

УДК 004.272.3

Р.В. Бредихин, аспирант, Юго-Западный государственный университет (Курск)**И.В. Зотов**, д-р техн. наук, профессор, Юго-Западный государственный университет (Курск)
(e-mail: zotovigor@yandex.ru)**РАСПРЕДЕЛЕННАЯ ПРОЦЕДУРА ЦИКЛИЧЕСКОЙ БАРЬЕРНОЙ СИНХРОНИЗАЦИИ
ДЛЯ ОДНОКРИСТАЛЬНЫХ МАТРИЧНЫХ МУЛЬТИКОМПЬЮТЕРОВ**

Представлена распределенная аппаратно-ориентированная процедура, описывающая процесс барьерной синхронизации в циклических параллельных программах, выполняемых однокристальными матричными мультимикомпьютерами, проведено исследование динамической корректности процедуры с использованием аппарата сетей Петри.

Ключевые слова: однокристальные мультимикомпьютеры, СБИС, параллельные процессы, барьерная синхронизация, сети Петри.

Современная технология дает возможность создания СБИС емкостью свыше 2 млрд транзисторов, позволяя реализовать на одной микросхеме мощные мультимикомпьютеры, содержащие 100 и более процессорных модулей. Типичными представителями таких однокристальных систем являются матричные мультимикомпьютеры фирмы Tiler, объединяющие от 32 до 100 процессорных элементов, связанных коммуникационной средой в двухмерную матричную структуру [1].

Функционирование мультимикомпьютеров сопряжено с выполнением ряда коммуникационных операций, к числу которых относится барьерная синхронизация. Она заключается в согласовании моментов завершения и запуска параллельных ветвей программы в определенной ее точке (называемой барьером) и предполагает приостановку выполнения некоторых ветвей (процессов) до наступления условия синхронизации, при котором все требуемые процессы достигли барьера [2]. Недостаточно эффективная организация синхронизации может значительно увеличить общее время выполнения параллельной программы, ограничивая производительность мультимикомпьютера.

Реализация барьерной синхронизации на программном уровне связана с необходимостью передачи большого числа служебных сообщений между взаимодей-

ствующими процессами, в связи с чем на практике большее распространение получила синхронизация на аппаратном уровне. Идея подобной аппаратной синхронизации уже нашла воплощение во многих коммерческих системах и широко представлена в литературе в форме различных аппаратно-ориентированных процедур синхронизации [3-6]. На современном этапе развития вычислительной техники особый интерес представляют распределенные аппаратные процедуры синхронизации, которые, с одной стороны, органично вписываются в архитектуру однокристальных мультимикомпьютеров, а с другой стороны, позволяют достичь более высокой скорости межпроцессорного взаимодействия при сохранении уровня гибкости и масштабируемости, присущего программно реализованным процедурам синхронизации [7,8].

В данной статье предлагается модификация известной распределенной процедуры барьерной синхронизации, основанной на применении многослойной координирующей среды [9], обеспечивающая дальнейшее ускорение межпроцессорного взаимодействия в матричных мультимикомпьютерах при реализации циклических параллельных программ.

**Характеристика задачи обеспечения
барьерной синхронизации**

Барьерная синхронизация заключается во временном согласовании этапов

завершения и запуска некоторого подмножества параллельных процессов (барьерной группы), размещенных в различных процессорных модулях. Процесс синхронизации состоит из двух последовательно протекающих фаз. Начало первой фазы определяется моментом завершения первого из процессов, входящих в барьерную группу. По завершении последнего синхронизируемого процесса первая фаза считается законченной (барьер достигнут). Далее сразу же начинается вторая фаза синхронизации, заключающаяся в запуске (активизации) группы процессов, которые должны выполняться после данного барьера. Время окончания второй фазы определяется моментом запуска последнего из активизируемых процессов.

