

## Программирование рекурсивных фильтров нижних частот на ассемблере контроллера i8051

Рассмотрим программирование цифрового фильтра, являющегося аналогом фильтра нижних частот на RC цепочке. Обозначив набор входных сигналов через  $X(N)$ , а выходных — через  $Y(N)$  (здесь  $N$  — номер отсчета по времени), запишем рекуррентное соотношение фильтра:

$$Y(N) = (1 - K) * Y(N-1) + K * X(N),$$

где коэффициент  $0 < K < 1$  должен быть выбран по желаемой постоянной времени с учетом периода обращения  $T$  к программе фильтрации. При постоянном значении входного сигнала  $X(N)$  выходной сигнал  $Y(N)$  независимо от его начального значения стремится к величине входного. Если при некотором ненулевом значении выходного сигнала постоянно подавать на вход нулевой входной сигнал, то множество значений выходного сигнала образует убывающую геометрическую прогрессию со знаменателем  $K = 1 - \exp(-T/(R*C))$ .

Для уменьшения количества умножений целесообразно привести уравнение фильтрации к виду

$$Y(N) = Y(N-1) + K * (X(N) - Y(N-1)).$$

Поскольку коэффициент фильтра положительный и не превышает единицы, то в дальнейшем предполагается, что запятая фиксируется перед 7 битом старшего байта коэффициента. При положительных сигналах и допустимости однобайтового представления сигналов и коэффициента программа фильтрации очень проста:

```

CLR C ;подготовка к вычитанию
MOV A, x ;загрузка вх. сигнала
SUBB A, y ;вычитание предыдущего вых. сигнала
MOV F0, C ;запоминание знака разности
JNC dp ;переход по положительной разности
CPL A ;вычисление обратного кода
INC A ;получение модуля разности
dp: MOV B, k ;загрузка коэффициента
MUL AB ;вычисление произведения
JNB A.7, nc; ;переход, если мл. байт < 1/2
INC v ;добавление 1 в старший байт
nc : MOV A, y ;загрузка предыдущего вых. сигнала
JNB F0, pos ; ;по положительному приращению
CLR C ;подготовка к вычитанию
SUBB A, B ;вычитание модуля приращения
SJMP str ;на запоминание вых. сигнала
pos: ADD A, B ;добавление модуля приращения
str: MOV y, A ;запоминание вых. сигнала

```

Для работы программы используется одна ячейка ОЗУ, в которой между обращениями к программе должно сохраняться значение выходного сигнала фильтра. В соответствии с масштабом коэффициента для вычисления выходного сигнала используется старший байт произведения. Для уменьшения погрешностей вычисления старший байт увеличивается на 1 в том случае, если старший разряд младшего байта произведения равен 1 (округление).

Фильтр нижних частот первого порядка имеет крутизну частотной характеристики вне полосы пропускания 3 дБ на октаву. Приведем пример программирования фильтра нижних частот второго порядка, имеющего два инерционных элемента и крутизну 6 дБ на октаву. Рекуррентное соотношение для выходного сигнала содержит два коэффициента и два

предыдущих отсчета сигнала. Для упрощения вычислений целесообразно записать это соотношение так, чтобы кроме предыдущего выходного сигнала запоминать и предыдущее его приращение:

$$Y(N) = Y(N-1) + K1 * (X(N) - Y(N-1)) * + K2 * (Y(N-1) - Y(N-2)).$$

Если выбрать граничную частоту полосы пропускания этого фильтра по уровню 3 дБ, соответствующую тому же значению произведения RC, что и в предыдущем случае, то выбор коэффициентов задается следующими формулами:

$$K1 = 1 - 2 * \exp(-0,7 * T / RC) * \cos(0,7 * T / RC) + \exp(-1,4 * T / RC) \quad K2 = \exp(-1,4 * T / RC)$$

Предполагая, что сигналы положительные однобайтовые и приращение выходного сигнала не превышает 127, можно хранить приращение в виде числа со знаком в ячейке dy.

```

MOV  A,  dy      ;предыдущее приращения вых. сигнала
CLR  FO         ;
JNB  A.7,  dyp  ;по положительному приращению
SETB FO        ;запоминание знака приращения
CPL  A         ;
INC  A         ;вычисление модуля
dyp: MOV  B,  k2  ;загрузка второго коэффициента
MUL  AB        ;вычисление произведения
JNB  A.7,  nc2  ;переход, если мл. байт < 1/2
INC  B         ;добавление 1 в старший байт
nc2: MOV  A,  B   ;запоминание модуля произведения
JNB  FO,  pp   ;по положительному приращению
CPL  A         ;вычисление дополнительного кода
INC  A         ;
pp:  MOV  dy,  A  ;временное запоминание произведения
CLR  C         ;подготовка к вычитанию
MOV  A,  x     ;загрузка вх. сигнала
SUBB A,  y     ;вычитание предыдущего вых. ;сигнала
MOV  FO,  C    ;запоминание знака разности
JNC  dp       ;переход по положительной ;разности
CPL  A         ;
INC  A         ;вычисление модуля разности
dp:  MOV  B,  k1  ;загрузка первого коэффициента
MUL  AB        ;вычисление произведения
JNB  A.7,  nc1 ;переход, если мл. байт < 1/2
INC  B         ;добавление 1 в старший байт
nc1 : MOV  A,  B  ;для вычисления приращения
JNB  FO,  pos  ;по положительному приращению
CPL  A         ;вычисление дополнительного кода
INC  A         ;
pos: ADD  A,  dy  ;вычисление приращения
MOV  dy,  A    ;запоминание приращения
ADD  A,  y     ;добавление предыдущего вых. сигнала
MOV  y,  A     ;запоминание вых. сигнала

```

## Литература

Каспер Э. Программирование на языке Ассемблера для микроконтроллеров семейства i8051. – М.: Горячая линия–Телеком, 2003. – 191 с.