МИКРОПРОЦЕССОРНЫЕ ТЕХНОЛОГИИ

ПРАКТИЧЕСКИЙ КУРС

Часть 3

Типовые программы

Пример 7. Осуществить операции со стеком. При записи в стек сначала происходит инкремент SP, а потом запись байта. При аппаратном сбросе указателю стека присваивается значение 07H. Для его переопределения используют команду `MOV SP,#d`, где d – требуемое значение указателя стека. Иными словами, стек может располагаться в любом месте РПД. Составим подпрограмму обработки прерывания. Подпрограмма должна сохранить в стеке содержимое тех регистров, которые сама использует, а перед возвратом – восстановить их значения. Пример подпрограммы обработки прерываний уровня 0. Обратить внимание на порядок записи в стек и извлечения из стека. Нарисовать размещение байтов в стеке в точке CONTR1.

```
        ORG   3             ; Вектор прерывания
Begin:  SJMP  SUBIN0        ; Переход к п/п
        ORG   30H           ; обработки прерываний
Bank1   EQU   08B           ; Банк 1
SUBIN0: PUSH  PSW           ; Сохранение в стеке
        PUSH  ACC           ;
        PUSH  B             ;
        PUSH  DPL           ;
        PUSH  DPH           ;
CONTR1: MOV   PSW,#Bank1    ; Выбор банка 1
        ...                 ; Обработка прерывания
        ...                 ;
        ...                 ;
        POP   DPH           ; Восстановление из стека
        POP   DPL           ;
        POP   B             ;
        POP   ACC           ;
        POP   PSW           ;
        RETI                ; Возврат и разрешение
        END                 ; прерываний
```

1.2.2. Команды арифметических операций

Пример 8. Сложить два двоичных многобайтных целых числа. Оба слагаемых находятся в РПД, начиная с младшего байта. Начальные адреса слагаемых заданы в R0 и R1. Формат слагаемых (количество байтов) задан в R2. Результат необходимо поместить на место первого слагаемого. Перенос между байтами учитывается при помощи команды ADDC. Перед началом цикла бит переноса сбрасывается командой CLR. При сложении беззнаковых чисел на переполнение укажет флаг C, а при сложении чисел со знаком – флаг OV.

```
        ORG   0H     ;
Begin:  CLR   C      ; Сброс бита переноса
LOOP:   MOV   A,@R0  ; Загрузка текущего
                     ; байта первого слагаемого
        ADDC  A,@R1  ; Сложение байтов с учетом C
        MOV   @R0,A  ; Сохранение результата
```

```
        INC   R0         ; Инкремент указателей
        INC   R1         ;
        DJNZ  R2,LOOP    ; Цикл
        END              ;
```

