

РАЗДЕЛ V. СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

имущества: - возможность исследовать "чистое" прецессионное движение, без искажения результатов измерений помехой, вызванной присутствием трения в осях подвеса гироскопа; - отсутствует необходимость ожидания интервала времени, требуемого для достижения ротором номинальных оборотов, поскольку это происходит мгновенно после запуска ротора в виртуальном пространстве; - возможность мгновенного изменения осевого момента инерции ротора (для реальной лабораторной установки потребовалось бы вносить корректировки в её конструкцию); - возможность замедления и ускорения времени в виртуальном пространстве позволяет наглядно изучать в динамике как сильно растянутые во времени процессы, так и носящие мгновенный характер; - возможность автоматического построения графиков зависимостей по окончании эксперимента, а также импорт полученных данных в другие среды разра-

ботки (такие, как AutoCAD, MATLAB, Mathcad) для дальнейшего исследования.

СПИСОК ЛИТЕРАТУРЫ

1. Описание стандарта X3D. [Электрон. данные]./ Режим доступа: www.web3d.org/x3d/
2. Кудряшов, Г.Н. Механика гироскопических систем./ Г.Н. Кудряшов, А.В.Сазонов, С.Ф. Скорина –СПб.: Изд-во ГУАП, 1992.
3. Агапонов, С.В. Средства дистанционного обучения. Методика, технология, инструментарий./ С.В. Агапонов, З.О. Джалиашивили, Д.Л. Кречман, и др. -СПб.: БХВ-Петербург, 2003.
4. Дари, К. Ajax и PHP. Разработка динамических веб-приложений./ К. Дари, Б. Бринзаре, Ф. Черчез-Тоза, М. Бусика –СПб.: Символ, 2006.
5. Вайк, Р.А. PHP. Справочник. / Р.А. Вайк, Д. Уокер, Р. Кокс. -Киев: ДияСофт, 2001.

*Аспирант, магистр техники и технологий
Афонин М. В., тел. 8-(921)-88-256-88, e-mail:
futures@inbox.ru, Государственный университет
аэрокосмического приборостроения (г. Санкт-Петербург).*

УДК 004.318-181.4 681.51:004.5

ОПТИМИЗАЦИЯ СХЕМЫ ТАКТОВОЙ СИНХРОНИЗАЦИИ ПРОГРАММНО-КОНФИГУРИРУЕМОГО ЦИФРОВОГО ПРИЕМНИКА НА ПРОЦЕССОРЕ TMS320VC55XX

В. В. Василевский, В. А. Панюшкин, П. И. Пузырёв

В статье рассмотрена оптимизация программы реализующей схему тактовой синхронизации для программно-конфигурируемого цифрового приемника на цифровом сигнальном процессоре серии TMS320VC55XX. Подробно рассмотрен процесс оптимизации и приведен оптимизированный код программы для интерполятора Фэрроу 3-го порядка и тактового генератора с числовым управлением.

Ключевые слова: оптимизация, цифровая обработка сигналов, вычислительная эффективность, восстановление тактовой синхронизации, программно-конфигурируемый цифровой приемник

к

При проектировании устройств на базе цифровых сигнальных процессоров (ЦСП) неизбежно возникает проблема выбора модели ЦСП и его производительности. С одной стороны ЦСП должен обеспечить выполнение программы с заданной скоростью, а с другой - иметь низкую стоимость, малое энергопотребление, тепловыделение и иметь все необходимые для устройства порты ввода/вывода. Для мобильных устройств одним из главных критериев выбора ЦСП является энергопотребление, при этом процессоры с меньшим энергопотреблением имеют меньшую производительность. Одним из способов уменьшения требований к производительности процессора является оптимизация алгоритма и кода программы. Как правило, опти-

мизируются только критические части кода, которые являются основными потребителями необходимого ресурса (процессорное время, память). Можно выделить три вида оптимизации программного обеспечения [1].

Оптимизация алгоритма. Является самым эффективным видом оптимизации и предопределяет теоретически достижимый порог оптимизации.

Машинно-независимая – оптимизация в пределах языка программирования высокого уровня не учитывающая особенности архитектуры процессора. Является промежуточным этапом оптимизации и среднем уровнем абстракции, позволяющим переносить программу без изменений на процессор с другой системой команд.