Барьерная синхронизация должна быть организована таким образом, чтобы выполнялись следующие условия:

$$T_{BS} \rightarrow \min; C_{BS} \rightarrow \max; E_{BS} \leq \bar{E}_{BS},$$

где T_{BS} – среднее время, затрачиваемое на синхронизацию; C_{BS} – комбинаторная гибкость средств синхронизации; E_{BS} – структурная сложность средств синхронизации (\bar{E}_{BS} – предельно допустимое значение).

Сущность предлагаемой процедуры барьерной синхронизации

Представим структуру матричного мультикомпьютера в виде графа $H = \langle M, U \rangle$, где M – множество вершин, соответствующее множеству процессорных модулей, а U – множество дуг, отражающее связи между ними. Для каждого модуля зададим порядковый номер (i, j) , $i = \overline{1, N_1}$, $j = \overline{1, N_2}$, где N_1 – число строк в матрице, а N_2 – число столбцов (считаем, что $N_1 > 1$, $N_2 > 1$ и в общем случае $N_1 \neq N_2$). Вершины графа H сопоставим множеству модулей так, что

$$\begin{cases} (m_{i,j}, m_{(i+1),j}) \in U, i = \overline{1, (N_1 - 1)}, j = \overline{1, N_2}; \\ (m_{i,j}, m_{i,(j+1)}) \in U, j = \overline{1, (N_2 - 1)}, i = \overline{1, N_1}, \end{cases}$$

где $m_{i,j}$ – модуль с номером (i, j) .

Основу предлагаемой процедуры составляют следующие принципы. Процесс синхронизации базируется на формировании и циклическом распространении координирующих сигналов, являющихся значениями двоичных индикаторных функций $Z_s(t) \in \{0, 1\}$, в множестве одноразрядных слоев синхронизации $\tilde{K}_u = \{K_1, K_2, \dots, K_n\}$, где n – максимально возможное количество параллельных барьеров. Каждый слой синхронизации K_l , $l = \overline{1, n}$ соответствует определенному барьеру v_s и представляет собой совокупность ячеек координирующей среды, распределенных между модулями и соединенных согласно матричной структуре.

Каждому модулю $m_{i,j}$ поставим в соответствие вектор $\Sigma^{i,j}(u) = (y_1^{i,j}(u), y_2^{i,j}(u), \dots, y_n^{i,j}(u))$, устанавливающий зависимость между модулями и барьерами u -й параллельной программы. Значение $y_s^{i,j}(u) = 0$, если $m_{i,j}$ выполняет ветвь программы u , завершающуюся барьером v_s , в противном случае $y_s^{i,j}(u) = 1$. Также введем признак завершения ветви $\zeta_e^s(t)$. Если $\zeta_e^s(t) = 0$, то $m_{i,j}$ еще не выполнил e -ю ветвь программы, завершающуюся барьером v_s , если же $\zeta_e^s(t) = 1$, то эта ветвь программы завершена.

Синхронизация протекает в три этапа. На первом этапе происходит выделение слоя координирующей среды, инициализация барьерной группы и формирование прямоугольной ограничивающей области (прямоугольника), внутри кото-

рой будет происходить распространение координирующих сигналов. На втором этапе синхронизации происходит циклический опрос и запуск модулей, входящих в барьерную группу. Третий этап синхронизации заключается в финализации барьерной группы и освобождении соответствующего слоя координирующей среды.

Каждому модулю поставим в соответствие признак его вхождения в ограничивающую область $k_s^{i,j}(t)$. Если $k_s^{i,j}(t) = 1$, то модуль находится в границах области, в противном случае он находится вне области и не принимает участия в синхронизации. Значение признака

$$k_s^{i,j}(t) = \begin{cases} 1 & \text{при } y_s^{i,j}(u) = 0; \\ (ht_s^{i,j}(t) \wedge hb_s^{i,j}(t)) \vee (ht_s^{i,j}(t) \wedge hr_s^{i,j}(t)) \vee \\ \vee (ht_s^{i,j}(t) \wedge hl_s^{i,j}(t)) \vee (hb_s^{i,j}(t) \wedge hr_s^{i,j}(t)) \vee \\ \vee (hb_s^{i,j}(t) \wedge hl_s^{i,j}(t)) \vee (hl_s^{i,j}(t) \wedge hr_s^{i,j}(t)) & \text{при } y_s^{i,j}(u) = 1. \end{cases}$$