Пример 9. Осуществить умножение. Команда MUL находит произведение двух целых беззнаковых чисел, хранящихся в регистрах (B) и (A). Регистр (B) содержит старший байт. Если после умножения (B)=0, то флаг OV сбрасывается, иначе – устанавливается. Пусть требуется умножить целое двоичное число произвольного формата на константу 173D. Исходное целое число размещается в РПД, адрес младшего байта находится в регистре R0. Формат числа в байтах хранится в R1.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Пример программы умножения
; 01020304H * 173D = 0AE5C09B4H
; 173D=0ADH
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ORG   0H          ;
Begin:  MOV   60H,#04H    ; Младший байт
        MOV   61H,#03H    ;
        MOV   62H,#02H    ;
        MOV   63H,#01H    ; Старший байт
        MOV   R0,#60H     ; Адрес мл. байта числа
        MOV   R1,#4       ; Кол-во байтов в числе
MULTY:  CLR   A           ; Очистка аккумулятора
        XCH   A,@R0       ; Загрузка множимого обменом
LOOP:   MOV   B,#173D     ; Загрузка множителя
        MUL   AB          ; Умножение одного байта
        ADD   A,@R0       ; Частичная сумма
        MOV   @R0,A       ; Запись младшего байта
                          ; частичного произведения
        MOV   A,B         ; Пересылка ст.байта в (A)
        ADDC  A,#0H       ; Сложить ст.байт и бит C
        INC   R0          ; Инкремент указателя байтов
        XCH   A,@R0       ; Формирование байта произв.
                          ; и загрузка байта множимого
        DJNZ  R1,LOOP     ; Цикл
        END               ;
```

Полученное произведение размещается на месте исходного числа и занимает в РПД на один байт больше. Составить блок-схему алгоритма программы и объяснить организацию переноса между байтами при умножении.

Пример 10. Осуществить деление. Команда DIV производит деление содержимого аккумулятора на содержимое регистра B. После деления аккумулятор содержит целую часть частного, а регистр B – остаток. Флаги C и OV сбрасываются. При делении на нуль устанавливается флаг переполнения, а частное остается неопределенным. Команда деления может быть использована для быстрого преобразования двоичных чисел в двоично-десятичный код (ДДК). В качестве примера рассмотрим программу, переводящую беззнаковое целое двоичное число, находящееся в аккумуляторе, в трехразрядный ДДК. Сотни будут размещаться в регистре R0, а десятки и единицы – в аккумуляторе.

```
        ORG   0H      ;
Begin:  MOV   B,#100  ; Вычисление кол-ва сотен
        DIV   AB      ; (A) содержит кол-во сотен
        MOV   R0,A    ; Сохранение числа сотен
        XCH   A,B     ; Пересылка остатка в A
        MOV   B,#10   ; Вычисление кол-ва десятков
        DIV   AB      ; (A) содержит число десятков
                      ; регистр B -- число единиц
        SWAP  A       ; Размещение числа десятков
                      ; в старшей тетраде
        ADD   A,B     ; Подсуммирование числа единиц
        END           ;
```

1.2.3. Команды логических операций

Пример 11. Команды логических операций используют для выполнения операций над отдельными битами. Ниже приведено шесть примеров, иллюстрирующих различные битовые операции над содержимым регистров и портов.

```
        ORG  0H                ;
Begin:  ANL  P2,#10111010B     ; Сброс битов 0,2,6 P2
        ORL  P1,#00001111B     ; (P1.0-P1.3) <-- 1111
        ANL  PSW,#11100111B    ; Выбор банка 0
        XRL  P1,A              ; (P1) <-- (P1) XOR (A)
        XRL  A,#0FH            ; (A) <-- (A) XOR 0FH
        XRL  P0,#11100000B     ; (P0) <-- (P0) XOR #d
        END                    ;
```

Пример 12. Осуществить управление группой битов порта. В РПД находится массив распакованных ДДК. Требуется передать массив внешнему устройству в соответствии с протоколом: P1.4 – вывод строба «Данные готовы» (высокий уровень), P1.5 – ввод строба «Данные приняты» (высокий уровень). Выводимое ДДК (4 бита) число выводится в P1.0 – P1.3. Исходные параметры: начальный адрес массива (R0) и длина массива (R1). Неиспользуемые биты порта P1 сохранять в неизменном виде.

```
        ORG   0H               ;
Begin:  ORL   P1,#00100000B    ; Настройка P1.5 на ввод
LOOP:   ANL   P1,#11101111B    ; Сброс строба P1.4
        JB    P1.5,$           ; Ожидание ответа '0'
        MOV   A,@R0            ; Загрузка байта в (A)
        ANL   A,#00001111B     ; Маскиров. ст.тетрады
        ANL   P1,#11110000B    ; Сброс данных P1.0-P1.3
        ORL   P1,A             ; Выдача данных
        ORL   P1,#00010000B    ; Выдача строба P1.4
        JNB   P1.5,$           ; Ожидание ответа '1'
        INC   R0               ; Инкремент указателя
        DJNZ  R1,LOOP          ; Цикл
        END                    ;
```