

РАЗДЕЛ I. ОБЩИЕ ВОПРОСЫ СИНТЕЗА И АНАЛИЗА ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫХ СИСТЕМ

2. Немудров, В., Мартин Г. Системы на кристалле. Проблемы проектирования и развития./ В. Немудров, Г.М. Мартин: М.: Техносфера. 2004.
3. Оленев, В. Методы межмодульного взаимодействия при моделировании протоколов встроенных систем/ В. Оленев, Л.Онищенко, А. Еганян // Научная сессия ГУАП. –С-Пб: СПбГУАП. 2008. С. 98-99.
4. Кларк, Э. Верификация моделей программ: Model Checking /Э.Кларк и др.- М.: Московский центр непрерывного математического образования, 2002.
5. Питерсон, Дж. Теория сетей Петри и моделирование систем / Дж. Питерсон. - N.J.: Prentice-Hall. 1981.
6. Котов, В.Е. Сети Петри/ В.Е. Котов.- М.: Наука, 1984.

Аспирант, магистр техники и технологии
Оленев В.Л., тел. 8-911-797-0533, e-mail: Valentin.Olenev@guap.ru, Государственный университет аэрокосмического приборостроения (г. Санкт-Петербург).

УДК 004.322

СТАТИЧЕСКИЕ МОДЕЛИ ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ УСТРОЙСТВ КАНАЛЬНОГО УРОВНЯ

Е.Н.Яблоков

В данной статье рассмотрены существующие статические модели для оценки устройств канального уровня. В качестве альтернативы предложена новая статическая модель оценки канала, приближенная к физической реализации, что позволяет более точно смоделировать задержки при передаче данных.

Ключевые слова: канальный уровень, приемник, передатчик, статическая модель, анализ.

С помощью статических моделей можно оценить производительность системы как функцию, зависящую от различных параметров. Самая главная проблема при построении такой модели – определение необходимых параметров и их взаимодействия. Составление модели производится в несколько этапов. В самом начале берутся задержки на самом высоком уровне синтеза для встроенных систем реального времени. Это позволяет рассмотреть задержки после отображения всех высокоуровневых задач на процессорные узлы, для проверки того, что реальные задержки не превышают заданных ограничений. Если задачи на одном процессорном узле должны взаимодействовать – то добавляются небольшие задержки сверху. В то же время, если задачам разных процессорных узлов надо взаимодействовать - им необходимо соединиться друг с другом или путем шины, или соединениями точка-точка с соответствующими задержками. Эти задержки зависят от количества информации, необходимой для передачи между процессорными узлами. Обычно при таком походе общее количество передаваемых данных на линии оценивается статически, которое потом изменяется путем умножения на масштабный коэффициент коммуникационной архитектуры для получения общих коммуникационных затрат [1].

Последующие подходы фокусируются на оценке коммуникационных задержек, чтобы

удостоверится, что программное обеспечение и аппаратная часть удовлетворяет ограничениям по производительности. Можно попробовать описать простую модель и описывающие коммуникационные задержки для протокола, используя простое выражение, включающее в себя количество информации, которую необходимо передать, отношение ширины канала к размеру передаваемого элемента данных, и время передачи одного элемента данных. Также можно попробовать расширить данную модель путем создания более детальной оценочной модели для программно-аппаратной архитектуры [2].

Обзор известных моделей

На рисунке 1а показана модель Кнудсена и Мадсена [3, 4], разделенная на 3 части – модель передатчика, модель канала, модель приемника. Такой подход позволяет оценить не только задержки, наложенные каналом, но и так же позволит оценить понижение производительности, наложенное программно/аппаратными драйверами. Время на установку соединения не учитывается (например, арбитраж) не моделируются, так как они не важны для соединений точка-точка.

Рисунок 1б показывает параметры драйвера, которые будут использоваться при расчете задержек на передающей модели драйвера. Драйвер получает n_t слов от процессорного узла для передачи через канал и соответственно делает из них n_s слов канала.