Признак $k_s^{i,j}(t)$ принимает значение 1, если модуль выполняет ветвь программы u , завершающуюся барьером v_s , либо он получил любые два признака наличия соседних модулей. В целом процесс формирования ограничивающей области и установка признаков $k_s^{i,j}(t)$ происходят поэтапно, начиная с модулей, входящих в барьерную группу, а затем посредством распространения признаков наличия соседних модулей $ht_s^{i,j}(t)$, $hb_s^{i,j}(t)$, $hr_s^{i,j}(t)$, $hl_s^{i,j}(t)$. Это позволяет логически пометить не участвующие в синхронизации модули без каких-либо данных о размещении синхронизируемых ветвей.

Второй этап синхронизации начинается тогда, когда определяются значения $k_s^{i,j}(t)$ для всех модулей мультимпьютера. Опрос и запуск модулей организуются в виде последовательности циклов

$k_s^{i,j}(t)$ формируется на основании значений следующих признаков наличия соседних модулей: $ht_s^{i,j}(t)$ – для соседа сверху, $hb_s^{i,j}(t)$ – для соседа снизу, $hr_s^{i,j}(t)$ – для соседа справа, $hl_s^{i,j}(t)$ – для соседа слева. Указанные признаки вычисляются согласно следующим условиям:

$$\begin{aligned} ht_s^{i,j}(t) &= ht_s^{(i-1),j}(t) \vee k_s^{(i-1),j}(t); \\ hb_s^{i,j}(t) &= hb_s^{(i+1),j}(t) \vee k_s^{(i+1),j}(t); \\ hr_s^{i,j}(t) &= hr_s^{i,(j+1)}(t) \vee k_s^{i,(j+1)}(t); \\ hl_s^{i,j}(t) &= hl_s^{i,(j-1)}(t) \vee k_s^{i,(j-1)}(t), \end{aligned}$$

распространения координирующих сигналов в слое $K_s \in \bar{K}_u$. Каждый такой цикл включает две фазы: 1) опрос модулей о завершении ими ветвей программы и формирование признака синхронизации $Z_s = 1$; 2) запуск модулей по достижении барьера и восстановление исходного значения $Z_s = 0$. Распространение признака синхронизации происходит от модуля, находящегося в верхнем левом углу ограничивающей области, и до модуля, расположенного в ее нижнем правом углу. Обозначим эти угловые модули как m_{i^*,j^*} и $m_{i^{**},j^{**}}$ соответственно.

Выделим из множества M подмножество вершин $m_{i,j} \in M^*$, удовлетворяющее условию $(\forall i, j \in M^* : (i \geq i^*) \wedge (j \geq j^*) \wedge (i \leq i^{**}) \wedge (j \leq j^{**}))$. Такое подмножество M^* будет определять область структуры мультимпьютера, внутри которой про-

исходит распространение координирующих сигналов.

Индикаторную функцию $Z_s(t)$ s -го барьера определим в виде $Z_s(t) = \bigwedge_{m_{i,j} \in M^*} z_s^{i,j}(t)$, где $z_s^{i,j}(t) = y_s^{i,j}(t) \vee \zeta_e^s(t)$ – текущее состояние процессора. Для определения значения функции $Z_s(t)$ на выходе каждого модуля введем понятие большего и меньшего порядкового номера.

Будем считать, что модуль m_{i_1,j_1} имеет меньший порядковый номер, чем m_{i_2,j_2} , т.е. $m_{i_1,j_1} < m_{i_2,j_2}$, если $(i_1 \leq i_2) \wedge (j_1 \leq j_2) \wedge ((i_1 \neq i_2) \vee (j_1 \neq j_2))$. Аналогично будем считать, что модуль m_{i_1,j_1} имеет больший порядковый номер, чем m_{i_2,j_2} , т.е. $m_{i_1,j_1} > m_{i_2,j_2}$, если $(i_1 \geq i_2) \wedge (j_1 \geq j_2) \wedge ((i_1 \neq i_2) \vee (j_1 \neq j_2))$.

