

ПРАКТИЧЕСКИЙ КУРС

4

1.2.4. Команды операций с битами

Использование булевого процессора дает возможность осуществлять битовые операции без использования логических команд. Это позволяет строить весьма компактные и эффективно работающие программы. Рассмотрим несколько примеров на использование команд булевого процессора.

1. Программирование на языке ассемблера

Пример 13. Выдать содержимое аккумулятора в последовательном коде младшими битами вперед через нулевую линию порта 1, оставляя без изменения остальные линии порта. Время исполнения программы 41 мкс, время передачи бита – 5 мкс, скорость передачи 200 кбит/с.

```
ORG 30H ;
Begin: MOV R7, #8 ; Счетчик цикла
LOOP:  RRC A ; C <-- ACC.0
      MOV P1.0, C ; Передача бита
      DJNZ R7, LOOP ; Цикл
      END ;
```

Пример 14. Организовать последовательную передачу данных из аккумулятора на P2.0. Передачу необходимо вести в манчестерском коде (каждый бит передается двумя интервалами: первый интервал содержит инверсию бита, второй интервал – прямое значение). Передача выполняется младшими битами вперед. Длительность одного интервала равна шести машинным циклам (6 мкс), время передачи бита равно 12 мкс, время передачи байта – 96 мкс, скорость 83 кбит/с или 10.4 кбайт/с.

```
ORG 30H ;
MOV R0, #8 ; Счетчик битов
LOOP:  RRC A ; C <-- ACC.0
      CPL C ; Инверсия бита
      MOV P2.0, C ; Передача инверсного бита
      CPL C ; Восстановление прямого бита
      NOP ; Три команды NOP для выравнивания интервалов
      NOP ;
      NOP ;
      MOV P2.0, C ; Передача прямого бита
      DJNZ R0, LOOP ; Цикл
      END ;
```

Пример 15. Вычислить булеву функцию трех переменных X, Y, Z: $F(X, Y, Z) = \text{NOT}(X \& Y + X \& \text{NOT}(Z))$. Переменные X, Y и Z поступают по линиям порта P0.2, P0.1 и P0.0 соответственно. Результат $F(X, Y, Z)$ необходимо вывести на линию P0.3. Сигналы квитирования: сигнал «Данные готовы» (уровень логической единицы) поступает на P0.3 от внешнего устройства. Сигнал «Данные приняты» (уровень ло-

гической единицы) выдает МК на линию P0.4. Флаг F0 используется для временного хранения промежуточного результата.

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Тестовая программа реализации логических ф-й
;  $F(X,Y,Z)=\text{NOT}(X \& Y + X \& \text{NOT}(Z))$ .
; Разрешение -- High; подтверждение -- High
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
      ORG    0H      ;
Begin: SJMP  FUNC   ; Переход на начало программы
      ORG    30H     ;

; Настройка P0.0...P0.3 на ввод, установка P0.4 = 0
FUNC:  MOV   P0, #00001111b ;
WAIT0: JNB   P0.3, WAIT0   ; ожидание разрешения
      MOV   C, P0.2        ; C = X
      ANL   C, P0.1        ; C = X&Y
; Запись промежуточного результата в битовую память
      MOV   F0, C          ;
      MOV   C, P0.2        ; C = X
      ANL   C, /P0.0       ; C = X& /Z
; Выдача сигнала подтверждения конца ввода
      SETB  P0.4           ;
; Ожидание снятия сигнала разрешения на P0.3
WAIT1: JB    P0.3, WAIT1   ;
      CLR   P0.4           ;
      ORL   C, F0          ; C = X&Y+X& /Z
      CPL   C              ; C = /(X&Y+X&/Z)
      MOV   P0.5, C        ; выдача результата
      JMP   BEGIN         ; переход на начало
      END                   ;

```

1.2.5. Опрос двоичного датчика

Пример 16. Осуществить ожидание статического сигнала (уровень). Типовая процедура состоит из следующих действий: ввод сигнала с датчика, анализ значения сигнала и передача управления в зависимости от состояния датчика. Конструкция датчика может быть различной. На рис. 1.2, а представлен возможный вариант организации двоичного датчика. Он может быть подключен к любой линии

1. Программирование на языке ассемблера

порта. Например, датчик подключен к линии P1.3 (рис. 1.2). Процедура опроса основана на использовании команды JNB (ожидание единичного уровня) или JB (ожидание нулевого уровня).

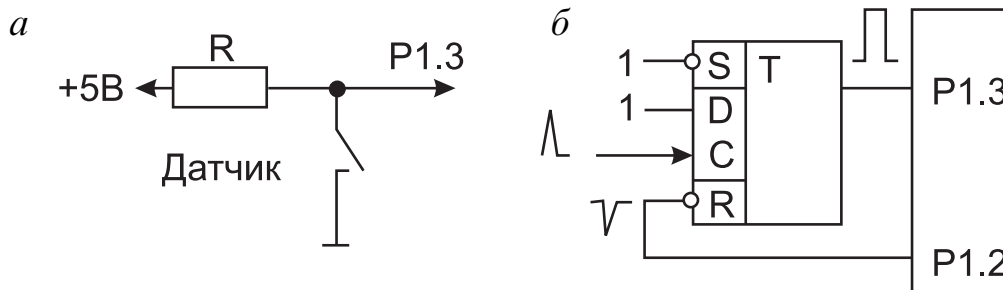


Рис. 1.2. Принципиальные схемы простого двоичного датчика (а) и двоичного датчика с флажковым триггером (б)

```
;;;;;;;;;;;;;
; Процедура ожидания единичного уровня
;;;;;;;;;;;;;
      ORG    30H      ;
Begin: SETB   P1.3    ; Програмуем P1.3 на ввод
      JNB   P1.3,$   ; Ожидание ед. уровня с датчика
      END                ;
```

```
;;;;;;;;;;;;;
; Процедура ожидания нулевого уровня
;;;;;;;;;;;;;
      ORG    30H      ;
Begin: SETB   P1.3    ; Програмуем P1.3 на ввод
      JB    P1.3,$   ; Ожидание нул. уровня с датчика
      END                ;
```

Пример 17. Осуществить ожидание импульсного сигнала (1–0–1), а не уровня. Микроконтроллер должен обнаружить не только факт появления, но и факт окончания сигнала. Процедура ожидания импульсного сигнала представляет собой последовательную комбинацию двух процедур ожидания нулевого и единичного уровней.