Для этого может потребоваться преобразование ширины слов, приходящих драйверу в слова ширины канала. Если частота процессорного узла равна f_t , количество тактов канала, которые требуются для инициализации драйвером канала (настройки канала и т.д.) равно c_{tc} , а количество тактов частоты процессорного узла, которое требуется на передачу одного слова драйвера равно c_{tp} , то общая задержка передачи драйвера может быть рассчитана как:

$$t_{td} = (c_{tc} + c_{tp}n_t) / f_t$$

Параметр w_g показывает время, которое было затрачено на преобразование из слов драйвера в слова канала и это время напрямую влияет на количество тактов процессорного элемента (c_{tp}). Варианты преобразования могут быть очень различны – от прямого преобразования (количество бит пакета драйвера совпадает с количеством бит пакета канала) до однобитного преобразования (каждый пакет драйвера передается побитно). Каждый подход имеет достоинства и недостатки, например, при прямом преобразовании в канале будет много пакетов, что приведет к дополнительным задержкам, а при однобитном преобразовании меньше слов будет передано драйвером, таким образом, необходим тщательный анализ.

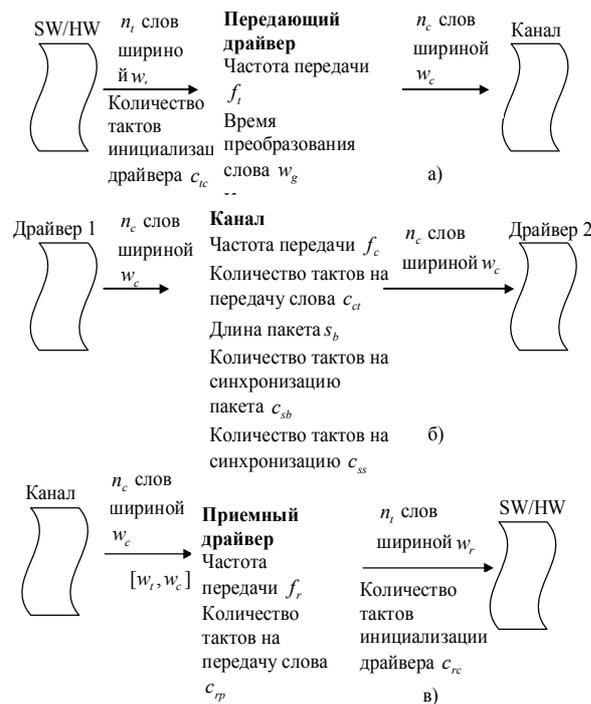


Рисунок 1 - Модель Кнудсена и Мадсена

Рисунок 1в показывает параметры канала, которые используются в модели канала. Количество слов канала n_c передаются пакетом $(n_b - 1)$ слов канала шириной s_b , и конечной частью s_r , причем $0 \leq s_r \leq s_b$:

$$n_c = (n_b - 1)s_b + s_r$$

Число n_b выбирается следующим образом. Пусть b_m определяет 1 из 3-х видов передачи – фиксированная, обозначенная как fixed (все пакеты одной длины), максимальная, обозначенная как max (существует максимальная длина пакета, однако пакеты меньшей длины так же могут быть) и бесконечная, обозначенная как inf (нет ограничения на размер пакета). В таком случае параметр n_b будет вычисляться как:

$$n_b = \begin{cases} 1 & b_m = \text{inf} \\ n_{cd}s_b & b_m = \text{fixed}, \text{max} \end{cases}$$

где n_{cd} - число слов канала в отношении к количеству входящих слов драйвера n_t шириной w_t , которые были преобразованы в слова канала шириной w_c .

