТАНОВКИ И СБРОСА БИТОВ" ПОЗВОЛЯЕТ "ВКЛЮЧИТЬ" ИЛИ "ВЫКЛЮЧИТЬ" ПО ЖЕЛАНИЮ ПРОИЗВОЛЬНЫЙ БИТ, ИЛИ ДАЖЕ ПРОСТО ПОСМОТРЕТЬ НА ЗАДАННЫЙ БИТ, ЧТОБЫ ПРОВЕРИТЬ, В КАКОМ ОН СОСТОЯНИИ,

ДАВАЙТЕ ПОСМОТРИМ НА ПЕРВЫЙ НАБОР КОМАНД:

```
SET N,R
SET N,(HL)
SET N,(IX +Д )
SET N,(IY +Д )
```

КОМАНДА "SET" "ВКЛЮЧАЕТ" (T,E, =1) БИТ С НОМЕРОМ "N" (В ОБОЗНАЧЕНИЯХ Ø - 7) В РЕГИСТРЕ "R" ИЛИ ЗАДАННОЙ ЯЧЕЙКЕ ПАМЯТИ,

НИКАКИЕ ФЛАГИ НЕ ИЗМЕНЯЮТСЯ,

ГРУППА КОМАНД "RESET" ДЕЙСТВУЕТ НА ТУ ЖЕ В ТОЧНОСТИ ГРУППУ РЕГИСТРОВ ИЛИ ЯЧЕЕК ПАМЯТИ, НО ВМЕСТО "ВКЛЮЧЕНИЯ" БИТОВ ОНА ИХ "ВЫКЛЮЧАЕТ" (T,E, = Ø),

КОМАНДЫ "BIT" НА САМОМ ДЕЛЕ НУЖНО ЧИТАТЬ [illegible]
ФУНКЦИЯ ЭТОЙ КОМАНДЫ СОСТОИТ В ПРОВЕРКЕ СОДЕРЖИМОГО УКАЗАННОГО БИТА,

НИКАКИХ ИЗМЕНЕНИЙ В РЕГИСТРЫ ИЛИ ПАМЯТЬ НЕ ВНОСЯТСЯ, НО ФЛАГ НУЛЯ ИЗМЕНЯЕТСЯ В СООТВЕТСТВИИ С СОСТОЯНИЕМ ПРОВЕРЯЕМОГО БИТА,

ЕСЛИ БИТ = Ø, ТО ФЛАГ НУЛЯ УСТАНАВЛИВАЕТСЯ (= 1),
ЕСЛИ БИТ = 1, ТО ФЛАГ НУЛЯ СБРАСЫВАЕТСЯ (= Ø),

ЭТО НА ПЕРВЫЙ ВЗГЛЯД МОЖЕТ ПОКАЗАТЬСЯ ЗАПУТАННЫМ, НО НУЖНО ПОСМОТРЕТЬ НА ЭТО СЛЕДУЮЩИМ ОБРАЗОМ: ЕСЛИ БИТ РАВЕН НУЛЮ, ТО УСТАНАВЛИВАЕТСЯ ФЛАГ НУЛЯ; ЕСЛИ БИТ УСТАНОВЛЕН, ТО ,ЕСТЕСТВЕННО ,ФЛАГ НУЛЯ ПОДНИМАТЬСЯ НЕ БУДЕТ,

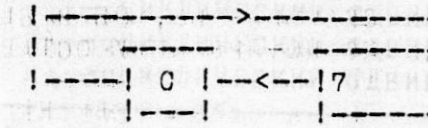
С Д В И Г И И Ц И К Л И Ч Е С К И Е С Д В И Г И

ВЫ МОЖЕТЕ СДВИГАТЬ РЕГИСТРЫ ВЛЕВО, ВПРАВО, КАК ЗАХОТИТЕ,

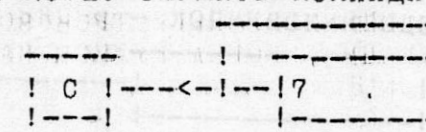
СЛОЖНОСТЬ СОСТОИТ В ТОМ, ЧТОБЫ РАЗЛИЧАТЬ РАЗЛИЧНЫЕ СДВИГИ И ЦИКЛИЧЕСКИЕ СДВИГИ С ТЕМ, ЧТОБЫ ЗНАТЬ, КАКИЕ И КОГДА НУЖНО ИСПОЛЬЗОВАТЬ, И ПОМНИТЬ, ЧТО БИТ ПЕРЕНОСА ЧАСТО МОЖНО РАССМАТРИВАТЬ В КАЧЕСТВЕ 9-ГО БИТА РЕГИСТРОВ, (Т.Е. БИТ ПЕРЕНОСА ЯВЛЯЕТСЯ ВОСЬМЫМ БИТОМ, ЕСЛИ БИТЫ ПРОНУМЕРОВАНЫ ОТ Ø ДО 7),

НЕКОТОРЫЕ КОМАНДЫ ЦИКЛИЧЕСКОГО СДВИГА ЗАХВАТЫВАЮТ БИТ ПЕРЕНОСА (В КАЧЕСТВЕ ДЕВЯТОГО БИТА), ТАК ЧТО ЦИКЛИЧЕСКИЙ СДВИГ В ЦЕЛОМ СОСТАВЛЯЕТ ЦИКЛ ИЗ 9 БИТОВ,

НАПРИМЕР, ДАВАЙТЕ РАССМОТРИМ КОМАНДУ "RLA" (СМЫСЛ КАЖДОЙ КОМАНДЫ БУДЕТ ОБ"ЯСНЕН НИЖЕ В ЭТОЙ ЖЕ ГЛАВЕ):



ДРУГИЕ ЦИКЛИЧЕСКИЕ СДВИГИ ЗАХВАТЫВАЮТ ТОЛЬКО 8-БИТОВЫЙ ЦИКЛ, ХОТЯ ФЛАГ ПЕРЕНОСА ИЗМЕНЯЕТСЯ В СООТВЕТСТВИИ СО ЗНАЧЕНИЕМ БИТА, КОТОРОМУ ПРИХОДИТСЯ "ПРОДЕЛЫВАТЬ КРУЖНОЙ ПУТЬ", ПРИМЕРОМ МОЖЕТ СЛУЖИТЬ КОМАНДА "RLCA":



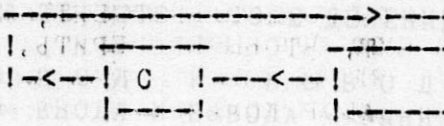
ЭТО ОЗНАЧАЕТ, ЧТО ПРИ ЦИКЛИЧЕСКОМ СДВИГЕ ВЛЕВО, КАК, НАПРИМЕР, ПОКАЗАНО ВЫШЕ, СОДЕРЖИМОЕ БИТА Ø ПЕРЕМЕЩАЕТСЯ В БИТ 1, БИТА 1 - В БИТ 2 И ТАК ДАЛЕЕ, А СОДЕРЖИМОЕ БИТА 7 ПЕРЕМЕЩАЕТСЯ И В БИТ Ø, И В БИТ ПЕРЕНОСА, СРАВНИТЕ ЭТО С КОМАНДОЙ "RLA", ОПИСАННОЙ ВЫШЕ, ГДЕ БИТ 7 ПЕРЕМЕЩАЕТСЯ В БИТ ПЕРНОСА, А БИТ ПЕРЕНОСА - В БИТ Ø,

ЦИКЛИЧЕСКИЕ СДВИГИ ВЛЕВО

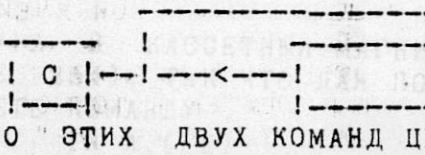
В ОСНОВНОМ ЕСТЬ ДВА ТИПА ЦИКЛИЧЕСКИХ СДВИГОВ ВЛЕВО:

Ц И К Л И Ч Е С К И Й С Д В И Г В Л Е В О Р Е Г И С Т Р О В (ЦИКЛИЧЕСКИЙ СДВИГ ВЛЕВО КРУГОВОЙ - "КРУГОВОЙ" ОЗНАЧАЕТ, ЧТО ЦИКЛ ОХВАТЫВАЕТ ТОЛЬКО 8 БИТОВ, КАК В КОМАНДЕ RLC A, ОПИСАННОЙ ВЫШЕ) - ЭТО 9-БИТОВЫЙ СДВИГ ВЛЕВО, АНАЛОГИЧНЫЙ ПОКАЗАННОМУ ВЫШЕ ДЛЯ "RL A",

RL A - "ЦИКЛИЧЕСКИЙ СДВИГ ВЛЕВО НАКОПИТЕЛЯ";
RL R - "ЦИКЛИЧЕСКИЙ СДВИГ ВЛЕВО РЕГИСТРА R"



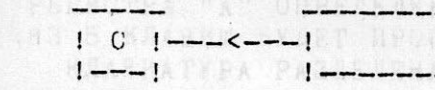
RLC A - ЦИКЛИЧЕСКИЙ СДВИГ ВЛЕВО КРУГОВОЙ РЕГИСТРА "A"
RLC R - ЦИКЛИЧЕСКИЙ СДВИГ ВЛЕВО КРУГОВОЙ РЕГИСТРА "R"
RLC (HL) - ЦИКЛИЧЕСКИЙ СДВИГ ВЛЕВО КРУГОВОЙ РЕГИСТРА (HL)
RLC (IX+ Д) - ЦИКЛИЧЕСКИЙ СДВИГ ВЛЕВО КРУГОВОЙ (IX +)
RLC (IY+ Д) - ЦИКЛИЧЕСКИЙ СДВИГ ВЛЕВО КРУГОВОЙ (IY +)



ПОМИМО ЭТИХ ДВУХ КОМАНД ЦИКЛИЧЕСКОГО СДВИГА ВЛЕВО, ЕСТЬ ЕЩЕ КОМАНДА СДВИГА ВЛЕВО, НО ОНА МОЖЕТ ОБРАБАТЫВАТЬ ТОЛЬКО

РЕГИСТР "A",

SL A СДВИГ НАКОПИТЕЛЯ ВЛЕВО



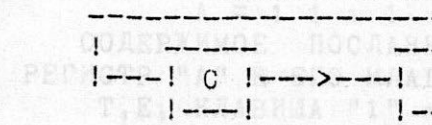
ОТЛИЧИЕ ЭТОЙ КОМАНДЫ СОСТОИТ В ТОМ, ЧТО СОДЕРЖИМОЕ БИТА ПЕРЕНОСА ТЕРЯЕТСЯ, А НУЛЕВОЙ БИТ ЗАПОЛНЯЕТСЯ НУЛЕМ, ПО-СУЩЕСТВУ ЭТО СВОДИТСЯ К УМНОЖЕНИЮ "A" НА 2, ПОСКОЛЬКУ НИЧЕГО В НАКОПИТЕЛЬ НЕ ПЕРЕНОСИТСЯ ,(ОБДУМАЙТЕ РАБОТУ КОМАНДЫ "SL A" В СЛУЧАЕ A = 80H),

ЦИКЛИЧЕСКИЕ СДВИГИ ВПРАВО

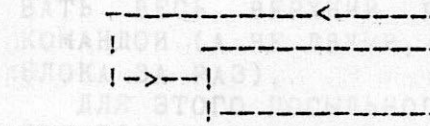
ВНОВЬ У НАС ДВА ОСНОВНЫХ РЕЖИМА ЦИКЛИЧЕСКОГО СДВИГА, НО НА ЭТОТ РАЗ ВПРАВО, ВРАЩЕНИЕ ВПРАВО ОХВАТЫВАЕТ ТОТ ЖЕ ДИАПАЗОН ДОПУСТИМЫХ ЯЧЕЕК ПАМЯТИ И ВРАЩЕНИИ (ТАК В ОРИГИНАЛЕ, ВОЗМОЖНО, ЗДЕСЬ ИМЕЕТСЯ ОПЕЧАТКА, СКОРЕЕ ВСЕГО НУЖНО: РЕГИСТРОВ,(ПРИМЕЧ, ПЕР,)), ЧТО И ВРАЩЕНИЯ ВЛЕВО ,

RR A – ЦИКЛИЧЕСКИЙ СДВИГ НАКОПИТЕЛЯ ВПРАВО

RR R – ЦИКЛИЧЕСКИЙ СДВИГ РЕГИСТРА ВПРАВО

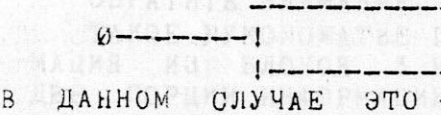


RRC A – ЦИКЛИЧЕСКИЙ СДВИГ ВПРАВО КРУГОВОЙ РЕГИСТРА "A"
RRC R – ЦИКЛИЧЕСКИЙ СДВИГ ВПРАВО КРУГОВОЙ РЕГИСТРА "R"
RRC (HL) – ЦИКЛИЧЕСКИЙ СДВИГ ВПРАВО КРУГОВОЙ РЕГИСТРА (HL)
RRC (IX+ Д) – ЦИКЛИЧЕСКИЙ СДВИГ ВПРАВО КРУГОВОЙ (IX +)
RRC (IY+ Д) – ЦИКЛИЧЕСКИЙ СДВИГ ВПРАВО КРУГОВОЙ (IY +)



ИМЕЕТСЯ СДВИГ ВПРАВО, АНАЛОГИЧНЫЙ СДВИГУ ВЛЕВО:

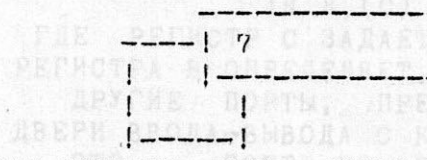
SRL R – ЛОГИЧЕСКИЙ СДВИГ ВПРАВО РЕГИСТРА "R"



В ДАННОМ СЛУЧАЕ ЭТО – ЧИСТОЕ ДЕЛЕНИЕ НА 2, ЕСЛИ МЫ ИСПОЛЬЗУЕМ ЧИСЛА БЕЗ ЗНАКА (Т,Е, ДИАПАЗОН ЧИСЕЛ ,КОТОРЫЕ МЫ ХОТИМ ПРЕДСТАВЛЯТЬ, СОСТАВЛЯЕТ 0 – 255),

ПОСКОЛЬКУ В НЕКОТОРЫХ ПРИКЛАДНЫХ ЗАДАЧАХ МЫ ПРИМЕНЯЕМ СОГЛАШЕНИЕ О ТОМ, ЧТО ДЛЯ ЗАДАНИЯ ОТРИЦАТЕЛЬНЫХ ЧИСЕЛ БИТ 7 УСТАНАВЛИВАЕТСЯ РАВНЫМ 1 (ЧТО ДАЕТ НАМ ДИАПАЗОН ОТ –128 ДО +127), ИМЕЕТСЯ ДОПОЛНИТЕЛЬНАЯ (В ОРИГИНАЛЕ, ОЧЕВИДНО ,ОПЕЧАТКА ADDITION SHIFT (ПРИМЕЧ, ПЕР,)) КОМАНДА СДВИГА ВПРАВО, НАЗЫВАЮЩАЯСЯ

SRA R – АРИФМЕТИЧЕСКИЙ СДВИГ ВПРАВО РЕГИСТРА "R"



КАК ВИДИТЕ, ЭТО – ТОЖЕ ДЕЛЕНИЕ НА 2, НО ОНО СОХРАНЯЕТ БИТ ЗНАКА,

В В О Д И В Ы В О Д

ВОПРОСЫ ВВОДА И ВЫВОДА – ОДНИ ИЗ САМЫХ ПРОСТЫХ В ПРОГРАММИРОВАНИИ НА МАШИННОМ ЯЗЫКЕ,

ЦП Z80 ИМЕЕТ 256 ПОРТОВ ВВОДА/ВЫВОДА,

КЛАВИАТУРА И КАССЕТНЫЙ МАГНИТОФОН ИМЕЮТ ПОРТ ВВ/ВЫВ FEH (ДЕСЯТИЧНОЕ 254), ТАК ЧТО ДЛЯ ПОЛУЧЕНИЯ ДАННЫХ С КЛАВИАТУРЫ ВЫ ПРИМЕНЯЕТЕ КОМАНДУ

IN A,(FE)

ТЕПЕРЬ ВЫ МОЖЕТЕ ЗАДАТЬ СЕБЕ ВОПРОС, КАК 40 КЛАВИШ НА КЛАВИАТУРЕ ОРГАНИЗОВАНЫ ТАК, ЧТО ОНИ ПРЕДСТАВЛЯЮТСЯ 8-БИТОВЫМИ

БИТАМИ,

ОТВЕТ ДЛЯ ВАС БУДЕТ СКОРЕЕ ВСЕГО НЕОЖИДАННЫМ - КЛАВИАТУРА ЗА ОДИН РАЗ ПЕРЕДАЕТ ИНФОРМАЦИЮ ТОЛЬКО ОТ 5 КЛАВИШ, ЗНАЧЕНИЕ РЕГИСТРА "A" ОПРЕДЕЛЯЕТ В МОМЕНТ ОТКРЫТИЯ ПОРТА КАКАЯ ГРУППА ИЗ 5 КЛАВИШ БУДЕТ ПРОСМАТРИВАТЬСЯ!

КЛАВИАТУРА РАЗДЕЛЕНА НА 4 РЯДА, КАЖДЫЙ ИЗ КОТОРЫХ СОДЕРЖИТ ДВА БЛОКА ПО 5 КЛАВИШ:

```
3 =)  1  2 3 4 5   6 7 8 9  Ø  (= 4
2 =)  Q  W E R T   Q U I O  P  (= 5
1 =)  A  S D F G   H J K L N/L (= 6.
Ø =) SFT Z X C V   B N M , SPC
```

ЛЕГКО ВИДЕТЬ, ЧТО ИМЕЕТСЯ 8 БЛОКОВ ЛИТЕР, И МЫ ПОЭТОМУ МОЖЕМ УВЯЗАТЬ ИХ С 8 БИТАМИ РЕГИСТРА "A",

ИМЕННО ТАК И ОБСТОИТ ДЕЛО:

ВСЕ БИТЫ РЕГИСТРА "A" УСТАНАВЛИВАЮТСЯ В ПОЛОЖЕНИЕ "ВКЛЮЧЕНО" ЗА ИСКЛЮЧЕНИЕМ ОДНОГО БИТА, ЗАДАЮЩЕГО ПОДЛЕЖАЩИЙ ЧТЕНИЮ БЛОК,

ТАК, ЧТОБЫ ПРОЧИТАТЬ КЛАВИШИ ИЗ БЛОКА "1 2 3 4 5", ДОЛЖЕН БЫТЬ ВЫКЛЮЧЕН БИТ 3 РЕГИСТРА "A":

A = 1 1 1 1 Ø 1 1 1 = F7

СОДЕРЖИМОЕ ПОСЛАННОЙ С КЛАВИАТУРЫ ИНФОРМАЦИИ ПОСТУПАЕТ В РЕГИСТР "A" В ЕГО МЛАДШИЕ БИТЫ:

```
Т,Е, КЛАВИША "1" -) БИТ Ø РЕГИСТРА "A"
     КЛАВИША "2" -) БИТ 1 РЕГИСТРА "A"
```

ВЫ МОЖЕТЕ СЧИТАТЬ, ЧТО ИНФОРМАЦИЯ ПОСТУПАЕТ В РЕГИСТР "A" СНАЧАЛА С ВНЕШНИХ СТОРОН КЛАВИАТУРЫ, ТАК ЧТО И "Ø" И "1" ПОПАДУТ В БИТ "Ø" РЕГИСТРА "A",

ДЛЯ НЕКОТОРЫХ ИГР ВАМ МОЖЕТ ПОНАДОБИТЬСЯ ПОЗВОЛИТЬ СЧИТЫВАТЬ ВЕСЬ ВЕРХНИЙ РЯД, ПРИЧЕМ ВЕСЬ ЕГО МОЖНО ЧИТАТЬ ОДНОЙ КОМАНДОЙ (А НЕ ДВУМЯ, КАК ПОТРЕБОВАЛОСЬ БЫ ПРИ ЧТЕНИИ ОДНОГО БЛОКА ЗА РАЗ),

ДЛЯ ЭТОГО ПОСЫЛЬНОГО ОБМАНЫВАЮТ, ЧТОБЫ ОН ВЫДАЛ ВАМ СРАЗУ ДВЕ ПОРЦИИ ИНФОРМАЦИИ:

НАПРИМЕР, A = 1 1 1 Ø Ø 1 1 1 = E7

ОБРАТИТЕ ВНИМАНИЕ, ЧТО ОБА БИТА "3" И "4" "ВЫКЛЮЧЕНЫ",

ТАКОЕ РУКОПОЖАТИЕ ГОВОРИТ ПОСЫЛЬНОМУ, ЧТО ЦП НУЖНА ИНФОРМАЦИЯ ИЗ БЛОКОВ 3 И 4, И ИМЕННО ЕЕ ОН И ПОЛУЧИТ, КОНЕЧНО, ДВЕ ПОРЦИИ ИНФОРМАЦИИ ОКАЗЫВАЮТСЯ ПЕРЕМЕШАННЫМИ, ТАК ЧТО ВЫ НЕ СМОЖЕТЕ СКАЗАТЬ, БЫЛА НАЖАТА КЛАВИША "Ø" ИЛИ "1", НАПРИМЕР, ОБЕ УСТАНОВЯТ БИТ Ø РЕГИСТРА "A",

```
Т,Е  "1"  ИЛИ "Ø"   -)  БИТ Ø РЕГИСТРА A
     "2"  ИЛИ "9"   -)  БИТ 1 РЕГИСТРА A
```

ЭТО ПОЛЕЗНО ДЛЯ ДИНАМИЧЕСКИХ ИГР, ПОСКОЛЬКУ ПОЗВОЛЯЕТ ИСПОЛЬЗОВАТЬ КЛАВИШИ "5" И "8" В КАЧЕСТВЕ УКАЗАТЕЛЯ НАПРАВЛЕНИЯ ДВИЖЕНИЯ НАПРАВО И НАЛЕВО, ХОТЯ ОНИ И НАХОДЯТСЯ В РАЗНЫХ БЛОКАХ КЛАВИАТУРЫ,

ОБРАТИТЕ ВНИМАНИЕ, ЧТО ЕСЛИ ВЫ ПРИМЕНЯЕТЕ КОМАНДУ

IN R,(C) ,

ГДЕ РЕГИСТР С ЗАДАЕТ ИСПОЛЬЗУЕМЫЙ ВАМИ ПОРТ, ТО СОДЕРЖИМОЕ РЕГИСТРА В ОПРЕДЕЛЯЕТ, КАКОЙ БЛОК КЛАВИАТУРЫ ВЫБИРАЕТСЯ,

ДРУГИЕ ПОРТЫ, ПРЕДСТАВЛЯЮЩИЕ ДЛЯ ВАС ИНТЕРЕС, ОЧЕВИДНО, ДВЕРИ ВВОДА-ВЫВОДА С КАССЕТНОГО МАГНИТОФОНА,

ЭТО - ПОРТ FE, КАК СКАЗАНО ВЫШЕ, ОСНОВНАЯ ВОЗНИКАЮЩАЯ ПРОБЛЕМА СВЯЗАНА С СИНХРОНИЗАЦИЕЙ ВВОДА И ВЫВОДА ДАННЫХ; ПРОБЛЕМЫ ТАКОГО ТИПА ТРЕБУЮТ БОЛЬШОГО ОПЫТА ПРОГРАММИРОВАНИЯ НА МАШИННОМ ЯЗЫКЕ И ВЫЧИСЛЕНИЯ ВРЕМЕНИ ВЫПОЛНЕНИЯ КАЖДОЙ КОМАНДЫ,

КОМАНДА OUT ТАКЖЕ ПРИМЕНЯЕТСЯ ДЛЯ ГЕНЕРИРОВАНИЯ ЗВУКОВОГО СИГНАЛА НА "СПЕКТРУМ" И УСТАНОВЛЕНИЯ ЦВЕТА ОКАЙМЛЕНИЯ,

БИТЫ ДЛЯ КОМАНДЫ OUT : Ø,1,2 - ЦВЕТ ОКАЙМЛЕНИЯ

```
3 - ПОСЫЛАЕТ ИМПУЛЬС НА ГНЕЗДА MIC И EAR
4 - ПОСЫЛАЕТ ИМПУЛЬС НА ВНУТРЕННИЙ ГРОМКОГОВОРИТЕЛЬ
```

ЧТОБЫ ИЗМЕНИТЬ ЦВЕТ ОКАЙМЛЕНИЯ, НУЖНО ЗАГРУЗИТЬ В РЕГИСТР A СООТВЕТСТВУЮЩЕЕ ЗНАЧЕНИЕ ЦВЕТА, А ЗАТЕМ ВЫПОЛНИТЬ КОМАНДУ OUT (FE),A, ОБРАТИТЕ ВНИМАНИЕ, ЧТО ЭТО - ЛИШЬ TEMPORARY (ВРЕМЕННОЕ) ИЗМЕНЕНИЕ ЦВЕТА ОКАЙМЛЕНИЯ, ЧТОБЫ ПОСТОЯННО ИЗМЕНИТЬ ЦВЕТ ОКАЙМЛЕНИЯ, ВЫ ДОЛЖНЫ ВЫПОЛНИТЬ ПРИВЕДЕННУЮ ВЫШЕ

КОМАНДУ OUT И ТАКЖЕ ИЗМЕНИТЬ ЗНАЧЕНИЕ ЯЧЕЙКИ ПАМЯТИ 23624, ПРЕДСТАВЛЯЮЩУЮ СОБОЙ ПЕРЕМЕННУЮ BORDCR ОПЕРАЦИОННОЙ СИСТЕМЫ

ПРИЧИНА ЭТОГО СОСТОИТ В ТОМ, ЧТО АППАРАТУРА "СПЕКТРУМ" (ЧИП ULA "СПЕКТРУМ") УПРАВЛЯЕТ ЦВЕТОМ ОКАИМЛЕНИЯ И ОНА ПОЛУЧАЕТ ИНФОРМАЦИЮ О НЕМ, ПРОСМАТРИВАЯ СОДЕРЖИМОЕ ЭТОЙ ЯЧЕЙКИ ПАМЯТИ. Вы МОЖЕТЕ ПРЕДОТВРАТИТЬ ВСЮ ЭТУ ВОЛЫНКУ ЦП С ЦВЕТОМ ОКАЙМЛЕНИЯ ТОЛЬКО ЗАБЛОКИРОВАВ ВСЕ ПРЕРЫВАНИЯ (КОМАНДА DI), ОБРАТИТЕ ВНИМАНИЕ, ЧТО НЕКОТОРЫЕ ПОДПРОГРАММЫ В ПЗУ РАЗБЛОКИРУЮТ ПРЕРЫВАНИЯ (КОМАНДА EI).

САМОСТОЯТЕЛЬНАЯ ГЕНЕРАЦИЯ ЗВУКОВОГО СИГНАЛА

ВЫ МОЖЕТЕ СГЕНЕРИРОВАТЬ СВОЙ СОБСТВЕННЫЙ ЗВУКОВОЙ СИГНАЛ НА "СПЕКТРУМ",НО ДЛЯ ПОЛЬЗОВАТЕЛЕЙ, ИМЕЮЩИХ ТОЛЬКО 16К ПАМЯТИ С ПРОИЗВОЛЬНЫМ ДОСТУПОМ ИМЕЮТСЯ ОПРЕДЕЛЕННЫЕ ОГРАНИЧЕНИЯ, ПОРОЖДАЕМЫЕ ОСОБЕННОСТЯМИ АППАРАТУРЫ.

ПОСКОЛЬКУ ЭКРАН ПОСТОЯННО ОБНОВЛЯЕТСЯ, АППАРАТУРА РЕГУЛЯРНО ПРЕРЫВАЕТ ВЫПОЛНЕНИЕ ЗАДАНИЙ Z80 ДЛЯ ТОГО, ЧТОБЫ ПОКАЗАТЬ, ЧТО НАХОДИТСЯ В ФАЙЛЕ ДИСПЛЕЯ, ЭТО ВЫПОЛНЯЕТСЯ ПУТЕМ СНИЖЕНИЯ НАПРЯЖЕНИЯ В СХЕМЕ WAIT.

В РЕЗУЛЬТАТЕ ОКАЗЫВАЕТСЯ НЕВОЗМОЖНЫМ СОЗДАНИЕ КАКОЙ БЫ ТО НИ БЫЛО ПРОГРАММЫ, ТРЕБУЮЩЕЙ ТОЧНОЙ ИЛИ РЕГУЛЯРНОЙ СИНХРОНИЗАЦИИ,ПОСКОЛЬКУ НЕВОЗМОЖНО ПРЕДСКАЗАТЬ РЕЗУЛЬТАТЫ СИНХРОНИЗАЦИИ С ПОМОЩЬЮ ЭТИХ ПРЕРЫВАНИЙ WAIT, КОНСТРУКЦИЯ "СПЕКТРУМ" ТАКОВА, ЧТО ПРЕРЫВАНИЕ Z80 ОСУЩЕСТВЛЯЕТСЯ ПРИ ПОПЫТКЕ ОБРАБОТКИ ИНФОРМАЦИИ, СОДЕРЖАЩЕЙСЯ В ПЕРВЫХ 16К ПАМЯТИ С ПРОИЗВОЛЬНЫМ ДОСТУПОМ, ТАКОГО ПРЕРЫВАНИЯ НЕ ПРОИСХОДИТ, ЕСЛИ ПРОГРАММА И ДАННЫЕ, К КОТОРЫМ ОСУЩЕСТВЛЯЕТСЯ ДОСТУП Z80, НАХОДЯТСЯ В ПЗУ ИЛИ ВЕРХНИХ 32К ПАМЯТИ.

ОБОБЩИТЬ ВСЕ ЭТО В ТЕРМИНАХ, ПОНЯТНЫХ НЕПОСВЯЩЕННЫМ, МОЖНО СЛЕДУЮЩИМ ОБРАЗОМ: ВЫ МОЖЕТЕ ВОСПРОИЗВОДИТЬ ЗВУКИ И ШУМЫ С ПОМОЩЬЮ КОМАНДЫ OUT НА ЭВМ С 16К ПАМЯТИ, НО ЭТО НЕ БУДУТ ЧИСТЫЕ НОТЫ, (ОБОЙТИ ЭТО МОЖНО С ПОМОЩЬЮ ПРОГРАММЫ BEEP ИЗ ПЗУ, СМ, ГЛАВУ О СРЕДСТВАХ "СПЕКТРУМ").

