УДК 621.391.260

В. Г. РУДЕНКО, В. В. ТРЕГУБ
(Харьков)

БЫСТРОЕ ФОРМИРОВАНИЕ ЗНАКОВОЙ КОРРЕЛЯЦИОННОЙ ФУНКЦИИ НА ВЫЧИСЛИТЕЛЬНЫХ СРЕДСТВАХ С МНОГОРАЗРЯДНЫМ АРИФМЕТИКО-ЛОГИЧЕСКИМ УСТРОЙСТВОМ

Знаковая корреляционная функция (ЗКФ) как характеристика стохастической связи случайных процессов получила в последние годы широкое распространение в измерительной технике благодаря простоте аппаратурной реализации цифровых знаковых корреляторов. Однако развитие современной микросхемотехники в направлении повышения быстродействия и увеличения разрядности микропроцессорных вычислительных средств позволяет в ряде случаев заменить аппаратные средства вычисления ЗКФ программными, реализуемыми даже на микропроцессорах с ограниченным набором команд. Такая замена целесообразна для расширения возможностей многофункциональной измерительной аппаратуры типа «разумных» осциллографов и им подобной. Ее эффективность определяется, помимо уменьшения аппаратурных затрат, быстродействием вычислительного алгоритма.

В настоящей статье приводится оценка производительности известных алгоритмов последовательного и параллельного вычисления ЗКФ, анализируется причина низкой эффективности использования многоразрядного арифметико-логического устройства (АЛУ) при их реализации и предлагается алгоритм, быстродействие которого пропорционально разрядности АЛУ.

Знаковые корреляторы при определении оценок корреляционных функций $R_{XY}(kT)$ эргодичных стационарных центрированных случайных процессов $X(t)$, $Y(t)$ в виде [1]

$$R_{XY}(kT) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \operatorname{sign} x(iT) \cdot \operatorname{sign} y(iT + kT), \tag{1}$$

где $k = 0, 1, \ldots, M-1$; T — период квантования, базируются на следующем задании значений знаковой функции:

$$\operatorname{sign} x = \begin{cases} 1, & x \geqslant 0; \\ -1, & x < 0. \end{cases} \tag{2}$$

При вычислениях на ЭВМ в качестве значения знаковой функции целесообразно использовать уже имеющееся в представлении числа содержимое его знакового разряда:

$$\operatorname{sgn} x = \begin{cases} 0, & x \geqslant 0; \\ 1, & x < 0. \end{cases} \tag{3}$$

Тогда оценка ЗКФ $\widehat{R}_{XY}(kT)$ может быть получена как функция несовпадения знаков (НЗ) коррелируемых выборок:

$$\widehat{R}_{XY}(kT) = \frac{2}{N}\left[\frac{N}{2} - \sum_{i=1}^{N} \operatorname{sgn} x(iT) \oplus \operatorname{sgn} y(iT + kT)\right], \tag{4}$$

где знак $\oplus$ обозначает операцию сложения по $\bmod 2$, являющуюся наряду с операцией арифметического сложения типовой операцией, выполняемой АЛУ.

В современных экспериментальных исследованиях значение N обычно велико по сравнению с M:

$$2^{10} \leqslant N \leqslant 2^{15}, \quad 2^3 \leqslant M \leqslant 2^7 \tag{5}$$

