Особенности разработки БИХ-фильтра для системы на кристалле

Д. Н. Борисов, e-mail: borisov@sc.vsu.ru
В. М. Солодухин, e-mail: solodyxin.1997@gmail.com

Воронежский государственный университет

Аннотация. *В работе рассматриваются особенности проектирования БИХ-фильтра на системе-на-кристалле 1892ВМ14Я НПЦ ЭЛВИС. Произведена оценка производительности процессора MCom-02 в тесте при реализации ФНЧ фильтра Баттерворта 3-го порядка.*

Ключевые слова: *RISC процессор, DSP-ядра, БИХ-фильтр.*

Введение

Научно производственный центр электронные вычислительно-информационные системы (НПЦ «Элвис») является одним из немногих на Российском рынке, осуществляющий разработку и изготовление микропроцессорной техники, ряду изделий которого присвоен статус отечественных микросхем первого и второго уровня, что в условиях импортозамещения позволяет использовать их в оборонной промышленности РФ.

В работе рассматривается система-на-кристалле (или сокращенно СнК) на базе собственной платформы проектирования НПЦ «Элвис» – «Мультикор». Процессоры серии «Мультикор» – это однокристальные программируемые многопроцессорные СнК, созданные на базе библиотеки IP [1] – ядер платформы Мультикор.

Процессоры данной серии позволяют сочетать в себе качества двух классов приборов: цифровых процессоров обработки сигналов и микроконтроллеров, что позволяет одновременно решать несколько задач: контроля и высокоточной обработки информации.

В данной работе рассматривается особенности работы с DSP кластером семейства «Мультикор» – процессором 1892ВМ14Я [2] («Мультиком-02», MCom-02), особенности разработки ФНЧ фильтра Баттерворта 3-го порядка с данным СнК.

1. Система на кристалле и ее особенности

Микропроцессор «Мультиком-02» (MCom-02, 1892ВМ14Я) является 6-ядерным сигнальным микропроцессором с пониженным

энергопотреблением созданным для связных, мультимедийных, навигационных, а также встраиваемых мобильных приложений.

Микросхема изготовлена по КМОП-технологии с минимальными топологическими размерами элементов равным 40 нм.

Микроконтроллер обладает несколькими основными модулями:

– стандартным управляющим процессором представленный сдвоенным процессорным ядром – Dual CORTEX-A9 (CPU 0-1) с FPU-акселератором и NEON SIMD-акселератором (ARM);

– DSP-кластером [3] на основе двух DSP-ядер ELcore-30M [4] с полной программной совместимостью с микросхемами 1892ВМ10Я и 1892ВМ15АФ.

Кластер DSP представляет собой двухпроцессорную MIMD (multiple instruction, multiple data) систему. Каждое DSP ядро имеет в наличии собственную программную память и может работать независимо от остальных ядер. На верхнем уровне DSP-кластера имеется набор общих для всего кластера регистров управления и состояния.

Для синхронизации работы DSP ядер в кластере имеется в наличии два механизма: механизм прерываний и механизм обменов через XBUF в синхронном режиме. Каждое DSP ядро может формировать прерывание для любого другого ядра в кластере, используя соответствующие регистры. Ядро, получившее прерывание, переходит в состояние «выполнения» (RUN), причем, если оно было остановлено, то исполнение подпрограммы начинается с адреса, который хранится в специальном регистре этого ядра [5].

Для оперативных обменов данными между CPU, DSP0 – DSP1 в составе DELcore-30M имеется буфер обмена XBUF, состоящий из 32-х 64-разрядных регистров X0-X31, доступных для записи и чтения для всех процессорных ядер. Буфер обмена XBUF представляет собой многопортовую память и позволяет одновременное чтение одной и той же ячейки со стороны более одного абонента – CPU, DSP0 – DSP1. При одновременном запросе на запись в одну и ту же ячейку приоритет в первую очередь отдается CPU, затем – DSP0, и в последнюю очередь – DSP1. Обменный буфер может работать как в обычном режиме, так и в синхронном режиме, причем в обычном режиме при обмене данными через буфер блокировка чтение/запись не задействуется.