ДЛЯ ГЕНЕРАЦИИ ЗВУКА ВЫ ДОЛЖНЫ ПОСЛАТЬ ИМПУЛЬС ДЛЯ ВКЛЮЧЕНИЯ ГРОМКОГОВОРИТЕЛЯ (ИЛИ (И) НА ГНЕЗДО MIC, ЕСЛИ ЕГО НУЖНО УСИЛИВАТЬ), ЗАТЕМ, СПУСТЯ НЕКОТОРОЕ ВРЕМЯ, ВАМ НУЖНО ПОСЛАТЬ ДРУГОЙ ИМПУЛЬС ДЛЯ ЕГО ВЫКЛЮЧЕНИЯ, ЗАТЕМ ЕЩЕ СПУСТЯ НЕКОТОРОЕ ВРЕМЯ ВНОВЬ ВКЛЮЧИТЬ И Т. Д . ...,

ТАКИМ ОБРАЗОМ СОЗДАЕТСЯ ЗВУК, ИНТЕРВАЛ ВРЕМЕНИ МЕЖДУ ДВУМЯ ПОСЛЕДОВАТЕЛЬНЫМИ ВКЛЮЧЕНИЯМИ ГРОМКОГОВОРИТЕЛЯ ОПРЕДЕЛЯЕТ ЧИСТОТУЗВУКА, ПРОДОЛЖИТЕЛЬНОСТЬ ИНТЕРВАЛА ВРЕМЕНИ, В ТЕЧЕНИИ КОТОРОГО ГРОМКОГОВОРИТЕЛЬ ОСТАЕТСЯ ВКЛЮЧЕННЫМ, В ПРОТИВОПОЛОЖНОСТЬ ИНТЕРВАЛУ ВРЕМЕНИ МЕЖДУ ИМПУЛЬСАМИ, МОЖЕТ ДАТЬ ВАМ МИНИМАЛЬНУЮ СТЕПЕНЬ КОНТРОЛЯ ГРОМКОСТИ ЗВУКА.

ОБРАТИТЕ ВНИМАНИЕ, ЧТО ВЫ ДОЛЖНЫ ПРИМЕНЯТЬ ДЛЯ ВКЛЮЧЕНИЯ И ВЫКЛЮЧЕНИЯ ТАКОЕ ЗНАЧЕНИЕ A, ЧТОБЫ ЦВЕТ ОКАЙМЛЕНИЯ НЕ ИЗМЕНЯЛСЯ, В ПРОТИВНОМ СЛУЧАЕ ЧАСТОТНАЯ ХАРАКТЕРИСТИКА ПОЛУЧЕННОГО, ЗВУКА БУДЕТ НАПОМИНАТЬ ЗВУКИ ЗАГРУЗКИ

ПРЕДСТАВЛЕНИЕ ДВОИЧНО-КОДИРОВАННЫХ ДЕСЯТИЧНЫХ ЧИСЕЛ

ДВОИЧНО-КОДИРОВАННОЕ ДЕСЯТИЧНОЕ ПРЕДСТАВЛЕНИЕ ПО АНГЛИЙСКИ СОКРАЩАЕТСЯ КАК BCD (BINARY-CODED DECIMAL), ЭТО - ОДИН ИЗ СПОСОБОВ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ В ДЕСЯТИЧНОЙ ФОРМЕ.

ЧТОБЫ ЗАКОДИРОВАТЬ КАЖДУЮ ИЗ ЦИФР ОТ 0 ДО 9 НЕОБХОДИМО ВСЕГО 4 БИТА, И 6 ДОПУСТИМЫХ КОДОВ НЕ БУДУТ В ЭТОМ ПРЕДСТАВЛЕНИИ ИСПОЛЬЗОВАТЬСЯ.

ПОСКОЛЬКУ ДЛЯ КОДИРОВАНИЯ ДЕСЯТИЧНОЙ ЦИФРЫ НЕОБХОДИМО 4 БИТА, В КАЖДОМ БАЙТЕ МОЖНО ЗАКОДИРОВАТЬ ДВЕ ЦИФРЫ, ЭТО НАЗЫВАЕТСЯ ДВОИЧНО-КОДИРОВАННЫМ ДЕСЯТИЧНЫМ ПРЕДСТАВЛЕНИЕМ.

НАПРИМЕР,0000 0000 -ДВОИЧНО-КОДИРОВАННОЕ ДЕСЯТИЧНОЕ ПРЕДСТАВЛЕНИЕ ДЕСЯТИЧНОГО ЧИСЛА 00

1001 1001 -ДВОИЧНО-КОДИРОВАННОЕ ДЕСЯТИЧНОЕ ПРЕДСТАВЛЕНИЕ ДЕСЯТИЧНОГО ЧИСЛА 99.

АРИФМЕТИЧЕСКИЕ ДЕЙСТВИЯ НАД ДВОИЧНО-КОДИРОВАННЫМИ ДЕСЯТИЧНЫМИ ЧИСЛАМИ

ЭТА СТРАННАЯ СИСТЕМА ОБОЗНАЧЕНИЙ ДЛЯ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ МОЖЕТ ПОВЕСТИ К ПОТЕНЦИАЛЬНЫМ ПРОБЛЕМАМ ПРИ СЛОЖЕНИИ И ВЫЧИТАНИИ.

ПОПРОБУЕМ СЛОЖИТЬ СЛЕДУЮЩИЕ ЧИСЛА

```
BCD    08  0000 1000
BCD    03  0000 0011
-------------------
BCD    11  0000 1011
```

ВЫ ЗАМЕТИТЕ, ЧТО РЕЗУЛЬТАТ ВТОРОЙ ОПЕРАЦИИ НЕВЕРЕН И ЯВЛЯЕТСЯ НЕДОПУСТИМЫМ В ПРЕДСТАВЛЕНИИ ДВОИЧНО-КОДИРОВАННОМ ДЕСЯТИЧНЫМ ЧИСЛОМ, ДЛЯ КОМПЕНСАЦИИ ЭТИХ ТРУДНОСТЕИ НЕОБХОДИМО ПРИМЕНЯТЬ ОСОБУЮ КОМАНДУ, "DAA", АЗЫВАЕМУЮ "ДЕСЯТИЧНАЯ НАСТРОЙКА АРИФМЕТИЧЕСКИХ ДЕИСТВИЙ", ЧТОБЫ ИСПРАВИТЬ РЕЗУЛЬТАТ СЛОЖЕНИЯ,(Т,Е, ДОБАВЛЯТЬ 6, ЕСЛИ РЕЗУЛЬТАТ ПРЕВЫШАЕТ 9),

СЛЕДУЮЩУЮ ТРУДНОСТЬ ИЛЛЮСТРИРУЕТ ТОТ ЖЕ ПРИМЕР, БУДЕТ ГЕНЕРИРОВАТЬСЯ БИТ ПЕРЕНОСА ИЗ МЛАДШЕГО РАЗРЯДА ДВОИЧНО-КОДИРОВАННОГО ДЕСЯТИЧНОГО ЧИСЛА (САМОГО ПЕРВОГО) В САМЫЙ ЛЕВЫЙ, ЭТОТ ВНУТРЕННИИ ПЕРЕНОС ДОЛЖЕН БЫТЬ УЧТЕН И ДОБАВЛЕН ВО ВТОРОЙ РАЗРЯД ДВОИЧНО-КОДИРОВАННОГО ДЕСЯТИЧНОГО.

ДЛЯ ВЫЯВЛЕНИЯ ЭТОГО ПЕРНОСА ИСПОЛЬЗУЕТСЯ "ФЛАГ ПОЛОВИННОГО ПЕРЕНОСА", "H",

```
LD A,12H      ; LOAD LITERAL BCD "12"
ADD A,24H     ; ADD LITERAL BCD "24"
DAA           ; DECIMAL ADJUST RESULT
LD (ADDR),A   ; STORE RESULT
```

LOAD LITERAL - ЗАГРУЗИТЬ ЛИТЕРАЛ; ADD LITERAL - ДОБАВИТЬ ЛИТЕРАЛ; DECIMAL ADJUST RESULT - ДЕСЯТИЧНАЯ НАСТРОЙКА РЕЗУЛЬТАТА STORE RESULT - ЗАПОМИНАНИЕ РЕЗУЛЬТАТА,

ПРИ ПРГРАММИРОВАНИИ ВАМ ВРЯД ЛИ ПОТРЕБУЕТСЯ ДВОИЧНО- КОДИРОВАННОЕ ДЕСЯТИЧНОЕ ПРЕДСТАВЛЕНИЕ, НО НЕПЛОХО ЗНАТЬ, ЧТО ЧИП Z80 ВСЕ-ТАКИ ПОДДЕРЖИВАЕТ ЭТО ПРЕДСТАВЛЕНИЕ, И КОМАНДА DAA ОБЛЕГЧАЕТ ЖИЗНЬ НЕБОЛЬШОИ ГРУППЕ ПОЛЬЗОВАТЕЛЕЙ ДВОИЧНО-КОДИРОВАННОГО ДЕСЯТИЧНОГО ПРЕДСТАВЛЕНИЯ,

П Р Е Р Ы В А Н И Я

ПРЕРЫВАНИЕ -ЭТО ПОСЫЛАЕМЫЙ МИКРОПРОЦЕССОРУ СИГНАЛ, КОТОРЫЙ МОЖЕТ ПОЯВИТЬСЯ В ЛЮБОЕ ВРЕМЯ И, ВООБЩЕ ГОВОРЯ, ПРИОСТАНОВИТ ВЫПОЛНЕНИЕ ТЕКУЩЕЙ ПРОГРАММЫ (ТАК, ЧТО ПРОГРАММА ДАЖЕ НЕ УЗНАЕТ ОБ ЭТОМ),

Z80 ПРЕДОСТАВЛЯЕТ ТРИ МЕХАНИЗМА ОСУЩЕСТВЛЕНИЯ ПРЕРЫВАНИИ: ЗАПРОС ШИНЫ (BUSRG), НЕМАСКИРУЕМОЕ ПРЕРЫВАНИЕ (NMI) И ОБЫЧНОЕ ПРЕРЫВАНИЕ (INT),

С ТОЧКИ ЗРЕНИЯ ПРОГРАММИРОВАНИЯ МЫ РАССМОТРИМ ТОЛЬКО ОБЫЧНОЕ МАСКИРУЕМОЕ ПРЕРЫВАНИЕ (INT),

КОМАНДА DI (БЛОКИРОВКА ПРЕРЫВАНИЯ) ПРИМЕНЯЕТСЯ ДЛЯ СБРОСА (МАСКИРОВАНИЯ), А КОМАНДА EI (РАЗБЛОКИРОВАНИЕ ПРЕРЫВАНИЯ)ДЛЯ УСТАНОВКИ (РАЗМАСКИРОВАНИЯ),

В ОБЩЕМ СЛУЧАЕ ОБЫЧНОЕ ПРЕРЫВАНИЕ ПРИВЕДЕТ К ТОМУ, ЧТО ТЕКУЩИЙ СЧЕТЧИК КОМАНД БУДЕТ ПОМЕЩЕН В СТЕК, А УПРАВЛЕНИЕ (В ОРИ—ГИНАЛЕ ВОЗМОЖНО ОПЕЧАТКА, ПЕРЕВОД ДАЕТСЯ ПО СМЫСЛУ, (ПРИМЕЧ, ПЕР,) С ПОМОШЬЮ КОМАНДЫ RST БУДЕТ ПЕРЕДАНО НА НУЛЕВУЮ СТРАНИЦУ ПЗУ, ДЛЯ ВОЗВРАШЕНИЯ ИЗ ПРЕРЫВАНИЯ ТРЕБУЕТСЯ КОМАНДА RETI (ВОЗВРАТ ИЗ ПРЕРЫВАНИЯ),

ПРИ НОРМАЛЬНОМ РЕЖИМЕ РАБОТЫ У "СПЕКТРУМ " ПРЕРЫВАНИЯ РАЗБЛОКИРОВАНЫ (EI), И НА САМОМ ДЕЛЕ ПРОГРАММА ПРЕРЫВАЕТСЯ 50 РАЗ В СЕКУНДУ, ЭТО ПРЕРЫВАНИЕ ПОЗВОЛЯЕТ ПРОГРАММЕ ПЗУ ОСУШЕСТВЛЯТЬ СКАНИРОВАНИЕ КЛАВИАТУРЫ,

ВАМ МОЖЕТ ПОНАДОБИТЬСЯ ЗАБЛОКИРОВАТЬ ПРЕРЫВАНИЯ В ВАШЕЙ ПРОГРАММЕ, ПОСКОЛЬКУ ЭТО УСКОРЯЕТ ВЫПОЛНЕНИЕ, ВЫ ВСЕ-ТАКИ СМОЖЕТЕ ПОЛУЧАТЬ ИНФОРМАЦИЮ С КЛАВИАТУРЫ, ПРИ УСЛОВИИ ИСПОЛЬЗОВАНИЯ СВОЕИ СОБСТВЕННОЙ ПРОГРАММЫ ДЛЯ ЭТОГО, ОБЯЗАТЕЛЬНО РАЗБЛОКИРУИТЕ ПРЕРЫВАНИЯ, КОГДА ЗАКОНЧИТЕ ВЫПОЛНЕНИЕ ПРОГРАММЫ,ПОСКОЛЬКУ ИНАЧЕ СИСТЕМА НЕ СМОЖЕТ ПОЛУЧАТЬ ИНФОРМАЦИЮ С КЛАВИАТУРЫ,

К.О М А Н Д А Р Е С Т А Р Т А (R S T)

ЭТО - СКОРЕЕ ВСЕГО "РУДИМЕНТЫ"ЧИПА 8080, РЕАЛИЗОВАННЫЕ В ЦЕЛЯХ СОВМЕСТМОСТИ, ПОЭТОМУ ВЫ ВРЯД ЛИ СТАНЕТЕ ПРИМЕНЯТЬ КОМАНДУ RST В СВОЕЙ ПРОГРАММЕ,

КОМАНДА RST ВЫПОЛНЯЕТ ТЕ ЖЕ САМЫЕ ДЕЙСТВИЯ, ЧТО И ВЫЗОВ, НО ПОЗВОЛЯЕТ СОВЕРШАТЬ ПЕРЕХОД ТОЛЬКО ПО ВОСЬМИ АДРЕСАМ,

40

РАСПОЛОЖЕННЫМ В ПЕРВЫХ 256 ЯЧЕЙКАХ ПАМЯТИ: 00H, 08H, 10H, 18H, 20H, 28H, 30H, И 38H,

ПРЕИМУЩЕСТВО КОМАНДЫ RST СОСТОИТ В ТОМ, ЧТО ЧАСТО ИСПОЛЬЗУЕМЫЕ ПОДПРОГРАММЫ МОЖНО ВЫЗВАТЬ, ЗАТРАЧИВАЯ НА ЭТО ВСЕГО ОДИН БАЙТ, КРОМЕ ТОГО, КОМАНДА RST ЗАНИМАЕТ МЕНЬШЕ ВРЕМЕНИ, ЧЕМ КОМАНДА CALL,

НЕДОСТАТКОМ КОМАНДЫ RST ЯВЛЯЕТСЯ ТО, ЧТО ЕЕ МОЖНО ПРИМЕНЯТЬ ТОЛЬКО ДЛЯ ОБРАЩЕНИЯ К ОДНОЙ ИЗ ВОСЬМИ ПРИВЕДЕННЫХ ВЫШЕ ДОПУСИМЫХ ЯЧЕЕК,

ПОСКОЛЬКУ ВСЕ ЭТИ ЯЧЕЙКИ РАСПОЛОЖЕНЫ В ПЗУ, ВЫ МОЖЕТЕ ВОСПОЛЬЗОВАТЬСЯ ЭТИМ ПРЕИМУЩЕСТВОМ В СВОЕЙ СОБСТВЕННОЙ ПРОГРАММЕ, ЕСТЬ, ОДНАКО, ВОЗМОЖНОСТЬ ИСПОЛЬЗОВАТЬ ПОДПРОГРАММЫ ПЗУ, ЕСЛИ ВЫ ЗНАЕТЕ, ЧТО ОНИ ДЕЛАЮТ, И ТЕМ САМЫМ ИСПОЛЬЗОВАТЬ КОМАНДЫ RST,

ВЫ СМОЖЕТЕ БОЛЬШЕ УЗНАТЬ О КОМАНДАХ RST ИЗ НАШЕЙ КНИГИ " UNDERSTANDING YOUR SPECTRUM", НАПИСАННОЙ ДОКТОРОМ ЯНОМ ЛОГАНОМ,

Н А П И С А Н И Е П Р О Г Р А М М Д Л Я "С П Е К Т Р У М

ПЛАНИРОВАНИЕ ВАШЕЙ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ

ПРОГРАММИРОВАНИЕ НА МАШИННОМ ЯЗЫКЕ ОБЛАДАЕТ ЧЕРЕЗВЫЧАЙНОЙ ГИБКОСТЬЮ В ТОМ СМЫСЛЕ, ЧТО ПОЗВОЛЯЕТ ВАМ ДЕЛАТЬ ВСЕ, ЧТО УГОДНО,

ПОСКОЛЬКУ ОТ ВСЕХ ЯЗЫКОВ БОЛЕЕ ВЫСОКОГО УРОВНЯ В КОНЕЧНОМ СЧЕТЕ НУЖНО ПЕРЕХОДИТЬ К МАШИННОМУ ЯЗЫКУ, ВСЕ, ЧТО ВЫ МОЖЕТЕ ЗАПРОГРАММИРОВАТЬ НА ЯЗЫКЕ "ФОРТРАН" ИЛИ "КОБОЛ", ИЛИ ЛЮБОМ ДРУГОМ, МОЖНО ЗАПРОГРАММИРОВАТЬ НА МАШИННОМ ЯЗЫКЕ,

ДОПОЛНИТЕЛЬНЫМ ПРЕИМУЩЕСТВОМ БУДЕТ ТО, ЧТО ПРОГРАММА НА МАШИННОМ ЯЗЫКЕ ВЫПОЛНЯЕТСЯ БЫСТРЕЕ,

ЭТА АБСОЛЮТНАЯ ГИБКОСТЬ МОЖЕТ ,ОДНАКО, ОКАЗАТЬСЯ ЛОВУШКОЙ ДЛЯ НЕОСТОРОЖНОГО ПРОГРАММИСТА, ПРИ НАЛИЧИИ ТАКОЙ ПОЛНОЙ СВОБОДЫ МОЖНО ДЕЛАТЬ ВСЕ, ЧТО УГОДНО, В ОТЛИЧИЕ ОТ ОПЕРАЦИОННОЙ СИСТЕМЫ "СПЕКТРУМ" ДЛЯ ЯЗЫКА "БЕЙСИК", НАПРИМЕР, ОТСУТСТВУЮТ ПРОВЕРКИ ПРЕДЛОЖЕНИЙ НА ДОПУСТИМОСТЬ,

ПОСКОЛЬКУ ЛЮБЫЕ ВВОДИМЫЕ ВАМИ ЧИСЛА БУДУТ КОМАНДАМИ ТОГО ИЛИ ИНОГО ТИПА, ЧИП Z80 БУДЕТ ОБРАБАТЫВАТЬ ВСЕ НА СВЕТЕ,

НО ДАЖЕ НЕ БЕРЯ В РАСЧЕТ ПРОБЛЕМЫ ПРОВЕРКИ ДОПУСТИМОСТИ СИНТАКСИСА, НУЖНО ОТМЕТИТЬ, ЧТО ПРОГРАММИРОВАНИЕ НА МАШИННОМ ЯЗЫКЕ НЕ НАКЛАДЫВАЕТ ОГРАНИЧЕНИЙ НА ИСПОЛЬЗУЕМУЮ ВАМИ ЛОГИКУ: ВЫ МОЖЕТЕ ВЫПОЛНЯТЬ ФУНКЦИИ, ПЕРЕХОДЫ И Т,П,, КОТОРЫЕ БУДУТ АБСОЛЮТНО НЕДОПУСТИМЫ В ЛЮБОМ ЯЗЫКЕ БОЛЕЕ ВЫСОКОГО УРОВНЯ,

ПОЭТОМУ ОГРОМНОЕ ЗНАЧЕНИЕ ПРИОБРЕТАЕТ САМОДИСЦИПЛИНА ПРИ РАЗРАБОТКЕ ПРОГРАММ НА МАШИННОМ ЯЗЫКЕ, НЕВОЗМОЖНО ПРЕУВЕЛИЧИТЬ ЗНАЧЕНИЕ КОНЦЕПЦИИ "НИСХОДЯЩЕГО" ПОДХОДА В ПРОГРАММИРОВАНИИ В ЦЕЛОМ, НО В ОСОБЕННОСТИ ЭТО КАСАЕТСЯ ПРОГРАММИРОВАНИЯ НА МАШИН НОМ ЯЗЫКЕ,

"НИСХОДЯЩИЙ" ПОДХОД ЗАСТАВЛЯЕТ ВАС РАЗБИВАТЬ ЗАДАЧУ НА БОЛЕ МЕЛКИЕ ЕДИНИЦЫ И ПОЗВОЛЯЕТ ПРОВЕРИТЬ ЛОГИКУ ВАШЕЙ РАЗРАБОТКИ, НА ПРОТЯЖЕНИИ ДОЛГОГО ВРЕМЕНИ НЕ ЗАПИСЫВАЯ НИ ОДНОЙ СТРОКИ ТЕКСТА ПРОГРАММЫ,

ПРЕДПОЛОЖИМ, ВАМ ЗАХОТЕЛОСЬ НАПИСАТЬ ПРОГРАММУ ПОСАДКИ НА ЛУНУ:

САМЫЙ ПЕРВЫЙ ПУТЬ МОГ БЫ НАПОМНИТЬ НЕЧТО ПОДОБНОЕ:

```
INSTR   ВЫДАЧА НА ЭКРАН ИНСТРУКЦИЙ
        ПЕРЕХОД НАЗАД НА INSTR,ПОКА НЕ БУДЕТ НАЖАТА КЛАВИША
        ENTER
DRAW    НАРИСОВАТЬ ЛАНДШАФТ, НАЧАТЬ ДВИЖЕНИЕ СПУСКАЕМОГО
        АППАРАТА С ВЕРШИНЫ ЭКРАНА
LAND    ДВИЖЕНИЕ СПУСКАЕМОГО АППАРАТА
        ЕСЛИ ГОРЮЧЕЕ КОНЧИЛОСЬ, ПЕРЕЙТИ НА CRASH
        ПЕРЕЙТИ НАЗАД НА LAND, ЕСЛИ ПОВЕРХНОСТЬ НЕ
        ДОСТИГНУТА
GROUND  НАПЕЧАТАТЬ ПОЗДРАВЛЕНИЯ
        ПЕРЕЙТИ НАЗАД НА INSTR ДЛЯ СЛЕДУЮЩЕГО ПРОГОНА
CRASH   НАПЕЧАТАТЬ СОБОЛЕЗНОВАНИЯ ПО ПОВОДУ НЕУДАЧНОЙ ПО-
        САДКИ
```

ПЕРЕЙТИ НАЗАД НА INSTR ДЛЯ СЛЕДУЮЩЕГО ПРОГОНА,

ОБРАТИТЕ ВНИМАНИЕ, ЧТО ВСЯ ЭТА "ПРОГРАММА" НАПИСАНА НА РУССКОМ ЯЗЫКЕ, НА ЭТОМ ЭТАПЕ НЕ ПРИНИМАЛОСЬ НИКАКИХ РЕШЕНИЙ, БУДЕТ ЛИ ПРОГРАММА ПИСАТЬСЯ НА ЯЗЫКЕ "БЕЙСИК" ИЛИ НА МАШИННОМ ЯЗЫКЕ, ТАКОЕ РЕШЕНИЕ И НЕ НУЖНО ПРИНИМАТЬ - КОНЦЕПЦИЯ ПРОГРАММЫ ПОСАДКИ НА ЛУНУ НЕ ЗАВИСИТ ОТ СПОСОБА ЗАПИСИ ЕЕ ТЕКСТА,

ТЕПЕРЬ НАСТУПАЕТ ЭТАП ЛОГИЧЕСКОЙ ПРОВЕРКИ,

ВЫ ВЫПОЛНЯЕТЕ РОЛЬ ЭВМ И СМОТРИТЕ, ВСЕ ЛИ ВОЗМОЖНОСТИ, КОТОРЫЕ ВЫ ХОТЕЛИ ВКЛЮЧИТЬ В ПРОГРАММУ , ИМЕЮТСЯ В НАЛИЧИИ,

НЕТ ЛИ ПЕРЕХОДОВ НА ОБ"ЕКТЫ, КОТОРЫЕ ВЫ ХОТЕЛИ НАПИСАТЬ, НО ЗАБЫЛИ ? ВСЕ ЛИ ЕСТЬ ? НЕТ ЛИ ИЗБЫТОЧНЫХ ПРОГРАММ ? НЕ СЛЕДУЕТ ЛИ НЕКОТОРЫЕ ОБ"ЕКТЫ ПЕРЕНЕСТИ В ПОДПРОГРАММЫ ?

ДАВАЙТЕ СНОВА ПОСМОТРИМ НА "ПРОГРАММУ" - ОХ-ОХ-ОХ! - МЫ ЗАБЫЛИ КАК-НИБУДЬ ЗАКОНЧИТЬ ПРОГРАММУ!

ОПИСАННАЯ ВЫШЕ ЛОГИКА МОЖЕТ БЫТЬ ПРЕКРАСНОЙ ДЛЯ НЕКОТОРЫХ ПРИКЛАДНЫХ ЗАДАЧ, ТАКИХ КАК ИГРОВАЯ ЭВМ, НО В СВОЕЙ ПРОГРАММЕ ВАМ МОЖЕТ ПРИЙТИ В ГОЛОВУ, ЧТО НЕПЛОХО БЫ ИМЕТЬ ВОЗМОЖНОСТЬ ВЫКЛЮЧИТЬ РАБОТУ ПРОГРАММЫ,

ТЕПЕРЬ МЫ ИЗМЕНИМ ПОСЛЕДНЮЮ ЧАСТЬ ПРОГРАММЫ СЛЕДУЮЩИМ ОБРАЗОМ:

```
GROUND  НАПЕЧАТАТЬ ПОЗДРАВЛЕНИЯ
        ПЕРЕЙТИ НА
CRASH   НАПЕЧАТАТЬ СОБОЛЕЗНОВАНИЯ ПО ПОВОДУ НЕУДАЧНОЙ ПО-
        САДКИ
FINISH  СПРОСИТЬ ИГРОКА ,НУЖНО ЛИ ЗАКОНЧИТЬ
        ЕСЛИ НЕТ, ПЕРЕЙТИ НА INSTR
        ЕСЛИ ДА, STOP
```

ОБРАТИТЕ ВНИМАНИЕ, ЧТО МЫ ИСПОЛЬЗОВАЛИ МЕТКИ ДЛЯ ОПИСАНИЯ ОПЕРЕДЕЛЕННЫХ СТРОК ПРОГРАММЫ, МЕТКИ - ОЧЕНЬ ЦЕННЫЙ АППАРАТ, ОСОБЕННО ЕСЛИ ВЫ БУДЕТЕ ВЫБИРАТЬ ИХ КРАТКИМИ И СОДЕРЖАТЕЛЬНЫМИ

КОГДА ЭТОТ УРОВЕНЬ ЗАКОНЧЕН, ВЫ ПЕРЕМЕЩАЕТЕСЬ НА УРОВЕНЬ ГЛУБЖЕ, ЧТОБЫ ПРОДЕЛАТЬ ТО ЖЕ САМОЕ С ОДНОЙ ИЗ СТРОК ИЛИ МОДУЛЕЙ, ПРИВЕДЕННЫХ ВЫШЕ,

ИМЕННО ПОЭТОМУ ЭТОТ ПОДХОД НАЗЫВАЕТСЯ НИСХОДЯЩИМ,

НАПРИМЕР, МЫ МОЖЕМ СЛЕДУЮЩИМ ОБРАЗОМ РАСПИСАТЬ МОДУЛЬ "FINISH", ПРИВЕДЕННЫЙ ВЫШЕ:

```
FINISH  ОЧИСТИТЬ ЭКРАН
        НАПЕЧАТАТЬ: "ХОТИТЕ ЛИ ЗАКОНЧИТЬ?"
        ОПРОСИТЬ КЛАВИАТУРУ В ОЖИДАНИИ ОТВЕТА
        ЕСЛИ ОТВЕТ = ДА, ТО ЗАКОНЧИТЬ
        ПЕРЕЙТИ НА INSTR
```

ЕЩЕ ОДНО ПРЕИМУЩЕСТВО НИСХОДЯЩЕГО ПОДХОДА СОСТОИТ В ТОМ, ЧТО ВЫ МОЖЕТЕ ТЕСТИРОВАТЬ И ВЫПОЛНЯТЬ КОНКРЕТНЫЙ МОДУЛЬ АВТОНОМНО, ТАК ЧТО ОН БУДЕТ ОТЛАЖЕН ДЛЯ ВКЛЮЧЕНИЯ В ОКОНЧАТЕЛЬНЫЙ ТЕКСТ ПРОГРАММЫ,

ДАВАЙТЕ СПУСТИМСЯ ЕЩЕ НА ОДИН УРОВЕНЬ И ПОСМОТРИМ НА СТРОКУ ОЧИСТИТЬ ЭКРАН БОЛЕЕ ПОДРОБНО,

НА ЭТОМ ЭТАПЕ НАМ НУЖНО РЕШИТЬ, НА КАКОМ ЯЗЫКЕ МЫ БУДЕМ ПИСАТЬ ПРОГРАММУ, И ДАВАЙТЕ ВЫБЕРЕМ МАШИННЫЙ ЯЗЫК "СИНКЛЕРА"

,

ЕСЛИ БЫ ВЫ ПИСАЛИ ПРОГРАММУ НА ЯЗЫКЕ "БЕЙСИК", ТО ВАМ БЫЛО БЫ ДОСТАТОЧНО НАПИСАТЬ:

```
        900 CLS,
```

НО НА МАШИННОМ ЯЗЫКЕ ЭТО ПРОСТОЕ ПРЕДЛОЖЕНИЕ "ОЧИСТИТЬ ЭКРАН" МОЖЕТ ОКАЗАТЬСЯ ОБМАНЧИВЫМ,) (ТАК В ОРИГИНАЛЕ, ОТКРЫВАЮЩЕЙ СКОБКИ НЕТ, (ПРИМЕЧ, ПЕР.))