ПОЛЗУНОВСКИЙ ВЕСТНИК № 2 2010

ОПТИМИЗАЦИЯ СХЕМЫ ТАКТОВОЙ СИНХРОНИЗАЦИИ ПРОГРАММНО-КОНФИГУРИРУЕМОГО ЦИФРОВОГО ПРИЕМНИКА НА ПРОЦЕССОРЕ TMS320VC55XX

Машинно-зависимая – оптимизация на уровне машинного языка. С точки зрения эффективности является более предпочтительной по сравнению с машинно-независимой оптимизацией, поскольку именно с ее помощью можно учесть особенности конкретной вычислительной среды, однако машинно-зависимый оптимизатор непереносим.

Таким образом, оптимизируя программу по среднему времени выполнения, можно снизить требования к производительности процессора и применить в устройстве процессор с меньшей производительностью, и, следовательно, с меньшим энергопотреблением.

Постановка задачи

В работе рассмотрены все типы оптимизации программы, реализующей работу схемы тактовой синхронизации (рисунок 1) программно-конфигурируемого цифрового приемника для семейства ЦСП TMS320VC55XX.

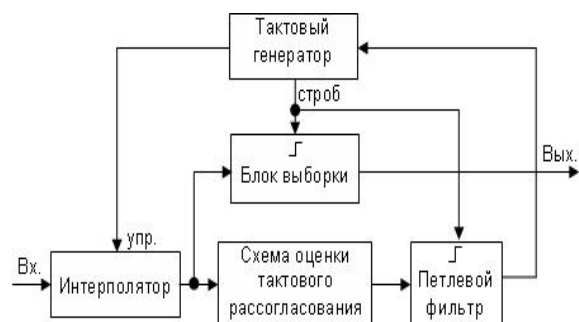


Рисунок 1 – Схема тактовой синхронизации

При разработке была использована имитационная модель схемы тактовой синхронизации, выполненная в программе Simulink Matlab.

Разработка и оптимизация программы для ЦСП выполнялись поэтапно:

1. Формирование структуры программы. Разбиение программы на подпрограммы с поблочной обработкой. Создание программного кода на языке С, согласно алгоритму работы имитационной модели, с учетом всех особенностей целочисленных процессоров (конечная точность, переполнение, особенности операций в формате Q15 и Q31) [2].

2. Отладка кода, с целью получения исходной программы для оптимизации. Профилирование (оценка скорости выполнения) кода до оптимизации.

3. Оптимизация алгоритма модели для снижения вычислительных затрат на ее обработку, подключение машинно-зависимых средств (intrinsics) компилятора языка С для

семейства ЦСП TMS320VC55XX [3]. Профилирование кода после оптимизации.

Результаты и их обсуждение

Схема тактовой синхронизации состоит из схемы восстановления тактовой частоты (СВТЧ) и блока выборки. СВТЧ необходима для формирования стробирующего импульса в момент, когда отклик на информационный символ максимален, а межсимвольные искажения минимальны [4]. СВТЧ состоит из интерполятора, схемы оценки тактового рассогласования, петлевого фильтра и тактового генератора с числовым управлением [5].

Структуру программы тактовой синхронизации можно условно разделить на следующие подпрограммы:

- интерполятор Ферроу
- схема оценки тактового рассогласования
- петлевой фильтр
- тактовый генератор с числовым управлением

По результатам профилирования неоптимизированной программы ЦСП были выделены критические части кода, являющиеся основными потребителями процессорного времени – интерполятор и тактовый генератор с числовым управлением. Рассмотрим подробно процесс оптимизации этих частей.

Подпрограмма интерполятора

В качестве интерполятора был взят фильтр Ферроу, аппроксимирующий кубический интерполятор Лагранжа, который достаточно просто реализуется в цифровых устройствах [6]. Выражение, описывающее работу фильтра Ферроу во временной области:

$$y(k) = h(3)x(k-3) + h(2)x(k-2) + h(1)x(k-1) + h(0)x(k) \quad (1)$$

где $y(k)$ – сигнал на выходе фильтра, $x(k)$ – сигнал на входе фильтра, $h(0)$, $h(1)$, $h(2)$, $h(3)$ – коэффициенты фильтра, зависящие от управляющего значения $\hat{\varepsilon}$ и определяющиеся следующими выражениями:

$$h(0) = -\frac{1}{6}(\hat{\varepsilon}-1)(\hat{\varepsilon}-2)(\hat{\varepsilon}-3) \quad (2)$$

$$h(1) = -\frac{1}{2}\hat{\varepsilon}(\hat{\varepsilon}-2)(\hat{\varepsilon}-3) \quad (3)$$

$$h(2) = -\frac{1}{2}\hat{\varepsilon}(\hat{\varepsilon}-1)(\hat{\varepsilon}-3) \quad (4)$$

$$h(3) = -\frac{1}{6}\hat{\varepsilon}(\hat{\varepsilon}-1)(\hat{\varepsilon}-2) \quad (5)$$

РАЗДЕЛ V. СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Кубический интерполятор можно представить параллельным соединением 3-х КИХ-

фильтров 3-го порядка (рисунок 2).