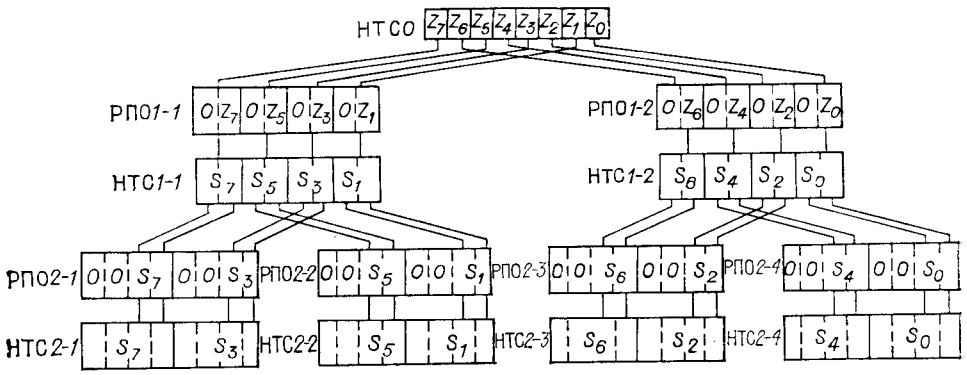

Рис. 1

— и основной объем команд при вычислении (4) расходуется на формирование суммы несовпадений знаков (СНЗ). Условимся считать затраты на формирование оценки M ординат ЗКФ в количестве команд арифметического сложения и сложения по mod 2. Будем считать также, что M равно разрядности АЛУ.

Последовательное вычисление СНЗ в (4) требует одной операции сложения по mod 2 и одной операции арифметического сложения на каждую пару коррелируемых выборок. Очевидно, что затраты последовательного алгоритма вычисления всех СНЗ в (4) составляют

$$K_{\text{п}} = 2MN \tag{6}$$

команд сложения.

Поскольку результат сложения по mod 2 имеет разрядность 1, то одной командой M-разрядного АЛУ можно сформировать M результатов НЗ — по одному биту для каждой ординаты ЗКФ. Количество операций арифметического сложения останется, однако, прежним, а общие затраты параллельного алгоритма вычисления M сумм НЗ уменьшатся до величины

$$K_{\text{пр}} = (M+1)N \tag{7}$$

команд сложения.

Общим недостатком упомянутых алгоритмов является крайне низкая эффективность использования M-разрядного АЛУ при накоплении СНЗ вследствие прибавления к текущей M-разрядной сумме, содержащей максимум $\log_2 N$ значащих бит, только одного бита НЗ.

В то же время M-разрядный арифметический сумматор можно представить в виде $G = M/m$ m-разрядных независимых сумматоров (секций) при устранении переноса между их примыкающими разрядами. Такая независимость обеспечивается алгоритмически ограничением количества накапливаемых операндов одинаковой разрядности p величиной