ПОЭТОМУ МЫ МОГЛИ БЫ СДЕЛАТЬ ЧТО-ТО В ТАКОМ РОДЕ:

```
CLEAR   НАЙТИ НАЧАЛО ЭКРАНА
        ЗАПОЛНИТЬ СЛЕДУЮЩИЕ 6144 ПОЗИЦИИ ПРОБЕЛАМИ
```

МЫ ВСЕ ЕЩЕ НЕ НАПИСАЛИ НИ СТРОЧКИ ТЕКСТА ПРОГРАММЫ, НО , ОЧЕВИДНО, ПОДХОД ОСНОВАН НА МАШИННОМ ЯЗЫКЕ, ДАВАЙТЕ ПОСМОТРИМ БОЛЕЕ ПРИСТАЛЬНО, ЧТО ДОЛЖНА ДЕЛАТЬ ЭТА ПРОГРАММА ОЧИСТКИ ЭКРАНА И ЧТО ОНА ДЕЛАЕТ НА САМОМ ДЕЛЕ,

ВЫ, ВОЗМОЖНО, ПОМНИТЕ ИЗ РУКОВОДСТВА ПО "СПЕКТРУМ", ЧТО ЭКРАН СОСТОИТ ИЗ 6144 ЯЧЕЕК И ЧТО ЕСТЬ ЕЩЕ 768 ЯЧЕЕК, ОПИСЫ-

ВАЮЩИХ АТРИБУТЫ ЭКРАНА: ЦВЕТ БУМАГИ, ЦВЕТ ЧЕРНИЛ И Т.П.

ПРИВЕДЕННОЕ ВЫШЕ КРАТКОЕ ОПИСАНИЕ ПРОГРАММЫ, КОНЕЧНО, ОЧИСТИТ ЧАСТЬ ЭКРАНА, НО НИКАК НЕ ПОВЛИЯЕТ НА ФАЙЛ АТРИБУТОВ, ЕСЛИ НЕ У ВСЕХ ПОЗИЦИЙ ЭКРАНА СОВПАДАЕТ ЦВЕТ БУМАГИ ИЛИ ЕСЛИ ПОЗИЦИИ НЕКОТОРЫХ ЛИТЕР ИМЕЮТ ВКЛЮЧЕННЫЕ АТРИБУТЫ МИГАНИЯ ИЛИ ЯРКОСТИ, ТО ЯСНАЯ ПРОГРАММА ОЧИСТКИ ЭКРАНА, ПРИВЕДЕННАЯ ВЫШЕ, ОКАЖЕТСЯ ЯВНО НЕАДЕКВАТНОЙ.

НАМ ПРИДЕТСЯ ТАКЖЕ ОБРАБОТАТЬ И ФАЙЛ АТРИБУТОВ, (ОБРАТИТЕ ВНИМАНИЕ, НАСКОЛЬКО СЛОЖНЕЕ ОКАЗЫВАЮТСЯ НЕКОТОРЫЕ ЗАДАЧИ НА МАШИННОМ ЯЗЫКЕ, ЧЕМ НА ЯЗЫКЕ "БЕЙСИК")

ПОЭТОМУ НАМ НУЖНО РАСШИРИТЬ ПРОГРАММУ ДО СЛЕДУЮЩЕГО ВИДА

```
НАЙТИ НАЧАЛО ЭКРАНА
ЗАПОЛНИТЬ СЛЕДУЮЩИЕ 6144 БАЙТА ПРОБЕЛАМИ
НАЙТИ НАЧАЛО ФАЙЛА АТРИБУТОВ
ЗАПОЛНИТЬ СЛЕДУЮЩИЕ 768 БАЙТОВ ТЕБУЕМЫМИ ЗНАЧЕНИЯМИ
АТРИБУТОВ БУМАГИ (ЧЕРНИЛ)
```

СЛЕДУЮЩИЙ НИЖЕЛЕЖАЩИЙ УРОВЕНЬ - ЭТО УЖЕ ТОТ, НА КОТОРОМ ВЫ ДОЛЖНЫ, НАКОНЕЦ, ПИСАТЬ ТЕКСТ ПРОГРАММЫ, ТАК ЧТО ДАВАЙТЕ ПОСМОТРИМ, КАК ЭКРАН ЗАПОЛНЯЕТСЯ ПРОБЕЛАМИ:

```
CLEAR   LD HL,SCREEN      : SCREEN START
        LD BC,6144        : BYTES TO CLEAR
        LD D,0            : D=BLANK
LOOP    LD (HL),D         : FILL  BLANK
        INC HL            : NEXT POSITION
        DEC BC            : REDUCE COUNT
        LD A,B
        OR C              : TEST IF BC=0
        JR NZ,LOOP        : AGAIN IF NOT END
```

SCREEN START-НАЧАЛО ЭКРАНА; BYTES TO CLEAR - ОЧИЩАЕМЫЕ БАЙТЫ; BLANK - ПРОБЕЛ; FILL BLANK - ЗАПОЛНЕНИЕ ПРОБЕЛОМ; NEXT POSITION - СЛЕДУЮЩАЯ ПОЗИЦИЯ; REDUCE COUNT - УМЕНЬШЕНИЕ СЧЕТЧИКА; TEST - ПРОВЕРКА; AGAIN IF NOT END - ПОВТОРЕНИЕ, ЕСЛИ НЕ ДОСТИГНУТ КОНЕЦ.

ТЕПЕРЬ ВЫ ДОСТАТОЧНО ЛЕГКО МОЖЕТЕ РАБОТАТЬ С ПРОГРАММАМИ ТАКОЙ ДЛИНЫ И ТАКИМ СПОСОБОМ СТРОИТЬ ДОСТАТОЧНО БОЛЬШИЕ ПРОГРАММЫ.

КСТАТИ, ВЫ ТЕПЕРЬ БЕЗ СОМНЕНИЯ ПОНИМАЕТЕ, ПОЧЕМУ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ ЧАСТО ТАК ВЕЛИКИ ПО ОБ"ЕМУ И ПОЧЕМУ ЛЮДИ ИЗОБРЕЛИ ПРОГРАММЫ НА ЯЗЫКАХ ВЫСОКОГО УРОВНЯ!

О Т В Е Т Ы

ЕСТЬ НЕСКОЛЬКО "ПРАВИЛЬНЫХ" ОТВЕТОВ - ЕДИНСТВЕННАЯ ПРОВЕРКА БУДЕТ ЛИ ЭТО РАБОТАТЬ ? ИНЫМИ СЛОВАМИ, ДЕЛАЕТ ЛИ ПРОГРАММА ТО, ЧТО ВАМ НУЖНО ?

С ПОМОЩЬЮ "DJNZ":

```
CLEAR   LD HL,SCREEN
        LD A,0
        LD B,24           : SET B=24
BIGLOOP PUSH BC           : SAVE VALUE
        LD B,A            : SET B=256
LITLOOP LD (HL),A         :
        INC HL            : FILL IN 256 BLANKS
        DJNZ LITLOOP
        POP BC            : GET BACK VALUE OF B
        DJNZ BIGLOOP      : DO IT UNTIL END
```

SET - УСТАНОВИТЬ; SAVE VALUE - ЗАПОМНИТЬ ЗНАЧЕНИЕ; FILL IN 256 BLANKS - ЗАПОЛНИТЬ 256 ПРОБЕЛОВ; GET BACK VALUE OF B - ПОЛУЧИТ НАЗАД ЗНАЧЕНИЕ B; DO IT UNTIL END - ВЫПОЛНЯТЬ ДО ДОСТИЖЕНИЯ КОНЦА.

МЫ СМОГЛИ ИСПОЛЬЗОВАТЬ 24 РАЗА ПО 256 (=6144) ДЛЯ ОЧИСТКИ ЭКРАНА.

НУЖНО ОТМЕТИТЬ СЛЕДУЮЩЕЕ:

МЫ МОЖЕМ ЗАДАТЬ B = 0, ЧТОБЫ ПРОЙТИ ЦИКЛ DJNZ 256 РАЗ, (ПОЧЕМУ?) ЭТА ПРОЦЕДУРА ОБЫЧНО НЕ БУДЕТ ПРИМЕНЯТЬСЯ В ПРОГРАММЕ, ЕСЛИ МЫ ТОЛЬКО НЕ СТАНЕМ ИСПОЛЬЗОВАТЬ РЕГИСТР C ДЛЯ ДРУГИХ ЦЕЛЕЙ.

ПРИМЕНЕНИЕ LDIR:

```
CLEAR  LD HL,SCREEN       : SOURCE
```

```
        PUSH HL
        POP DE
        INC DE              : DEST = HL + 1
        LD BC,6144          : HOW MANY
        LD (HL),Ø           : LAT POS = Ø
        LDIR                : MOVE IT
```

SOURCE - ИСТОЧНИК; HOW - СКОЛЬКО; LAT - ПЕРВАЯ; MOVE - ПЕРЕМЕЩЕНИЕ.

ОБРАТИТЕ ВНИМАНИЕ, ЧТО МЫ ПОЛУЧИЛИ DE =HL + 1, ЗАДАВАЯ DE= HL И ДАВАЯ ПРИРАЩЕНИЕ DE, ЭТО МОЖНО СДЕЛАТЬ ПРОЩЕ, ЗАГРУЖАЯ ЗНАЧЕНИЕ SCREEN + 1 В DE НЕПОСРЕДСТВЕННО, НО ДЛЯ ЭТОГО ТРЕБУЕТСЯ НА ОДИН БАЙТ БОЛЬШЕ!

ПРИЧИНА, ПО КОТОРОЙ ЭТА КОМАНДА LDIR СРАБАТЫВАЕТ, СОСТОИТ В ТОМ, ЧТО ИСПОЛЬЗУЕТСЯ ФАКТ, ЧТО В ПРОЦЕССЕ ОБРАБОТКИ ДАННЫЕ ПЕРЕЗАПИСЫВАЮТСЯ В БЛОК, ЗДЕСЬ ПРОИСХОДИТ ПРИМЕНЕНИЕ С ПОЛОЖИТЕЛЬНЫМ РЕЗУЛЬТАТОМ ЗАДАЧИ, РАССМОТРЕННОЙ НАМИ В ГЛАВЕ О ПЕРЕМЕЩЕНИИ БЛОКОВ.

ЕСЛИ ПРОСУММИРОВАТЬ ПОТРЕБНУЮ ПАМЯТЬ, ТО ЛРИ ПЕРВОМ МЕТОДЕ ТРЕБУЕТСЯ 14 БАЙТОВ, ПРИ ВТОРОМ - 16 БАЙТОВ, А ПРИ ПОСЛЕДНЕМ 13 БАЙТОВ.

С Р Е Д С Т В А "ZX С П Е К Т Р У М"

НАСТАЛО ВРЕМЯ РАССМОТРЕТЬ СРЕДСТВА ВАШЕГО "ZX СПЕКТРУМ", ПОЛЕЗНЫЕ ПРИ РАЗРАБОТКЕ ДЛЯ НЕГО ПРОГРАММ НА МАШИННОМ ЯЗЫКЕ.

В В О Д - К Л А В И А Т У Р А

ЧТО КАСАЕТСЯ ВВОДА ИНФОРМАЦИИ В "СПЕКТРУМ", МЫ БУДЕМ ИГНОРИРОВАТЬ ВВОД С КАССЕТНОГО МАГНИТОФОНА И СКОНЦЕНТРИРУЕМСЯ НА КЛАВИАТУРЕ.

КЛАВИАТУРА - ЕДИНСТВЕННЫЙ ИСТОЧНИК ИНФОРМАЦИИ, ПРЕДОСТАВЛЯЮЩИЙ СВЯЗЬ, В РЕАЛЬНОМ МАСШТАБЕ ВРЕМЕНИ, ОНА МОЖЕТ ДИНАМИЧЕСКИ ВЛИЯТЬ НА ВЫПОЛНЕНИЕ ЛЮБОЙ ПРОГРАММЫ, КАК ОПЕРАЦИОННОЙ СИСТЕМЫ В ПЗУ, ТАК И ПОЛЬЗОВАТЕЛЬСКОЙ ПРОГРАММЫ В ПАМЯТИ С ПРОИЗ ВОЛЬНЫМ ДОСТУПОМ.

ЛОГИЧЕСКИ МЫ МОЖЕМ РАССМАТРИВАТЬ КЛАВИАТУРУ КАК ДВУМЕРНУЮ МАТРИЦУ С ВОСЕМЬЮ РЯДАМИ И ПЯТЬЮ СТОЛБЦАМИ, КАК В ПРИЛОЖЕНИИ А.

КАЖДОЕ ИЗ 40 ПЕРЕСЕЧЕНИЙ ПРЕДСТАВЛЯЕТ КЛАВИШУ КЛАВИАТУРЫ, В НОРМАЛЬНОМ СОСТОЯНИИ (КОГДА ОНИ НЕ НАЖАТЫ) КАЖДОЕ ПЕРЕСЕЧЕНИЕ УСТАНАВЛИВАЕТСЯ РАВНЫМ 1.

ПРИ НАЖАТИИ КОНКРЕТНОЙ КЛАВИШИ "НАЖАТОЕ" ПЕРЕСЕЧЕНИЕ, СООТВЕТСТВУЮЩЕЕ ЭТОЙ КЛАВИШЕ, БУДЕТ СБРОШЕНО В Ø.

ЗНАЯ СВЯЗЬ МЕЖДУ КЛАВИАТУРОЙ И ЭТОЙ ВНУТРЕННЕЙ МАТРИЦЕЙ ПРЕДСТАВЛЕНИЯ, МЫ МОЖЕМ ВЫВЕСТИ ЛОГИЧЕСКИЙ СПОСОБ ПРОВЕРКИ НАЖАТИЯ КЛАВИШИ, КОТОРЫЙ МОЖНО ПРИМЕНЯТЬ В ПРОГАММИРОВАНИИ НА МАШИННОМ ЯЗЫКЕ.

В ЯЗЫКЕ "БЕЙСИК" ПРИ СКАНИРОВАНИИ КЛАВИАТУРЫ НАМ НУЖНО ЗАДАТЬ АДРЕС ТОЙ КОНКРЕТНОЙ ПОЛОВИНЫ РЯДА КЛАВИАТУРЫ, ГДЕ РАСПОЛАГАЕТСЯ НУЖНАЯ КЛАВИША, ПРЕЖДЕ ЧЕМ ИСПОЛЬЗОВАТЬ ФУНКЦИЮ IN.

АНАЛОГИЧНЫМ ОБРАЗОМ В ПРОГРАММЕ НА МАШИННОМ ЯЗЫКЕ МЫ ДОЛЖНЫ ЗАГРУЗИТЬ В НАКОПИТЕЛЬ ЗНАЧЕНИЕ, СООТВЕТСТВУЮЩЕЕ АДРЕСУ ПОЛУРЯДА КЛАВИШ, КОТОРЫЙ МЫ ХОТИМ ПРОВЕРИТЬ, ТРЕБУЕМОЕ ЗНАЧЕНИЕ ДЛЯ КАЖДОЙ ПОЛОВИНЫ РЯДА ПРИВОДИТСЯ В САМОЙ ЛЕВОЙ КОЛОНКЕ ТАБЛИЦЫ В ПРИЛОЖЕНИИ А.

НАПРИМЕР, ДЛЯ "H - ENTER" (ПОЛУРЯДА) МЫ ЗАГРУЖАЕМ В РЕГИСТР А ЗНАЧЕНИЕ

```
LD A,BFH,
```

ЗАТЕМ ЗНАЧЕНИЕ А БУДЕТ ИСПОЛЬЗОВАТЬСЯ ДЛЯ ПОИСКА БАЙТА, СОДЕРЖАЩЕГО СОСТОЯНИЕ ТОЙ КОНКРЕТНОЙ ПОЛОВИНЫ РЯДА КЛАВИШ И ВОЗВРАТА В А ПРИ ЗАДАНИИ КОМАНДЫ INPUT.

НАПРИМЕР, ИСПОЛЬЗУЕТСЯ ПОРТ FEH

```
IN A,(FEH),
```

ПОСКОЛЬКУ В ПОЛОВИНЕ РЯДА ИМЕЕТСЯ ПЯТЬ КЛАВИШ, НАС ИНТЕРЕСУЮТ ТОЛЬКО ПЯТЬ МЛАДШИХ БИТОВ БАЙТА, ВОЗВРАЩАЕМОГО В РЕГИСТРЕ А.

ЕСЛИ В ЭТОЙ ПОЛОВИНЕ РЯДА НИКАКИЕ КЛАВИШИ НАЖАТЫ НЕ БЫЛИ, ТО ЗНАЧЕНИЕ ПЯТИ МЛАДШИХ БИТОВ БУДЕТ (2**4 + 2**3 + 2**2 + 2**1 +

2**0 T,E, 16 + 8 + 4 + 2 + 1 = 31),
РЕГИСТР A = XXX11111, КОГДА НЕТ НАЖАТЫХ КЛАВИШ,
ЕСЛИ МЫ ХОТИМ ПРОВЕРИТЬ, НАЖАТ ЛИ САМЫЙ ПЕРВЫЙ БИТ, ТО МЫ ПРОВЕРЯЕМ, СБРОШЕН ЛИ ОН,
ЕСТЬ ДВА СПОСОБА ПРОВЕРИТЬ ЭТО:
1, С ПОМОЩЬЮ КОМАНДЫ ПРОВЕРКИ БИТА, НАПРИМЕР BIT 0,A
ЕСЛИ БИТ СБРОШЕН (НЕ УСТАНОВЛЕН), ТО БУДЕТ УСТАНОВЛЕН ФЛАГ НУЛЯ,
2, С ПОМОЩЬЮ КОМАНД ЛОГИЧЕСКОГО И AND 1
ЕСЛИ БИТ СБРОШЕН (НЕ УСТАНОВЛЕН), ТО РЕЗУЛЬТАТ БУДЕТ НУЛЕВЫМ И ФЛАГ НУЛЯ БУДЕТ УСТАНОВЛЕН,
ПЕРВЫЙ СПОСОБ ПРОЩЕ, ПОСКОЛЬКУ КОНКРЕТНЫЙ БИТ, ПОДЛЕЖАЩИЙ ПРОВЕРКЕ, УКАЗАН НЕПОСРЕДСТВЕННО В КОМАНДЕ ПРОВЕРКИ БИТА, ЕГО НЕДОСТАТОК СОСТОИТ В ТОМ, ЧТО ЕСЛИ НАМ НУЖНО ПРОВЕРИТЬ ДВЕ КЛАВИШИ В ЭТОЙ ПОЛОВИНЕ РЯДА, НАМ ПРИДЕТСЯ ПРИМЕНЯТЬ ДВЕ КОМАНДЫ ПРОВЕРКИ БИТА, И, ВОЗМОЖНО, ДВА ОТНОСИТЕЛЬНЫХ ПЕРЕХОДА,
НАПРИМЕР, ЧТОБЫ ПРОВЕРИТЬ БИТЫ 0 И 1 С ПОМОЩЬЮ ПЕРВОГО СПОСОБА

```
        BIT 0,A          : TEST BIT 0 OF A SET OR NOT
         JR Z,NPRESS     : JUMP IF NOT PRESSED
        BIT 1,A          : TEST BIT 1 OF A SET OR NOT
         JR Z,NPRESS     * JUMP IF NOT PRESSED
              ,
              ,
         DO WHATEVER IF BOTH ARE PRESSED
              ,
     NPRESS ,
              ,
```

TEST BIT 0 OF SET OR NOT - ПРОВЕРИТЬ, УСТАНОВЛЕН ИЛИ НЕТ БИТ 0 РЕГИСТРА A; JUMP IF NOT PRESSED - ПЕРЕХОД, ЕСЛИ НЕ НАЖАТА; TEST BIT 1 OF SET OR NOT - ПРОВЕРИТЬ, УСТАНОВЛЕН ИЛИ НЕТ БИТ 1 РЕГИСТРА A; DO WHATEVER IF BOTH ARE PRESSED - ВЫПОЛНИТЬ ТО, ЧТ СЛЕДУЕТ, ЕСЛИ ОБЕ КЛАВИШИ НАЖАТЫ,
ВТОРОЙ СПОСОБ ПРОВЕРКИ С ПОМОЩЬЮ ЛОГИЧЕСКОГО И ТРЕБУЕТ НЕСКОЛЬКО БОЛЬШЕ ЛОГИЧЕСКИХ УХИЩРЕНИЙ, ДЛЯ ПРОВЕРКИ БИТА 0 МЫ ИСПОЛЬЗУЕМ "AND 1" ; ДЛЯ ПРОВЕРКИ БИТА 1 - "AND 2"; ДЛЯ ПРОВЕРКИ БИТА 2 - "AND 4" И Т,Д,
ДЛЯ ПРОВЕРКИ ДВУХ КЛАВИШ МЫ ПРИМЕНЯЕМ "AND X", ГДЕ X - СУММ ЗНАЧЕНИЙ, ИСПОЛЬЗУЕМЫХ ДЛЯ ПРОВЕРКИ КАЖДОЙ ИЗ КЛАВИШ В ОТДЕЛЬНОСТИ,
НАПРИМЕР, ЧТОБЫ ПРОВЕРИТЬ, ЧТО БИТ 0 И БИТ 1 РЕГИСТРА A УСТАНОВЛЕНЫ:

```
        AND  1           : TEST BOTH BIT 0 AND BIT 1 IS SET
         CP  1           : TEST IF BOTH SET
         JK  NZ,NBOTH    : JUMP IF NOT BOTH PRESSED
```

TEST BOTH BIT 0 AND BIT 1 IS SET - ПРОВЕРИТЬ, ЧТО ОБА БИТА 0 И 1 УСТАНОВЛЕНЫ; TEST IF BOTH SET - УСТАНОВЛЕНЫ ЛИ; JUMP IF NOT BOTH PRESSED - ПЕРЕХОД, ЕСЛИ НЕ ОБЕ КЛАВИШИ НАЖАТЫ,
ЧТОБЫ ПРОВЕРИТЬ, ЧТО УСТАНОВЛЕН ОДИН ИЗ БИТОВ 0 ИЛИ 1 РЕГИСТРА A,

```
        AND  3           : TEST ELTHER BIT 0 AND BIT 1 IS SET
         JR  Z,NOTONE    : JUMP IF NOT ONE IS PRESSED
              ,
              ,
```

TEST ELTHER BIT 0 AND BIT 1 IS SET - ПРОВЕРИТЬ, ЧТО ОДИН ИЗ БИТОВ 0 ИЛИ 1 УСТАНОВЛЕН; JUMP IF NOT ONE IS PRESSED - ПЕРЕХОД, ЕСЛИ НИ ОДНА КЛАВИША НЕ НАЖАТА,
ПРИВОДИМАЯ НИЖЕ ПРОГРАММА НА МАШИННОМ ЯЗЫКЕ ПОКАЗЫВАЕТ СПОСОБ ОРГАНИЗАЦИИ ПАМЯТИ ЭКРАНА В "СПЕКТРУМ": 210040 LD HL, 4000H : LOAD HL WITH START OF DISPLAY FILE 36FF LD (HL),FFH ; FILL THET SCREEN LOCATION 110140 LD DE,4001H : LOAD DE WITH BYTE IN DISPLAY 010100 LD BC,1 : BC CONTAINS NUMBER OF BYTES TO BE
: TRANSFERRED EDB0 LDIR : MOVE A BLOCK LENGTH BC FROM (HL) : TO (DE) C9 RET : END OF PROGRAM
LOAD HL WITH START OF DISPLAY FILE - ЗАГРУЗИТЬ В HL НАЧАЛО

ФАЙЛА ДИСПЛЕЯ; FILL THET SCREEN LOCATION - ЗАПОЛНИТЬ ЭТУ ЯЧЕЙКУ ЭКРАНА; LOAD DE WITH BYTE IN DISPLAY - ЗАГРУЗИТЬ В DE СЛЕДУЮЩИЙ БАЙТ ДИСПЛЕЯ; BC CONTAINS NUMBER OF BYTES TO BE TRANSFERRED В BC СОДЕРЖИТСЯ ЧИСЛО ПЕРЕДАВАЕМЫХ БАЙТОВ; MOVE A BLOCK LENGTH BC FROM (HL) TO (DE) - ПЕРЕМЕСТИТЬ БЛОК ДЛИНОЙ BC ИЗ (HL) В (DE); END OF PROGRAM - КОНЕЦ ПРОГРАММЫ.

ЗАГРУЗИТЕ ПРИВЕДЕННУЮ ВЫШЕ ПРОГРАММУ В СВОИ "СПЕКТРУМ" И ВЫПОЛНИТЕ ПРОГРАММУ НА МАШИННОМ ЯЗЫКЕ, В ТОМ ВИДЕ, КАК ОНА НАПИСАНА ВЫШЕ, ИЗ (HL) В(DE) БУДЕТ ПЕРЕМЕЩЕН ВСЕГО ОДИН БАЙТ.

ТЕПЕРЬ ДАВАЙТЕ ИЗМЕНИМ ЧЕТВЕРТУЮ СТРОЧКУ ТАК: LD BC,31 (Ø1 1FØØ), ВАС, ВОЗМОЖНО, УДИВИТ, КАКИЕ БУДУТ ВЫСВЕЧЕНЫ ПЕРВЫЕ 32 БАЙТА НА ЭКРАНЕ, ОБРАТИТЕ ВНИМАНИЕ, ВВЕРХУ ЭКРАНА БУДЕТ ПРОВЕДЕНА ОЧЕНЬ ТОНКАЯ ПОЛОСА, ПЕРВЫЕ 32 БАЙТА ЭКРАННОЙ ПАМЯТИ ОТНОСЯТСЯ К ПЕРВОМУ БАЙТУ КАЖДОГО ИЗ ПЕРВЫХ 32 СИМВОЛОВ.

ТЕПЕРЬ ИЗМЕНИМ ЭТУ СТРОКУ ТАК: LD BC,255 (Ø1FFØØ), ВНОВЬ ВЫ, ВОЗМОЖНО, УДИВИТЕСЬ, СЛЕДУЮЩИЙ БАЙТ ПОСЛЕ 32-ГО НЕ ОКАЖЕТСЯ ВО ВТОРОМ РЯДУ ТОЧЕК НА ЭКРАНЕ! ЭТО - ПЕРВЫЙ БАЙТ 32-ГО СИМВОЛА! И ТАК ДАЛЕЕ ВПЛОТЬ ДО 256-ГО СИМВОЛА.

СМОЖЕТЕ ЛИ ВЫ ПРЕДСКАЗАТЬ, КУДА ПОПАДЕТ СЛЕДУЮЩИЙ БАЙТ? ИЗМЕНИТЕ ЭТУ СТРОКУ ТАК: LD BC,2047 (Ø1FF07) И ВЫПОЛНИТЕ ПРОГРАММУ, ВЫ ОБНАРУЖИТЕ, ЧТО ЗАПОЛНЕННОЙ ОКАЗАЛАСЬ ТОЛЬКО ВЕРХНЯЯ ТРЕТЬ ЭКРАНА.

ВЫ МОЖЕТЕ ПОЭКСПЕРИМЕНТИРОВАТЬ С ЭТИМ, ПОЛЬЗУЯСЬ РАЗНЫМИ ЗНАЧЕНИЯМИ BC ВПЛОТЬ ДО LD BC,6143 (Ø1FF17), ТАКИМ СПОСОБОМ ВЫ МОЖЕТЕ ПОСМОТРЕТЬ, КАК "СПЕКТРУМ" ОРГАНИЗУЕТ ЭКРАН.

ЭКРАННАЯ ПАМЯТЬ НА САМОМ ДЕЛЕ РАЗДЕЛЕНА НА ТРИ ЧАСТИ.

1. ПАМЯТЬ С 4000H ПО 47FFH (===) ПЕРВЫЕ ВОСЕМЬ СТРОК.
2. ПАМЯТЬ С 4800H ПО 4FFFH (===) ВТОРЫЕ ВОСЕМЬ СТРОК.
3. ПАМЯТЬ С 5000H ПО 57FFH (===) ТРЕТЬИ ВОСЕМЬ СТРОК.

НО НЕ ТОЛЬКО ЭТО, ВЫ ЕЩЕ ВСПОМНИТЕ, ЧТО В "СПЕКТРУМ" КАЖДАЯ ЛИТЕРА СОСТОИТ ИЗ ВОСЬМИ 8-БИТОВЫХ БАЙТА, ЧТО СОСТАВЛЯЕТ 64 ТОЧКИ.

НАПРИМЕР, ДЛЯ ЛИТЕРЫ "!" ПРЕДСТАВЛЕНИЕ ИМЕЕТ ВИД:

Ø	ØØØØØØØØ	ØH
16	ØØØ1ØØØØ	1ØH
16	ØØØ1ØØØØ	1ØH
16	ØØØ1ØØØØ	1ØH
16	ØØØ1ØØØØ	1ØH
Ø	ØØØØØØØØ	ØH
16	ØØØ1ØØØØ	1ØH
Ø	ØØØØØØØØ	ØH

СТРУКТУРА ПАМЯТИ ЭКРАНА ДИСПЛЕЯ "СПЕКТРУМ" ТАКОВА, ЧТО ПЕРВЫЕ 256 БАЙТОВ С 4000H ПО 40FFH СООТВЕТСТВУЮТ ПЕРВЫМ БАЙТАМ КАЖДОЙ ИЗ 256 8-БАЙТОВЫХ ЛИТЕР ПЕРВЫХ ВОСЬМИ СТРОК.