Тогда значение функции $Z_s(t)$ на выходе (i, j) -й ячейки слоя K_s для модуля $m_{i,j}$ будет равно 1, если этот модуль завершил свою ветвь программы (или не участвует в s -м барьере) и все модули с меньшими порядковыми номерами к этому моменту также завершили ветви программы, относящиеся к s -му барьеру (или не участвуют в нем). Соответственно, момент времени t' завершения всех ветвей барьера v_s задается условием

$$k_s^{i,j}(t) = \begin{cases} 0 & \text{при } y_s^{i,j}(u) = 0; \\ (ht_s^{i,j}(t) \wedge hb_s^{i,j}(t) \wedge hr_s^{i,j}(t) \wedge hl_s^{i,j}(t)) & \text{при } y_s^{i,j}(u) = 1. \end{cases}$$

Таким образом, признак вхождения $k_s^{i,j}(t)$ сбрасывается, если модуль $m_{i,j}$ выполняет ветвь программы u , завершающуюся барьером v_s , либо хотя бы один из признаков наличия соседних модулей для $m_{i,j}$ равен 0.

$Z_s(t') = Z_s^{i^*,j^*}(t') = 0 \rightarrow 1$ (условие окончания первой фазы синхронизации).

Значение функции $Z_s(t)$ на выходе (i, j) -й ячейки слоя K_s для модуля $m_{i,j}$ не зависит от его текущего состояния и равно 1, только если на выходах модулей с большими порядковыми номерами уже установлено единичное значение.

Условие окончания второй фазы синхронизации:

$$Z_s(t'') = Z_s^{i^*,j^*}(t'') = 1 \rightarrow 0.$$

На третьем этапе синхронизации путем обнуления признаков $k_s^{i,j}(t)$ всех модулей и признаков $ht_s^{i,j}(t)$, $hb_s^{i,j}(t)$, $hr_s^{i,j}(t)$, $hl_s^{i,j}(t)$ происходит логическое удаление ограничивающей области. Сброс признаков наличия соседних модулей происходит согласно выражениям:

$$\begin{aligned} ht_s^{i,j}(t) &= ht_s^{(i-1),j}(t) \wedge k_s^{(i-1),j}(t); \\ hb_s^{i,j}(t) &= hb_s^{(i+1),j}(t) \wedge k_s^{(i+1),j}(t); \\ hr_s^{i,j}(t) &= hr_s^{i,(j+1)}(t) \wedge k_s^{i,(j+1)}(t); \\ hl_s^{i,j}(t) &= hl_s^{i,(j-1)}(t) \wedge k_s^{i,(j-1)}(t). \end{aligned}$$

Признаки $k_s^{i,j}(t)$ обнуляются согласно следующим условиям:

Исследование динамической корректности процедуры

Для оценки корректности функционирования разработанной процедуры синхронизации использовался аппарат сетей Петри [10].

Исследование динамики формирования ограничивающей области на первом этапе синхронизации осуществлялось с

использованием сети Петри $\Theta_{i,j}$, изображенной на рис. 1 и включающей ряд подсетей.

Подсети $\Lambda_{i,j}^h$, $h = \{t, b, r, l\}$ моделируют получение и распространение признаков $ht_s^{i,j}(t)$, $hb_s^{i,j}(t)$, $hr_s^{i,j}(t)$, $hl_s^{i,j}(t)$. Подсеть $\Lambda_{i,j}^k$ отображает формирование признака вхождения модуля в ограничивающую область $k_s^{i,j}(t)$. Соединенные между собой согласно двумерной матричной топологии сети Петри $\Theta_{i,j}$ образуют объединенную сеть Θ , моделирующую формирование ограничивающей области в отдельном слое координирующей среды на первом этапе синхронизации.