$$n = (2^m - 1)/(2^p - 1). \tag{8}$$

В дальнейшем полагается

$$m = 2p,\ p = 2^{v-1}, \tag{9}$$

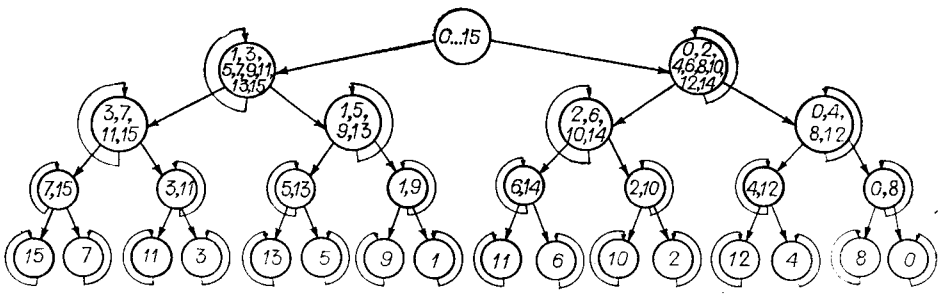

Рис. 2

```
/ *   АЛГОРИТМ ВЫЧИСЛЕНИЯ 16 ОРДИНАТ ЗНАКОВОЙ КОРРЕЛЯЦИОННОЙ ФУНКЦИИ  * /
/ *          НА ЭВМ С 16-РАЗРЯДНЫМ АРИФМЕТИКО-ЛОГИЧЕСКИМ УСТРОЙСТВОМ,  * /
/ *          МОДЕЛИРУЕМЫМ СРЕДСТВАМИ ЯЗЫКА ПЛ/1 ЕС ЭВМ.                * /

SUMMA: PROCEDURE (U,J_OUT) RECURSIVE;
        DECLARE                                  / *    П А Р А М Е Т Р Ы:                    * /
          (U,                                    / *  ТЕКУЩИЙ УРОВЕНЬ ГРАФА АЛГОРИТМА СУММИРОВ.  * /
          J_OUT) BIN FIXED;                      / *  ИНДЕКС НАЧАЛА ПОДМАССИВА НТС В МАССИВЕ S    * /
        DECLARE                                  / *  Г Л О Б А Л Ь Н Ы Е  МАССИВЫ И СКАЛЯРЫ:    * /
          ((X, Y)   (*) DEC FLOAT,               / *  ВЫБОРКИ КОРРЕЛИРУЕМЫХ ПРОЦЕССОВ            * /
          S       (31) BIN FIXED,                / *  НАКОПИТЕЛИ ТЕКУЩИХ СУММ (НТС) ДЛЯ ВСЕХ U    * /
          MASK   (4) BIT     (31),               / *  МАСКИ ДЛЯ ПРОРЕЖИВАНИЯ ОПЕРАНДОВ И          * /
          NSUM   (4) BIN FIXED,                  / *  ПРЕДЕЛЫ СУММИРОВАНИЯ НА КАЖДОМ ИЗ УРОВНЕЙ   * /
          Z          BIT     (31),               / *  ПРАВЫЕ 16 БИТ — ЗНАКИ ВЫБОРОК ПРОЦЕССА Y     * /
          ZERO       BIT     (15),               / *  =(15)'0'B — ТОЛЬКО ДЛЯ МОДЕЛИРОВАНИЯ!       * /
          I          BIN FIXED                   / *  ИНДЕКС КОРРЕЛИРУЕМЫХ ВЫБОРОК                 * /
          )          EXTERNAL;
        DECLARE                                  / *  Л О К А Л Ь Н Ы Е  ПЕРЕМЕННЫЕ — ДЛЯ РЕКУРСИИ  * /
          (N,                                    / *  СЧЕТЧИК СУММИРУЕМЫХ ОПЕРАНДОВ               * /
          J_IN,                                  / *  ИНД. НАЧАЛА ПОДМАССИВА ПРОРЕЖИВ. ОПЕРАНДОВ  * /
          J_END,                                 / *  ИНДЕКС КОНЦА ПОДМАССИВА НТС В МАССИВЕ S      * /
          W_IN,                                    *  КОЛИЧЕСТВО ПРОРЕЖИВАЕМЫХ ОПЕРАНДОВ           * /
          W_OUT,                                 / *  КОЛИЧЕСТВО ОБСЛУЖИВАЕМЫХ НТС                 * /
          J       )      BIN FIXED,              / *  СЧЕТЧИК ОБСЛУЖИВАЕМЫХ НТС                    * /
         (R1, R2)        BIT     (31);           / *  РЕГИСТРЫ ПРОРЕЖЕННЫХ ОПЕРАНДОВ               * /
        W_IN = 2 **(U — 1);                      / *  КОЛИЧ. ПРОРЕЖИВАЕМЫХ ОПЕРАНДОВ НА УРОВНЕ U   * /
        W. OUT = 2 *W_IN;                        / *  КОЛИЧ. НАКОПИТЕЛЕЙ ТЕКУЩИХ СУММ НА УРОВНЕ U  * /
        J_IN = J_OUT + W_OUT;                    / *  НАЧАЛО U-ГО ПОДМАССИВА ПРОРЕЖИВ. ОПЕРАНДОВ   * /
        DO J = J_OUT TO J_IN — 1; S(J)= 0; END;          / *    ОЧИСТКА ПОДМАССИВА НТС             * /
        DO N = 1 TO NSUM(U);                     / *  ЦИКЛ НАКОПЛЕНИЯ ПРОРЕЖЕННЫХ ОПЕРАНДОВ:       * /
          IF U > 1                               / *  ДОСТИГНУТ УРОВЕНЬ 1?                         * /
           THEN CALL SUMMA (U — 1, J_IN);                / *  НЕТ, ПРОДОЛЖИТЬ РЕКУРСИЮ.            * /
           ELSE DO;                                      / *  ДА, ГРУППА ОПЕРАЦИЙ 1-ГО УРОВНЯ:     * /
```

```
SUBSTR (Z, 1, 1)='0'B;                    /*  МОДЕЛЬ ЛОГИЧЕСКОГО СДВИГА ВЛЕВО СОДЕР-  */
Z = Z + Z                                 /*  ЖИМОГО 16-РАЗРЯДНОГО РЕГИСТРА ЗНАКОВ     */
  +(SIGN(Y(1))= - 1B);                    /*  И ЗАНЕСЕНИЕ В МЛАДШИЙ РАЗРЯД SGN(Y(1)).  */
IF X(I)  <0                               /*  СЛОЖЕНИЕ ПО MOD2 ТРИВИАЛЬНО. ЕГО РЕЗУЛЬТАТ:  */
   THEN UNSPEC(S(J_IN))= SUBSTR(Z, 16);   /*  ЭТО 16 ЗНАКОВ                            */
   ELSE UNSPEC(S(J_IN))=¬SUBSTR(Z, 16);   /*  ИЛИ ИХ ИНВЕРСИЯ                          */
I = I + 1;                                /*  ИНКРЕМЕНТ ИНДЕКСА КОРРЕЛИРУЕМЫХ ВЫБОРОК  */
END;                                      /*  КОНЕЦ ГРУППЫ ОПЕРАЦИЙ ПЕРВОГО УРОВНЯ.    */
DO J = 0 TO W_IN - 1;                     /*  ЦИКЛ ПРОРЕЖИВАНИЯ И НАКОПЛЕНИЯ:          */
   R1 = ZERO!! UNSPEC(S(J_IN + J));       /*  ЗАГРУЗКА ПРОРЕЖИВАЕМЫХ ОПЕРАНДОВ         */
   R2 = R1/2**(2**(U - 1));               /*  И ВЫРАВНИВАНИЕ НА ПРАВУЮ ГРАНИЦУ          */
   R1 = R1 & MASK(U); R2 = R2 & MASK(U);  /*  ПРОРЕЖИВАНИЕ ОПЕРАНДОВ                   */
   S(J_OUT + 2*J)=(ZERO!! UNSPEC(S(J_OUT + 2*J))) + R1;   /*  И ИХ НА-               */
   S(J_OUT + 2*J + 1)=(ZERO!! UNSPEC(S(J_OUT + 2*J + 1)))+ R2;   /*  КОПЛЕНИЕ       */
END;                                      /*  ЦИКЛА СУММИРОВАНИЯ ПРОРЕЖЕННЫХ ОПЕРАНДОВ  */
END;                                      /*  ЦИКЛА ФОРМИРОВАНИЯ НТС НА УРОВНЕ U        */
END SUMMA;
```