ДАЛЕЕ СЛЕДУЮЩИЕ 256 БАЙТОВ ЯЧЕЕК ПАМЯТИ С 4100H ПО 42FFH СООТВЕТСТВУЮТ ВТОРЫМ БАЙТАМ КАЖДОЙ 256 8-БАЙТОВЫХ ЛИТЕР ПЕРВЫХ ВОСЬМИ СТРОК И Т.Д.

ТАКИМ ОБРАЗОМ, РАСПОЛОЖЕНИЕ В ПАМЯТИ ВОСЬМИ БАЙТОВ, СООТВЕТСТВУЮЩИХ ПЕРВОЙ ЛИТЕРЕ НА ЭКРАНЕ, БУДЕТ СЛЕДУЮЩИМ:

1ST BYTE	4000H
2ND BYTE	4100H
3RD BYTE	4200H
4TH BYTE	4300H
5TH BYTE	4400H
6TH BYTE	4500H
7TH BYTE	4600H
8TH BYTE	4700H

1ST BYTE - 1-И БАЙТ

СТРАННО, НЕ ТАК ЛИ? НО ПРИХОДИТСЯ ПРИНИМАТЬ "СПЕКТРУМ" ТАКИМ, КАК ЕГО СКОНСТРУИРОВАЛИ.

СМОЖЕТЕ ЛИ ВЫ ЗАПИСАТЬ ВОСЕМЬ БАЙТОВ, СООТВЕТСТВУЮЩИХ 31-И ЛИТЕРЕ ТРЕТЬЕЙ СТРОКИ ЭКРАНА ? ВЫ МОЖЕТЕ ОБРАТИТЬСЯ К ПРИЛОЖЕНИЮ В, КАРТЕ ПАМЯТИ ЭКРАНА.

(405EH, 415EH, 425EH,,,,,475EH)

В СООТВЕТСВИИ С ПРИНЯТЫМ НАМИ СПОСОБОМ ВЫДАЧИ НА ЭКРАН

ЯЧЕЕЕК ПАМЯТИ,СООТВЕТСТВУЮЩИЕ ПЕРВОЙ ЛИТЕРЕ ВТОРОЙ ГРУППЫ ИЗ ВОСЬМИ СТРОК, БУДУТ ТАКИМИ:

4800H, 4900H, 4A00H, 4B00H, 4C00H, 4D00H, 4E00H, 4F00H

АНАЛОГИЧНЫМ ОБРАЗОМ, ПЕРВАЯ ЛИТЕРА ТРЕТЬЕЙ ГРУППЫ ИЗ ВОСЬМИ СТРОК ОТОБРАЖАЕТСЯ ВОСЬМЬЮ БАЙТАМИ В СЛЕДУЮЩИХ ЯЧЕЙКАХ:

5000H, 5100H, 5200H, 5300H, 5400H, 5500H, 5600H, 5700H

В ПРИМЕНЕНИИ МАШИННОГО ЯЗЫКА , ТЕМ НЕ МЕНЕЕ ,ЕСТЬ ОПРЕДЕЛЕННЫЕ ПРЕИМУЩЕСТВА, ОЧЕВИДНЫЕ ТРУДНОСТИ СТОИТ ПРЕОДОЛЕВАТЬ, ВОТ ТРИВИАЛЬНЫЙ ПРИМЕР ИЗ ЯЗЫКА "БЕЙСИК", ЕСЛИ ВЫ ПОПЫТАЕТЕСЬ ВЫПОЛНИТЬ КОМАНДУ PRINT ДЛЯ ВВОДНОЙ ЧАСТИ ЭКРАНА (НИЖНИЕ ДВЕ СТРОКИ), ТО СИСТЕМА "БЕЙСИК" ЭТОМУ РЕЗКО ВОСПРОТИВИТСЯ, НО НА МАШИННОМ ЯЗЫКЕ У ВАС ИМЕЕТСЯ ПОЛНЫЙ ДОСТУП КО ВСЕМУ ЭКРАНУ,

ЕСЛИ ВЫ БОЛЕЕ ВНИМАТЕЛЬНО ПРИСМОТРИТЕСЬ К ОРГАНИЗАЦИИ ЭКРАНА ДИСПЛЕЯ, ВЫ УВИДИТЕ, ЧТО СТАРШИЙ БАЙТ ПЕРВОГО БАЙТА (НØ ВFВ) КАЖДОЙ ЛИТЕРЫ ОПРЕДЕЛЯЕТ, К КАКОЙ ИЗ ТРЕХ ГРУПП ПАМЯТИ ЛИТЕРА ОТНОСИТСЯ,

НАПРИМЕР:

ЕСЛИ 40H = (НØBFB (41H ЛИТЕРА НАХОДИТСЯ В ПЕРВОЙ ГРУППЕ ИЗ ВОСЬМИ СТРОК

ЕСЛИ 48H = (НØBFB (49H ЛИТЕРА НАХОДИТСЯ ВО ВТОРО ГРУППЕ ИЗ ВОСЬМИ СТРОК

ЕСЛИ 50H = (НØBFB (51H ЛИТЕРА НАХОДИТСЯ В ТРЕТЬЕ ГРУППЕ ИЗ ВОСЬМИ СТРОК

ПОМИМО ЭТОГО ТРИ МЛАДШИХ БИТА НОВ (HIGH ORDER BYTE , СТАРШЕГО БАЙТА) ОПРЕДЕЛЯЮТ, К КАКОМУ ИЗ ВОСЬМИ БАЙТОВ ЛИТЕРЫ ОН ПРИНАДЛЕЖИТ,

СИТУАЦИЯ НЕСКОЛЬКО ПРОЯСНЯЕТСЯ? ПОСМОТРИТЕ ПРИЛОЖЕНИЕ В И ПОСТАРАИТЕСЬ ПОНЯТЬ СВЯЗЬ МЕЖДУ ЯЧЕЙКАМИ ПАМЯТИ И ЭКРАНОМ ДИСПЛЕЯ,

РАССМОТРИМ СЛЕДУЮЩИЙ ПРИМЕР;

ПРЕДПОЛОЖИМ, НАМ ЗАДАН АДРЕС 4A36H, СТАРШИЙ БАЙТ АДРЕСА БУДЕТ 4AH, ТАК ЧТО:

1) МЫ ЗНАЕМ, ЧТО ОН НАХОДИТСЯ В ПРЕДЕЛАХ ПАМЯТИ ЭКРАНА ДИСПЛЕЯ, ПОСКОЛЬКУ ЕГО ЗНАЧЕНИЕ НАХОДИТСЯ МЕЖДУ 40H И 58H;

2) ЕГО ДВОИЧНОЕ ПРЕДСТАВЛЕНИЕ ИМЕЕТ ВИД Ø1ØØ1Ø1Ø;

3) ПО МЛАДШИМ ТРЕМ БИТАМ МЫ ЗНАЕМ, ЧТО ОН ПРИНАДЛЕЖИТ ТРЕТЬЕМУ БАЙТУ ПОЗИЦИИ ЛИТЕРЫ НА ЭКРАНЕ;

4) ЕСЛИ МЫ СДЕЛАЕМ МЛАДШИЕ ТРИ БИТА РАВНЫМ НУЛЮ, ТО ЗНАЧЕНИЕ НОВ БУДЕТ РАВНО 48H, ТАКИМ ОБРАЗОМ МЫ ЗНАЕМ, ЧТО ОН ПРИНАД ЛЕЖИТ ВТОРОЙ ГРУППЕ ИЗ ВОСЬМИ СТРОК, Т,Е, СРЕДНЕЙ ПОРЦИИ ЭКРАНА,

МЫ МОЖЕМ ПРИЙТИ К ВЫВОДУ, ЧТО ЗАДАННЫЙ БАЙТ ОТНОСИТСЯ К ТРЕТЬЕМУ БАЙТУ ЛИТЕРЫ В СРЕДНЕЙ ПОРЦИИ ПАМЯТИ ДИСПЛЕЯ,

КАКОЙ ЛИТЕРЕ ИЗ СЕРЕДИННОЙ ПОРЦИИ ЭТОТ БАЙТ ПРИНАДЛЕЖИТ? ДЛЯ ОТВЕТА НА ЭТОТ ВОПРОС НАМ ПОТРЕБУЕТСЯ ЗНАТЬ ЗНАЧЕНИЕ МЛАДШЕГО БАЙТА АДРЕСА,

МЫ ЗНАЕМ, ЧТО МЛАДШИЙ БАЙТ АДРЕСА РАВЕН 36H, ТАК ЧТО АДРЕС ОТНОСИТСЯ К ЛИТЕРЕ 36H (48 + 6),Т,Е К 54 ПОЗИЦИИ, СЧИТАЯ ОТ ПЕРВОЙ ЛИТЕРЫ СЕРЕДИННОЙ ПОРЦИИ,

ПОСКОЛЬКУ В КАЖДОЙ СТРОКЕ 32 ЛИТЕРЫ, ЗАДАННАЯ ПОЗИЦИЯ НАХОДИТСЯ ВО ВТОРОЙ СТРОКЕ СЕРЕДИННОЙ ПОРЦИИ ЭКРАНА ДИСПЛЕЯ И БУДЕТ (54 - 32 + 1) - Й ЛИТЕРОЙ В ЭТОЙ СТРОКЕ,

ВЫВОД, КОТОРЫЙ МЫ МОЖЕМ СДЕЛАТЬ, - ЗАДАННЫЙ БАЙТ ЯВЛЯЕТСЯ ТРЕТЬИМ БАЙТОМ 23-Й ЛИТЕРЫ 10-Й СТРОКИ ОТ НАЧАЛА ЭКРАНА,

АТРИБУТЫ ТЕЛЕВИЗИОННОГО ЭКРАНА

ПАМЯТЬ, ПРЕДНАЗНАЧЕНАЯ ДЛЯ ХРАНЕНИЯ АТРИБУТОВ ЭКРАНА, ЛЕГЧЕ ДЛЯ ПОНИМАНИЯ, ЧЕМ САМА ПАМЯТЬ ДИСПЛЕЯ, ПОСКОЛЬКУ У НЕЕ ИМЕЕТСЯ ВЗАИМНО ОДНОЗНАЧНОЕ СООТВЕТСТВИЕ С ЛИТЕРАМИ ДИСПЛЕЯ,

ФАЙЛ АТРИБУТОВ РАСПОЛОЖЕН В ПАМЯТИ С 5800H И БAFFH, ОН СОДЕРЖИТ 768 БАЙТОВ, ЧТО СООТВЕТСТВУЕТ 24 СТРОКАМ ПО 32 ЛИТЕРЫ КАЖДАЯ, ИНЫМИ СЛОВАМИ, ИМЕЕТСЯ ПО ОДНОМУ БАЙТУ - АТРИБУТУ ДЛЯ КАЖДОЙ ПОЗИЦИИ ЛИТЕРЫ,

ТАК, 5800H СООТВЕТСТВУЕТ АТРИБУТУ ПЕРВОЙ ЛИТЕРЫ ПЕРВОЙ СТРОКИ, 5801H - ВТОРОЙ ЛИТЕРЫ, 5802H - ТРЕТЬЕЙ,.....,581FH ТРИДЦАТЬ ВТОРОЙ ЛИТЕРЕ ПЕРВОЙ СТРОКИ,

МЫ ЗНАЕМ, ЧТО ДЛЯ КАЖДОЙ ПОЗИЦИИ ЛИТЕРЫ НА ЭКРАНЕ ИМЕЕТСЯ

СООТВЕТСТВУЮЩИЙ БАЙТ-АТРИБУТ В ПАМЯТИ АТРИБУТОВ, СОСТАВЛЕННЫЙ СЛЕДУЮЩИМ ОБРАЗОМ:

БАЙТ-АТРИБУТ.

БИТЫ 0 -2 ПРЕДСТАВЛЯЕТ ЦВЕТ ЧЕРНИЛ ОТ 0 ДО 7,

БИТЫ 3 -5 ПРЕДСТАВЛЯЕТ ЦВЕТ БУМАГИ ОТ 0 ДО 7

БИТ 6 ПОВЫШЕННАЯ ЯРКОСТЬ- ЕСЛИ 1, НОРМАЛЬНАЯ- ЕСЛИ 0

БИТ 7 МИГАНИЕ - ЕСЛИ 1, ОТСУТСТВИЕ МИГАНИЯ - ЕСЛИ 0

У П Р А Ж Н Е Н И Е

СМОЖЕТЕ ЛИ ВЫ НАПИСАТЬ ПОДПРОГРАММУ, ПРЕОБРАЗУЮЩУЮ ЗАДАННЫЙ АДРЕС НА ЭКРАНЕ В АДРЕС СООТВЕТСТВУЮЩЕГО ЕМУ АТРИБУТА, НАПРИМЕР , 4529H.

ПО СУЩЕСТВУ ВАМ НУЖНО ОПРЕДЕЛИТЬ ,КАКОЙ ЛИТЕРЕ ЭКРАНА СООТВЕТСТВУЕТ ЭТОТ АДРЕС, А ЗАТЕМ ПРИБАВИТЬ ЭТО ЗНАЧЕНИЕ К 5800H.

ПРИВОДИМАЯ НИЖЕ ПРОГРАММА ПОКАЗЫВАЕТ, КАК ЭТОГО ДОСТИЧЬ БЫСТРО.

```
LD  HL,4529H    : ЗАГРУЗИТЬ ЗАДАННЫЙ АДРЕС В HL
LD  A,H         : ЗАГРУЗИТЬ СТАРШИЙ БАЙТ В A
AND 18H         : ЛОВУШКА ДЛЯ БИТОВ 3 И 4 ДЛЯ ОПЕРЕДЕЛЕНИЯ
                : ТОГО, КАКОЙ ЧАСТИ ЭКРАНА СООТВЕТСТВУЕТ
                : АДРЕС
SRA A           : СДВИГ НАКАПЛИВАЮЩЕГО РЕГИСТРА ВПРАВО
SRA A           : ТРИ РАЗА, Т.Е, ДЕЛЕНИЕ НА 8
SRA A           : РЕЗУЛЬТАТ МОЖЕТ БЫТЬ РАВЕН 0,1 ИЛИ 2
                : В ЗАВИСИМОСТИ ОТ ТОГО, ЧЕМУ РАВНЯЛСЯ H:
                :   48H, 50H ИЛИ 58H (ТАК В ОРИГИНАЛЕ,ОЧЕ-
                : ВИДНО,ОПЕЧАТКА (ПРИМЕЧ.ПЕР.))
ADD A,58H       : ПРЕОБРАЗОВАНИЕ В ПАМЯТЬ АТРИБУТОВ
LD H,A          : В HL - АДРЕС АТРИБУТА,Т.Е H = 58H,59H ИЛИ
                : 60H, L ОСТАЕТСЯ НЕИЗМЕННЫМ!!!
```

ЗВУКОВОЙ СИГНАЛ

ЕЩЕ ОДНО СРЕДСТВО СВЯЗИ В РЕАЛЬНОМ МАСШТАБЕ ВРЕМЕНИ, ПРЕДОСТАВЛЯЕМЕ ВАШЕЙ МИКО-ЭВМ "СПЕКТРУМ" - ЗВУКОВОЙ СИГНАЛ, БЫЛО БЫ ГЛУПО НЕ ВОСПОЛЬЗОВАТЬСЯ ЭТИМ СРЕДСТВОМ В ПОЛНОЙ МЕРЕ.

В МАШИННОМ ЯЗЫКЕ "СПЕКТРУМ" ЕСТЬ ДВА ОСНОВНЫХ СПОСОБА ГЕНЕРИРОВАНИЯ ЗВУКА,

1. ПОСЫЛКА СИГНАЛОВ НА ВЫХОДНОЙ ПОРТ 254 ДЛЯ КАССЕТНОГО МАГНИТОФОНА В ТЕЧЕНИЕ ОПРЕДЕЛЕННОГО ПРОМЕЖУТКА ВРЕМЕНИ С ПОМОЩЬЮ КОМАНДЫ OUT 254,

НАПРИМЕР, OUT (254),A

2. УСТАНОВИТЬ ОПРЕДЕЛЕННЫЕ ЗНАЧЕНИЯ В HL,DE И ВЫЗВАТЬ ПРОГРАММУ ЗВУКОВОГО СИГНАЛА ИЗ ПЗУ ДЛЯ ГЕНЕРАЦИИ СИГНАЛА.

ВХОДНЫЕ ПАРАМЕТРЫ:

DE - ПРОДОЛЖИТЕЛЬНОСТЬ В СЕКУНДАХ * ЧАСТОТА

HL - (437 500 / ЧАСТОТА) - 30,125

ПОТОМ

CALL 03B5H

ПРЕИМУЩЕСТВОМ. ПЕРВОГО СПОСОБА ГЕНЕРАЦИИ ЗВУКА ЯВЛЯЕТСЯ ОТСУТСТВИЕ ОБРАЩЕНИЙ К ПЗУ, ОН ВЫПОЛНЯЕТСЯ БОЛЕЕ БЫСТРО, НО...

ПОСКОЛЬКУ АЛУ (В ОРИГИНАЛЕ ULA) ПОСТОЯННО ОБРАЩАЕТСЯ К ПЕРВЫМ 16К ПАМЯТИ С ПРОИЗВОЛЬНЫМ ДОСТУПОМ ДЛЯ ВЫПОЛНЕНИЯ ВЫВОДА НА ЭКРАН, ВАША ПРОГРАММА ,ЕСЛИ ОНА РАЗМЕЩЕНА В ПЕРВЫХ 16К ,БУДЕТ ЧАСТО ПОДВЕРГАТЬСЯ КРАТКОВРЕМЕННЫМ ПРЕРЫВАНИЯМ.

ЕСЛИ ПРОГРАММА ГЕНЕРИРУЕТ ЗВУКОВОЙ СИГНАЛ ,ТО ЗВУК БУДЕТ ИЗДАВАТЬСЯ В ВИДЕ НЕПРЕДСКАЗУЕМЫХ ПО ПРОДОЛЖИТЕЛЬНОСТИ ГУДКОВ, ОДИН ИЗ СПОСОБОВ ПРЕОДОЛЕНИЯ ЭТОЙ ТРУДНОСТИ СОСТОИТ В ПЕРЕМЕЩЕНИИ ТОЙ ЧАСТИ ПРОГРАММЫ, КОТОРАЯ ГЕНЕРИРУЕТ ЗВУК, В ОБЛАСТЬ БОЛЬШИХ ЗНАЧЕНИЙ АДРЕСОВ ПАМЯТИ, ЕСЛИ У ВАС ЭВМ С ОБ"ЕМОМ ПАМЯТИ 48К,

ЕСЛИ ЖЕ У ВАС НЕТ ЭВМ С ОБ"ЕМОМ ПАМЯТИ 48К, ТО ВЫ ВСЕ-ТАКИ МОЖЕТЕ ГЕНЕРИРОВАТЬ ЗВУК ЭТИМ МЕТОДОМ, НО ЭТО НЕ БУДЕТ "ЧИСТЫЙ ЗВУК", ВАМ ПРИДЕТСЯ ПРИМЕНЯТЬ ВТОРОЙ СПОСОБ ГЕНЕРАЦИИ ЗВУКА (С ПОМОЩЬЮ ВЫЗОВА ПРОГРАММЫ ИЗ ПЗУ), ЧТОБЫ ДОБИТЬСЯ НУЖНОГО РЕЗУЛЬТАТА.

ОБРАТИТЕ ВНИМАНИЕ, ЧТО ПРИ ПОСЫЛКЕ ЗНАЧЕНИЙ НА ПОРТ ВЫВОДА 254 ОНИ БУДУТ ТАКЖЕ ВЛИЯТЬ НА ЦВЕТ ОКАЙМЛЕНИЯ, И ВКЛЮЧАТЬ

MIC, А ТАКЖЕ ГРОМКОГОВОРИТЕЛЬ, В ЗАВИСИМОСТИ ОТ ПОСЫЛАЕМОГО ЗНАЧЕНИЯ,

С ДРУГОЙ СТОРОНЫ, ПРОГРАММА ИЗ ПЗУ ДЛЯ ГЕНЕРИРОВАНИЯ ЗВУКА ПО СУЩЕСТВУ ПОЗВОЛЯЕТ ВАМ ПРИМЕНЯТЬ ИЗ СВОЕЙ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ КОМАНДУ BEEP, ВЫ МОЖЕТЕ СЧИТАТЬ, ЧТО В ПАРЕ РЕГИСТРОВ СОДЕРЖИТСЯ ЗНАЧЕНИЕ ПРОДОЛЖИТЕЛЬНОСТИ ЗВУКОВОГО СИГНАЛА, А В HL – ЗНАЧЕНИЕ ЧАСТОТЫ, ПОЭКСПЕРИМЕНТИРУЙТЕ С РАЗЛИЧНЫМИ ЗНАЧЕНИЯМИ HL И DE, ПОКА НЕ ПОЛУЧИТЕ НУЖНЫЙ ВАМ ЗВУК,

ОГРАНИЧЕННОСТЬ ЭТОГО МЕТОДА ,КОНЕЧНО, СОСТОИТ В ТОМ, ЧТО ВЫ НЕ МОЖЕТЕ ВЫЙТИ ЗА ПРЕДЕЛЫ ТОГО ДИАПАЗОНА ЗВУКОВ, КОТОРЫЕ ДАЕТ ВОЗМОЖНОСТЬ ИЗДАВАТЬ КОМАНДА BEEP,

В В Е Д Е Н И Е В М О Н И Т О Р Н Ы Е П Р О Г Р А М М Ы Н А М А Ш И Н Н О М Я З Ы К Е

М О Н И Т О Р Н А Я П Р О Г Р А М М А E Z C O D E

ЭТО– МОНИТОРНАЯ ПРОГРАММА НА МАШИННОМ ЯЗЫКЕ, ПОЗВОЛЯЮЩАЯ ВАМ

1, ВВОДИТЬ СВОЙ ПРОГРАММНЫЙ МОДУЛЬ, НАПИСАННЫЙ НА МАШИННОМ ЯЗЫКЕ ЛИБО

В ПОЛНОСТЬЮ АССЕМБЛИРОВАННОМ ВИДЕ;

ЛИБО В ПОЛУАССЕМБЛИРОВАННОМ ВИДЕ, КОГДА ВСЕ ОТНОСИТЕЛЬНЫЕ ПЕРЕХОДЫ И АБСОЛЮТНЫЕ ПЕРЕХОДЫ ВЫРАЖАЮТСЯ ЧЕРЕЗ НОМЕРА СТРОК,.

2, РАСПЕЧАТЫВАТЬ ИСХОДНЫЙ ВВОДИМЫЙ ПРОГРАММНЫЙ МОДУЛЬ,

3, ДЕЛАТЬ ДАМП ВХОДНОГО ПРОГРАММНОГО МОДУЛЯ ПО ЗАДАННОМУ АДРЕСУ ПАМЯТИ,

4, ПРОСМАТРИВАТЬ ЯЧЕЙКИ ПАМЯТИ ИЗ ОПРЕДЕЛЕННОГО ДИАПОЗОНА,

5, ЗАПОМИНАТЬ ЛИБО "ИСХОДНЫЙ МОДУЛЬ", ЛИБО ДАМП ПРОГРАММЫ, ПОЛНОСТЬЮ ПЕРЕВЕДЕННОЙ НА МАШИННЫЙ ЯЗЫК,

6, ЗАГРУЖАТЬ ЗАПИСАННУЮ "ИСХОДНУЮ ПРОГРАММУ" С КАССЕТЫ,

7, ВЫПОЛНЯТЬ ДАМП МОДУЛЯ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ,

ПРЕДВАРИТЕЛЬНЫЕ ТРЕБОВАНИЯ ДЛЯ КОДА EZ

ПРЕЖДЕ ЧЕМ ПРИМЕНЯТЬ ЭТУ МОНИТОРНУЮ ПРОГРАММУ ДЛЯ ВВОДА КАКИХ-ЛИБО ПРОГРАММ НА МАШИННОМ ЯЗЫКЕ, ВЫ ДОЛЖНЫ АССЕМБЛИРОВАТЬ СВОЮ ПРОГРАММУ НА ЯЗЫКЕ АССЕМБЛЕРА, ВАМ НЕ НУЖНО ВЫЧИСЛЯТЬ ОТНОСИТЕЛЬНЫЕ ИЛИ АБСОЛЮТНЫЕ ПЕРЕХОДЫ!

ВАШ ПРОГРАММНЫЙ МОДУЛЬ НЕ ДОЛЖЕН ПРЕВЫШАТЬ 800 БАЙТОВ ИЛИ 200 КОМАНД,

ВЫ НЕ ДОЛЖНЫ ЗАГРУЖАТЬ ОКОНЧАТЕЛЬНЫЙ ТЕКСТ ПРОГРАММЫ ПО АДРЕСАМ, МЕНЬШИМ 31499 (ЧТОБЫ НЕ СТЕРЕТЬ ПРОГРАММУ КОДА EZ),

ИДЕЙНЫЕ ОСНОВАНИЯ ПРОГРАММЫ КОДА EZ

ОСНОВНАЯ ИДЕЯ ЭТОЙ ПРОГРАММЫ СОСТОИТ В ТОМ, ЧТОБЫ ДАТЬ ВОЗМОЖНОСТЬ ВВОДИТЬ КОМАНДЫ МАШИННОГО ЯЗЫКА В ВИДЕ НУМЕРОВАННЫХ СТРОК, АНАЛОГИЧНО РАСПЕЧАТКЕ ПРОГРАММЫ НА ЯЗЫКЕ "БЕЙСИК",

КАЖДАЯ СТРОКА "ИСХОДНОЙ ПРОГРАММЫ" (ТАК НАЗЫВАЮТСЯ СТРОКИ ТЕКСТА ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ) ИМЕЕТ НОМЕР И СОДЕРЖИТ ДО 4 БАЙТОВ ТЕКСТА,

ОСНОВНОЕ ПРЕИМУЩЕСТВО ЭТОГО ПРОГРАММНОГО СРЕДСТВА, ТАКИМ ОБРАЗОМ, СОСТОИТ В ВОЗМОЖНОСТИ "РЕДАКТИРОВАНИЯ" ЛЮБОЙ СТРОКИ, "ИСХОДНАЯ ПРОГРАММА" МОЖЕТ ТАКЖЕ ОТДЕЛЬНО ЗАПИСЫВАТЬСЯ НА ЛЕНТ ЧТО ПОЗВОЛЯЕТ ЗАПОМИНАТЬ РАБОТУ НА КОНКРЕТНЫХ ЭТАПАХ,

ОСНОВНАЯ НОВИЗНА ЭТОЙ ПРОГРАММЫ – В ВОЗМОЖНОСТИ ВСТАВКИ ОТНОСИТЕЛЬНЫХ И АБСОЛЮТНЫХ ПЕРЕХОДОВ БЕЗ НЕОБХОДИМОСТИ ПЕРЕСЧЕТА СООТВЕТСТВУЮЩИХ ЧИСЕЛ, ЛЮБОЙ ПЕРЕХОД МОЖНО СДЕЛАТЬ, ПРОСТО СОСЛАВШИСЬ НА НОМЕР СТРОКИ, НА КОТОРУЮ ВЫ ХОТИТЕ ПЕРЕЙТИ!

