МИКРОПРОЦЕССОРНЫЕ ТЕХНОЛОГИИ

ПРАКТИЧЕСКИЙ КУРС

Часть 6

Типовые программы

1.2.6. Формирование временных задержек

Пример 21. Осуществить формирование временной задержки малой длительности на основе таймера. Логика работы управляющих цепей таймера показана на рис. 1.3. Старт таймера происходит при запуске программы, а останов – в результате выполнения подпрограммы обслуживания прерывания по переполнению счетчика-таймера T/C0.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Формирование временной задержки на основе
; таймера T/C0. Рабочий диапазон: 1--65536 мкс
; В примере формируем задержку в 50 мс
; Тактовая частота F = 12 МГц.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
       ORG   0H          ;
TIME   EQU   50000       ; Величина задержки в мкс
       SJMP  Begin       ; Стартовый адрес программы
;.............................................
       ORG   0BH         ; Вектор прерываний T/C0
       CLR   TCON.4      ; Останов T/C0
       RETI              ;
;.............................................
       ORG   30H         ;
Begin: MOV   TMOD,#01H   ; Режим 1 для T/C0
       MOV   TL0,#LOW(NOT(TIME) + 1)
       MOV   TH0,#HIGH(NOT(TIME) + 1)
       SETB  TCON.4      ; Старт T/C0
       SETB  IE.1        ; Разрешение прерываний T/C0
       SETB  PCON.0      ; Режим холостого хода
NEXT:  ...               ;
       END               ;
```

Пример 22. Измерить временной интервал импульса положительной полярности, подаваемого на вход $\overline{\text{INT0}}$. Метод измерения – заполнение временного интервала импульсами с известной частотой. Число импульсов в счетчике будет пропорционально длительности временного интервала. Верхний предел измерения 65536 мкс, а максимальная погрешность 1 мкс. Логика работы управляющих цепей таймера показана на рис. 1.3. Для выбранного режима нужно установить GATE=1.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Измерение временного интервала импульса
; положительной полярности, подаваемого
; на вход INT0.
; Результат -- количество импульсов в T/C0
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ORG   0           ;
        SJMP  Begin       ;
;.............................................
        ORG   30H         ;
Begin:  MOV   TMOD,#09H   ; Настройка T/C0
                          ; Режим 1, GATE=1
        MOV   A,#0        ;
        MOV   TH0,A       ; Сброс T/C0
        MOV   TL0,A       ;
        SETB  TCON.4      ; Старт T/C0
        JNB   P3.2,$      ;
        JB    P3.2,$      ;
        CLR   TCON.4      ; Стоп T/C0
        MOV   B,TH0       ; Сохранение
        MOV   A,TL0       ; результата
        END               ;
```

1.2.7. Аналого-цифровое преобразование

Пример 23. Реализовать аппаратно-программный аналого-цифровой преобразователь (АЦП) с поразрядным взвешиванием (метод последовательных приближений, метод побитного уравновешивания). Данный метод преобразования характеризуется средним быстродействием (несколько медленнее чисто аппаратного преобразования), средней стоимостью (использует дополнительные компоненты, такие как цифро-аналоговый преобразователь (ЦАП) и компаратор). На рис. 1.4 приведена схема реализации АЦП с поразрядным взвешиванием на основе микроконтроллера x51. Алгоритм работы преобразователя заключается в следующем. Микроконтроллер готовит и выдает пробный код, по которому ЦАП формирует опорный уровень $U_{оп}$ и строб готовности к взвешиванию, подаваемый на P1.6. Компаратор сравнивает U_x и опорный уровень. Сравнение проводится для каждого бита кода по отдельности, начиная со старшего бита. Для этого в старший