В синхронном режиме для конкретного регистра XBUF [5] обязательно должны чередоваться операции чтения и записи. Если какое-либо ядро пытается осуществить запись, то после выполнения операции чтения/записи – оно блокируется. Обмен через XBUF в синхронном режиме является дополнительным программным способом

синхронизации ядер DSP. Программная память и память данных кластера DSP физически организована как двухпортовая. По одному порту производятся внешние обращения от RISC ядра и контроллеров DMA, по другому порту производятся обращения от ядер DSP. Такая организация позволяет производить бесконфликтный фоновый обмен данными между памятью кластера DSP и внешними устройствами.

2. Проектирование фильтра

Проектирование ФНЧ фильтра Баттерворта 3-го порядка и генерация коэффициентов проводилась с использованием средства Filter Design and Analysis Toolbox (проектирования и анализа фильтров) MATLAB. Использовалась частота дискретизации 8400 Гц и частота среза 50 Гц. На рис. 1 представлены АЧХ и ФЧХ разрабатываемого фильтра.

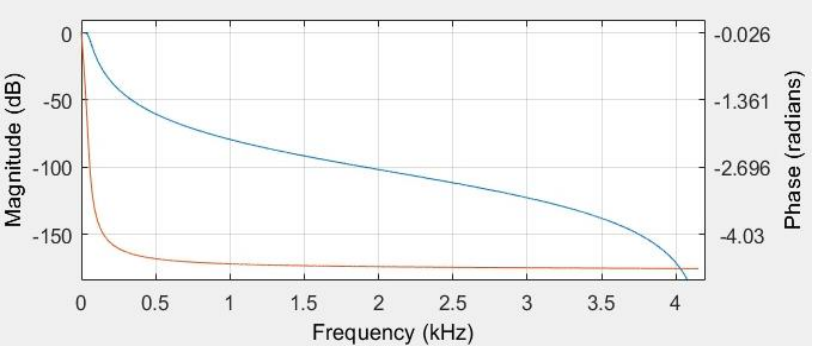

Рис. 1. Графики АЧХ (синий) и ФЧХ (оранжевый) фильтра

Рассчитанные коэффициенты фильтра представлены на рис. 2.

```
Num:
0.000006301299621181513574950678463482134
0.000018903898863544541571884982644746742
0.000018903898863544541571884982644746742
0.000006301299621181513574950678463482134
Den:
1
-2.925204528808932114003482638509012758732
2.853180108439186568602963234297931194305
-0.927925169233285163095104053354589268565
```

Рис. 2. Коэффициенты ФНЧ фильтра Баттерворта 3-го порядка

3. Программная реализация фильтра

Программная реализация БИХ-фильтра осуществлялась для СнК «Мультиком-02», которая реализуется в виде набора отладочного модуля Салют-ЭЛ24Д1 и процессорного модуля Салют-ЭЛ24ОМ1 (НПЦ «ЭЛВИС»). Для программирования используется проприетарная среда (IDE) MCStudio4, построенная на основе свободной интегрированной среды разработки Eclipse, совместно с эмулятором MC-USB-JTAG, использующий JTAG-порт СнК.

Особенностью построения программ для СнК заключается в необходимости программирования DSP-ядер, код для которого пишется на ассемблере, и программирования RISC процессора, код для которого код пишется на языке С. Данная особенность построения кода связана с архитектурой СнК: CPU является управляющим звеном, но для того чтобы использовать DSP-ядро (ядра) необходимо из программы написанной на языке С обращаться к соответствующим регистрам DSP-ядра, используя функции прерывания, останавливающие (запускающие) работу ядер.

При реализации БИХ-фильтров для системы на кристалле «Мультиком-02» используются регистр DCSR и регистр SR [6], которые отвечают за состояние ядер (в регистре SR первые 7 разрядов регистра доступны для чтения, остальные для записи).

После инициализации базовых регистров, используются коэффициенты фильтра, полученные на стадии проектирования в Matlab. Коэффициенты задаются в виде массива чисел с плавающей точкой, причем количество элементов массивов, формируется таким образом, чтобы он был кратен 2, для корректного выделения памяти.