ЭТО ОЗНАЧАЕТ, ЧТО ИЗМЕНЕНИЯ МОЖНО ВНОСИТЬ БЕЗ ТРУДА ДАЖЕ В ПРЕДЕЛАХ ДИАПАЗОНА ОТНОСИТЕЛЬНОГО ПЕРЕХОДА,

ТЕКСТ "ИСХОДНОЙ ПРОГРАММЫ" ПЕРЕДАЕТСЯ В ПАМЯТЬ ПО КОМАНДЕ "ДАМП", ПОЛУЧАЮЩИЙСЯ В РЕЗУЛЬТАТЕ ТЕКСТ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ МОЖНО ТАКЖЕ ЗАПИСАТЬ В ПАМЯТЬ,

СВОДКА КОМАНД КОДА EZ

ОБРАТИТЕ ВНИМАНИЕ, ЧТО ПЕРВЫЙ ВОПРОС, ЗАДАВАЕМЫЙ ВАМ ПРОГ-

РАММОЙ;
"АДРЕС ЗАГРУЗКИ",
ЭТО АДРЕС, ПО КОТОРОМУ ВЫ ХОТИТЕ РАЗМЕСТИТЬ ТЕКСТ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ, ОН НЕ МОЖЕТ БЫТЬ МЕНЬШЕ 31500,

ВВОД СТРОК

1, ДЛЯ ВВОДА СТРОК "ИСХОДНОЙ ПРОГРАММЫ":
(НОМЕР-СТРОКИ) (ПРОБЕЛ)(НЕ БОЛЕЕ 4 ШЕСТНАДЦАТЕРИЧНЫХ БАЙТОВ)(ENTER)
НАПРИМЕР, 1 210040 ПРИВЕДЕТ К ВВОДУ МАШИННОЙ КОМАНДЫ
LD HL, 4000H В СТРОКУ НОМЕР 1,
2, ДЛЯ РЕДАКТИРОВАНИЯ СТРОКИ:
(НОМЕР СТРОКИ)(ПРОБЕЛ)(ЗАНОВО ВВЕСТИ НОВЫЕ БАЙТЫ)(ENTER)
НАПРИМЕР, 1 210140 ПРИВЕДЕТ К ЗАМЕНЕ СТРОКИ НОМЕР 1 КОМАНДОЙ LD HL,4001H
3, ДЛЯ УДАЛЕНИЯ СТРОКИ КОМАНДЫ:
(НОМЕР-СТРОКИ)(ENTER)
НАПРИМЕР, 1 (ENTER) ПРИВЕДЕТ К УДАЛЕНИЮ СТРОКИ НОМЕР 1,
4, ДЛЯ ЗАДАНИЯ ОТНОСИТЕЛЬНОГО ИЛИ АБСОЛЮТНОГО ПЕРЕХОДА
(НОМЕР-СТРОКИ)(ПРОБЕЛ)(КОМАНДА ПЕРЕХОДА)(""L" В НИЖНЕМ РЕГИСТРЕ")(НОМЕР-СТРОКИ)(ENTER)
НАПРИМЕР, 1 C312 ПРЕДСТАВЛЯЕТ КОМАНДУ JP НА СТРОКУ 2,
2 1811 ПРЕДСТАВЛЯЕТ КОМАНДУ JR НА СТРОКУ 1,

К О М А Н Д Ы

1, DUMP(ENTER)
* ДАМП ИСХОДНОЙ РАСПЕЧАТКИ В ПАМЯТЬ, НАЧИНАЯ С ЗАДАННОГО АДРЕСА ЗАГРУЗКИ,
* ЭТО НЕОБХОДИМО ПРОДЕЛАТЬ ПЕРЕД ВЫПОЛНЕНИЕМ ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ,
СОКРАЩЕНИЕ: DU
2, EXIT(ENTER)
* ВЫХОД ИЗ КОДА EZ И ПОВТОРНЫЙ ВХОД В СИСТЕМУ "БЕЙСИК",
СОКРАЩЕНИЕ: EX
3, LIST(ENTER)
* РАСПЕЧАТКА ПЕРВЫХ 22 СТРОК КОМАНД РАСПЕЧАТКИ ИСХОДНОГО ТЕКСТА
* НАЖМИТЕ ЛЮБУЮ КЛАВИШУ, КРОМЕ "M" И "BREAK",ЧТОБЫ ПРОДОЛЖИТЬ РАСПЕЧАТКУ,
СОКРАЩЕНИЕ: LI
LIST#(ENTER)
* РАСПЕЧАТКА 22 СТРОК РАСПЕЧАТКИ ИСХОДНОГО ТЕКСТА НАЧИНАЯ СО СТРОКИ N#, ГДЕ НОМЕР - ОТ 1 ДО 200 ВКЛЮЧИТЕЛЬНО,
СОКРАЩЕНИЕ: БЕЗ СОКРАЩЕНИЯ
4, LOAD(ENTER)
* ЗАГРУЗИТЬ МОДУЛЬ С РАСПЕЧАТКОЙ ИСХОДНОГО ТЕКСТА С КАССЕТЫ, ЗАМЕНИВ ИМ СУЩЕСТВУЮЩИЙ МОДУЛЬ,
СОКРАЩЕНИЕ: LD
5, MEM(ENTER)
ПОДСКАЗКА: АДРЕС НАЧАЛА;
* ВВЕДИТЕ АДРЕС ПАМЯТИ С КОТОРОГО ВЫ ХОТИТЕ НАЧАТЬ ВЫДАЧУ,
* МОЖЕТ БЫТЬ ОТ 0 ДО 32767 ДЛЯ "СПЕКТРУМ" С ОБ"ЕМОМ ПАМЯТИ 16K ИЛИ ОТ 0 ДО 65535 ДЛЯ "СПЕКТРУМ С ОБ"ЕМОМ ПАМЯТИ 48K,
* НАЖМИТЕ "M", ЧТОБЫ ВЫЙТИ ИЗ РЕЖИМА ПРОСМОТРА ПАМЯТИ,
СОКРАЩЕНИЕ: ME
6, NEW(ENTER)
* УДАЛИТЬ ТЕКУЩИЙ МОДУЛЬ И ЗАНОВО СТАРТОВАТЬ КОД EZ,
* ЭТА КОМАНДА ПОЛЕЗНА ДЛЯ ТОГО, ЧТОБЫ НАЧАТЬ ВВОД ТЕКСТА ЕЩЕ ОДНОГО ПРОГРАММНОГО МОДУЛЯ,
СОКРАЩЕНИЕ: NE
7, RUN(ENTER)
* ВЫПОЛНИТЬ ПОДВЕРГНУТЫЙ ДАМПУ ПРОГРАММНЫЙ МОДУЛЬ, НАЧИНАЯ С АДРЕСА ЗАГРУЗКИ,ЗАДАННОГО ВАМИ, КОГДА СТАРТОВАЛИ ПРОГРАММУ КОДА EZ ИЛИ ПРИ ЗАГРУЗКЕ НОВОЙ РАСПЕЧАТКИ ИСХОДНОГО ТЕКСТА,
СОКРАЩЕНИЕ: RU
8, SAVE(ENTER)
* ЗАПИСАТЬ НА КАССЕТУ-РАСПЕЧАТКУ ИСХОДНОГО ТЕКСТА ИЛИ ДАМП ТЕКСТА МАШИННОЙ ПРОГРАММЫ,

ПОДСКАЗКА: ВВЕСТИ ИМЯ:
ВВЕДИТЕ ИМЯ, КОТОРОЕ ВЫ ХОТИТЕ ИСПОЛЬЗОВАТЬ, ИСХОДНЫЙ ТЕКСТ ИЛИ МАШИННЫЙ ЯЗЫК: (S ИЛИ M)
ВВЕДИТЕ S ДЛЯ ЗАПИСИ РАСПЕЧАТКИ ИСХОДНОГО ТЕКСТА;
ВВЕДИТЕ M ДЛЯ ЗАПИСИ ТЕКСТА НА МАШИННОМ ЯЗЫКЕ;
ЗАПУСТИТЕ ЛЕНТУ, ПОТОМ НАЖМИТЕ ЛЮБУЮ КЛАВИШУ, УБЕДИТЕСЬ, ЧТО КАССЕТА ПРАВИЛЬНО ЗАПРАВЛЕНА, НАЖМИТЕ ЛЮБУЮ КЛАВИШУ, КОГДА КАССЕТА ГОТОВА,
СОКРАЩЕНИЕ: SA

З А М Е Ч А Н И Я

1, ЕСЛИ ВЫ НЕ ХОТИТЕ, ЧТОБЫ ПОСЛЕ ВЫПОЛНЕНИЯ ПРОГРАММЫ ВОЗВРАЩАЛСЯ РЕЗУЛЬТАТ В РЕГИСТРЕ BC, ИЗМЕНИТЕ СТРОКУ 3090 СЛЕДУЮЩИМ ОБРАЗОМ:

```
3090 IF K$ = "RU"THEN LET L = USR R
```

2, ДЛЯ РЕСТАРТА ПРОГРАММЫ КОДА EZ: ЛИБО ПРИМЕНИТЕ RUN, В РЕЗУЛЬТАТЕ ЧЕГО ВСЕ ПЕРЕМЕННЫЕ БУДУТ ИНИЦИАЛИЗИРОВАНЫ; ЛИБО ПРИМЕНИТЕ GO TO 2020, В РЕЗУЛЬТАТЕ ЧЕГО ВЫДАЕТСЯ ПОДСКАЗКА: "COMMAND OR LINE (###)",

3, ВСЕ ВВОДИМЫЕ ЧИСЛА, ЗА ИСКЛЮЧЕНИЕМ ТЕКСТА МАШИННЫХ КОМАНД, ДОЛЖНЫ БЫТЬ В ДЕСЯТИЧНОМ ФОРМАТЕ,

4, ЧТОБЫ ВЫ МОГЛИ ВСТАВЛЯТЬ ДОПОЛНИТЕЛЬНЫЕ СТРОКИ В ТЕКУЩУЮ РАСПЕЧАТКУ, ПОЛЕЗНО В РАСПЕЧАТКЕ ПРОПУСКАТЬ НОМЕРА СТРОК, Т,Е, ВМЕСТО ТОГО ЧТОБЫ ВВОДИТЬ СТРОКИ КОМАНД С НОМЕРАМИ 1,2, 3, ВВОДИТЕ 1,5,10 И Т,Д,

ЭТО ПРИДАСТ БОЛЬШУЮ ГИБКОСТЬ ВВОДИМОМУ МОДУЛЮ,

У П Р А Ж Н Е Н И Е Н А К О Д E Z

ВВЕДИТЕ СЛЕДУЮЩИЙ ТЕКСТ ПРОГРАММЫ,

```
210040          LD HL,4000H         ;FILL SCREEN
110140          LD DE,4001H
01FF17          LD BC,6143
3EFF            LD A, 0FFH
77              LD (HL),A
EDB0            LDIR
3E7F     LOOP:  LD A, 7FH           ; TRAP BREAK KEY
DBFE            IN A,(0FEH)
E601            AND 1
20F8            JR NZ,LOOP
C9              RET
```

FILL SCREEN-ЗАПОЛНЕНИЕ ЭКРАНА;TRAP BREAK KEY -ПРЕРЫВАНИЕ ПО КЛАВИШЕ,

ЧТОБЫ ВВЕСТИ ПРИВЕДЕННЫЙ ВЫШЕ ТЕКСТ С ПОМОЩЬЮ ПРОГРАММЫ КОДА EZ

```
(RUN)
COMMAND OR LINE(###): 1 210040(ENTER)
COMMAND OR LINE(###): 5 110140(ENTER)
COMMAND OR LINE(###): 10 01FF17(ENTER)
COMMAND OR LINE(###): 15 3EFF(ENTER)
COMMAND OR LINE(###): 20 77(ENTER)
COMMAND OR LINE(###): 25 EDB0(ENTER)
COMMAND OR LINE(###): 30 3E7F(ENTER)
COMMAND OR LINE(###): 35 DBFE(ENTER)
COMMAND OR LINE(###): 40 E601(ENTER)
COMMAND OR LINE(###): 45 20130(ENTER)
```

(ЭТО БУДЕТ 20, ПОТОМ "L" В НИЖНЕМ РЕГИСТРЕ, ПОТОМ 30, Т,Е, JR NZ,LINE 30).

```
COMMAND OR LINE(###): 50 C9(ENTER)
COMMAND OR LINE(###): LIST(ENTER)
COMMAND OR LINE(###): DUMP(ENTER)
COMMAND OR LINE(###): MEM(ENTER)
STARTING ADDRESS: 31500(ENTER)
                  (THIS IS THE KEY TO EXIT THE MEMORY
                   DISPLAY MODE)
COMMAND OR LINE(###): RUN(ENTER)
                      (BREAK)
```

RUN - ВЫПОЛНИТЬ; LOADING ADDRESS - АДРЕС ЗАГРУЗКИ; COMMAND OR LINE - КОМАНДА ИЛИ СТРОКА; STARTING ADDRESS - АДРЕС НАЧАЛА; THIS IS THE KEY TO EXIT THE MEMORY DISPLAY MODE - ЭТО -

```
КЛАВИШ ДЛЯ ВЫХОДА ИЗ РЕЖИМА ПРОСМОТРА ПАМЯТИ,
 ОБРАТИТЕ ВНИМАНИЕ, ЧТО ПОСЛЕ НОМЕРОВ СТРОК ДОЛЖЕН ИДТИ ПРО-
БЕЛ,
*EZCODE :
 COPYRIGHT (C) 1982
 100 REM MACHINE
 110 REM MACHINE CODE MONITOR
 120 GO TO 9000
 130 DEF FN D(S$) =(S$>"9"*(CODE S$-55)
       +(S$<="9")*(CODE S$-48)-(S$ > " ")*32
 140 DEF FN O(O$) = ((O$ = "CA")+(O$ = "DA")
       +(O$ = "EA")+(O$ = "FA")+(O$ = "C2")
       +(O$ = "D2")+(O$ = "E2")+(O$ = "F2")
       +(O$ = "C3"))-((O$ = "38")+(O$ = #30")
       +(O$ = "28")+(O$ = "20")+(O$ = "18")
       +(O$ = "10"))
 1000 REM
 1010 REM INV LINE PRINTING ROUTINE IRU
       ---                          ---
 1020 CLS : PRINT AT ZE,25;INVERSE ON;FLASH ON;"LISTING IRU"
 1030 LET F = ZE : PRINT AT ZE, ZE;
 1040 FOR J = PL1 TO PL2
 1050 IF C$(J,ON) = "  " THEN GO TO 1110
 1060 PRINT TAB TR- LEN STR$ J; J; TAB FR; " ";
 1070 IF C$(J, TW, ON TO ON) ="1"
       THEN PRINT C$(J, ON)+" "+C$(J, TW)+C$(J, TR)
     : GO TO 1090)
 1080 PRINT C$(J, ON); " "; C$(J, TW); " "
       ; C$(J, TR); " "; C$(J, TR)
 1090 LET F =F+ON
 1100 IF F = 22 THEN GO TO 1120
 1110 NEXT J
 1120 PRINT AT ZE, 25; "       "
 1130 RETURN
 2000 REM INV MAIN ROUTINE IRU
       ---                 ---
 2020 INPUT "COMMAND OR LINE(###) : "; A$
 2030 IF A$( TO FR) ="    "THEN GO TO MR
 2040 IF A$(ON) > "9" THEN GO TO 3000
 2050 LET K$ = "": FOR K = ON TO FR
 2060 IF A$(K TO K) = " " THEN GO TO 2090
 2070 LET K$ = K$+A$(K TO K)
 2080 NEXT K
 2090 IF K = 5 OR VAL K$ = ZE OR VAL K$ > IN
       THEN GO TO MR
 2100 LET J = VAL K$ : LET N = J
     : REM LINE NUMBER MUST BE 3 BYTES
 2110 LET A$ = A$(K+ON TO)
 2120 LET K$ = ""
 2130 FOR K = ON TO LEN A$
 2140 IF A$(K TO K) <> " "
       THEN LET K$ =K$+A$(K TO K)
 2150 NEXT K
 2160 LET A$ = K$
 2162 IF A$(ON) = "1" THEN GO TO MR
 2170 CLS : FOR I = ON TO 7 STEP TW
 2180 LET K = INT (I/TW+ON)
 2190 LET C$(J, K) = A$(I TO 1+ON)
 2200 NEXT I
 2210 IF C$(N, ON) = "  " THEN GO TO 2250
 2220 IF N < TP THEN LET TP = N
 2230 IF N > BP THEN LET BP = N
 2240 GO TO 2320
 2250 IF N <> BP THEN GO TO 2280
 2260 IF BP = ON OR C$(BP, ON) <> "  "
       THEN GO TO 2320
 2270 LET BP = BP-ON : GO TO 2260
```

```
2280 IF N <> TP THEN GO TO 2320
2290 IF C$(TP, ON) <> "  " THEN GO TO 2320
2300 IF TP <> BP AND TP <> IN THEN LET TP = TP+ON
     : GO TO 2290
2310 LET TP = ON
2320 LET PP = N
2330 IF N < TP THEN LET PP = TP : GO TO 2380
2340 LET NUMLP = ZE
2350 IF PP = TP OR NUMLP = 11 THEN GO TO 2380
2360 IF C$(PP, ON) <> "  "
     THEN LET NUMLP = NUMLP+ON
2370 LET PP = PP-ON : GO TO 2350
2380 LET PL1 = PP : LET PL2 = BP
2390 GO SUB 1000
    * REM PRINT A BLOCK OF LINES
2400 GO TO MR
3000 REM
3010 REM  INV COMMANDS********** IRU
       ---                      ---
3020 LET K$ = A$( TO TW)
3030 IF K$ = "DU" THEN GO TO 5000
3040 IF K$ = "EX" THEN STOP
3050 IF K$ = "LI" THEN GO TO 4000
3060 IF K$ = "LO" THEN GO TO 7000
3070 IF K$ = "ME" THEN GO TO 6000
3080 IF K$ = "NE" THEN RUN
3090 IF K$ = "RU" THEN PRINT USR R
3100 IF K$ = "SA" THEN GO TO 8000
3110 GO TO MR
4000 REM
4000 REM INV LIST  ROUTINE********** IRU
     ---                        ---
4020 LET PL1 = TP : LET PL2 = BP
4030 LET N1 = CODE A$(6 TO 6)
4040 IF LEN A$ > FR AND NL > 47 AND N1 < 58
       THEN LET PL1 = VAL A$(5 TO 8)
4050 GO SUB 1000
4060 GO TO MR
5000 REM
5010 REM  INV  DUMP ROUTINE********** IRU
       ---                        ---
5020 CLS : PRINT AT ZE, 25; INK ON; INVERSE ON
       ; FLASH ON; "DUMPING" : LET G = R
5030 PRINT AT ON, ZE;
5040 FOR J = TP TO BP
5050 IF C$(J, ON) = "  " THEN GO TO 5470
5060 IF C$(J, TW, ON TO ON) <> "1" THEN GO TO 5380
5070 POKE G, ZE : POKE G+ON, ZE : POKE G+TW, ZE
     ; POKE G+TR, ZE
5080 LET JL = VAL (C$(J, TW, TW TO TW)+C$(J, TR))
5090 PRINT TAB TR- LEN STR$ J; INVERSE ON; J
     ; TAB FR; INVERSE ZE; " "
     ; C$(J, ON)+" "+C$(J, TW)+C$(J, TR)
     ; " =  > ";
5100 IF JL < ZE OR JL > LN THEN GO TO 5460
5110 LET CJ = FN O(C$(J, ON))
5120 PRINT TAB 17- LEN STR$ JL; INVERSE ON; JL
       ; TAB 18; INVERSE ZE; " "; C$(JL, ON)
       ; " "; C$(JL, TW); " "; C$(JL, TR); " "
       ; C$(JL, FR);
5130 IF ABS CJ <> ON THEN GO TO 5460
5140 LET DD = (JL > J)-(JL < J)
5150 LET JA = G : LET DP = ZE
5170 LET CL = J+DD
5180 LET NL = ZE : IF C$(CL, ON) = "  "
       THEN GO TO 5220
5190 IF C$(CL, TW, ON TO ON) <> "1"
```

```
        THEN LET NL = ON+(C$(CL, TW) <> "  ")
             +(C$(CL, TR) <> "  ")
             +(C$(CL, FR) <> "  ")
     : GO TO 5220
5200 LET TJ = FN O(C$(CL, ON))
5210 LET N1 = (TJ = ON)*TR+(TJ = -ON)*TW
5220 IF CL = JL AND DD > ZE THEN GO TO 5270
5230 LET DP = DP+N1
5240 IF CL = JL THEN GO TO 5270
5250 LET CL = CL+DD
5260 GO TO 5180
5270 IF CJ = ON THEN LET JA = JA+DD*DP+(DD > ZE)*TR
     : GO TO 5310
5280 IF DD > ZE THEN LET DP = DP+2
5290 IF DP > 126 AND DD < ZE THEN GO TO 5460
5300 IF DP > 129 AND DD > ZE THEN GO TO 5460
5310 LET V = 16* FN D(C$(J, ON, ON TO ON))
     + FN D(C$(J, ON, TW TO TW))
5320 POKE G, V : LET G = G+ON
5330 IF CJ = ON THEN POKE G, JA- INT (JA/QK)*QK
     : LET G = G+ON : POKE G, INT (JA/QK)
     : LET G = G+ON : GO TO 5360
5340 IF DD < ZE THEN LET DP = -DP
5350 LET DP = DP-TW : POKE G, DP : LET G = G+ON
5360 PRINT "OK"
5370 GO TO 5470
5380 FOR I = ON TO 7 STEP TW
5390 LET K = INT (I/TW+ON)
5400 LET V = 16* FN D(C$(J, K, ON TO ON))
        + FN D(C$(J, K, TW TO TW))
5410 IF V < ZE THEN GO TO 5440
5420 POKE G, V
5430 LET G = G+ON
5440 NEXT I
5450 GO TO 5470
5460 PRINT "**"
5470 NEXT J
5480 PRINT AT ZE, 25; "        "
     : GO TO MR
6000 REM
6010 REM INV MEMORY DISPLAY********* IBU
       ---                          ---
6020 INPUT "STARTING ADDRESS : "; DM
6030 CLS : PRINT AT ZE, ZE ;
6040 LET G = DM : LET F = ZE
6050 LET F = F+ON
     : PRINT TAB 5-LEN STR$ G; G; TAB 6;
6060 FOR I= ON TO FR
6070 LET V = PEEK G
6080 LET H = INT (V/16)
6090 LET L = V-16*H
6100 PRINT D$(H+ON) ; D$(L+ON ; " ";
6110 LET G = G+ON
6120 NEXT I
6130 PRINT " "
6140 IF F<> 22 THEN GO TO 6050
6050 LET K$ = INKEY$ : IF K$ ="" THEN GO TO 6150
6160 IF K$ <> "M" AND K$ <> "M" THEN LET F = ZE
     : POKE 23692, QK-ON : GO TO 6050
6200 POKE 23692, ON : PAUSE 20 : GO TO MR
7000 REM
7010 REM INV LOAD ****************** IRU
       ---                          ---
7020 CLS
7030 INPUT
       "LOAD ARRAY : PRESS ANY KEY WHEN   READY, "
       ; K$
```

```
 7040 PRINT AT ZE, 25; INVERSE ON; FLASH ON; "LOADING"
 7050 LOAD "SOURCE" DATA C¤()
 7060 FOR I = ON TO 1N
 7070 LET TP = I
 7080 IF C¤(I, ON) <>  "  " THEN GO TO 7100
 7090 NEXT I
 7100 FOR I = 1N TO ON STEP -1
 7110 LET BP = I
 7120 IF C¤(I, ON) <> "  " THEN GO TO 7140
 7130 NEXT I
 7140 PRINT AT ZE, 25; "        "
 7150 GO TO 9150
 8000 REM
 8010 REM INV SAVE****************** IRU
        ---                          ---
 8020 INPUT "ENTER NAME : "; N¤
 8030 IF N¤ = ""THEN GO TO 8020
 8040 INPUT
        "SOURCE OR MACHIEN CODE : (S OR M)"
        ; K¤
 8050 IF K¤ <> "S" AND K¤ <> "M" THEN GO TO 8040
 8060 IF K¤ = "S" THEN SAVE N¤ DATA C¤() : GO TO MR
 8070 INPUT "STARTING ADDRESS :  "; SS
 8080 INPUT "FINISHING ADDRESS :  "; SF
 8090 LET SB = SF-SS+ON
 8100 SAVE N¤ CODE SS, SB
 8110 GO TO MR
 9000 REM
 9010 REM INITIALISATION
 9020 LET ZE = PI - PI : LET ON = PI / PI
      : LET TW = ON+ON : LET TR = ON+TW
      : LET FR = TW+TW : LET QK = 256
      : LET MR = 2020 : LET 1N = 200
 9025 BORDER 7 : PAPER 7 : INK ON : INVERSE ZE
      ; OVER ZE : FLASH ZE : BRIGHT ZE
      : BEEP ,25, 24 : BEEP ,25, 12
 9030 DIM A¤(15) : DIM O¤(TW)
 9040 LET TP = LN : LET BP = ON : REM LINE NUMBER BUFFER
 9050 DIM C¤(LN, FR, TW) : REM HOLDS CODE
 9060 PRINT AT ZE, 20; INVERSE ON; FLASH ON
        ; "INITIALISING"
 9070 FOR I = ON TO LN
 9080 FOR J = ON TO FR
 9090 LET C¤(I, J) = " "
 9100 NEXT J
 9110 BEEP ,01, 20
 9120 NEXT I
 9130 PRINT AT ZE, 20; "              "
 9140 LET D¤ = "0123456789ABCDEF"
 9150 CLS : PRINT "LOWEST ADDRESS :  "; 31500
 9160 INPUT "LOADING ADDRESS :  "; R : PAUSE 20
 9170  IF R < 31500 THEN GO TO 9160
 9180 CLS : GO TO MR
```

COPYRIGHT – АВТОРСКИЕ ПРАВА; MACHINE CODE MONITOR – МОНИТОР МАШИННОГО ЯЗЫКА; ROUTINE – ПРОГРАММА; MAIN ROUTINE – ОСНОВНАЯ ПРОГРАММА; COMMAND OR LINE – КОМАНДА ИЛИ СТРОКА; NUMBER MUST BE 3 BYTES – НОМЕР СТРОКИ ДОЛЖЕН СОСТАВЛЯТЬ 3 БАЙТА; PRINT A BLOCK OF LINES – НАПЕЧАТАТЬ БЛОК СТРОК; COMMANDS – КОММАНДЫ; MEMORY DISPLAY – РАСПЕЧАТКА ПАМЯТИ; STARTING ADDRESS – НАЧАЛЬНЫЙ АДРЕС; LOAD – ЗАГРУЗКА; LOAD ARRAY : PRESS ANY KEY WHEN READY – ЗАГРУЗКА МАССИВА: НАЖМИТЕ ЛЮБУЮ КЛАВИШУ, КОГДА БУДЕТЕ ГОТОВЫ; SAVE – ЗАПИСЬ; ENTER NAME – ВВЕДИТЕ ИМЯ; SOURCE OR MAC INE CODE – ИСХОДНЫЙ ТЕКСТ ИЛИ ТЕКСТ НА МАШИННОМ ЯЗЫКЕ; STARTIN ADDRESS – НАЧАЛЬНЫЙ АДРЕС; FINISHING ADDRESS – КОНЕЧНЫЙ АДРЕС; INITIALISATION – ИНИЦИАЛИЗАЦИЯ; LINE NUMBER

РЕС; LOADING ADDRESS - АДРЕС ЗАГРУЗКИ,

МОНИТОРНАЯ ПРОГРАММА ЗАГРУЗКИ ТЕКСТА ПРОГРАММЫ НА МАШИННОМ ЯЗЫКЕ В ШЕСТНАДЦАТЕРИЧНОМ ФОРМАТЕ HEXLOAD

ЭТА НАПИСАННАЯ НА ЯЗЫКЕ "БЕЙСИК" ПРОГРАММА МОЖЕТ БЫТЬ МОНИТОРОМ САМА ПО СЕБЕ, ПОСКОЛЬКУ ОНА МОЖЕТ ЗАПИСЫВАТЬ ДАННЫЕ В ШЕСТНАДЦАТЕРИЧНОМ ФОРМАТЕ В ПАМЯТЬ, РАСПЕЧАТЫВАТЬ СОДЕРЖИМОЕ ПАМЯТИ, ПЕРЕМЕЩАТЬ СОДЕРЖИМОЕ ПАМЯТИ, ЗАПИСЫВАТЬ СОДЕРЖИМОЕ ПАМЯТИ НА КАССЕТУ И ЗАГРУЖАТЬ ЕГО С КАССЕТЫ,

С ДРУГОЙ СТОРОНЫ, МЫ МОЖЕМ ПРИМЕНЯТЬ ПРОГРАММУ NEXLOAD В КАЧЕСТВЕ ПОЛУСВЯЗЫВАЮЩЕГО ЗАГРУЗЧИКА (SIME-LINKING LOADER) ДЛЯ ТЕКСТА ПРОГРАММЫ, СОЗДАННОГО ПРОГРАММОЙ КОДА EZ, ТАК ПОЛУЧАЕТСЯ ПОТОМУ, ЧТО ПРОГРАММА КОДА EZ МОЖЕТ ПРИМЕНЯТЬСЯ ТОЛЬКО ДЛЯ ВВОДА НЕБОЛЬШИХ МОДУЛЕЙ, НЕ ПРЕВЫШАЮЩИХ 8ØØ БАЙТОВ ИЛИ 2ØØ КОМАНД,

ТАК ЧТО ДЛЯ БОЛЬШИХ ПРОГРАММ МЫ ПРИМЕНЯЕМ ПРОГРАММУ КОДА EZ ДЛЯ РАЗРАБОТКИ МОДУЛЕЙ И ЗАПИСЫВАЕМ КАЖДЫЙ МОДУЛЬ В ВИДЕ ТЕКСТА НА МАШИННОМ ЯЗЫКЕ НА КАССЕТУ,

ЗАТЕМ МЫ ИСПОЛЬЗУЕМ ПРОГРАММУ HEXLOAD, ГОРАЗДО МЕНЬШУЮ ПО ОБ"ЕМУ ПРОГРАММУ НА ЯЗЫКУ "БЕЙСИК", ЧТОБЫ ЗАГРУЗИТЬ ЭТИ МОДУЛИ И СВЯЗАТЬ ИХ, ПЕРЕМЕСТИВ В ОТВЕДЕННЫЕ ИМ ЯЧЕЙКИ ПАМЯТИ.

МЫ ПРИМЕНИМ ЭТОТ МЕТОД НА ПРАКТИКЕ ПРИ РАЗРАБОТКЕ ПРОГРАММЫ FREEWAY FROG,

ИДЕЙНЫЕ ОСНОВАНИЯ ПРОГРАММЫ HEXLOAD

ИДЕИ, ЛЕЖАЩИЕ В ОСНОВЕ ПРОГРАММЫ HEXLOAD, ЧЕРЕЗВЫЧАЙНО ПРОСТЫ,

МОНИТОРНАЯ ПРОГРАММА НА САМОМ ДЕЛЕ УСТАНАВЛИВАЕТ КАМТОР СИСТЕМЫ "БЕЙСИК" РАВНЫМ 26999,

ЭТО ОЗНАЧАЕТ, ЧТО ВЫ МОЖЕТЕ ВВОДИТЬ СВОЮ ПРОГРАММУ НА МАШИННОМ ЯЗЫКЕ В ЛЮБОЕ МЕСТО МЕЖДУ ЯЧЕЙКАМИ 27ØØØ И 32578 ДЛЯ "СПЕКТРУМ" С 16К ОПЕРАТИВНОЙ ПАМЯТИ ИЛИ С 27ØØØ ПО 65343 ДЛЯ "СПЕКТРУМ" С 48К,

HEXLOAD - ДОСТАТОЧНО ПРОСТАЯ МОНИТОРНАЯ ПРОГРАММА ДЛЯ РАБОТЫ С ТЕКСТОМ НА МАШИННОМ ЯЗЫКЕ, ОНА ПРЕДСТАВЛЯЕТ ТАКИЕ БАЗИСНЫЕ ФУНКЦИИ УПРАВЛЕНИЯ:

WRITE - ЗАПИСЬ В ПАМЯТЬ В ШЕСТНАДЦАТЕРИЧНОМ ФОРМАТЕ;
SAVE - ЗАПИСЬ ИЗ ПАМЯТИ НА КАССЕТУ;
LOAD - ЗАПИСЬ С КАССЕТЫ В ПАМЯТЬ;
LIST - РАСПЕЧАТКА СОДЕРЖИМОГО ПАМЯТИ НАЧИНАЯ С НАЧАЛЬНОГО АДРЕСА;
MOVE - ПЕРЕМЕЩЕНИЕ СОДЕРЖИМОГО ПАМЯТИ ИЗ ОДНОЙ ГРУППЫ ЯЧЕЕК В ДРУГУЮ,

СВОДКА КОМАНД ПРОГРАММЫ

1, WRITE

ЗАПИСЬ ТЕКСТА ПРОГРАММЫ В ШЕСТНАДЦАТЕРИЧНОМ ФОРМАТЕ В ПАМЯТЬ, ПРОЦЕДУРА:

А) В ОТВЕТ НА ПОДСКАЗКУ ВВЕСТИ НАЧАЛЬНЫЙ АДРЕС ПАМЯТИ, С КОТОРОГО ВЫ ХОТИТЕ ВЕСТИ ЗАПИСЬ, В ДЕСЯТИЧНОМ ФОРМАТЕ, АДРЕС ДОЛЖЕН ЛЕЖАТЬ В ДИАПАЗОНЕ:

27ØØØ - 32578 ДЛЯ ПАМЯТИ ОБ"ЕМОМ 16К
27ØØØ - 65346 ДЛЯ ПАМЯТИ ОБ"ЕМОМ 48К
НАПРИМЕР: ЗАПИСЬ ПО АДРЕСУ: 27ØØØ(ENTER)

В) ВВЕДИТЕ ТЕКСТ ПРОГРАММЫ В ШЕСТНАДЦАТЕРИЧНОМ ФОРМАТЕ,

С) НАЖМИТЕ КЛАВИШУ "М", ЧТОБЫ ВЕРНУТЬСЯ К ОСНОВНОМУ МЕНЮ,

2, SAVE

ЗАПИСЬ СОДЕРЖИМОГО ПАМЯТИ НА КАССЕТУ, ПРОЦЕДУРА:

А) ВВОДИМОЕ ЗНАЧЕНИЕ НАЧАЛЬНОГО АДРЕСА ПАМЯТИ, С КОТОРОГО НАЧИНАЕТСЯ ЗАПИСЬ, МОЖЕТ БЫТЬ ЛЮБЫМ ИЗ СЛЕДУЮЩЕГО ДИАПАЗОНА

Ø - 32767 ДЛЯ ПАМЯТИ ОБ"ЕМОМ 16К
Ø - 65535 ДЛЯ ПАМЯТИ ОБ"ЕМОМ 48К

Б) ВВЕДИТЕ КОЛИЧЕСТВО ЗАПИСЫВАЕМЫХ БАЙТОВ,

В) ВВЕДИТЕ ИМЯ ЗАПИСЫВАЕМОГО МОДУЛЯ,

Г) НАЖМИТЕ ЛЮБУЮ КЛАВИШУ, КОГДА КАССЕТА БУДЕТ ГОТОВА,

Д) ВОЗМОЖНОСТЬ ПРОВЕРКИ МОДУЛЯ, ЗАПИСАННОГО НА КАССЕТУ.