При любой передаче требуется синхронизация между посылками пакетов (например такты выборки управляемых устройств, или обмен настройками). Число тактов, необходимых для синхронизации обозначим c_{sb} . Передача же всех n_c слов потребует дополнительной общей синхронизации перед всеми пакетами, которое мы обозначим c_{ss} . Общее количество тактов синхронизации для передачи n_c слов будет равно:

$$c_{cs} = [n_b c_{sb}] + c_{ss}$$

Таким образом, для частоты передачи канала f_c , при количестве тактов передачи через канал одного слова c_{ct} , и количестве передаваемых слов n_c , задержки в канале передачи будут равны:

$$t_{cd} = (c_{cs} + c_{ct}n_c) / f_c$$

Расчет данной задержки предполагает, что соединение уже установлено, таким об-

РАЗДЕЛ I. ОБЩИЕ ВОПРОСЫ СИНТЕЗА И АНАЛИЗА ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫХ СИСТЕМ

разом не берутся в расчет задержки на установку соединения и арбитраж.

Для построения модели будет сделано допущение, что приемник знает параметры W_t и W_g , таким образом зная принцип преобразования данных. Так же делается допущение, что $W_r \geq W_t$, чтобы каждое передаваемое слово помещалось в одно слово приемное слово. При частоте приемного процессорного узла f_r , числе тактов для инициализации приемного драйвера c_{rc} , и числа тактов приема одного не преобразованного слова данных c_{rp} , задержки на приемной части драйвера будут равны

$$t_{rd} = (c_{rc} + c_{rp}n_t) / f_r$$

Принимая в расчет то, что передающая часть, канал и приемная часть работают параллельно (туннельная передача), то можно сделать вывод, что самая большая задержка из вышеперечисленных даст нам задержку:

$$t_m = \max(t_{td}, t_{cd}, t_{rd})$$

А полную задержку можно высчитать по формуле

$$t_m = t_m + 2t_m / n_t$$

Второй компонент представляет собой приблизительную задержку на запуск и завершение передачи через туннель. Данная задержка является худшим случаем.

Таким образом, мы получили простую модель расчета задержек при соединении двух процессорных узлов.

Коммуникационная модель Ренера [5], показана на рисунке 2. Такая модель предполагает не только задержки приемной, канальной и передающей модели, но и задержки, которые появляются в результате выбора определенного протокола. Используя дополнительные задержки, определенные в протоколе, строятся модели приемной, передающей и канальной части.

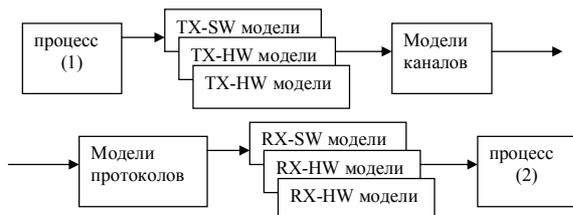


Рисунок 2 - Коммуникационная модель Ренера

Модель передающей стороны для расчета задержек:

$$t_o^T = \frac{n_{tc}^{SW}}{f_{tc}^{SW}} + \frac{n_{tc}^{HW}}{f_{tc}^{HW}}$$

где n_{tc} - количество тактов, необходимых для передачи данных в канал, а f_{tc} - частота передающей части драйвера (обозначение sw – означает программную реализацию драйвера, hw – аппаратную реализацию драйвера). Так как драйвер, в зависимости от реализации, может быть либо программным, либо аппаратным, то один из слагаемых вышеобозначенной модели будет равен 0. Данная передающая модель Ренера предусматривает добавление дополнительной задержки для преобразования/сжатия данных. Приемная модель идентична передающей модели.

Новая модель канала передачи данных

Предлагаемая модель для расчета задержек канала основана на структуре базы коммутационных элементов, включающую в себя задержки на программируемой логике (FPGA) и задержки на элементах дуплексной памяти (DPRAM), которые используются для задания различных задержек протоколов.

Модель канала представлена как:

$$t_p^c = kt_{sw} + \sum_i t_{fpga,i} + mt_{DPR} + \sum_i t_{wire,i}$$

где число k обозначает количество коммутационных элементов, t_{sw} - задержка элемента, t_{fpga} - задержки на элементах FPGA, число m – количество обращений к памяти, t_{ddr} - задержка на памяти, t_{wire} - задержки на шинах. В зависимости от типов соединения некоторые слагаемые будут равны 0. Параметры для данной модели задаются табличным способом.