Для эффективного взаимодействия с DSP ядром используется либо регистр памяти XBUFF, либо переменные, объявленные в файле ассемблера.

DSP процессоры всегда отличались своей высокой скоростью работы и особыми расширенными наборами команд для достижения максимальной производительности. DSP модуль СнК MCom-02 предоставляет возможность помимо использования специфических команд для вычислительных операций, также использовать параллельные команды, позволяющие выполнять более двух операций за одну инструкцию: две вычислительные операции и две операции пересылки. Но при этом для выполнения параллельных команд накладывается ограничение: его использование возможно только для работы на разных устройствах. В результате нельзя выполнить две операции сложения за одну инструкцию, так как они выполняются на одном арифметико-логическом устройстве. Поэтому при выполнении

операции свертки (как наиболее востребованной при реализации фильтрации) используется одновременное выполнение операции сложения и умножения с использованием пересылки в адресный регистр.

Чтобы оценить эффективность (по времени) реализации ФНЧ фильтра Баттерворта 3-го порядка по окончании работы фильтра подсчитаем количество тактов работы DSP-ядра.

В листинге 1 представлена программа расчета времени вычисления коэффициентов ФНЧ фильтра Баттерворта 3-го порядка в пакете Matlab.

Листинг 1

Расчет времени вычисления коэффициентов фильтра в Matlab

```
Num = [
    0.000006301299621181513574950678463482134,...
    0.000018903898863544541571884982644746742,...
    0.000018903898863544541571884982644746742,...
    0.000006301299621181513574950678463482134]
Den = [     1,...
    -2.925204528808932114003482638509012758732,...
     2.853180108439186568602963234297931194305,...
    -0.927925169233285163095104053354589268565,...]
x = [1, zeros(1,2047)];
tic;
y = filter(Num, Den, x);
toc
plot(y, 'r'); grid on; axis tight;
```

Время вычисления коэффициентов ФНЧ фильтра составляет t = 0.000154 секунды.

Для вычисления времени расчета коэффициентов ФНЧ фильтра на DSP-ядре вычислим тики работы с помощью соответствующего регистра. Для того что бы оценить время работы в секундах необходимо учитывать тактовую частоту процессора, которая составляет 720 МГц при рабочей нагрузке.

В листинге 2 и 3 представлены программы для оценки времени вычисления коэффициентов ФНЧ фильтра Баттерворта 3-го порядка в, написанных на языке С для RISC-процессора и на языке ассемблер для DSP-ядра соответственно.

Количество тиков при расчете коэффициентов ФНЧ фильтра (при многократном запуске программы) составляет в среднем 10500, что в итоге примерно равно t = 0.00001458 секундам.

Листинг 2

Фрагмент кода вычисления временных затрат для расчета коэффициентов фильтра на языке С под RISC

```
extern volatile float Out;
void filter(float* input, float* output, int length, float
B[], float A[])
{      b0 = B[0];
       b1 = B[1];
       b2 = B[2];
       b3 = B[3];
       _a0 = A[0];
       _a1 = A[1];
       _a2 = A[2];
       _a3 = A[3];
       PC(0) = ((unsigned int)&Reset_DSP- 0x3a600000) >> 2;
       DCSR(0) = 0x4000;
       unsigned int startupPointer = ((unsigned
int)&Start_DSP- 0x3a600000) >> 2;
       for (int i = 0; i < length; i++)
       {       In = input[i];

               //Run DSP core
               PC(0) = startupPointer;
               DCSR(0) = 0x4000;

               while (!(QSTR_DSP & (1 << 3)));

               output[i] = Out;       }  }
int main()
{   DCSR(0) = 0;
    SR(0) = 0;
    int length = 256 * 2 * 4;
    float B[] =
    {         0.00000630129962118151,
              0.0000189038988635445,
              0.0000189038988635445,
              0.00000630129962118151   };
    float A[] =
    {         1,
              -2.92520452880893,
              2.85318010843919,
              -0.927925169233285   };
    float signal[length];
    float result[length];
     for(int i = 0; i < length; i++)
    { signal[i] = 0;
      result[i] = 0;   }
    signal[0] = 1;
    TOTAL_CLK_CNT = 0;
    filter(signal, result, length, B, A);
    int dspRunTicks = TOTAL_RUN_CNT;
```

```
    int dspTotalTicks = TOTAL_CLK_CNT;
    int ticksCPU = utimer_tics_get(0);  }
```