НЕПЛОХО СДЕЛАТЬ ПРОВЕРКУ, ЧТОБЫ УДОСТОВЕРИТЬСЯ, ЧТО МОДУЛЬ НЕ ИСПОРТИЛСЯ ВО ВРЕМЯ ПРОЦЕДУРЫ ЗАПИСИ,

3, LOAD

ЗАГРУЗКА МОДУЛЯ НА МАШИННОМ ЯЗЫКЕ С КАССЕТЫ, ПРОЦЕДУРА:

А) ВВЕДИТЕ АДРЕС ПАМЯТИ, С КОТОРОГО МОДУЛЬ НАЧИНАЕТ ЗАГРУЖАТЬСЯ, АДРЕС ДОЛЖЕН ЛЕЖАТЬ В ТОМ ЖЕ ДИАПАЗОНЕ, ЧТО И ДЛЯ КОМАНД ЗАПИСИ,

Б) ВВЕДИТЕ ИМЯ, ИСПОЛЬЗОВАННОЕ ПРИ ЗАПИСИ МОДУЛЯ, ЕСЛИ ВЫ НЕ УВЕРЕНЫ В ИМЕНИ, ПРОСТО НАЖМИТЕ КЛАВИШУ (ENTER)

4, LIST

ВЫДАЧА СОДЕРЖИМОГО ПАМЯТИ, НАЧИНАЯ С НЕКОТОРОГО АДРЕСА, ПРОЦЕДУРА:

А) ВВЕДИТЕ АДРЕС НАЧАЛА РАСПЕЧАТКИ, ЭТО МОЖЕТ БЫТЬ ЛЮБОЙ АДРЕС КАК В ПРИВЕДЕННОЙ ВЫШЕ КОМАНДЕ SAVE,

Б) НАЖМИТЕ ЛЮБУЮ КЛАВИШУ ДЛЯ ПРОДОЛЖЕНИЯ ВЫДАЧИ,

С) НАЖМИТЕ КЛАВИШУ "M", ЧТОБЫ ВЕРНУТЬСЯ К ОСНОВНОМУ МЕНЮ,

5, MOVE

ПЕРЕМЕСТИТЬ СОДЕРЖИМОЕ ПАМЯТИ С НАЧАЛЬНОГО ПО КОНЕЧНЫЙ АДРЕС ПО НОВОМУ АДРЕСУ В ПАМЯТИ,

ПРОЦЕДУРА:

А) ВВЕСТИ НАЧАЛЬНЫЙ АДРЕС ПЕРЕМЕЩАЕМОЙ ПОРЦИИ, ЛЮБОЙ АДРЕС, КАК В ДИАПАЗОНЕ ДЛЯ КОМАНДЫ SAVE

Б) ВВЕСТИ КОНЕЧНЫЙ АДРЕС ПЕРЕМЕЩАЕМОЙ ПОРЦИИ, ЛЮБОЙ АДРЕС, КАК В ДИАПАЗОНЕ ДЛЯ КОМАНДЫ SAVE,

В) ВВЕДИТЕ АДРЕС ПАМЯТИ, КУДА ПЕРЕМЕЩАЕТСЯ ИНФОРМАЦИЯ, ДИАПОЗОН АДРЕСОВ, КАК В КОМАНДЕ WRITE,

Г) ВЫ МОЖЕТЕ ДАЖЕ С ПОМОЩЬЮ ЭТОЙ КОМАНДЫ КОПИРОВАТЬ ИЗ ПЗУ В ПАМЯТЬ С ПРОИЗВОЛЬНЫМ ДОСТУПОМ,
НАПРИМЕР;
ПЕРЕМЕСТИТЬ, НАЧИНАЯ С: 0(ENTER)
ПЕРЕМЕСТИТЬ, КОНЧАЯ: 1000(ENTER)
ПЕРЕМЕСТИТЬ ПО АДРЕСУ: 32000(ENTER)

ТАКАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ДЕЙСТВИЙ ПРИВЕДЕТ К ПЕРЕМЕЩЕНИЮ СОДЕРЖИМОГО ПЗУ С 0 ПО 1000 АДРЕС В ПАМЯТЬ С ПРОИЗВОЛЬНЫМ ДОСТУПОМ ПО АДРЕСУ 32000,

ЗАМЕЧАНИЯ: ЛЮБАЯ ПОПЫТКА ВВОДА В ПРИВЕДЕННЫХ ВЫШЕ КОМАНДАХ НАРУШАЮЩАЯ ДОПУСТИМЫЙ ДИАПОЗОН АДРЕСОВ, ПРИВЕДЕТ К ПОВТОРЕНИЮ ПОДСКАЗКИ ДЛЯ ВВОДА,

ПОПРОБУЙТЕ С ПОМОЩЬЮ ЭТОГО МОНИТОРА ВВЕСТИ МОДУЛЬ, РАЗРАБОТАННЫЙ НАМИ С ПОМОЩЬЮ ПРОГРАММЫ КОДА EZ,

```
HEXLOAD
COPYIRIHT (C) 1982
100 REM
110 REM MONITOR PROGRAM
120 CLEAR 26999 : LET ZE = PI - PI
    : LET ON = PI / PI : LET TW = ON+ON
    : LET QK = 256 : LET LM = 27000
    : LET MR = 140 : LET WL = 340
130 GO SUB 2000
140 CLS
    : PRINT "START OF,MACHINE CODE AREA = "
    ; LM
150 PRINT "MENU" : PRINT
    : PRINT
    "    WRITE MACHINE CODE,,,,,,,,,1"
160 PRINT
    : PRINT
    "    SAVE  MACHINE CODE,,,,,,,,,2"
170 PRINT
    : PRINT
    "    LOAD  MACHINE CODE,,,,,,,,,3"
180 PRINT
    : PRINT
    "    LIST  MACHINE CODE,,,,,,,,,4"
190 PRINT
    : PRINT
```

```
    "  . MOVE  MACHINE CODE,,,,,,,,,5"
200 PRINT
    : PRINT
    "PLEASE PRESS APPROPRIATE KEY,"
210 LET G$ = INKEY$
220 IF G$ = " " OR G$ = "M" THEN STOP
230 IF G$ = "" OR G$ < "1" OR G$ > "5"
    THEN GO TO 210
240 CLS
    : PRINT "START OF MACHINE CODE AREA = "
    ; LM
250 GO TO 300* VAL G$
300 REM  INV WRITE****************** IRU
         ---                        ---
310 INPUT "WRITE TO ADDRESS :"; D
320 IF D >. MM OR D < LM THEN GO TO 310
330 PRINT : PRINT "WRITE ADDRESS : "; D
    ; PRINT "TO RETURN TO MENU ENTER ""M"""
340 LET A$ = ""
350 IF A$ = "" THEN INPUT "ENTER HEX, CODE :"
    ; A$
360 IF A$(ON) =" " OR A$(ON) = "M"
    THEN GO TO MR
370 IF LEN A$/TW <> INT (LEN A$/TW)
    THEN PRINT "INCORRECT ENTRY ";
    : GO TO WL
380 LET C = ZE
390 FOR F = 16 TO ON STEP -15
400 LET A = CODE A$((F. = 16)+TW*(F = ON))
410 IF A < 48 OR A > 102 OR (A > 57 AND A < 65)
    OR (A > 70 AND A < 97)
    THEN PRINT "INCORRECT ENTRY ";
    : GO TO WL
420 LET C = C+F*((A < 58)*(A-48)
    +(A > 64 AND A < 71)*(A-55)+(A < 96)*(A-87))
430 NEXT F ; POKE D, C ; LET D = D+ON
440 PRINT A$(TO TW); "  ";
450 LET A$ =A$(3 TO )
460 IF D = UDG
    THEN PRINT
    "WARNING ;YOU ARE NOW IN THE USER
    GRAPHICS AREA!"
    : GO TO WL
470 IF D = UDR-20
    THEN PRINT
    "WARNING ; YOU ARE NOW IN ROUTINES
    MEMORY AREA!"
    ; GO TO WL
480 GO TO WL+ON
600 REM  INV SAVE****************** IRU
         ---                       ---
610 INPUT "SAVE M,C, FROM ADDRESS :"; A
620 INPUT "NUMBER OF BYTES TO BE SAVED :"; N
630 INPUT "NAME OF THE ROUTINE :"; A$
640 SAVE A$ CODE A, N
650 PRINT "DO YOU WISH TO VERITY?"
660 INPUT V$
670 IF V$ <> "Y" THEN GO TO MR
680 PRINT "REWIND TAPE AND PRESS ""PLAY""","
690 VERIFY A$ CODE A, N
700 PRINT "O,K," ; PAUSE 50
710 GO TO MR
900 REM  INV LOAD****************** IRU
    ---                          ---
910 INPUT
    "LOAD M,C, TO ADDRESS STARTING    : "
    ; A
```

```
920 IF A > MM OR A < LM THEN GO TO 910
930 INPUT "PROGRAM NAME :"; A$
940 PRINT ""PLAY"" ON TAPE,"
950 LOAD A$ CODE A : GO TO MR
1200 REM  INV LIST********************* IRU
    ---                              ---
1210 LET A$ = "0123456789ABCDEF"
1220 INPUT "LIST ADDRESS :"; D
1230 PRINT "PRESS ""M"" TO RETURN TO MENU,"
1240 LET A = INT (PEEK D/16)
     : LET B = PEEK D-16* INT ( PEEK D/16)
1250 PRINT D; TAB 7; A$(A+ON); A$(B+ON)
1260 LET D = D+ON
1270 IF INKEY$ = " " OR INKEY$ = "M" THEN GO TO MR
1280 GO TO 1240
1500 REM  INV MOVE******************** IRU
     ---                             ---
1510 INPUT "MOVE FROM MEMORY : "; FM
1520 INPUT "MOVE UNTIL MEMORY : "; UM
1530 INPUT "MOVE TO MEMORY : "; TM
1560 FOR I =
1570 POKE MP, PEEK I
1580 LET MP = MP+ON
1590 NEXT I
1600 GO TO MR
1610 LET MP = UM+TM-FM
1620 FOR I = UM TO FM STEP -ON
1630 POKE MP, PEEK I
1640 LET MP = MP-ON
1650 NEXT I
1660 GO TO MR
2000 LET RT = PEEK 23732+QK* PEEK 23733
2010 IF RT = 65535 THEN LET MM = 65347
     : LET UDG = 65367
2020 IF RT = 32767 THEN LET MM = 32579
     : LET UDG = 32599
2030 LET N1 = INT (UDG/QK)
2040 POKE 23675, UDG-N1*QK : POKE 23676, N1
2050 RETURN
```

COPYRIGHT – АВТОРСКИЕ ПРАВА; MONITOR PROGRAM – МОНИТОРНАЯ ПРОГРАММА; START OF,MACHINE CODE AREA – НАЧАЛО ОБЛАСТИ ТЕКСТА В МАШИННЫХ КОМАНДАХ; MENU – МЕНЮ; WRITE MACHINE CODE – ЗАПИСАТЬ ТЕКСТ НА МАШИННОМ ЯЗЫКЕ; SAVE MACHINE CODE – ЗАПОМНИТЬ ТЕКСТ НА МАШИННОМ ЯЗЫКЕ; LOAD MACHINE CODE – ЗАГРУЗИТЬ ТЕКСТ НА МАШИННОМ ЯЗЫКЕ; LIST MACHINE CODE – РАСПЕЧАТАТЬ ТЕКСТ НА МАШИННОМ ЯЗЫКЕ; MOVE MACHINE CODE – ПЕРЕМЕСТИТЬ ТЕКСТ НА МАШИННОМ ЯЗЫКЕ; PLEASE PRESS APPROPRIATE KEY – ПОЖАЛУЙСТА, НАЖМИТЕ СООТВЕТСТВУЮЩУЮ КЛАВИШУ; START OF MACHINE CODE AREA – НАЧАЛО ОБЛАСТИ ТЕКСТА В МАШИННЫХ КОМАНДАХ; WRITE TO ADDRESS – ЗАПИСАТЬ ПО АДРЕСУ; TO RETURN TO MENU ENTER – ЧТОБЫ ВЕРНУТЬСЯ К МЕНЮ, ВВЕДИТЕ; ENTER HEX, CODE – ВВЕДИТЕ В ШЕСТНАДЦАТЕРИЧНЫХ КОДАХ; INCORRECT ENTRYНЕВЕРНЫЙ ВВОД; WARNING; YOU ARE NOW IN USER GRAPHICS AREA ПРЕДУПРЕЖДЕНИЕ: ВЫ СЕЙЧАС В ГРАФИЧЕСКОЙ ОБЛАСТИ ПОЛЬЗОВАТЕЛЯ; WARNING: YOU ARE NOW IN THE ROUTINES MEMORY AREA – ПРЕДУПРЕЖДЕНИЕ: ВЫ СЕЙЧАС В ОБЛАСТИ ПАМЯТИ ПРОГРАММ1; SAVE M,C, FROM ADDRESS – ЗАПОМНИТЬ ТЕКСТ НА МАШИННОМ ЯЗЫКЕ; NUMBER OF BYTES TO BE SAVED – КОЛИЧЕСТВО ЗАПОМИНАЕМЫХ БАЙТОВ; NAME OF THE ROUTINEНАЗВАНИЕ ПРОГРАММЫ; DO YOU WISH TO VERIFY – ХОТИТЕ ПРОВЕРИТЬ? REWIND TAPE AND PRESS "PLAY" – ПЕРЕМОТАЙТЕ ПЛЕНКУ И НАЖМИТЕ "ПУСК"; LOAD M,C, TO ADDRESS STARTING – ЗАГРУЗИТЬ ТЕКСТ НА МАШИННОМ ЯЗЫКЕ ПО АДРЕСУ; PROGRAM NAME – НАЗВАНИЕ ПРОГРАММЫ; PRESS "PLAY" ON TAPE – НАЖМИТЕ "ПУСК" ДЛЯ ПЛЕНКИ; LIST ADDRESS – АДРЕС ДЛЯ РАСПЕЧАТКИ; PRESS "M" TO RETURN TO MENU – НАЖМИТЕ "M", ЧТОБЫ ВЕРНУТЬСЯ К МЕНЮ; MOVE FROM MEMORY

- ПЕРЕМЕСТИТЬ, НАЧИНАЯ С ЯЧЕЙКИ ПАМЯТИ, ... ПЕРЕМЕСТИТЬ ДО ЯЧЕЙКИ ПАМЯТИ; MOVE TO MEMORY - ПЕРЕМЕСТИТЬ ПО АДРЕСУ,

П Р И Л О Ж Е Н И Е А
ТАБЛИЦА КЛАВИШ "СПЕКТРУМ"

INPUT VALUE IN A FOR OFF H	D4	D3	D2	D1	D0
	V	C	X	Z	CAP
0FE H					SHIFT
	G	F	D	S	A
0FD H					
	7	R	E	W	Q
0FB H					
	5	4	3	2	1
0F7 H					
	6	7	8	9	0
0EF H					
	Y	U	I	O	P
0DF H					
	H	J	K	L	ENTER
0BF H					
	B	N	M	SYM	BREAK
				SHIFT	SPACE
07F H					
X:	16	8	4	2	1

INPUT VALUE IN A OFF - ВХОДНОЕ ЗНАЧЕНИЕ В А ДЛЯ; CAP - ПРОПИСНЫЕ БУКВЫ; SHIFT - СМЕНА РЕГИСТРА; SYM - СОКРАЩЕНИЕ РАСШИФРОВАТЬ НЕ УДАЛОСЬ; ENTER - ВВОД; BREAK - ПРЕРЫВАНИЕ; SPACE- ПРОБЕЛ,

ПРИМЕЧАНИЕ: ЧТОБЫ ВЫПОЛНИТЬ ПРЕРЫВАНИЕ ПО КЛАВИШЕ;

1, ЗАГРУЗИТЕ В РЕГИСТР А ВХОДНОЕ ЗНАЧЕНИЕ ИЗ СООТВЕТСТВУЮЩЕГО РЯДА,

```
      LD  A  07E          : НИЖНИЙ РЯД
```

2, ПРИМИТЕ ИНФОРМАЦИЮ С ВХОДНОГО ПОРТА 0FEH,

```
      IN  A  (0FEH)
```

3, ПРОВЕРЬТЕ, ЧТО ДЛЯ НУЖНОЙ КЛАВИШИ DX ИМЕЕТ НИЗКОЕ ЗНАЧЕНИЕ,

```
      AND  1              : ПРЕРЫВАНИЕ ПО КЛАВИШЕ
                          : BREAK/SPACE
```

4, ЕСЛИ НОЛЬ, ТО КЛАВИША НАЖАТА,

```
      JR  Z ,КЛАВИША НАЖАТА : В НОРМАЛЬНОМ СОСТОЯНИИ
                            : ЗНАЧЕНИЕ ВСЕГДА ВЫСОКОЕ
```

ASSEMBLER ZX SPECTRUM NOT COPI

60

П Р И Л О Ж Е Н И Е В

MEMORI IN HEX	ATTRIBUTE IN HEX	LINE	MEMORI IN HEX	ATTRIBUTE IN HEX
4000	5800	0	401F	581F
4020	5820	1	403F	583F
4040	5840	2	405F	585F
4060	5860	3	407F	587F
4080	5880	4	409F	589F
40A0	58A0	5	40BF	58BF
40C0	58C0	6	40DF	58DF
40E0	58E0	7	40FF	58FF
4800	5900	8	481F	591F
4820	5920	9	483F	593F
4840	5940	10	485F	595F
4860	5960	11	487F	597F
4880	5980	12	489F	599F
48A0	59A0	13	48BF	59BF
48C0	59C0	14	48DF	59DF
48E0	59E0	15	48FF	59FF
5000	5A00	16	501F	5A1F
5020	5A20	17	503F	5A3F
5040	5A40	18	505F	5A5F
5060	5A60	19	507F	5A7F
5080	5A80	20	509F	5A9F
50A0	5AA0	21	50BF	5ABF
50C0	5AC0	22	50DF	5ADF
50E0	5AE0	23	50FF	5AFF

MEMORY IN HEX – ПАМЯТЬ В ШЕСТНАДЦАТЕРИЧНОМ ФОРМАТЕ;
ATTRIBUTE IN HEX – АТРИБУТ В ШЕСТНАДЦАТЕРИЧНОМ ФОРМАТЕ;
LINE – СТРОКА

ТАБЛИЦА НАБОРА ЛИТЕР "СПЕКТРУМ"

HEX	НОВ	0	1	2	3	4	5	6	7
LOB	BITS	000	001	010	011	100	101	110	111
0	0000	NU	INK CTRL	SPACE	0		P		
1	0001	NU	RAPER CTRI	!	1	A	Q		
2	0010	NU	FLASH CTRI	"	2	B	R		
3	0011	NU	BRIGHT CTRI	#	3	C	S		
4	0100	NU	INVERSE CTRI	$	4	D	T		
5	0101	NU	OVER CTRI	%	5	E	U		
6	0110	PRINT	AT CTRI	&	6	F	V		
7	0111	EDIT	TAB CTRI	'	7	G	W		
8	1000	CURSOR LEFT	NU	(	8	H	X		
9	1001	CURSOR RIGHT	NU	)	9	I	Y		
A	1010	CURSOR DOWN	NU	*	:	J	Z		
B	1011	CRSOR UP	NU	+	;	K			
C	1100	DELETE	NU	,	<	L			
D	1101	ENTER	NU	-	=	M			
E	1110	NUMBER	NU	.	>	N			
F	1111	NU	NU	/	?	P	-		
		<---NON PRINTABLE------>		<-------PRINTABLE------>					

NB: NU = NOT USED,

NEX - ШЕСТНАДЦАТЕРИЧНОЕ ПРЕДСТАВЛЕНИЕ; LOB - МЛАДШИЙ БАЙТ; НОВ - СТАРШИЙ БАЙТ; INK CTRI - УПРАВЛЕНИЕ ЧЕРНИЛАМИ; SPACE - ПОБОЛ; PAPER CTRI - УПРАВЛЕНИЕ БУМАГОИ; FLASH CTRI - УПРАВЛЕНИЕ РЕЖИМОМ МИГАНИЯ; BRIGHT CTRI - УПРАВЛЕНИЕ ЯРКОСТЬЮ; INVERSE CTRI - УПРАВЛЕНИЕ ИНВЕРСИЕЙ ЯРКОСТИ; OVER CTRI- УПРАВЛЕНИЕ ПРЕВЫШЕНИЕМ; PRINT - ПЕЧАТЬ; AT CTRI - УПРАВЛЕНИЕ; EDIT - РЕДАКТИРОВАНИЕ; TAB CTRI - УПРАВЛЕНИЕ ТАБУЛЯЦИЕЙ; CURSOR LEFT - КУРСОР ВЛЕВО; CURSOR RIGHT - КУРСОР ВПРАВО; CURSOR DOWN - КУРСОР ВНИЗ; CURSOR UP - КУРСОР ВВЕРХ; DELETE - УДАЛЕНИЕ; ENTER - ВВОД; NUMBER - ЧИСЛО; NON PRINTABLE - НЕПЕЧАТНЫЕ; PRINTABLE - ПЕЧАТАЕМЫЕ; NB: - ЗАМЕЧАНИЕ; NU - НЕ ИСПОЛЬЗУЕТСЯ,

П Р И Л О Ж Е Н И Е D

ТАБЛИЦЫ ПРЕОБРАЗОВАНИЯ ДЕСЯТИЧНЫХ ЧИСЕЛ В ШЕСТНАДЦАТЕРИЧНЫЕ

NEX	0	1	2	3	4	5	6	7	8	9	A
0	0	1	2	3	4	5	6	7	8	9	10
1	16	17	18	19	20	21	22	23	24	25	25
2	32	33	34	35	36	37	38	39	40	41	42
3	48	49	50	51	52	53	54	55	56	57	58
4	64	65	66	67	68	69	70	71	72	73	74
5	80	81	82	83	84	85	86	87	88	89	90
6	96	97	98	99	100	101	102	103	104	105	1061
7	112	113	114	115	116	117	118	119	120	121	1221
8	128	129	130	131	132	133	134	135	136	137	1381
9	144	145	146	147	148	149	150	151	152	153	1541
A	160	161	162	163	164	165	166	167	168	169	1701
B	176	177	178	179	180	181	182	183	184	185	1861
C	192	193	194	195	196	197	198	199	200	201	2022
D	208	209	210	211	212	213	214	215	216	217	2182
E	224	225	226	227	228	229	230	231	232	233	2342
F	240	241	242	243	244	245	246	247	248	249	2502

B	C	D	E	F	00XX	XX00
11	12	13	14	15	0	0
27	28	29	30	31	256	4096
43	44	45	46	47	512	8192
59	60	61	62	63	768	12288
75	76	77	78	79	1024	16384
91	92	93	94	95	1280	20480
07	108	109	110	111	1536	24576
23	124	125	126	127	1792	28672
39	140	141	142	143	2048	32768
55	156	157	158	159	2304	36864
71	172	173	174	175	2560	40960
87	188	189	190	191	2816	45056
03	204	205	206	207	3072	49152
19	220	221	222	223	3328	53248
35	236	237	238	239	3564	57344
51	252	253	254	255	3840	61440

NEX – ШЕСТНАДЦАТЕРИЧНОЕ,

МЫ МОЖЕМ ПОКАЗАТЬ ПРИМЕНЕНИЕ ЭТОЙ ТАБЛИЦЫ НА ПРИМЕРЕ, ДАВАЙТЕ НАЙДЕМ ШЕСТНАДЦАТЕРИЧНЫЙ ЭКВИВАЛЕНТ ДЕСЯТИЧНОГО ЧИСЛА 6200, НАМ НУЖНО ОПРЕДЕЛИТЬ 16-БИТОВОЕ ДВОИЧНОЕ ЧИСЛО, Т, Е,,

0001BBBB BBBBBBBB
НОВ LOB

НОВ – СТАРШИЙ БАЙТ; LOB – МЛАДШИЙ БАЙТ

1, ИЗ САМОЙ ЛЕВОЙ КОЛОНКИ ТАБЛИЦЫ ПОД ЗАГОЛОВКОМ XX00 МЫ НАХОДИМ, ЧТО 6200 НАХОДИТСЯ МЕЖДУ 4096 И 8192, ТАК ЧТО МЫ ВЫБИРАЕМ МЕНЬШЕЕ ЗНАЧЕНИЕ 4096 И ИЗ ЗНАЧЕНИЯ РЯДА МЫ БЕРЕМ 4 САМЫХ СТАРШИХ БИТА СТАРШЕГО БАЙТА РАВНЫЕ 1, Т, Е, 01,

BBBBBBBB BBBBBBBB
НОВ LOB

НОВ – СТАРШИЙ БАЙТ; LOB – МЛАДШИЙ БАЙТ;

2, ВТОРОЙ ШАГ СОСТОИТ В ТОМ, ЧТО МЫ ОПРЕДЕЛЯЕМ СЛЕДУЮЩИЕ ПО СТАРШИНСТВУ 4 БИТА СТАРШЕГО БАЙТА, МЫ НАХОДИМ РАЗНОСТЬ МЕЖДУ 6200 И 4096, РАВНУЮ 2104, ПОСКОЛЬКУ РАЗНОСТЬ ВСЕ ЕЩЕ ПРЕВЫШАЕТ 255, МЫ ОБРАЩАЕМСЯ КО ВТОРОЙ СЛЕВА КОЛОНКЕ ТАБЛИЦЫ ПОД ЗАГОЛОВКОМ 00XX И ВЫЯСНЯЕМ, ЧТО 2104 НАХОДИТСЯ МЕЖДУ 2048 И 2304, ВНОВЬ МЫ ВЫБИРАЕМ МЕНЬШЕЕ ЗНАЧЕНИЕ 2048 И ПО ЗНАЧЕНИЮ РЯДА ПОЛУЧАЕМ, ЧТО СЛЕДУЮЩЕЕ ПО СТАРШИНСТВУ 4 БИТА СТАРШЕГ БАЙТА РАВНЫ 8, Т, Е, 1000,

00011000 BBBBBBBB
НОВ LOB

НОВ - СТАРШИЙ БАЙТ; LOB - МЛАДШИЙ БАЙТ.
3, ТРЕТИЙ ШАГ СОСТОИТ В ОПРЕДЕЛЕНИИ МЛАДШЕГО БАЙТА ЧИСЛА, МЫ ОБНАРУЖИВАЕМ, ЧТО РАЗНОСТЬ МЕЖДУ 2104 И 2048 РАВНА 56, Т.К. ЛЕЖИТ НА ПЕРЕСЕЧЕНИИ РЯДА 3 И КОЛОНКИ 8. ТАК ЧТО МЫ ПРИНИМАЕМ МЛАДШИЙ БАЙТ РАВНЫМ 38Н.

ØØØ1100 ØØ111ØØØ
НОВ LOB

ИТАК, ШЕСТНАДЦАТЕРИЧНОЕ ЗНАЧЕНИЕ ЧИСЛА 62ØØ РАВНО 1838Н.