Дополнительные задержки в протоколе обозначаются параметром t_p^p . Зависит данный параметр от расходов на арбитраж, проверку контрольной суммы и т.п. Общая задержка будет равна:

$$t_p = t_p^T + t_p^C + t_p^p + t_p^R$$

Однако, если соединение представляет из себя туннель, то общая задержка будет равна максимальной задержке из всех:

$$t_p = \max(t_p^T, t_p^C, t_p^p, t_p^R)$$

Особенности реализации модели

Рассмотрим модель более подробно. Отметим в данной модели есть свои недостатки, например, некоторые параметры, задаются статическими значениями, использовано ровно 3 этапа преобразования: драйвер-канал, канал, канал-драйвер.

Попробуем представить общую модель передачи данных (рисунок 3).

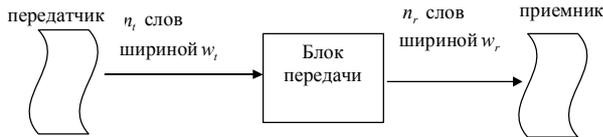


Рисунок 3 - Модель передачи данных

На этом рисунке число слов n_t шириной w_t передается от некоего передатчика в блок передачи (модель которого мы и строим), а от него в некоторый приемник с количеством слов n_r шириной w_r . Количество слов и ширина не обязательно должна совпадать, так как приемник и передатчик могут находиться в любой точке модели. Задача сводится к представлению работы модели блока передачи. Блок передачи может состоять из множества блоков с частотой, соединенных между собой (рисунок 4).

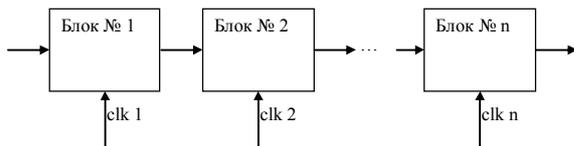


Рисунок 4 - Блоки передачи данных

На данной модели блок передачи распадается на большое количество подблоков, работающих на своей частоте. Каждый блок, преобразовав данные передает их в следующий блок для дальнейшей передачи. Каждый блок представляет собой синхронный блок, работающий на своей частоте. Однако, для передачи данных с одной частоты на другую требуется некоторое количество времени. Таким образом каждый блок можно разбить на 2 блока, один из которых будет отвечать за перевод данных с одной частоты на другую, а второй собственно само преобразование (рисунок 5).

Таким образом, можно сказать, что время задержки на каждом таком блоке можно высчитать по формуле

$t_m = t_d + t_c$, где t_d - время, которое затрачено на перевода данных с частоты на частоту, а t_c - время преобразования данных во втором блоке.

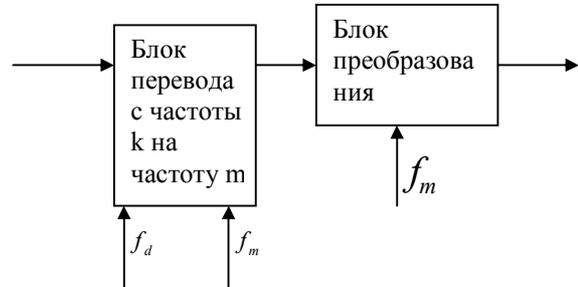


Рисунок 5 - Подблок модели

Если посмотреть на рисунок 4, то так как каждое преобразование в блоке будет выполняться параллельно с остальными преобразованиями. Время, затраченное на перевод на другую частоту можно рассчитывать по формуле:

$$t_d = c_{id} / f_d + c_{im} / f_m, \text{ где } c_{id} - \text{ количество тактов, потраченных на преобразование}$$

на тактовом сигнале f_d , c_{im} - количество тактов, потраченных на преобразование, на частоте f_m . Минимальное количество тактов обычно равно 2-м на приемной частоте, однако может быть и больше при более сложном преобразовании. Если же частоты одинаковые или зависят друг от друга, время t_d может равняться 0.