Рис. 3

гательного регистра, соответствующих младшим разрядам m-разрядных сумматоров, и очисткой остальных разрядов регистра. Назовем такое размещение накапливаемых операндов «прореживанием», а вспомогательный регистр — регистром прореженных операндов (РПО). Во избежание потери информации операнды, находящиеся в очищаемых разрядах РПО, предварительно копируются во второй РПО и после выравнивания их по правой границе накапливаемых m-разрядных сумм также прореживаются. Затем содержимое каждого РПО арифметически суммируется с содержимым соответствующих накопителей текущих сумм (НТС). Распределение разрядов 8-разрядных РПО и НТС на уровнях 1 и 2 показано на рис. 1, где НТС0 содержит однобитовые суммы по mod 2 знаков коррелируемых выборок, а стрелки указывают порядок прореживания и совмещения битов операндов. На рис. 2 приведен ориентированный четырехуровневый граф секционированного формирования СНЗ 16-разрядным АЛУ. Каждая вершина графа обозначает операцию арифметического сложения прореженного операнда, переносимого стрелкой, с предыдущим (возможно, нулевым) значением текущей суммы НЗ, переносимым дугой. Числа в вершинах графа обозначают номера формируемых ординат ВКФ. Прореживание и суммирование операндов на уровне U производятся после завершения формирования текущей суммы на уровне $U-1$, что определяет общее количество НТС величиной $Q = 1 + \sum_{U=1}^{4} 2^U$.