П Р И Л О Ж Е Н И Е Е

ТАБЛИЦА ПРЕОБРАЗОВАНИЯ ДЕСЯТИЧНЫХ ЧИСЕЛ В ФОРМЕ ДОПОЛНЕНИЯ ДО 2 В ШЕСТНАДЦАТЕРИЧНЫЙ ФОРМАТ

HEX	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	-128	-127	-126	-125	-124	-123	-122	-121	-12Ø	-119	-118	-117	-116	-115	-114	-113
9	-112	-111	-11Ø	-1Ø9	-1Ø8	-1Ø7	-1Ø6	-1Ø5	-1Ø4	-1Ø3	-1Ø2	-1Ø1	-1ØØ	-99	-98	-97
A	-96	-95	-94	-93	-92	-91	-9Ø	-89	-88	-87	-86	-85	-84	-83	-82	-81
B	-8Ø	-79	-78	-77	-76	-75	-74	-73	-72	-71	-7Ø	-69	-68	-67	-66	-65
C	-64	-63	-62	-61	-6Ø	-59	-58	-57	-56	-55	-54	-53	-52	-51	-5Ø	-49
D	-48	-47	-46	-45	-44	-43	-42	-41	-4Ø	-39	-38	-37	-36	-35	-34	-33
E	-32	-31	-3Ø	-29	-28	-27	-26	-25	-24	-23	-22	-21	-2Ø	-19	-18	-17
F	-16	-15	-14	-13	-12	-11	-1Ø	-9	-8	-7	-6	-5	-4	-3	-2	-1

HEX - ШЕСТНАДЦАТЕРИЧНОЕ

П Р И Л О Ж Е Н И Е F

ШЕСТНАДЦАТЕРИЧНАЯ ТАБЛИЦА СЛОЖЕНИЯ

+	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Ø	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	1Ø
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	1Ø	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	1Ø	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	1Ø	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	1Ø	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	1Ø	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	1Ø	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	1Ø	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	1Ø	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	1Ø	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	1Ø	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	1Ø	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	1Ø	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	1Ø	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	1Ø	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

П Р И Л О Ж Е Н И Е G

СВОДНАЯ ТАБЛИЦА ИЗМЕНЕНИЯ ФЛАГОВ

INSTRUCTION	I	C	I	Z	I	P/V	I	S	I	N	I	H	I	COMMENTS
ADC NL, SS	I	#	I	#	I	V	I	#	I	Ø	I	X	I	16-BIT ADD WITH CARRY
ADX S, ADD S	I	#	I	#	I	V	I	#	I	Ø	I	#	I	8-BIT ADD OR ADD WITH
	I		I		I		I		I		I		I	CARRY
ADD DD, SS	I	#	I	-	I	-	I	-	I	Ø	I	X	I	16-BIT ADD
AND S	I	Ø	I	#	I	P	I	#	I	Ø	I	1	I	LOGICAL OPERATIONS
BIT B,S	I	-	I	#	I	X	I	X	I	Ø	I	1	I	STATE OF BIT B OF LO-
	I		I		I		I		I		I		I	CATION S IS COPIED
	I		I		I		I		I		I		I	INTO THE Z FLAG
CCF	I	#	I	-	I	-	I	-	I	Ø	I	X	I	COMPLEMENT CARRY
CPD:CPDR:	I	-	I	#	I	#	I	X	I	1	I	X	I	BLOK SEARCH INSTRUC-
CPI:CPIR	I		I		I		I		I		I		I	TION Z=1;IF A=(HL),
	I		I		I		I		I		I		I	ELSE Z=Ø P/V=1; IF
	I		I		I		I		I		I		I	BC#G,OTHERVISE P/V=Ø
CP S	I	#	I	#	I	V	I	#	I	1	I	#	I	COMPARE ACCUMULATOR
CPL	I	-	I	-	I	-	I	-	I	1	I	1	I	COMPLEMENT ACCUM-TOR
DAA	I	#	I	#	I	P	I	#	I	-	I	#	I	DECIMAL ABJUST ACCU-
	I		I		I		I		I		I		I	MULATOR
DEC S	I	-	I	#	I	V	I	#	I	1	I	#	I	8-BIT DECREMENT
IN R,(C)	I	-	I	#	I	P	I	#	I	Ø	I	Ø	I	INPUT REGISTER INDI-
	I		I		I		I		I		I		I	RECT
INC S	I	-	I	#	I	V	I	#	I	Ø	I	#	I	8-BIT INCREMENT
IND; INI	I	-	I	#	I	X	I	X	I	1	I	X	I	BLOK INPUT Z=Ø IF B#Ø
	I		I		I		I		I		I		I	ELSE Z=1
INDR; INIR	I	-	I	1	I	X	I	X	I	1	I	X	I	BLOK INPUT Z=Ø IF B#Ø
	I		I		I		I		I		I		I	ELSE Z=1
LDA,I;LDA,R	I	-	I	#	I	IFF	I	#	I	Ø	I	Ø	I	CONTENT OF INTERRUPT
	I		I		I		I		I		I		I	ENADLE FLIP-FLOP IS
	I		I		I		I		I		I		I	COPIED INTO THE P/V
	I		I		I		I		I		I		I	FLAG
LDD; LDI	I	-	I	X	I	#	I	X	I	Ø	I	Ø	I	BLOCK TRANSFER
	I		I		I		I		I		I		I	INSTRUCTIONS
LDDR; LDIR	I	-	I	X	I	Ø	I	X	I	Ø	I	Ø	I	P/V=1 IF BC#Ø,OTHER-
	I		I		I		I		I		I		I	WISE P/V=Ø
NEG	I	#	I	#	I	V	I	#	I	1	I	#	I	NEGATE ACCUMULATOR
OR S	I	Ø	I	#	I	P	I	#	I	Ø	I	Ø	I	LOGICAL OR ACCUM-TOR
OTDR;OTIR	I	-	I	1	I	X	I	X	I	1	I	X	I	BLOK OUTPUT;Z=Ø IF
	I		I		I		I		I		I		I	B#Ø OTHERWISE Z=1
OUTD;OUTI	I	-	I	#	I	X	I	X	I	1	I	X	I	BLOK OUTPUT;Z=Ø IF
	I		I		I		I		I		I		I	B#Ø OTHERWISE Z=1
RLA;RLCA;	I	#	I	-	I	-	I	-	I	Ø	I	Ø	I	ROTATE ACCUMULATOR
RRA;RRCA	I		I		I		I		I		I		I	
RLD; RRD	I	-	I	#	I	P	I	#	I	Ø	I	/	I	ROTATE DIGIT LEFT AND
	I		I		I		I		I		I		I	RIGHT
RLS;RLC S;RR SI	#	I	#	I	P	I	#	I	Ø	I	Ø	I		WOTATE AND SHIFT
RRC S;SLA S;	I		I		I		I		I		I			LOCATION S
SRA S	I		I		I		I		I		I			
SRL S	I		I		I		I		I		I			
SBC HL; SS	I	#	I	#	I	V	I	#	I	1	I	X	I	16-BIT SUBTRACT WITH
	I		I		I		I		I		I		I	CARRY
SCF	I	1	I	-	I	-	I	-	I	Ø	I	Ø	I	SET CARRY
SBC S; SUB S	I		I		I	V	I		I	1	I		I	8-BIT SUBTRACT WITH
	I		I		I		I		I		I		I	CARRY
XOR X	I	Ø	I		I	P	I		I	Ø	I	Ø	I	EXCLUSIVE OR ACCUMU-
	I		I		I		I		I		I		I	LATOR

INSTRUCTION -КОМАНДА; COMMENTS -КОМЕНТАРИИ;16-BIT ADD WITH CARRY-16-БИТОВОЕ СЛОЖЕНИЕ С ПЕРЕНОСОМ;8-BIT ADD OR ADD WITH CARRY-8-БИТОВОЕ СЛОЖЕНИЕ ИЛИ СЛОЖЕНИЕ С ПЕРЕНОСОМ;16-BIT ADD-16-БИТОВОЕ СЛОЖЕНИЕ; LOGICAL OPERATION-ЛОГИЧЕСКИЕ ОПЕРАЦИИ;STATE OF BIT B OF LOCATION S IS COPED INTO THE Z FLAG-СОСТОЯНИЕ БИТА В ЯЧЕЙКИ S КОПИРУЕТСЯ В ФЛАГ Z;COMPLEMENT CARR

-ДОПОЛНЕНИЕ ПЕРЕНОСА;BLOCK SEARCH INSTRUCTION-КОМАНДА БЛОЧ
НОГО ПОИСКА;IF-ЕСЛИ;ELSE-ИНАЧЕ;OTHERWISE - В ПРОТИВНОМ
СЛУЧАЕ;COMPARE ACCUMULATOR-СРАВНЕНИЕ НАКАПЛИВАЮЩЕГО РЕГИСТРА
COMPLEMENT ACCUMULATOR-ДОПОЛНЕНИЕ НАКАПЛИВАЮЩЕГО РЕГИСТРА;
DECIMAL ADJUST ACCUMULATOR-ДЕСЯТИЧНАЯ НАСТРОЙКА НАКАПЛИВАЮЩЕ
ГО РЕГИСТРА;8-BIT DECREMENT-8-БИТОВОЕ УМЕНЬШЕНИЕ;INPUT RE-
GISTER INDIRECT-КОСВЕННЫЙ ВВОД РЕГИСТРА; 8-BIT INCREMENT -
8-БИТОВОЕ УВЕЛИЧЕНИЕ; BLOCK INPUT - ВВОД БЛОКА; CONTENT OF
INTERRUPT ENABLE FLIP-FLOP IS COPIED INTO THE P/V FLAG-СОДЕР
ЖИМОЕ ТРИГГЕРА ПРЕРЫВАНИЯ КОПИРУЕТСЯ В P/V ФЛАГ;BLOCK TRANS-
FER INSTRUCTIONS -КОМАНДЫ ПЕРЕДАЧИ БЛОКА; NEGATE ACCUMULATOR
ОТРИЦАНИЕ ДЛЯ НАКАПЛИВАЮЩЕГО РЕГИСТРА;LOGICAL OR ACCUMULATOR
ЛОГИЧЕСКОЕ ИЛИ ДЛЯ НАКАПЛИВАЮЩЕГО РЕГИСТРА; BLOCK OUTPUT -
ВЫВОД БЛОКА; ROTATE ACCUMULATOR - ЦИУЛИЧЕСКИЙ СДВИГ НАКАПЛИ-
ВАЮЩЕГО РЕГИСТРА; ROTATE DIGIT LEFT ANT RIGHT - ЦИКЛИЧЕСКИЙ
СДВИГ РАЗРЯДА ВЛЕВО ИЛИ ВПРАВО; ROTATE AND SHIFT LOCATION -
ЦИКЛИЧЕСКИЙ СДВИГ И СДВИГ ЯЧЕЙКИ;16-BIT SUBTRACT WITH CARRY-
16-БИТОВОЕ ВЫЧИТАНИЕ С ПЕРЕНОСОМ; SET CARRY-УСТАНОВКА ПЕРЕНО
СА;8-BIT SUBTRACT WITH CARRY -8-БИТОВОЕ ВЫЧИТАНИЕ С ПЕРЕНОСО
EXCLUSIVE OR ACCUMULATOR -ИСКЛЮЧАЮЩЕЕ ИЛИ ДЛЯ НАКАПЛИВАЮЩЕГО
РЕГИСТРА,

СИМВОЛ	ОПЕРАЦИЯ
C	ФЛАГ ПЕРЕНОСА, C=1, ЕСЛИ ОПЕРАЦИЯ ПРИВЕЛА К ПЕРЕНОСУ ИЗ САМОГО ЗНАЧАЩЕГО БИТА ОПЕРАНДА ИЛИ РЕЗУЛЬТАТА,
Z	ФЛАГ НУЛЯ, Z=1 ,ЕСЛИ РЕЗУЛЬТАТ ОПЕРАЦИИ НУЛЕВОЙ,
S	ФЛАГ ЗНАКА, S=1, ЕСЛИ САМЫЙ ЗНАЧАЩИЙ БИТ РЕЗУЛЬТАТА РАВЕН ЕДИНИЦЕ, Т.Е, ЧИСЛО ОТРИЦАТЕЛЬНОЕ,
P/V	ФЛАГ ЧЕТНОСТИ ИЛИ ПЕРЕПОЛНЕНИЯ, ЧЕТНОСТЬ (P) И ПЕРЕПОЛНЕНИЕ (Ø) ОТНОСЯТСЯ К ОДНОМУ И ТОМУ ЖЕ ФЛАГУ, ДЛЯ ЛОГИЧЕСКИХ ОПЕРАЦИЙ ЭТОТ ФЛАГ ЗАДАЕТ ЧЕТНОСТЬ РЕЗУЛЬТАТА, А ДЛЯ АРИФМЕТИЧЕСКИХ - ПЕРЕПОЛНЕНИЕ, ЕСЛИ В P/V ХРАНИТСЯ ЧЕТНОСТЬ: P/V=1, ЕСЛИ РЕЗУЛЬТАТ ОПЕРАЦИИ ЧЕТНЫЙ, P/V=Ø, ЕСЛИ РЕЗУЛЬТАТ НЕЧЕТНЫЙ, ЕСЛИ P/V СОДЕРЖИТ ПЕРЕПОЛНЕНИЕ: P/V=1, ЕСЛИ В РЕЗУЛЬТАТЕ ОПЕРАЦИИ ПОЛУЧИЛОСЬ ПЕРЕПОЛНЕНИЕ,
H	ФЛАГ ПОЛОВИННОГО ПЕРЕНОСА, H=1, ЕСЛИ ПРИ ОПЕРАЦИИ СЛОЖЕНИЯ ИЛИ ВЫЧИТАНИЯ ПРОИЗОШЕЛ ПЕРЕНОС ИЛИ ЗАЕМ В ЧЕТВЕРТОМ БИТЕ НАКАПЛИВАЮЩЕГО РЕГИСТРА,
N	ФЛАГ СЛОЖЕНИЯ (ВЫЧИТАНИЯ), N=1, ЕСЛИ ПРЕДЫДУЩЕЙ ОПЕРАЦИЕЙ БЫЛО ВЫЧИТАНИЕ, ФЛАГ N И H ИСПОЛЬЗУЮТСЯ В СОВОКУПНОСТИ С КОМАНДОЙ ДЕСЯТИЧНОЙ НАСТРОЙКИ (DAA) ДЛЯ ИСПРАВЛЕНИЯ РЕЗУЛЬТАТА В ПРАВИЛЬНОЙ УПАКОВАННОЙ ДВОИЧНО-КОДИРОВАННЫЙ ДЕСЯТИЧНЫЙ ФОРМАТ ПОСЛЕ СЛОЖЕНИЯ ИЛИ ВЫЧИТАНИЯ С ПРИМЕНЕНИЕМ ОПЕРАНДОВ В УПАКОВАННОМ ДВОИЧНО-КОДИРОВАННОМ ДЕСЯТИЧНОМ ФОРМАТЕ,
#	ФЛАГ УСТАНАВЛИВАЕТСЯ В СООТВЕТСТВИИ С РЕЗУЛЬТАТОМ ОПЕРАЦИИ,
-	ФЛАГ НЕ ИЗМЕНЯЕТСЯ В РЕЗУЛЬТАТЕ ОПЕРАЦИИ,
Ø	ФЛАГ СБРАСЫВАЕТСЯ (=Ø) ОПЕРАЦИЕЙ,
1	ФЛАГ УСТАНАВЛИВАЕТСЯ (=1) ОПЕРАЦИЕЙ,
X	ФЛАГ РЕЗУЛЬТАТА НЕИЗВЕСТЕН,
V	ДЕЙСТВИЕ НА ФЛАГ P/V СООТВЕТСТВУЕТ РЕЗУЛЬТАТУ: ПЕРЕПОЛНЕНИЯ ПРИ ОПЕРАЦИИ,
P	ДЕЙСТВИЕ НА ФЛАГ P/V СООТВЕТСТВУЕТ ЧЕТНОСТИ РЕЗУЛЬТАТА ОПЕРАЦИИ,
R	ЛЮБОЙ ИЗ РЕГИСТРОВ ЦП: A, B, C, D, E, H, L,
S	ЛЮБАЯ 8-БИТОВАЯ ЯЧЕЙКА ДЛЯ ВСЕХ РЕЖИМОВ АДРЕСАЦИИ, ДОПУСТИМЫХ ДЛЯ КОНКРЕТНОЙ КОМАНДЫ,
SS	ЛЮБАЯ 16-БИТОВАЯ ЯЧЕЙКА ДЛЯ ВСЕХ РЕЖИМОВ АДРЕСАЦИИ, ДОПУСТИМЫХ ДЛЯ ЭТОЙ КОМАНДЫ,
R	ОБНОВЛЕНИЕ РЕГИСТРА,
N	8-БИТОВОЕ ЗНАЧЕНИЕ ИЗ ДИАПОЗОНА Ø - 255 ,

ИТОВОЕ ЗНАЧЕНИЕ ИЗ ДИАПОЗОНА Ø - 65535,

П Р И Л О Ж Е Н И Е I

КОМАНДЫ ЦП Z80 В ПОРЯДКЕ КОДОВ ОПЕРАЦИИ

HEXADECIMAL	MNEMONIC	HEXADECIMAL	MNEMONIC	HEXADECIMAL	MNEM
00	NOP	38XX	JR C,DIS	70	LD (HL),B
01 XXXX	LD BC,NN	39	ADD HL,SP	71	LD (HL),C
02	LD(BC),A	3AXXXX	LD A,(NN)	72	LD (HL),D
03	INC BC	3B	DEC SP	73	LD (HL),E
04	INC B	3C	INC A	74	LD (HL),H
05	DEC B	3D	DEC A	75	LD (HL),L
06 XX	LD B,N	3EXXXX	LD A	76	HALT
07	RLCA	3F	CCF	77	LD (HL),A
08	EX AF,AF"	40	LD B,B	78	LD A,B
09	ADD HL,BC	41	LD B,C	79	LD A,C
0A	LD A,(BC)	42	LD B,D	7A	LD A,D
0B	DEC BC	43	LD B,E	7B	LD A,E
0C	INC C	44	LD B,H	7C	LD A,H
0D	DEC C	45	LD B,L	7D	LD A,L
0E XX	LD C,N	46	LD B,(HL)	7E	LD A,(HL)
0F	RRCA	47	LD B,A	7F	LD A,A
10 XX	DJNZ DIS	48	LD C,B	80	ADD A,B
11 XXXX	LD DE,NN	49	LD C,C	81	ADD A,C
12	LD (DE),A	4A	LD C,D	82	ADD A,D
13	INC DE	4B	LD C,E	83	ADD A,E
14	INC D	4C	LD C,H	84	ADD A,H
15	DEC D	4D	LD C,L	85	ADD A,L
16 XX	LD D,N	4E	LD C,(HL)	86	ADD A,(HL)
17	RLA	4F	LD C,A	87	ADD A,A
18 XX	JR DIS	50	LD D,B	88	ADC A,B
19	ADD HL,DE	51	LD D,C	89	ADC A,C
1A	LD A,(DE)	52	LD D,D	8A	ADC A,D
1B	DEC DE	53	LD D,E	8B	ADC A,E
1C	INC E	54	LD D H	8C	ADC A,H
1D	DEC E	55	LD D,L	8D	ADC A,L
1E XX	LD E,N	56	LD D,(HL)	8E	ADC A,(HL)
1F	RRA	57	LD D,A	8F	ADC A,A
20 XX	JR NZ,DIS	58	LD E,B	90	SUB B
21 XXXX	LD HL,NN	59	LD E,C	91	SUB C
22 XXXX	LD(NN),HL	5A	LD E,D	92	SUB D
23	INC HL	5B	LD E,E	93	SUB E
24	INC H	5C	LD E,H	94	SUB H
25	DEC H	5D	LD E,L	95	SUB L
26 XX	LD H,N	5E	LD E,(HL)	96	SUB (HL)
27	DAA	5F	LD E,A	97	SUB A
28 XX	JR Z,DIS	60	LD H,B	98	SBC A,B
29	ADD HL,HL	61	LD H,C	99	SBC A,C
2A XXXX	LD HL(NN)	62	LD H,D	9A	SBC A,D
2B	DEC HL	63	LD H,E	9B	SBC A,E
2C	INC L	64	LD H,H	9C	SBC A,H
2D	DEC L	65	LD H,L	9D	SBC A,L
2E XX	LD L,N	66	LD H,(HL)	9E	SBC A,(HL)
2F	CPL	67	LD H,A	9F	SBC A,A
30 XX	JR NC,DIS	68	LD L,B	A0	AND B
31 XXXX	LD SP,NN	69	LD L,C	A1	AND C
32 XXXX	LD (NN),A	6A	LD L,D	A2	AND D
33	INC SP	6B	LD L,E	A3	AND E
34	INC (HL)	6C	LD L,H	A4	AND H
35	DEC (HL)	6D	LD L,L	A5	AND L
3620XX	LD (HL),N	6E	LD L,(HL)	A6	AND (HL)
37	SCF	6F	LD L,A	A7	AND A
A8	XOR B	E5	PUSH HL	CB22	SLA D
A9	XOR C	E6XX	AND N	CB23	SLA E
AA	XOR D	E7	RST 20H	CB24	SLA H
AB	XOR E	E8	RET PE	CB25	SLA L
AC	XOR H	E9	JP (HL)	CB26	SLA (HL)
AD	XOR L	EAXXXX	JE PE NN	CB27	SLA A

AE	XOR (HL)	!EB	EX DE,HL	!CB28	SRA B
AF	XOR A	!ECXXXX	CALL PE,NN	!CB29	SRA C
B0	OR B	!EEXX	XOR N	!CB2A	SRA D
B1	OR C	!EF	RST 28H	!CB2B	SRA E
B2	OR D	!F0	RET P	!CB2C	SRA H
B3	OR E	!F1	POP AF	!CB2D	SRA L
B4	OR H	!F2XXXX	JR P,NN	!CB2E	SRA (HL)
B5	OR L	!F3	D1	!CB2F	SRA A
B6	OR (HL)	!F4XXXX	CALL P,NN	!CB38	SRL B
B7	OR A	!F5	RUSH AF	!CB39	SRL C
B8	CP B	!F620XX	OR N	!CB3A	SRL D
B9	CP C	!F7	RST 30H	!CB3B	SRL E
BA	CP D	!F8	RET N	!CB3C	SRL H
BB	CP E	!F9	LD SP,HL	!CB3D	SRL L
BC	CP H	!FAXXXX	JP N,NN	!CB3E	SRL (HL)
BD	CP L	!FB	E1	!CB3F	SRL A
BE	CP (HL)	!FCXXXX	CALL M,NN	!CB40	BIT 0,B
BF	CP A	!FE20XX	CP N	!CB41	BIT 0,C
C0	RET NZ	!FF	RST 38H	!CB42	BIT 0,D
C1	POP BC	!CB00	RLC 8	!CB43	BIT 0,E
C2XXXX	JP NZ,NM	!CB01	RLC C	!CB44	BIT 0,H
C3XXXX	JP NM	!CB02	RLC D	!CB45	BIT 0,L
C4XXXX	CALL NZ,NM	!CB03	RLC E	!CB46	BIT 0,(HL)
C5	PUSH BCHL	!CB04	RLC H	!CB47	BIT 0,A
C6XX	ADD A,N	!CB05	RLC L	!CB48	BIT 1,B
C7	RST 0	!CB06	RLC (HL)	!CB49	BIT 1,C
C8	RET Z	!CB07	RLC A	!CB4A	BIT 1,D
C9	RET	!CB08	RRC B	!CB4B	BIT 1,E
CAXXXX	JP Z,NM	!CB09	RRC C	!CB4C	BIT 1,H
CCXXXX	CALL Z,NN	!CB0A	RRC D	!CB4D	BIT 1,L
CDXXXX	CALL NN	!CB0B	RRC E	!CB4E	BIT 1,(HL)
CEXX	ADC A,N	!CB0C	RRC H	!CB4F	BIT 1,A
CF	RST 8	!CB0D	RRC L	!CB50	BIT 2,B
D0	RET NC	!CB0E	RRC (HL)	!CB51	BIT 2,C
D1	POP DE	!CB0F	RRC A	!CB52	BIT 2,D
D2XXXX	JP NC,NN	!CB10	RL B	!CB53	BIT 2,E
D3XX	OUT (N),A	!CB11	RL C	!CB54	BIT 2,H
D4XXXX	CALL NC,NN	!CB12	RL D	!CB55	BIT 2,L
D5	PUSH DE	!CB13	RL E	!CB56	BIT 2,(HL)
D6XX	SUB N	!CB14	RL H	!CB57	BIT 2,A
D7	RST 10H	!CB15	RL L	!CB58	BIT 3,B
D8	RET C	!CB16	RL (HL)	!CB59	BIT 3,C
D9	EXX	!CB17	RL A	!CB5A	BIT 3,D
DAXXXX	JP C,NN	!CB18	RR B	!CB5B	BIT 3,E
DBXX	IN A,(N)	!CB19	RR C	!CB5C	BIT 3,H
DCXXXX	CALL C,NN	!CB1A	RR D	!CB5D	BIT 3,L
DEXX	SBC A,N	!CB1B	RR E	!CB5E	BIT 3,(HL)
DF	RST 18H	!CB1C	RR H	!CB5F	BIT 3,A
E0	RET PO	!CB1D	RR L	!CB60	BIT 4,B
E1	POP HL	!CB1E	RR (HL)	!CB61	BIT 5,C
E2XXXX	JP PO,NN	!CB1F	RR A	!CB62	BIT 4,D
E3	EX (SP),HL	!CB20	SLA B	!CB63	BIT 4,E
E4XXXX	CALL PO,NN	!CB21	SLA C	!CB64	BIT 4,H
CB65	BIT 4,L	!CB9F	RES 3,A	!CBD9	SET 3,C
CB66	BIT 4,(HL)	!CBA0	RES 4,B	!CBDA	SET 3,D
CB67	BIT 4,A	!CBA1	RES 4,C	!CBDB	SET 3,E
B68	BIT 5,B	!CBA2	RES 4,D	!CBDC	SET 3,H
CB69	BIT 5,C	!CBA3	RES E,E	!CBDD	SET 3,L
CB6A	BIT 5,D	!CBA4	RES E,H	!CBDE	SET 3,(HL)
CB6B	BIT 5,E	!CBA5	RES 4,L	!CBDF	SET 3,A
CB6C	BIT 5,H	!CBA6	RES 4,(HL)	!CBE0	SET 4,B
CB6D	BIT 5,L	!CBA7	RES 4,A	!CBE1	SET 4,C
CB6E	BIT 5,(HL)	!CBA8	RES 5,B	!CBE2	SET 4,D
CB6F	BIT 5,A	!CBA9	RES 5,C	!CBE3	SET 4,E
CB70	BIT 6,B	!CBAA	RES 5,D	!CBE4	SET 4,H
CB71	BIT 6,C	!CBAB	RES 5,E	!CBE5	SET 4,L
CB72	BIT 6,D	!CBAC	RES 5,H	!CBE6	SET 4,(HL)

CB73	BIT 6,E	!CBAD	RES 5,L	!CBE7	SET 4,A
CB74	BIT 6,H	!CBAE	RES 5,(HL)	!CBE8	SET 5,B
CB75	BIT 6,L	!CBAF	RES 5,A	!CBE9	SET 5,C
CB76	BIT 6,(HL)	!CBB0	RES 6,B	!CBEA	SET 5,D
CB77	BIT 6,A	!CBB1	RES 6,C	!CBEB	SET 5,E
CB78	BIT 7,B	!CBB2	RES 6,D	!CBEC	SET 5,H
CB79	BIT 7,C	!CBB3	RES 6,E	!CBED	SET 5,L
CB7A	BIT 7,D	!CBB4	RES 6,H	!CBEE	SET 5,(HL)
CB7B	BIT 7,E	!CBB5	RES 6,L	!CBEF	SET 5,A
CB7C	BIT 7,H	!CBB6	RES 6,(HL)	!CBF0	SET 6,B
CB7D	BIT 7,L	!CBB7	RES 6,A	!CBF1	SET 6,C
CB7E	BIT 7,(HL)	!CBB8	RES 7,B	!CBF2	SET 6,D
CB7F	BIT 7,A	!CBB9	RES 7,C	!CBF3	SET 6,E
CB80	RES 0,B	!CBBA	RES 7,D	!CBF4	SET 6,H
CB81	RES 0,C	!CBBB	RES 7,E	!CBF5	SET 6,L
CB82	RES 0,D	!CBBC	RES 7,H	!CBF6	SET 6,(HL)
CB83	RES 0,E	!CBBD	RES 7,L	!CBF7	SET 6,A
CB84	RES 0,H	!CBBE	RES 7,(HL)	!CBF8	SET 7,B
CB85	RES 0,L	!CBBF	RES 7,A	!CBF9	SET 7,C
CB86	RES 0,(HL)	!CBC0	SET 0,B	!CBFA	SET 7,D
CB87	RES 0,A	!CBC1	SET 0,C	!CBFB	SET 7,E
CB88	RES 1,B	!CBC2	SET 0,D	!CBFC	SET 7,H
CB89	RES 1,C	!CBC3	SET 0,E	!CBFD	SET 7,L
CB8A	RES 1,D	!CBC4	SET 0,H	!CBFE	SET 7,(HL)
CB8B	RES 1,E	!CBC5	SET 0,L	!CBFF	SET 7,A
CB8C	RES 1,H	!CBC6	SET 0,(HL)	!DD09	ADD IX,BC
CB8D	RES 1,L	!CBC7	SET 0,A	!DD19	ADD IX,DE
CB8E	RES 1,(HL)	!CBC8	SET 1,B	!DD21XXXX	LD IX,NN
CB8F	RES 1,A	!CBC9	SET 1,C	!DD22XXXX	LD (NN),IX
CB90	RES 2,B	!CBCA	SET 1,D	!DD23	INC IXC),L
CB91	RES 2,C	!CBCB	SET 1,E	!DD29	ADD IX,IX
CB92	RES 2,D	!CBCC	SET 1,H	!DD2AXXXX	LD IX,(NN)
CB93	RES 2,E	!CBCD	SET 1,L	!DD2B	DEC IX
CB94	RES 2,H	!CBCE	SET 1,(HL)	!DD34XX	INC (IX+Д)
CB95	RES 2,L	!CBCF	SET 1,A	!DD35XX	DEC (IX+Д)
CB96	RES 2,(HL)	!CBD0	SET 2,B	!DD36XX20	LD (IX+Д),N
CB97	RES 2,A	!CBD1	SET 2,C	!DD39	ADD IX,SP
CB98	RES 3,B	!CBD2	SET 2,D	!DD46XX	LD B,(IX+Д)
CB99	RES 3,C	!CBD3	SET 2,E	!DD4EXX	LD C,(IX+Д)
CB9A	RES 3,D	!CBD4	SET 2,H	!DD56XX	LD D,(IX+Д)
CB9B	RES 3,E	!CBD5	SET 2,L	!DD5EXX	LD E,(IX+Д)
CB9C	RES 3,H	!CBD6	SET 2,(HL)	!DD66XX	LD H,(IX+Д)
CB9D	RES 3,L	!CBD7	SET 2,A	!DD6EXX	LD L,(IX+Д)
CB9E	RES 3,(HL)	!CBD8	SET 3,B	!DD70XX	LD (IX+Д),B
DD71XX	LD (IX+Д),C	!ED47	LD 1,A	!FD39	ADD IY,CP
DD72XX	LD (IX+Д),D	!ED48	INC C,(C)	!FD46XX	LD B,(IY+Д)
DD73XX	LD (IX+Д),E	!ED49	OUT (C),C	!FD4EXX	LD C,(IY+Д)
DD74XX	LD (IX+Д),H	!ED4A	ADC HL,BC	!FD56XX	LD D,(IY+Д)
DD75XX	LD (IX+Д),L	!ED4BXXXX	LD BC,(NN)	!FD5EXX	LD E,(IY+Д)
DD77XX	LD (IX+Д),A	!ED4D	RET 1	!FD66XX	LD H,(IY+Д)
DD7EXX	LD A,(IX+Д)	!ED50	IN D,(C)	!FD6EXX	LD L,(IY+Д)
DD86XX	ADD A,(IX+Д)	!ED51	OUT (C),D	!FD70XX	LD (IY+Д),B
DD8EXX	ADC A,(IX+Д)	!ED52	SBC HL,DE	!FD71XX	LD (IY+Д),C
DD96XX	SUB (IX+Д)	!ED53XXXX	LD (NN),DE	!FD72XX	LD (IY+Д),D
DD9EXX	SBC A,(IX+Д)	!ED56	IN 1	!FD73XX	LD (IY+Д),E
DDA6XX	AND (IX+Д)	!ED57	LD A,1	!FD74XX	LD (IY+Д),H
DDAEXX	XOR (IX+Д)	!ED58	IN E,(C)	!FD75XX	LD (IY+Д),L
DDB6XX	OR (IX+Д)	!ED59	OUT (C),E	!FD77XX	LD (IY+Д),A
DDBEXX	CP (IX+Д)	!ED5A	ADC HL,DE	!FD7EXX	LD A,(IY+Д)
DDE1	POP IX	!ED5BXXXX	LD DE,(NN)	!FD86XX	ADD A,(IY+Д)
DDE3	EX (SP),IX	!ED5E	IN 2	!FD8EXX	ADC A,(IY+Д)
DDE5	PUSH IX	!ED60	IN H,(C)	!FD96XX	SUB (IY+Д)
DDE9	JP (IX)	!ED61	OUT (C),H	!FD9EXX	SBC A,(IY+Д)
DDF9	LD SP,IX	!ED62	SBC HL,HL	!FDA6XX	AND (IY+Д)
DDCBXX06	RLC (IX+Д)	!ED67	RRD	!FDAEXX	XOR (IY+Д)
DDCBXX0E	RRC (IX+Д)	!ED68	IN L,(C)	!FDB6XX	OR (IY+Д)
DDCBXX16	RL (IX+Д)	!ED69	OUT (C),L	!FDBEXX	CP (IY+Д)