Динамика функционирования отдельной ячейки координирующей среды на втором этапе синхронизации описывается подсетями Петри $\Lambda'_{i,j}$, $\Lambda''_{i,j}$, приведенными на рис. 2. Данные подсети моделируют соответственно функционирование ячейки в первой и второй фазах синхронизации на втором этапе.

Переход $\pi_{i,j}^-$ отображает завершение выполнения ветви программы модулем

$m_{i,j}$; $\pi_{i,j}^+$ – запуск модуля по завершении барьера; $\bar{\pi}_{i,j}$ и $\pi_{i,j}$ – прохождение сигнала синхронизации через ячейку координирующей среды. Соединение подсетей $\Lambda'_{i,j}$ и $\Lambda''_{i,j}$ согласно двумерной матричной топологии приводит к получению двух подсетей: Λ' и Λ'' . Подсеть Λ' отражает динамику формирования признака синхронизации внутри ограничивающей области, а подсеть Λ'' – передачу данного признака и запуск модулей. Соединенные между собой подсети Λ' и Λ'' образуют сеть Λ , моделирующую поведение отдельного слоя координирующей среды на втором этапе синхронизации.

Динамика функционирования ячейки координирующей среды на третьем этапе синхронизации определяется сетью Петри $\Xi_{i,j}$, которая показана на рис. 3. Соединенные между собой согласно двумерной матричной топологии сети $\Xi_{i,j}$ образуют сеть Петри Ξ , моделирующую функционирование слоя координирующей среды на третьем этапе синхронизации.

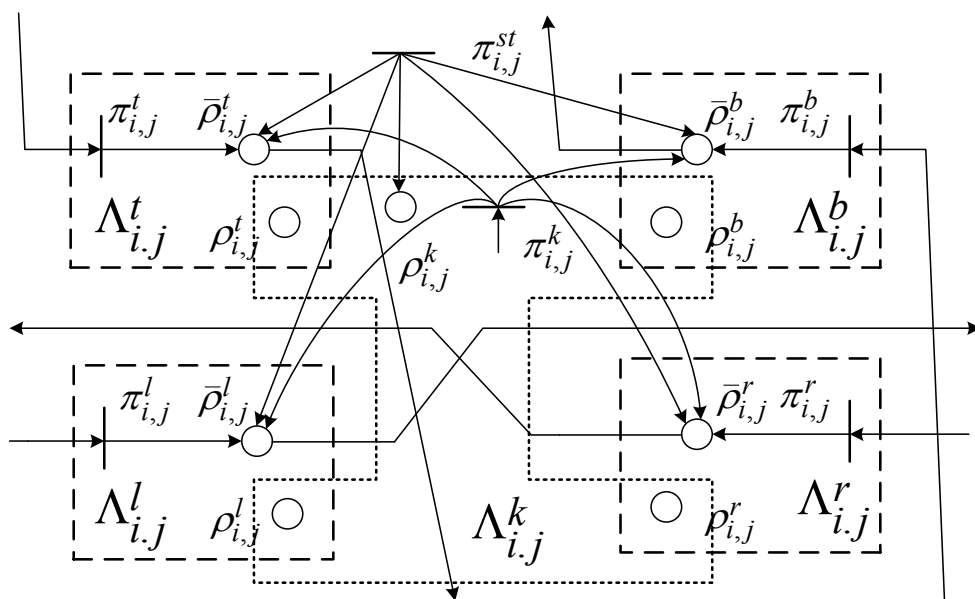
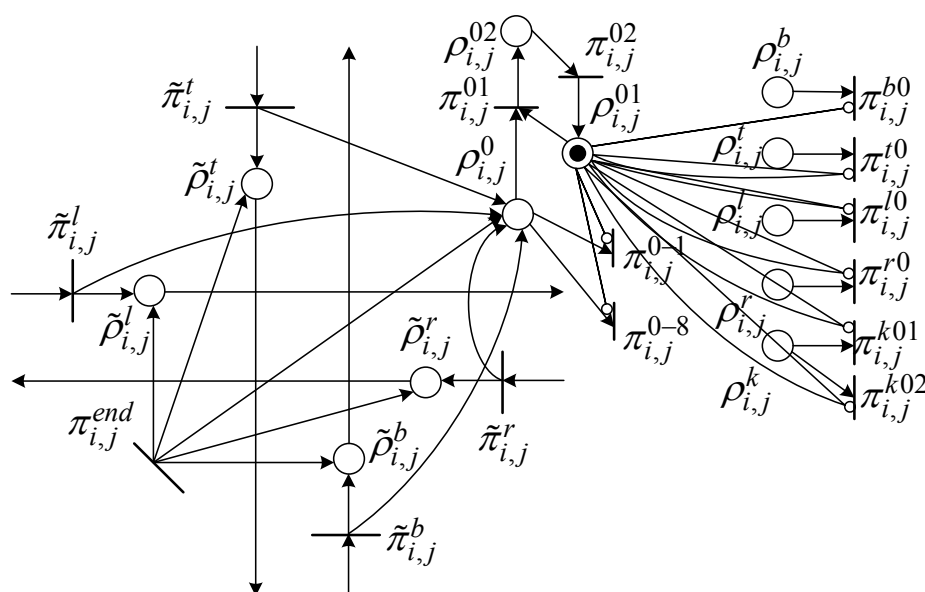