Если рассмотреть полную модель, состоящую из целого ряда блоков и учитывая, что преобразование идет одновременно, то получается формула:

$$t_o = \sum_{i=1}^{i=n} t_{d_i} + \max(t_{c_1}, t_{c_2}, t_{c_3}, \dots, t_{c_n})$$

где n – количество блоков.

Данная модель состоит из 2-х слагаемых – первая сумма времени перевода с частоты на частоту и максимальное время преобразования блоков данных. Таким образом, после расчета времени каждого блока – будет получена общая модель.

Каждый блок можно представить виде набора входных слов и выходных (рисунок 6). Рассмотрим данную модель более подробно. Данная модель похожа на представленную выше за исключением того, что эта модель является более общей и убрано время на

РАЗДЕЛ I. ОБЩИЕ ВОПРОСЫ СИНТЕЗА И АНАЛИЗА ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫХ СИСТЕМ

инициализацию. Мы передаем n_t слов шириной w_t и получаем некоторое количество слов n_r шириной w_r .

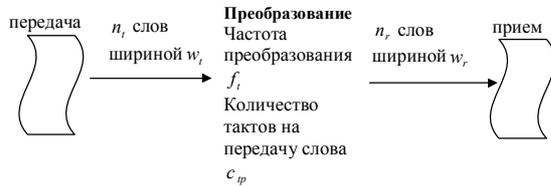


Рисунок 6 - Блок передачи данных (входные и выходные параметры)

Время, которое тратится на посылку каждого слова равно c_p . Время на передачу этих слов равно:

$t_c = c_p n_t / f_t$, где c_p - количество тактов, которое требуется на обработку 1 слова данных на частоте f_t . Однако, данный параметр можно не просто предположить, а рассчитать более точно. Предположим, что ширина входного слова больше ширины выходного. Тогда количество этапов преобразования слова будет кратно целой части w_t / w_r , т.е. количество передаваемых слов будет на w_t / w_r больше числа принимаемых. Если домножить на количество тактов задержки внутри блока (пусть m), то мы получим время задержки внутри блока:

$$t_c = [w_t / w_r] m n_t / f_t,$$

а количество получаемых слов будет равно

$$n_r = [w_t / w_r] n_t$$

Если же ширина выходного слова больше ширины передаваемого, то задержка внутри блока будет высчитываться по формуле:

$$t_c = m n_t / f_t,$$

а количество полученных выходных слов будет равно

$$n_r = [(w_t / w_r) n_t]$$

Заменим $[w_t / w_r]$ на коэффициент W_o , такой, что

$$W_o = \begin{cases} [w_t / w_r], & \text{если } w_t / w_r \geq 1 \\ 1, & \text{если } w_t / w_r < 1, \text{ конвейер} \\ (w_r / w_t), & \text{если } w_t / w_r < 1 \end{cases}$$

Тогда формула задержки будет

$$t_c = W_o m n_t / f_t$$

Время задержки между приемной и передающей частью

$$t_k = \max(t_{m_1}, t_{m_2}, t_{m_3}, \dots, t_{m_n})$$

Полная задержка данной модели будет высчитываться по формуле

$$t_o = t_k + 2t_k / n_t$$

У данной модели есть следующие преимущества:

- Не зависит от исследуемой структуры
- Простота моделирования
- Прозрачность параметров
- Возможность разделения на подблоки в сложных случаях

В каналах точка-точка (пример узла на рисунке 7) может быть реализовано несколько временных доменов.

Использование вышеописанной модели серьезно упростит создание блоков, рассмотренных на рисунках 4-7.

В качестве еще одного параметра для характеристики подобных устройств используется пропускная способность.