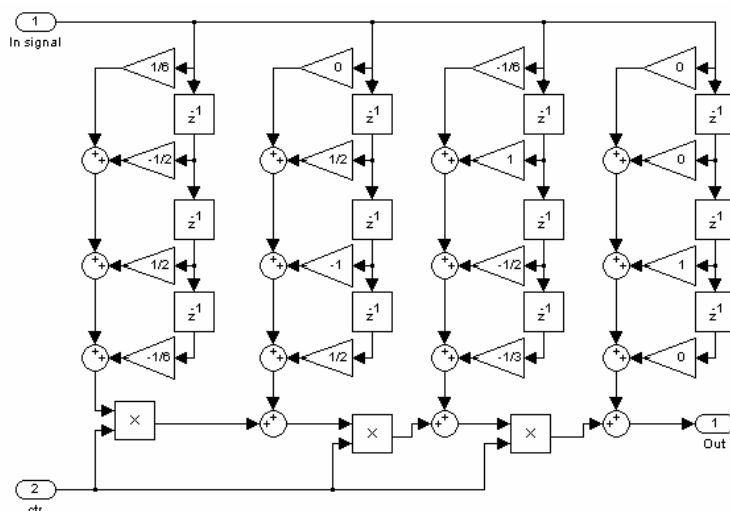


Рисунок 2 – Модель интерполятора Фэрроу 3-го порядка

Блок содержит 15 операций суммирования и 19 операций умножения. По рисунку нетрудно посчитать, что для расчета всех коэффициентов полинома требуется 9 нетривиальных умножений (умножения на ноль и ± 1 считаются тривиальными), а также 11 операций суммирования. Профилирование показало, что обработка буфера из 82 отсчетов занимает 5400 тактов процессора. Нетрудно заметить, что фильтры имеют схожие коэффициенты (равные по модулю), поэтому мож-

но сократить количество умножений. Модель, полученная после сокращения количества умножений, показана на рисунок 3 [7].

Оптимизированная модель (рисунок 3) содержит 6 операций умножения (2 из которых заменены сдвигами), 11 операций сложения/вычитания. Профилирование показало, что обработка 82 отсчетов занимает 940 тактов процессора. Эффективность кода возросла в 5,74 раза.

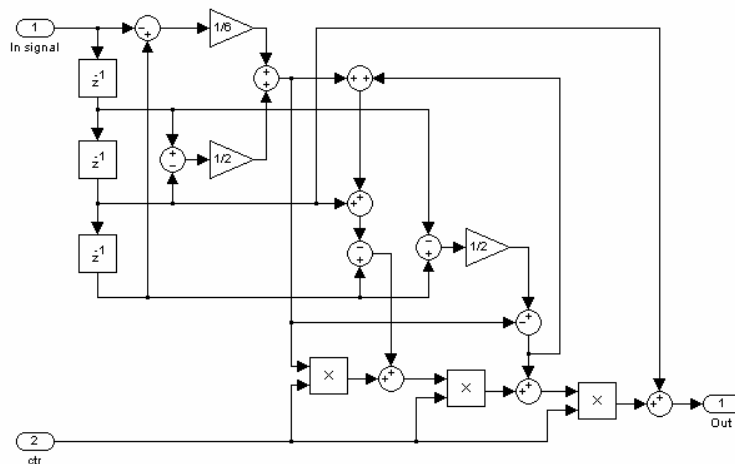


Рисунок 3 – Оптимизированная модель интерполятора Фэрроу 3-го порядка

Подпрограмма тактового генератора с числовым управлением

Структурная схема тактового генератора с числовым управлением приведена на рисунке 4. Тактовый генератор построен таким

образом, чтобы во всех частях схемы значения по модулю не превышало «1», что соответствует предельным значениям формата Q15.