бит выдается единица, если компаратор показал на выходе логический ноль ($U_{оп} < U_{х}$), то эта единица в данном разряде сохраняется, а если данный компаратор показал на выходе логическую единицу ($U_{оп} > U_{х}$), то в данном разряде записывают ноль. После этого переходят к взвешиванию следующего более младшего разряда. Этот процесс продолжают до тех пор, пока не достигнут самого младшего разряда. Сколько разрядов в коде, столько и взвешиваний необходимо произвести.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; АЦП с поразрядным взвешиванием с 8-битным ЦАП и
; компаратором в цепи обратной связи.
; R3 -- бегущая единица для взвешивания;
; R4 -- цифровой эквивалент изм.величины;
; R5 -- счетчик битов;
; ЦАП подключен к P0;
; Строб 'Готов' от ЦАП 0-1-0 на линию P1.6;
; Компаратор подключен к P1.7: 1 == U(ADC) > Ux;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
       ORG   0H         ;
       SJMP  Begin      ;
       ORG   30H        ;
Begin:                  ;
;Настройка P1.7 и P1.6 на ввод
       MOV   P1,#11000000B
;Бегущая единица для взвешивания
       MOV   R3,#1      ;
       MOV   R4,#0      ; Очистка регистра кода
       MOV   R5,#8D     ; Счетчик битов
LOOP:  MOV   A,R3       ; Формирование ед. во
       RR    A          ; взвешиваемом разряде
       MOV   R3,A       ; сохранение маски
;Пробный код = старый код + маска
       ORL   A,R4       ;
       MOV   P0,A       ; Вывод пробн. кода
       JNB   P1.6,$     ; Ожидание 1
       JB    P1.6,$     ; Ожидание 0
       JB    P1.7,OMIT  ; Взвешивание
       MOV   R4,A       ; ст. код <-- пробн.код
OMIT:  DJNZ  R5,LOOP    ;
       END              ;
```

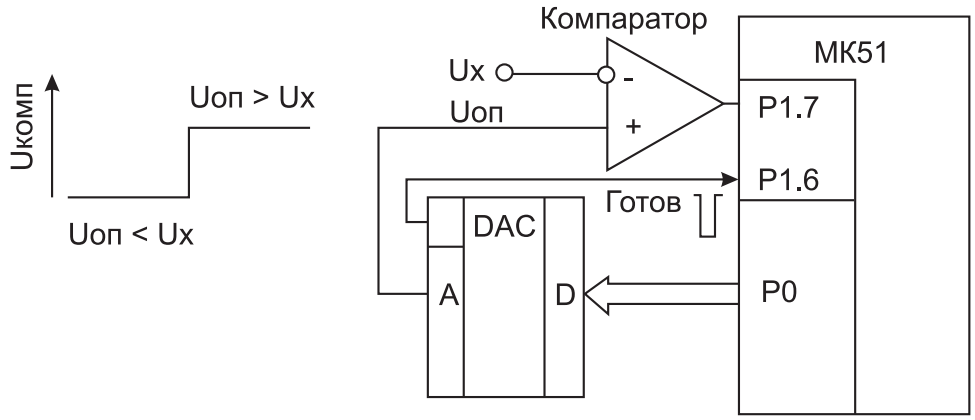

Рис. 1.4. Состояния компаратора и структурная схема АЦП

Пример 24. Программно-аппаратурный метод двойного интегрирования – это самый дешевый, но самый медленный способ преобразования, который может обеспечить очень высокую точность преобразования. Дополнительное оборудование: операционный усилитель, компаратор и аналоговый коммутатор (AMUX) на два входа. Первоначально на вход интегратора подают положительный уровень $E_{оп}$. На выходе интегратора имеем постоянный минус. Затем подаем неизвестный отрицательный уровень U_x – на выходе интегратора будет линейно-изменяющееся напряжение (ЛИН), растущее от отрицательного исходного уровня. Момент пересечения нулевого уровня ЛИНом считается t_0. В момент t_1 на вход интегратора подают снова положительное $E_{оп}$, тогда ЛИН уменьшается и в момент t_2 пересекает нулевой уровень. Временной интервал $T_1 = t_1 - t_0$ задается фиксированным и отсчитывается таймером, а интервал $T_2 = t_2 - t_0$ измеряют таймером. Измеряемое напряжение вычисляют по формуле $U_x = E_{оп} \cdot (T_2 / T_1)$.

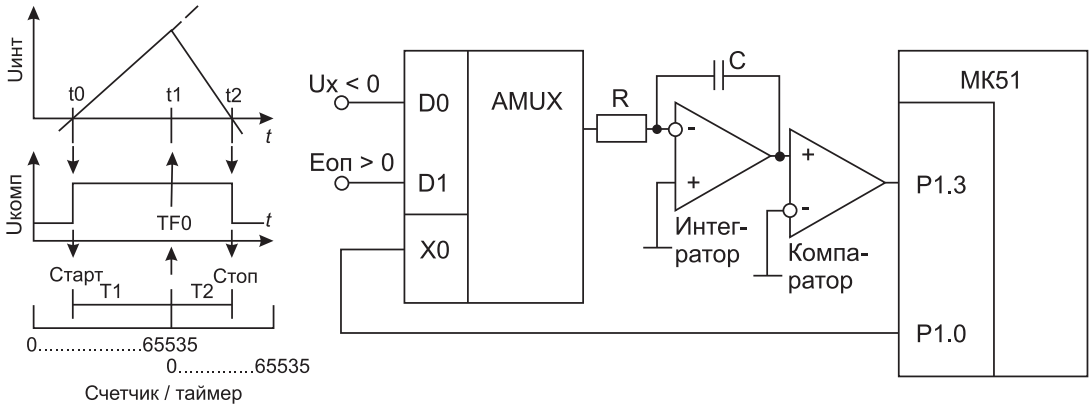

Рис. 1.5. Временная диаграмма работы и структурная схема АЦП

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; АЦП по методу двойного интегрирования
; Выход компаратора подключен к P1.3
; Управление AMUX -- к P1.0
; (1 -- выбор Eоп; 0 -- выбор Ux)
; Интервал T1 -- задает таймер T/C0
; Интервал T2 -- измеряет таймер T/C0
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ORG   0H          ;
        SJMP  Begin       ;
        ORG   30H         ;
Begin:  MOV   TMOD,#01H   ; T/C0 в режиме 1 (16 бит)
        MOV   TH0,#HIGH(NOT(T1) +1)
        MOV   TL0,#LOW(NOT(T1) +1)
        SETB  P1.1        ; P1.3 (выход комп.) на ввод
        SETB  P1.0        ; P1.0 (упр. MUX) Выбор Eоп
;Ожидание смены знака (0) компаратора
        JB    P1.3,$      ;
        CLR   P1.0        ; Выбор Ux на вх. MUX
;Ожидание смены знака (1) компаратора (момент t0)
        JNB   P1.3,$      ;
        SETB  TCON.4      ; Старт T/C0
;Ожидание переполнения T/C0 (момент t1)
        JNB   TCON.5,$    ; Ожидание момента t1
        SETB  P1.0        ; Выбор Eоп на входе MUX
;Ожидание смены знака (0) компаратора (момент t2)
        JB    P1.3,$      ;
        CLR   TCON.4      ; Стоп T/C0
        CLR   TCON.5      ; Сброс флага TF0
        MOV   B,TH0       ; Ст. байт T2
        MOV   A,TL0       ; Мл. байт T2
        END               ;
```

Программа позволяет сформировать 16-битный код, эквивалентный входному сигналу в диапазоне от 0 до -10 В. Это очень высокая точность (около 0.002 % относит.погрешн.). Максимальное время преобразования составляет 2×65.535 мс.