Из (8), (9) следует, что на уровнях 1—3 выполняется

соответственно по 3, 5, 17 операций сложения для каждого из 2^v НТС. Количество операций суммирования на уровне 4 ограничивается не столько разрядностью секции суммирования ($m_4 = 16$), сколько числом коррелируемых выборок $N < 2^{16}$. Допустим, что $N = 255n_4$. Тогда формирование СНЗ может быть выполнено по рекурсивному алгоритму (рис. 3).

В головной программе резервируется память для накопителей текущих сумм — массив S (1 : 31) 16-разрядных слов, для выборок процессов — массивы X (1 : $255n_4$), Y (1 : $255n_4$). Резервируется память и инициализируется содержимое массива масок для прореживания операндов MASK (1 : 4) и массива количества операций суммирования на каждом уровне NSUM (1 : 4).

После формирования массивов X и Y задается начальное значение индекса $I = 1$ и вызывается рекурсивная программа SUMMA (4, 1). В результате первые 16 слов массива S будут содержать СНЗ для вычисления ординат ЗКФ.

Из рис. 2 видно, что индексы СНЗ в массиве S не соответствуют порядковым номерам ординат ЗКФ. Техника упорядочения элементов массива S в порядке возрастания ординат ЗКФ хорошо известна из практики реализации алгоритмов быстрого преобразования Фурье [2]. Можно также воспользоваться табличным заданием индексов r ординат ВКФ в массиве S, вычисляемых однократно по рекуррентным формулам [3]:

$$r_1 = 1, \quad z = 1;$$
$$r_j = r_{j-v} + 2^{w-z}, \quad z = 2, 3, \ldots, w, \quad w = \log_2 M;$$
$$v = 2^{z-2}, \quad j = v + 1, \ldots, 2v,$$

и применяемых для упорядочения элементов массива S по правилу

$$\text{если } 2j - 1 < r_j, \text{ то } S(2j-1) \rightleftarrows S(r_j);$$
$$\text{если } 2j < r_j + M/2, \text{ то } S(2j) \rightleftarrows S(r_j + M/2),$$
$$j = 1, 2, \ldots, M/2,$$

где знак $\rightleftarrows$ указывает на обмен содержимого двух ячеек.

Общее количество операций сложения, затрачиваемых секционированным алгоритмом на формирование M сумм несовпадений знаков, равно

$$K_c = N\left(1 + \sum_{U=1}^{4} \frac{2^U}{2^{2^U - 1}}\right) = N\left(1 + \frac{2}{1} + \frac{4}{3} + \frac{8}{15} + \frac{16}{255}\right) \approx 5N. \qquad (10)$$

Этот результат показывает, что значение K_c не зависит от числа параллельно накапливаемых сумм в пределах разрядности АЛУ, следовательно, затраты команд на вычисление одной ординаты ЗКФ уменьшаются с ростом M. Сравнивая (6) и (7) с (10) для $M = 32$, получаем, что секционированный алгоритм экономичнее последовательного примерно в 13 раз и параллельного — в 6,6 раза.