ОПТИМИЗАЦИЯ СХЕМЫ ТАКТОВОЙ СИНХРОНИЗАЦИИ ПРОГРАММНО-КОНФИГУРИРУЕМОГО ЦИФРОВОГО ПРИЕМНИКА НА ПРОЦЕССОРЕ TMS320VC55XX

Основной проблемой при создании программного кода стала функция нахождения остатка от деления (mod), поскольку эта функция работает несколько иначе, чем аналогичная функция fmod из стандартной библиотеки math.h компилятора языка C [8]. В функции mod пакета Matlab вычисление коэффициента кратности q выполняется с округлением в сторону минус бесконечности:

$$q = \left\lfloor \frac{x}{y} \right\rfloor \quad (6)$$

$$\text{mod}(x, y) = x - y \times q \quad (7)$$

Такая функция отсутствует в библиотеках math.h и DspLib и ее необходимо реализовывать самостоятельно. Благодаря ограниченности динамического диапазона чисел в формате с фиксированной запятой [-1,1) функция сводится к следующему виду:

$$\text{mod}(x, y) = \begin{cases} x, & \text{при } 0 \leq x < y \\ x - y, & \text{при } y \leq x < 1 \\ y + x, & \text{при } -y \leq x < 0 \\ 2 * y + x, & \text{при } -1 \leq x < -y \end{cases} \quad (8)$$

При прямой реализации этой функции с использованием условных переходов на обработку буфера из 82 отсчетов подпрограмме тактового генератора требовалось 5800 тактов процессора.

В процессе оптимизации условные переходы были заменены логическими операциями и операциями суммирования/вычитания:

$$\text{nco_mod} = (\text{nco_out} \& 0x80000000) + \text{nco_out} - (\text{nco_out} \& 0x40000000).$$

Функционально представленная выше строка кода полностью соответствует функции нахождения остатка от деления на 0.5, реализованной в Simulink. Профилирование показало, что эффективность кода после оптимизации возросла в 6,2 раза.

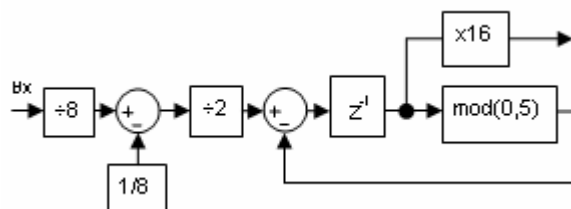


Рисунок 4 – Тактовый генератор с числовым управлением

В результате профилирования схемы тактовой синхронизации было определено

количество тактов процессора (N), необходимое для обработки буфера из 82 отсчетов.

Результаты профилирования приведены в таблице 1.

Таблица 1. Количество тактов на обработку (N) и эффективность оптимизации (K)

ОБЪЕКТ	N ДО ОПТИМ ИЗАЦИИ	N ПОСЛЕ ОПТИМИЗ АЦИИ	K
ИНТЕР-ПОЛЯТОР ТАКТОВЫЙ ГЕНЕРАТОР С ЧУ СХЕМА ТАКТОВОЙ СИНХРОНИЗАЦИИ	5400	940	5,74
	5900	950	6,21
	18400	9000	2,05

Ниже приведен фрагмент исходного текста оптимизированной нами программы.

```
// ----- Interpolator Farrow -----//
// Входные значения i_in и q_in
static s16 del[6]={0,0,0,0,0,0};
static s16 sum[6]={0,0,0,0,0,0};
static s16 out_interpol_i, out_interpol_q, i_in, q_in;
static s32 ctr=0;
sum[0] = _smpy((del[2]-i_in),0x1555)+((del[0]-del[1])>>1);
sum[1] = (del[0]-del[2])>>1-sum[0];
sum[2] = del[2]-sum[0]-sum[1]-del[1];
sum[3] = _smpy((del[5]-q_in),0x1555)+((del[3]-del[4])>>1);
sum[4] = (del[3]-del[5])>>1-sum[3];
sum[5] = del[5]-sum[3]-sum[4]-del[4];
sum[1]+=_smpy(sum[0],(s16)(ctr>>16));
sum[2]+=_smpy(sum[1],(s16)(ctr>>16));
out_interpol_i=_smpy(sum[2],(s16)(ctr>>16))+del[1];
sum[4]+=_smpy(sum[3],(s16)(ctr>>16));
sum[5]+=_smpy(sum[4],(s16)(ctr>>16));
out_interpol_q=_smpy(sum[5],(s16)(ctr>>16))+del[4];
del[5]=del[4];
del[4]=del[3];
del[3]=q_in;
del[2]=del[1];
del[1]=del[0];
del[0]=i_in;
//----- end of interpolator -----//
//----- NCO -----//
// вход nco_in (s32)
// выход overflow (s32)
static s32 nco_in, nco_out, nco_mod;
static s32 overflow;
```