DDCBXX1E	RR (IX+Д)	!ED6A	ADC HL,HL	!FDE1	POP IY
DDCBXX26	SLA (IX+Д)	!ED6F	RLD	!FDE3	EX (SP),IY
DDCBXX2E	SRA (IX;D)	!ED72	SBC HL,SP	!FDE5	RUSH IY
DDCBXX3E	SRL (IX+Д)	!ED73XXXX	LD (NN),SP	!FDE9	JP (IY)
DDCBXX46	BIT0,(IX+Д)	!ED78	IN A,(C)	!FDF9	LD SP,IY
DDCBXX4E	BIT1,(IX+Д)	!ED79	OUT (C),A	!FDCBXX06	RLC (IY+Д)
DDCBXX56	BIT2,(IX+Д)	!ED7A	ADC HL,SP	!FDCBXX0E	RRC (IY+Д)
DDCBXX5E	BIT3,(IX+Д)	!ED7BXXXX	LD SP,(NN)	!FDCBXX16	RL (IY+Д)
DDCBXX66	BIT4,(IX+Д)	!EDA0	LDI	!FDCBXX1E	RR (IY+Д)
DDCBXX6E	BIT5,(IX+Д)	!EDA1	CPI	!FDCBXX26	SLA (IY+Д)
DDCBXX76	BIT6,(IX+Д)	!EDA2	INI	!FDCBXX2E	SRA (IY+Д)
DDCBXX7E	BIT7,(IX+Д)	!EDA3	OUTI	!FDCBXX3E	SRL (IY+Д)
DDCBXX86	RES0,(IX+Д)	!EDA8	LDD	!FDCBXX46	BIT 0,(IY+Д)
DDCBXX8E	RES1,(IX+Д)	!EDA9	CP0	!FDCBXX4E	BIT 1,(IY+Д)
DDCBXX96	RES2,(IX+Д)	!EDAA	IND	!FDCBXX56	BIT 2,(IY+Д)
DDCBXX9E	RES3,(IX+Д)	!EDAB	OUTD	!FDCBXX5E	BIT 3,(IY+Д)
DDCBXXA6	RES4,(IX+Д)	!ED80	LDIR	!FDCBXX66	BIT 4,(IY+Д)
DDCBXXAE	RES5,(IX+Д)	!ED81	CPIR	!FDCBXX6E	BIT 5,(IY+Д)
DDCBXXB6	RES6,(IX+Д)	!ED82	INIR	!FDCBXX76	BIT 6,(IY+Д)
DDCBXXBE	RES7,(IX+Д)	!ED83	OTIR	!FDCBXX7E	BIT 7,(IY+Д)
DDCBXXC6	SET0,(IX+Д)	!ED88	LDDR	!FDCBXX86	RES 0,(IY+Д)
DDCBXXCE	SET1,(IX+Д)	!ED89	CPDR	!FDCBXX8E	RES 1,(IY+Д)
DDCBXXD6	SET2,(IX+Д)	!ED8A	INDR	!FDCBXX96	RES 2,(IY+Д)
DDCBXXDE	SET3,(IX+Д)	!ED8B	OTDR	!FDCBXX9E	RES 3,(IY+Д)
DDCBXXE6	SET4,(IX+Д)	!ED09	ADD IV,BC	!FDCBXXA6	RES 4,(IY+Д)
DDCBXXEE	SET5,(IX+Д)	!ED19	ADD IV,DC	!FDCBXXAE	RES 5,(IY+Д)
DDCBXXF6	SET6,(IX+Д)	!ED21XXXX	LD IV,NN	!FDCBXXB6	RES 6,(IY+Д)
DDCBXXFE	SET7,(IX+Д)	!FD22XXXX	LD (NN),IV	!FDCBXXBE	RES 7,(IY+Д)
ED40	IN B,(C)	!FD23	INC IY	!FDCBXXC6	SET 0,(IY+Д)
ED41	OUT (C),B	!FD29	ADD IY,IY	!FDCBXXCE	SET 1,(IY+Д)
ED42	SBC HL,BC	!FD2AXXXX	LD IY,(NN)	!FDCBXXD6	SET 2,(IY+Д)
ED43XXXX	LD (NN),IX	!FD2B	DEC IY	!FDCBXXDE	SET 3,(IY+Д)
ED44	NEG	!FD34XX	INC (IY+Д)	!FDCBXXE6	SET 4,(IY+Д)
ED45	RETN	!FD35XX	DEC (IY+Д)	!FDCBXXEE	SET 5,(IY+Д)
ED46	IM 0	!FD36XX20	LD (IY+Д),N	!FDCBXXF6	SET 6,(IY+Д)
		!		!FDCBXXFE	SET 7,(IY+Д)

П Р И Л О Ж Е Н И Е Н

КОМАНДЫ ЦП Z80 В ПОРЯДКЕ ВОЗРАСТАНИЯ МНЕМОНИЧЕСКИХ ОБОЗНАЧЕНИЙ

MNEMONIC	HEXADECIMAL	MNEMONIC	HEXADECIMAL	MNEMONIC	HEXADECIMA
ADC A,(HL)	8E	BIT Ø,B	CB4Ø	BIT 5,E	CB6B
ADC A,(IX+Д)	DD8EXX	BIT Ø,C	CB41	BIT 5,H	CB6C
ADC A,(IY+Д)	FD8EXX	BIT Ø,D	CB42	BIT 5,L	CB6D
ADC A,A	8F	BIT Ø,E	CB43	BIT 6,(HL)	CB76
ADC A,B	88	BIT Ø,H	CB44	BIT6,(IX+Д)	DDCBXX7
ADC A,C	89	BIT Ø,L	CB45	BIT6,(IY+Д)	FDCBXX7
ADC A,D	8A	BIT 1,(HL)	CB4E	BIT 6,A	CB77
ADC A,N	CEXX	BIT 1,(IX+Д)	DDCBXX4E	BIT 6,B	CB7Ø
ADC A,E	8B	BIT 1,(IY+Д)	FDCBXX4E	BIT 6,C	CB71
ADC A,H	8C	BIT 1,A	CB4F	BIT 6,D	CB72
ADC A,L	8D	BIT 1,B	CB48	BIT 6,E	CB73
ADC HL,BC	ED4A	BIT 1,C	CB49	BIT 6,H	CB74
ADC HL,DE	ED5A	BIT 1,D	CB4A	BIT 6,L	CB75
ADC HL,HL	ED6A	BIT 1,E	CB4B	BIT 7,(HL)	CB7E
ADD HL,SP	ED7A	BIT 1,H	CB4C	BIT7,(IX+Д)	DDCBXX7
ADD A,(HL)	86	BIT 1,L	CB4D	BIT7,(IY+Д)	FDCBXX7
ADD A,(IX+Д)	DD86XX	BIT 2,(HL)	CB56	BIT 7,A	CB7F
ADD A,(IY+Д)	FD86XX	BIT 2,(IX+Д)	DDCBXX56	BIT 7,B	CB78
ADD A,A	87	BIT 2,(IY+Д)	FDCBXX56	BIT 7,C	CB79
ADD A,B	8Ø	BIT 2,A	CB57	BIT 7,D	CB7A
ADD A,C	81	BIT 2,B	CB5Ø	BIT 7,E	CB7B
ADD A,D	82	BIT 2,C	CB51	BIT 7,H	CB7C
ADD A,N	C6XX	BIT 2,D	CB52	BIT 7,L	CB7D
ADD A,E	83	BIT 2,E	CB53	CALL ADDR	CDXXXX
ADD A,H	84	BIT 2,H	CB54	CALL C,ADDR	DCXXXX
ADD A,L	85	BIT 2,L	CB55	CALL M,ADDR	FCXXXX
ADD HL,BC	Ø9	BIT 3,(HL)	CB5E	CALL NC,ADDR	D4XXXX
ADD HL,DE	19	BIT 3,(IX+Д)	DDCBXX5E	CALL NZ,ADDR	C4XXXX
ADD HL,HL	29	BIT 3,(IY+Д)	FDCBXX5E	CALL P,ADDR	F4XXXX
ADD HL,SP	39	BIT 3,A	CB5F	CALL PE,ADDR	ECXXXX
ADD IX,BC	DDØ9	BIT 3,B	CB58	CALL PO,ADDR	E4XXXX
ADD IX,DE	DD19	BIT 3,C	CB59	CALL Z,ADDR	CCXXXX
ADD IX,IX	DD29	BIT 3,D	CB5A	CPF	3F
ADD IX,SP	DD39	BIT 3,E	CB5B	CP (HL)	BE
ADD IY,BC	FDØ9	BIT 3,H	CB5C	CP (IX+Д)	DDBEXX
ADD IY,DE	FD19	BIT 3,L	CB5D	CP (IY+Д)	FDBEXX
ADD IY,IY	FD29	BIT 4,(HL)	CB66	CP A	BF
ADD IY,SP	FD39	BIT 4,(IX+Д)	DDCBXX66	CP B	B8
AND (HL)	A6	BIT 4,(IY+Д)	FDCBXX66	CP C	B9
AND (IX+Д)	DDA6XX	BIT 4,A	CB67	CP D	BA
AND (IY+Д)	FDA6XX	BIT 4,B	CB6Ø	CP N	FEXX
AND A	A7	BIT 4,C	CB61	CP E	BB
AND B	AØ	BIT 4,D	CB62	CP H	BC
AND C	A1	BIT 4,E	CB63	CP L	BD
AND D	A2	BIT 4,H	CB64	CPD	EDA9
AND N	E6XX	BIT 4,L	CB65	CPDR	EDB9
AND E	A3	BIT 5,(HL)	CB6E	CPI	EDA1
AND H	A4	BIT 5,(IX+Д)	DDCBXX6E	CPIR	EDB1
AND L	A5	BIT 5,(IY+Д)	FDCBXX6E	CPL	2F
BIT Ø,(HL)	CB46	BIT 5,A	CB6F	DAA	27
BITØ,(IX+Д)	DDCBXX46	BIT 5,B	CB68	DEC (HL)	35
BITØ,(IY+Д)	FDCBXX46	BIT 5,C	CB69	DEC (IX+Д)	DD35XX
BIT Ø,A	CB47	BIT 5,D	CB6A	DEC (IY+Д)	FD35XX
DEC A	3D	JP C,ADDR	DAXXXX	LD A,E	7B
DEC B	Ø5	JP M,ADDR	FAXXXX	LD A,H	7C
DEC BC	ØB	JP NC,ADDR	D2XXXX	LD A,I	ED57
DEC C	ØD	JP NZ,ADDR	C2XXXX	LD A,L	7D
DEC D	15	JP P,ADDR	F2XXXX	LD A,R	ED5F
DEC DE	1B	JP PE,ADDR	EAXXXX	LD B,(HL)	46
DEC E	1D	JP PO,ADDR	E2XXXX	LD B,(IX+Д)	DD46XX
DEC H	25	JP Z,ADDR	CAXXXX	LD B,(IY+Д)	FD46XX

DEC HL	2B	!JR C,Д	38XX	!LD B,A	47
DEC IX	DD2B	!JR Д	18XX	!LD B,B	40
DEC IY	FD2B	!JR NC,Д	30XX	!LD B,C	41
DEC L	2D	!JR NZ,Д	20XX	!LD B,D	42
DEC SP	3B	!JR Z,Д	28XX	!LD B,N	06XX
DI	F3	!LD (ADDR),A	32XXXX	!LD B,E	43
DJNZ,Д	10XX	!LD(ADDR),BC	ED43XXXX	!LD B,H	44
E1	F8	!LD(ADDR),DE	ED53XXXX	!LD B,L	45
EX (SP),HL	E3	!LD(ADDR),HL	ED63XXXX	!LD BC,(ADDR)	ED4BXXXX
EX (SP),IX	DDE3	!LD(ADDR),HL	22XXXX	!LD BC,NN	01XXXX
EX (SP),IY	FDE3	!LD(ADDR),IX	DD22XXXX	!LD C,(HL)	4E
EX AF,AF"	08	!LD(ADDR),IY	FD22XXXX	!LD C,(IX+Д)	DD4EXX
EX DE,HL	EB	!LD(ADDR),SP	ED73XXXX	!LD C,(IY+Д)	FD4EXX
EXX	D9	!LD (BC),A	02	!LD C,A	4F
HALT	76	!LD (DE),A	12	!LD C,B	48
IM 0	ED46	!LD (HL),A	77	!LD C,C	49
IM 1	ED56	!LD (HL),B	70	!LD C,D	4A
IM 2	ED5E	!LD (HL),C	71	!LD C,N	0EXX
IN A,(C)	ED78	!LD (HL),D	72	!LD C,E	4B
IN A,PORT	D8XX	!LD (HL),N	36XX	!LD C,H	4C
IN B,(C)	ED40	!LD (HL),E	73	!LD C,L	4D
IN C,(C)	ED48	!LD (HL),H	74	!LD D,(HL)	56
IN D,(C)	ED50	!LD (HL),L	75	!LD D,(IX+Д)	DD56XX
IN E,(C)	ED58	!LD (IX+Д),A	DD77XX	!LD D,(IY+Д)	FD56XX
IN H,(C)	ED60	!LD (IX+Д),B	DD70XX	!LD D,A	57
IN L,(C)	ED68	!LD (IX+Д),C	DD71XX	!LD D,B	50
INC (HL)	34	!LD (IX+Д),D	DD72XX	!LD D,C	51
INC (IX+Д)	DD34XX	!LD (IX+Д),N	DD36XXXX	!LD D,D	52
INC (IY+Д)	FD34XX	!LD (IX+Д),E	DD73XX	!LD D,N	16XX
INC A	3C	!LD (IX+Д),H	DD74XX	!LD D,E	53
INC B	04	!LD (IX+Д),L	DD75XX	!LD D,H	54
INC BC	03	!LD (IY+Д),A	FD77XX	!LD D,L	55
INC C	0C	!LD (IY+Д),B	FD70XX	!LD DE,(ADDR)	ED5BXXXX
INC D	14	!LD (IY+Д),C	FD71XX	!LD DE,NN	11XXXX
INC DE	13	!LD (IY+Д),D	FD72XX	!LD E,(HL)	5E
INC E	1C	!LD (IY+Д),N	FD36XXXX	!LD E,(IX+Д)	DD5EXX
INC H	24	!LD (IY+Д),E	FD73XX	!LD E,(IY+Д)	FD5EXX
INC HL	23	!LD (IY+Д),H	FD74XX	!LD E,A	5F
INC IX	DD23	!LD (IY+Д),L	FD75XX	!LD E,B	58
INC IY	FD23	!LD A,(ADDR)	3AXXXX	!LD E,C	59
INC L	2C	!LD A,(BC)	0A	!LD E,D	5A
INC SP	33	!LD A,(DE)	1A	!LD E,N	1EXX
IND	EDAA	!LD A,(HL)	7E	!LD E,E	5B
INCR	EDBA	!LD A,(IX+Д)	DD7EXX	!LD E,H	5C
INI	EDA2	!LD A,(IY+Д)	FD7EXX	!LD E,L	5D
INIR	EDB2	!LD A,A	7F	!LD H,(HL)	66
JP (HL)	E9	!LD A,B	78	!LD H,(IX+Д)	DD66XX
JP (IX)	DDE9	!LD A,C	79	!LD H,(IY+Д)	FD66XX
JP (IY)	FDE9	!LD A,D	7A	!LD H,A	67
JP ADDR	C3XXXX	!LD A,N	3EXX	!LD H,B	60
LD H,C	61	!OUTD	EDAB	!RES 4,B	CBA0
LD H,D	62	!OUTI	EDA3	!RES 4,C	CBA1
LD H,N	26XX	!POP AF	F1	!RES 4,D	CBA2
LD H,E	63	!POP BC	C1	!RES 4,E	CBA3
LD H,H	64	!POP DE	D1	!RES 4,H	CBA4
LD H,L	65	!POP HL	E1	!RES 4,L	CBA5
LDHL,(ADDR)	ED68XXXX	!POP IX	DDE1	!RES 5,(HL)	CBAE
LDHL,(ADDR)	2AXXXX	!POP IY	FDE1	!RES5,(IY+Д)	DDCBXXAE
LD HL,NN	21XXXX	!PUSN AF	F5	!RES5,(IY+Д)	FDCBXXAE
LD I,A	ED47	!PUSH BC	C5	!RES 5,A	CBAF
LDIX,(ADDR)	DD2AXXXX	!PUSH DE	D5	!RES 5,B	CBA8
LDIX,NN	DD21XXXX	!PUSH HL	E5	!RES 5,C	CBA9
LDIY,(ADDR)	FD2AXXXX	!PUSH IX	DDE5	!RES 5,D	CBAA
LD IY,NN	FD21XXXX	!PUSH IY	FDE5	!RES 5,E	CBAB
LD L,A	6F	!RES 0,(HL)	CB86	!RES 5,H	CBAC
LD L,B	68	!RES0,(IX+Д)	DDCBXX86	!RES 5,L	CBAD
LD L,C	69	!RES0,(IY+Д)	FDCBXX86	!RES 6,(HL)	CBB6

71

LD L,D	6A	!RES Ø,A	CB87	!RES6,(IX+Д)	DDCBXXB6
LD L,N	2EXX	!RES Ø,B	CB8Ø	!RES6,(IY+Д)	FDCBXXB6
LD L,E	6B	!RES Ø,C	CB81	!RES 6,A	CBB7
LD L,(HL)	6E	!RES Ø,D	CB82	!RES 6,B	CBBØ
LDL,(IX+Д)	DD6EXX	!RES Ø,E	CB83	!RES 6,C	CBB1
LDL,(IY+Д)	FD6EXX	!RES Ø,H	CB84	!RES 6,D	CBB2
LD L,H	6C	!RES Ø,L	CB85	!RES 6,E	CBB3
LD L,L	6D	!RES 1,(HL)	CB8E	!RES 6,H	CBB4
LD R,A	ED4F	!RES1,(IX+Д)	DDCBXX8E	!RES 6,L	CBB5
LDSP,(ADDR)	ED7BXXXX	!RES1,(IY+Д)	FDCBXX8E	!RES 7,(HL)	CBBE
LD SP,NN	31XXXX	!RES 1,A	CB8F	!RES7,(IX+Д)	DDCBXXBE
LD SP,HL	F9	!RES 1,B	CB88	!RES7,(IY+Д)	FDCBXXBE
LD SP,IX	DDF9	!RES 1,C	CB89	!RES 7,A	CBBF
LD SP IY	FDF9	!RES 1,D	CB8A	!RES 7,B	CBB8
LDD	EDA8	!RES 1,E	CB8B	!RES 7,C	CBB9
LDDR	ED88	!RES 1,H	CB8C	!RES 7,D	CBBA
LDI	EDAØ	!RES 1,L	CB8D	!RES 7,E	CBBB
LDIR	EDBØ	!RES 2,(HL)	CB96	!RES 7,H	CBBC
NEG	ED44	!RES2,(IX+Д)	DDCBXX96	!RES 7,L	CBBD
NOP	ØØ	!RES2,(IY+Д)	FDCBXX96	!RET	C9
OR (HL)	B6	!RES 2,A	CB97	!RET C	D8
OR (IX+Д)	DDB6XX	!RES 2,B	CB9Ø	!RET M	F8
OR (IY+Д)	FDB6XX	!RES 2,C	CB91	!RET NC	DØ
OR A	B7	!RES 2,D	CB92	!RET NZ	CØ
OR B	BØ	!RES 2,E	CB93	!RET P	FØ
OR C	B1	!RES 2,H	CB94	!RET PE	E8
OR D	B2	!RES 2,L	CB95	!RET PO	EØ
OR N	F6XX	!RES 3,(HL)	CB9E	!RET Z	C8
OR E	B3	!RES3,(IX+Д)	DDCBXX9E	!RETI	ED4D
OR H	B4	!RES3,(IY+Д)	FDCBXX9E	!RETN	ED45
OR L	B5	!RES 3,A	CB9F	!RL (HL)	CB16
OTDR	EDBB	!RES 3,B	CB98	!RL (IX+Д)	DDCBXX16
OTIR	EDB3	!RES 3,C	CB99	!RL (IY+Д)	FDCBXX16
OUT (C),A	ED79	!RES 3,D	CB9A	!RL A	CB17
OUT (C),B	ED41	!RES 3,E	CB9B	!RL B	CB1Ø
OUT (C),C	ED49	!RES 3,H	CB9C	!RL C	CB11
OUT (C),D	ED51	!RES 3,L	CB9D	!RL D	CB12
OUT (C),E	ED59	!RES 4,(HL)	CBA6	!RL E	CB13
OUT (C),H	ED61	!RES4,(IX+Д)	DDCBXXA6	!RL H	CB14
OUT (C),L	ED69	!RES4,(IY+Д)	FDCBXXA6	!RL L	CB15
OUT PORT,A	D3PORT	!RES 4,A	CBA7	!RLA	17
RLC (HL)	CBØ6	!SBC HL,BC	ED42	!SET 4,L	CBE5
RLC (IX+Д)	DDCBXXØ6	!SBC HL,DE	ED52	!SET 5,(HL)	CBEE
RLC (IY+Д)	FDCBXXØ6	!SBC HL,HL	ED62	!SET5,(IX+Д)	DDCBXXEE
RLC A	CBØ7	!SBC HL,SP	ED72	!SET5,(IY+Д)	FDCBXXEE
RLC B	CBØØ	!SCF	37	!SET 5,A	CBEF
RLC C	CBØ1	!SET Ø,(HL)	CBC6	!SET 5,B	CBE8
RLC D	CBØ2	!SETØ,(IX+Д)	DDCBXXC6	!SET 5,C	CBE9
RLC E	CBØ3	!SETØ,(IY+Д)	FDCBXXC6	!SET 5,D	CBEA
RLC H	CBØ4	!SET Ø,A	CBC7	!SET 5,E	CBEB
RLC L	CBØ5	!SET Ø,B	CBCØ	!SET 5,H	CBEC
RLCA	Ø7	!SET Ø,C	CBC1	!SET 5,L	CBED
RLD	ED6F	!SET Ø,D	CBC2	!SET 6,(HL)	CBF6
RR (HL)	CB1E	!SET Ø,E	CBC3	!SET6,(IX+Д)	DDCBXXF6
RR (IX+Д)	DDCBXX1E	!SET Ø,H	CBC4	!SET6,(IY+Д)	FDCBXXF6
RR (IY+Д)	FDCBXX1E	!SET Ø,L	CBC5	!SET 6,A	CBF7
RR A	CB1F	!SET 1,(HL)	CBCE	!SET 6,B	CBFØ
RR B	CB18	!SET1,(IX+Д)	DDCBXXCE	!SET 6,C	CBF1
RR C	CB19	!SET1,(IY+Д)	FDCBXXCE	!SET 6,D	CBF2
RR D	CB1A	!SET 1,A	CBCF	!SET 6,E	CBF3
RR E	CB1B	!SET 1,B	CBC8	!SET 6,H	CBF4
RR H	CB1C	!SET 1,C	CBC9	!SET 6,L	CBF5
RR L	CB1D	!SET 1,D	CBCA	!SET 7,(HL)	CBFE
RRA	1F	!SET 1,E	CBCB	!SET7,(IX+Д)	DDCBXXFE
RRC (HL)	CBØE	!SET 1,H	CBCC	!SET7,(IY+Д)	FDCBXXFE
RRC (IX+Д)	DDCBXXØE	!SET 1,L	CBCD	!SET 7,A	CBFF
RRC (IY+Д)	FDCBXXØE	!SET 2,(HL)	CBD6	!SET 7,B	CBF8

RRC A	CBØF	!SET2,(IX+Д)	DDCBXXD6	!SET 7,C	CBF9
RRC B	CBØ8	!SET2,(IY+Д)	FDCBXXD6	!SET 7,D	CBFA
RRC C	CB09	!SET 2,A	CBD7	!SET 7,E	CBFB
RRC D	CBØA	!SET 2,B	CBDØ	!SET 7,H	CBFC
RRC E	CBØB	!SET 2,C	CBD1	!SET 7,L	CBFD
RRC H	CBØC	!SET 2,D	CBD2	!SLA (HL)	CB26
RRC L	CBØD	!SET 2,E	CBD3	!SLA (IX+Д)	DDCBXX26
RRCA	ØF	!SET 2,H	CBD4	!SLA (IY+Д)	FDCBXX26
RRD	ED67	!SET 2,L	CBD5	!SLA A	CB27
RST ØØ	C7	!SET 3,(HL)	CBDE	!SLA B	CB2Ø
RST Ø8	CF	!SET3,(IX+Д)	DDCBXXCE	!SLA C	CB21
RST 1Ø	D7	!SET3,(IY+Д)	FDCBXXCE	!SLA D	CB22
RST 18	DF	!SET 3,A	CBDF	!SLA E	CB23
RST 2Ø	E7	!SET 3,B	CBD8	!SLA H	CB24
RST 28	EF	!SET 3,C	CBD9	!SLA L	CB25
RST 3Ø	F7	!SET 3,D	CBDA	!SRA (HL)	CB2E
RST 38	FF	!SET 3,E	CBDB	!SRA (IX+Д)	DDCBXX2E
SBC A,(HL)	9E	!SET 3,H	CBDC	!SRA (IY+Д)	FDCBXX2E
SBC A,(IX+Д)	DD9EXX	!SET 3,L	CBDD	!SRA A	CB2F
SBC A,(IY+Д)	FD9EXX	!SET 4,(HL)	CBE6	!SRA B	CB28
SBC A,A	9FXX	!SET4,(IX+Д)	DDCBXXE6	!SRA C	CB29
SBC A,B	98	!SET4,(IY+Д)	FDCBXXE6	!SRA D	CB2A
SBC A,C	99	!SET 4,A	CBE7	!SRA E	CB2B
SBC A,D	9A	!SET 4,B	CBEØ	!SRA H	CB2C
SBC A,N	DEXX	!SET 4,C	CBE1	!SRA L	CB2D
SBC A,E	9B	!SET 4,D	CBE2	!SRL (HL)	CB3E
SBC A,H	9C	!SET 4,E	CBE3	!SRL (IX+Д)	DDCBXX3E
SBC A,L	9D	!SET 4,H	CBE4	!SRL (IY+Д)	FBCBXX3E
SRL A	CB3F	!			
SRL B	CB38	!			
SRL C	CB39	!			
SRL D	CB3A	!			
SRL E	CB3B	!			
SRL H	CB3C	!			
SRL L	CB3D	!			
SUB (HL)	96	!			
SUB (IX+Д)	DD96XX	!			
SUB (IY+Д)	FD96XX	!			
SUB A	97	!			
SUB B	9Ø	!			
SUB C	91	!			
SUB D	92	!			
SUB N	D6XX	!			
SUB E	93	!			
SUB H	94	!			
SUB L	95	!			
XOR (HL)	AE	!			
XOR (IX+Д)	DDAEXX	!			
XOR (IY+Д)	FDAEXX	!			
XOR A	AF	!			
XOR B	A8	!			
XOR C	A9	!			
XOR D	AA	!			
XOR N	EEXX	!			
XOR E	AB	!			
XSOR H	AC	!			
XOR L	AD	!			

MNEMONIC - МНЕМОНИКА; HEAXDECIMAL - ШЕСТНАДЦАТЕРИЧНЫЙ КОД,
Д - ПРИРАЩЕНИЕ