Листинг 3

Фрагмент кода вычисления временных затрат для расчета коэффициентов фильтра на ассемблере под DSP

```
Reset_DSP:
            MOVE     0, R6.L
            MOVE     0, R8.L
            MOVE     0, R10.L
            MOVE     0, R12.L
       STOP
Start_DSP:
            MOVE In, A0
            MOVE b0, A1
            MOVE _a0, A2
            MOVE (A0), R0.L ; Input
            MOVE (A1), R2.L ; B0
            MOVE (A2), R4.L ; A0
            FMPY R0.L, R2.L, R6.L
            MOVE b1, A1
            MOVE _a1, A2
            MOVE (A1), R2.L ; B1
            FMPY R0.L, R2.L, R14.L FADD R8.L, R6.L   (A2),
R4.L ; A1
            FMPY R6.L, R4.L
            FSUB R4.L, R14.L, R8.L
            MOVE b2, A1
            MOVE _a2, A2
            MOVE (A1), R2.L ; B2
            FMPY R0.L, R2.L, R14.L FADD R10.L, R8.L (A2), R4.L
; A2
            FMPY R6.L, R4.L
            FSUB R4.L, R14.L, R10.L
            MOVE b3, A1
            MOVE _a3, A2
            MOVE (A1), R2.L ; B3
            FMPY R0.L, R2.L, R14.L FADD R12.L, R10.L (A2),
R4.L ; A3
            FMPY R6.L, R4.L
            FSUB R4.L, R14.L, R12.L
            MOVE _a0, A2
            MOVE (A2), R4.L
            MOVE Out, A2
            MOVE (A2), R14.L
            FMPY R4.L, R6.L, R14.L
            MOVE R14.L, (A2)
        STOP
 .data
In:  .real 0
b0:  .real 0
```

```
b1:  .real 0
b2:  .real 0
b3:  .real 0
_a0: .real 0
_a1: .real 0
_a2: .real 0
_a3: .real 0
Out: .real 0
.end
```

Полученные оценки времени расчета коэффициентов фильтра показали, что эффективность работы фильтра на DSP-ядрах системы-на-кристалле на порядок выше, чем с использованием унифицированных средств разработки (например, таких как Matlab).

Заключение

В работе рассматривались особенности проектирования БИХ-фильтра на системы-на-кристалле 1892ВМ14Я НПЦ ЭЛВИС. Произведена оценка производительности процессора MCom-02 в тесте для реализованного ФНЧ фильтра Баттерворта 3-го порядка в сравнении со временем расчетов в среде Matlab, которая показала значительное превосходство по эффективности использования системы-на-кристалле 1892ВМ14Я.

Список литературы

1. Baer, J. Microprocessor Architecture: From Simple Pipelines to Chip Multiprocessors / J. Baer. – Cambridge : Cambridge University Press, 2009. – 382 p.
2. Микросхема интегральная 1892ВМ14Я [Электронный ресурс] : Руководство пользователя – Режим доступа : https://multicore.ru/mc/data_sheets/Manual_1892VM14YA.pdf
3. DSP-кластер DELcore-30M. Архитектура [Электронный ресурс] : документация. – Режим доступа : https://multicore.ru/mc/data_sheets/Manual_DELcore-30M_031210.pdf
4. DSP-Ядро Elcore-30M. Система инструкций [Электронный ресурс] : Руководство пользователя. – Режим доступа : https://multicore.ru/mc/data_sheets/Manual_App1_DELcore-30M.pdf
5. Уильямс, Г. Отладка микропроцессорных систем / Г. Уильямс. – М. : Энергоатомиздат, 1988. – 253 с.
6. Иванова, В. Цифровая обработка сигналов и сигнальные процессоры [Электронный ресурс] : учебное пособие / В. Иванова, А. Тяжев. – Самара : Поволжский государственный университет телекоммуникаций и информатики, 2017. – 253 с.