РАЗДЕЛ V. СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

```
nco_out=nco_mod-((0x20000000-  
(nco_in>>2))>>1); // nco_mod-(1/4-nco_in/4)/2  
nco_mod=(nco_out&0x80000000)+nco_out-  
(nco_out&0x40000000); // mod om 0.5
```

```
if (nco_out<=0)  
{  
    overflow = 0x7FFFFFFF;  
} else overflow=0;  
//----- end of NCO -----//
```

Выводы

В результате комплексного подхода к оптимизации программы ЦСП был достигнут выигрыш по среднему времени выполнения более чем в два раза по сравнению с неоптимизированной программой. Благодаря использованию машинно-зависимых средств (intrinsics) компилятора языка C для семейства ЦСП TMS320VC55XX ЦОС программа сохранила высокий уровень переносимости на процессор с другой системой команд. При переносе следует заменить эти средства на соответствующие другому компилятору языка C, а в случае их отсутствия подключить виде внешних функций. Дальнейшая оптимизация программы может быть достигнута путем частичного или полного перевода программы на язык ассемблера, однако при этом переносимость будет полностью потеряна.

СПИСОК ЛИТЕРАТУРЫ

1. Касьянов, В.Н. Оптимизирующие преобразования программ./ В.Н. Касьянов– М. : Наука, 1988.
2. Сергиенко, А.Б. Цифровая обработка сигналов./ А.Б. Сергиенко. – СПб.: Питер, 2003.
3. TMS320C55x optimizing C/C++ compiler : User's guide, SPRU281F, 2003. [Электрон.ресурсы]. URL: <http://focus.ti.com/lit/ug/spru281f/spru281f.pdf>
4. Gardner, F.M. A BPSK/QPSK timing error detector for sampled receivers. // F.M. Gardner IEEE Trans. Commun.- 1986.- Vol. 34, P. 423–429.
5. Nezami, M.K. RF architectures & digital signal processing: Aspects of digital wireless transceivers. // M.K. Nezami IEEE Trans. Commun. - 2003.
6. Farrow, C.W. A continuously variable digital delay element. // IEEE Int. Symp. "Circuits and Systems".- Espoo, Finland, 1988.- P. 2641-2645.
7. Фильтры Фарроу (Farrow filters) на примере фильтра третьего порядка. Ресэмплинг сигналов. // DSPSYSTEM. Теория и практика цифровой обработки сигналов. [Электрон.ресурсы].URL: <http://www.dspsystem.narod.ru/content/farrow>
8. Modulo operation. // Википедия: свободная электронная энциклопедия: на английском языке. [Электрон. ресурсы]. URL: http://en.wikipedia.org/wiki/Modulo_operation

М.н.с., аспирант **Василевский В.В.**, тел. 8-960-990-1248, e-mail: altego@omgtu.ru; инженер, **Панюшкин В.А.**, тел. 8-908-792-30-07, e-mail: panyushkin@rtuisd.ru; инженер, **Пузырёв П.И.**, тел. 8-908-105-48-51, e-mail: puzyrev@rtuisd.ru Омский государственный технический университет, НИЧ, кафедра «Радиотехнические устройства и системы диагностики» (г. Омск).

УДК 681.2.001.66:004.5

ОБЩИЕ ПОДХОДЫ ПРИ РАЗРАБОТКЕ СПЕЦИАЛИЗИРОВАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ВИРТУАЛИЗИРОВАННЫХ ИЗМЕРИТЕЛЬНЫХ ПРИБОРОВ

С.Ф. Дмитриев, Д.Н. Лященко, А.В. Новоженев, А.В. Ишков

Авторами выработаны общие подходы, лежащие в основе создания специализированного программного обеспечения виртуализированных измерительных приборов. Общие подходы при разработке приборных приложений основаны на использовании интерфейса низкого уровня для доступа к буферу с данными в wave-формате, микшерной подсистемы Windows для контроля параметров звукового устройства и модульном принципе построения алгоритма прибора. Приведено описание алгоритмов и основных функций wave-модуля, модуля инициализации и калибровки и модуля основного цикла приборов ИЭНМ-20М и ВВТ-2М.

Ключевые слова: виртуализированный измерительный прибор, программное обеспечение, интерфейс низкого уровня, микшерная подсистема, программный модуль.

Современные подходы к реализации измерительного прибора (ИП) базируются на нескольких положениях, связанных с общими

концепциями измерений, средствами их осуществления, объектами и их моделям, а также метрологическом эксперименте [1, 2].