Однако вследствие примерного равенства времен выполнения команд сложения и вспомогательных команд, реализующих алгоритмы накопления СНЗ (обмен с оперативной памятью, сдвиг, сравнение, ветвление и т. п.), реальные объемы выполняемых команд отличаются от приведенных выше. Для их определения каждый из алгоритмов был запрограммирован на языке Ассемблера ЕС ЭВМ с использованием команд формата «регистр — регистр» и «регистр — индексируемая память». Наиболее экономичные варианты программ (в частности, нерекурсивная программа секционированного вычисления СНЗ) дали следующие оценки командных затрат:

$$K'_{\text{п}} \approx 7{,}5\ MN;$$
$$K'_{\text{пр}} \approx 7{,}5\,MN + 10{,}5N;$$
$$K'_{\text{с}} \approx 23{,}6N.$$

Отсутствие реальной экономии команд при реализации параллельного алгоритма объясняется неэффективностью поразрядной распаковки M-разрядного результата суммирования знаков по модулю 2. Секционированный алгоритм при $M = 32$ экономичнее последовательного более чем в 10 раз.

Таким образом, рациональный выбор вида знаковой функции, параллельное формирование как битов несовпадения знаков коррелируемых процессов, так и сумм переменной разрядности позволили реализовать алгоритм формирования знаковой корреляционной функции, производительность которого пропорциональна разрядности используемого арифметико-логического устройства. Для 32-разрядного АЛУ достигнуто десятикратное повышение скорости вычислений по сравнению с последовательным алгоритмом вычисления ЗКФ. Кроме того, разработанный алгоритм параллельного накопления независимых сумм однобитовых операндов может быть использован в многочисленных приложениях, связанных с параллельной обработкой импульсных потоков.

ЛИТЕРАТУРА

1. **Мирский Г. Я.** Характеристики стохастической взаимосвязи и их измерения.— М.: Энергоиздат, 1982.
2. **Рабинер Л., Гоулд Б.** Теория и применение цифровой обработки сигналов.— М.: Мир, 1978.
3. **Руденко В. Г.** Реализация алгоритмов БПФ и БПУ без двоично-инверсного счетчика // АСУ и приборы автоматики. Харьков: Вища школа, 1982, вып. 63.

Поступила в редакцию 11 ноября 1985 г.

УДК 621.391.1

Б. Ф. БЕЗРОДНЫЙ, А. В. САВИЧ, Я. А. ФОМИН

(Москва)

ОПТИМИЗАЦИЯ СИСТЕМЫ РАСПОЗНАВАНИЯ ОДНОМЕРНЫХ НОРМАЛЬНЫХ СОВОКУПНОСТЕЙ

Задача распознавания двух одномерных нормальных совокупностей может быть сформулирована как задача определения принадлежности ансамбля независимых случайных величин $x^n = (x_1, \ldots, x_n)$, называемого контрольной выборкой, к одному из двух нормальных законов $N_1 = N(a_1, \sigma_1^2)$ или $N_2 = N(a_2, \sigma_2^2)$.

В общем случае, когда средние a_1, a_2 и дисперсии σ_1^2, σ_2^2 неизвестны, правило распознавания записывается в виде [1]; если справедливо

$$V = \sum_{i=1}^{n} \left[\frac{1}{\widehat{\sigma}_1^2} (x_i - \widehat{a}_1)^2 - \frac{1}{\widehat{\sigma}_2^2} (x_i - \widehat{a}_2)^2 \right] + n \ln \frac{\widehat{\sigma}_1^2}{\widehat{\sigma}_2^2} \geqslant 2k, \qquad (1)$$

принимается решение $x^n \in N_2$, в противном случае — решение $x^n \in N_1$. Здесь

$$\widehat{a}_i = \frac{1}{m_i} \sum_{j=1}^{m_i} x_j^{(i)}; \quad \widehat{\sigma}_i^2 = \frac{1}{m_i - 1} \sum_{j=1}^{m_i} (x_j^{(i)} - \widehat{a}_i)^2, \quad i = 1, 2,$$

— МП-оценки: соответствующих неизвестных параметров [2], полученные по обучающим выборкам $(x_1^{(i)}, \ldots, x_{m_i}^{(i)}) \in N_i$.