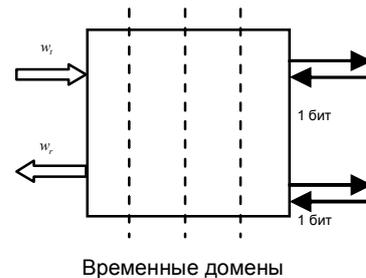


Рисунок 7 - Узел «точка-точка»

Анализ пропускной способности модели

Для расчета данного параметра мы воспользуемся выведенной выше формулой задержек в канале передачи данных. Таким образом пропускная способность равна

$B = L / t_s$, где t_s - время на передачу информации, L - количество бит на передачу

Можно представить количество бит на передачу в виде произведения количества по-

сылаемых слов (n_t) и ширины посылаемого слова (w_t), то формула преобразуется в виде

$$B = n_t w_t / t_s$$

Время на передачу информации также зависит от двух параметров – время на передачу самой информации, которое было посчитано выше (t_o) и временем на ответ (t_a), то окончательная формула будет выглядеть так:

$$B_o = n_t w_t / (t_o + t_a)$$

Время на ответ может быть посчитано исходя из множества факторов так же как и время на задержку, в зависимости от длины и ширины ответного пакета.

Однако задержки на передачу пакета – это не единственные временные характеристики, которые учитываются при отсылке. Предположим, мы посылаем некоторое количество пакетов некоторой длины и ширины. Тогда часть времени будет затрачено на подготовку связи, то есть на установку соединения. Формулу пропускной способности можно преобразовать в виде

$$B_o = (q n_t w_t / ((q(t_o + t_a) + t_n)))$$

где q – это число пакетов на отсылку в канал, t_n - время, затрачиваемое на установку соединения.

Если при передаче пакета возникнет ошибка, то возможно искажение данных и пакет необходимо передать заново. В результате ошибке связь разрывается и установку соединения необходимо производить заново.

Таким образом, если при передаче q пакетов произойдет s ошибок, то формула трансформируется в

$$B_o = (q n_t w_t / ((q(t_o + t_a) + t_n) + s((t_o + t_a) / 2 + t_n)))$$

где второе слагаемое в знаменателе – время, затраченное на передачу ошибочных пакетов.

Время отправки и ответа поделено пополам для простоты, ибо ошибка может быть как в начале так и в конце пакета и в среднем пакет передается на половину времени. Необходимо отметить, что s зависит от многих параметров. Данное число можно представить как зависимость от некоторого вероят-

ностного параметра ошибки в пакете. То есть параметр s может быть представлен в виде:

$$s = \text{round}(q p_q)$$

где P_q - это вероятность ошибке в пакете.

Однако, если число s будет посчитано по этой формуле – оно не будет учитывать ошибки в повторенных пакетах. Так как ошибки могут возникать как в оригинальных пакетах, так и в отправленный после восстановления после ошибки. То есть каждое последующее число ошибок (и соответственно число перепосланных пакетов) можно высчитать с помощью геометрической прогрессии

$$S_{n+1} = P_q S_n$$

А общее количество всех перепосланных пакетов можно посчитать в виде суммы всех членов прогрессии

$$s = \text{round}(q p_q) + \text{round}(q p_q p_q) + \text{round}(q p_q p_q p_q) + \dots = \sum_{i=1}^{\infty} \text{round}(q p_q^i)$$

Преобразовав формулу для пропускной способности, мы получим такое выражение

$$B_o = (q n_t w_t / ((q(t_o + t_a) + t_n) + \sum_{j=1}^{\infty} \text{round}(q p_q^j) ((t_o + t_a) / 2 + t_n)))$$

Более точно можно посчитать вероятность искажения пакета. Для этого можно воспользоваться значением, описываемым во всех стандартах, а именно вероятностью для искажения бита

$$p_q = (1 - (1 - p_b)^{n_t w_t})$$

Если подставить данный параметр, то окончательная формула для расчета пропускной способности будет

$$B_o = (q n_t w_t / ((q(t_o + t_a) + t_n) + \sum_{i=1}^{\infty} \text{round}(q(1 - (1 - p_b)^{(n_t w_t + c_t n_t)})^i) \cdot ((t_o + t_a) / 2 + t_n)))$$

В результате мы можем вывести два параметра для расчетов:

а) общего времени задержки при передаче -

$$t_o = t_k + 2t_k / n_t, u$$

РАЗДЕЛ I. ОБЩИЕ ВОПРОСЫ СИНТЕЗА И АНАЛИЗА ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫХ СИСТЕМ

б) пропускной способности -

$$B_o = (qn_i w_i / ((q(t_o + t_a) + t_n) + \sum_{i=1}^{\infty} \text{round}(q(1 - (1 - p_b)^{(n_i w_i + c_i n_i)})^i) \cdot ((t_o + t_a) / 2 + t_n))$$

Заключение

В статье были рассмотрены статические методы анализа аппаратных устройств. Они дают очень быстрые результаты, которые позволяют оценить целесообразность создаваемого устройства канального уровня еще на этапе проектирования. Однако для более точного анализа необходимо использовать динамические или смешанные модели.

СПИСОК ЛИТЕРАТУРЫ

1. Gogniat, G. "Communication synthesis and HW/SW integration for embedded system design/

- M. Auguin, L. Bianco, A. Pegatoquet.- " CODES 1998, pp 49-53.
2. Kalla, P. A flexible framework for communication evaluation in SoC design / P. Kalla, X. Hu and J. Henkel, - журнал "Parallel Programming on Embedded Processors", Vol. 36, No. 5, October 2008, pp. 457-477.
3. Knudsen, P. Communication Estimation for Hardware/Software Codesign/ P. Knudsen, J. Madsen.-" CODES 1998, pp. 55-59.
4. Knudsen, P. Integrating communication protocol selection with hardware/software codesign/ P. Knudsen, J. Madsen. - ISSS 1998, pp. 111-116.
5. Renner, F-M. Automated Communication Synthesis for Architecture-Precise Rapid Prototyping of Real-Time Embedded Systems/ F-M. Renner, J. Becker, M. Glesner - RSP 2000, pp. 154-159.

Аспирант, магистр техники и технологий
Яблоков Е.Н., тел. (812) 571-15-22, e-mail: Yablokov@guar.ru, Государственный университет аэрокосмического приборостроения, каф. 51 (г. Санкт-Петербурга).

УДК 681.586.773

РАЗРАБОТКА МНОГОЭЛЕМЕНТНЫХ ИЗМЕРИТЕЛЬНЫХ УСТРОЙСТВ С ИСПОЛЬЗОВАНИЕМ СВЯЗАННЫХ КОЛЕБАНИЙ ПЬЕЗОРЕЗОНАТОРОВ

А.О. Тицнер, А.В. Балыков, А.С. Назаров, А.А. Тятюхин, М.Ю. Ларионов, В.Н.Седалищев

В статье рассматриваются новые принципы построения многоэлементных первичных измерительных преобразователей, основанные на использовании связанных колебаний составных пьезорезонаторов. Устройства данного типа могут найти практическое применение для измерения и контроля различных физических величин.

Ключевые слова: составной пьезорезонатор, ансамбль взаимодействующих осцилляторов, связанные колебания, многоэлементный первичный измерительный преобразователь

Современные тенденции развития отрасли производства датчиков определяются в основном использованием прогрессивных технологий их производства, а также использованием новых физических принципов их функционирования. Основными задачами, которые приходится решать при создании новых типов датчиков, являются: повышение быстродействия измерений, точности измерений, термоустойчивости, надежности, пожаро- и взрывобезопасности, обеспечение дистанционности измерений. Для снижения неопределенности результатов измерений приходится производить обработку массивов данных, полученных от большого числа датчиков, расположенных в различных точках объекта измерения. С усложнением измери-

тельных систем растет объем измерительной информации, особенно если приходится следить за изменением параметров во времени. Производить быструю обработку таких объемов информации современными средствами вычислительной техники уже сейчас является достаточно трудоемкой задачей.

Решение проблемы

Выходом из этой ситуации может послужить объединение процессов получения, передачи и обработки измерительной информации в одном интеллектуальном измерительном устройстве. Такие системы уже на начальных этапах получения измерительной информации могут обеспечивать ее предварительную обработку, например, осуществ-