Рис. 1. Сеть Петри $\Theta_{i,j}$

Рис. 2. Подсеть $\Lambda'_{i,j}$ (а); подсеть $\Lambda''_{i,j}$ (б)Рис. 3. Сеть Петри $\Xi_{i,j}$

Анализ достижимых маркировок сетей Петри Θ , Λ , Ξ показал, что обладают свойством живости, а максимальная емкость позиций равна 3, 1 и 8, соответственно. Это доказывает отсутствие тупиковых ситуаций как при построении/удалении области распространения координирующих сигналов, так и при опросе и запуске процессорных модулей.

Работа выполнена при поддержке гранта Президента РФ МД-2218.2011.8.

Список литературы

1. TILE-Gx 3036 [Электронный ресурс] // Tiler Corporation, 2012. URL: <http://www.tiler.com/sites/default/files/productbriefs/Tile-Gx%203036%20SB012-01.pdf> (дата обращения: 15.09.2012).
2. Panda D.K. Fast Barrier Synchronization in Wormhole k-ary n-cube Networks // First IEEE Symposium on High Performance Computer Architecture, 1995. P. 200–209.
3. Four-ary tree-based barrier synchronization for 2d meshes without nonmember involvement / S. Moh, C. Yu, B. Lee [et al.] // IEEE Trans. Comput. 2001. Vol. 50, № 8. P. 811–823.
4. Efficient barrier and allreduce on InfiniBand clusters using hardware multicast and adaptive algorithms [Текст] / A.R. Mamidala, J. Liu, D.K. Panda // Proceedings of the 2004 IEEE International Conference on Cluster Computing, 2004. P. 135–144.
5. Efficient synchronization for embedded on-chip multiprocessors / M. Monchic-

ro, G. Palermo, C. Silvano, O. Villa // IEEE Transactions on very large scale integration (VLSI) systems. 2006. Vol. 14, № 10. P. 1049–1062.

6. Аль-Хади А.М. Зотов И.В., Волобуев С.В. Методы распределенной барьерной синхронизации параллельных процессов в матричных СБИС-системах // Нейрокомпьютеры: разработка, применение. 2009. № 12. С. 46–52.

7. Волобуев С.В., Зотов И.В. Распределенный механизм барьерной синхронизации на основе параллельно-конвейерной координирующей среды // Информационно-измерительные и управляющие системы. 2010. №7. С. 35–39.

8. Zotov I.V. Distributed Virtual Bit-Slice Synchronizer: A Scalable Hardware Barrier Mechanism for n-Dimensional Meshes // IEEE Trans. Comput. 2010. Vol. 59, № 9. P. 1187–1199.

9. Аль-Хади А.М., Муратов С.А., Зотов И.В. Распределенная процедура барьерной синхронизации с динамическим ограничением области распространения координирующих сигналов // Телекоммуникации. 2011. № 5. С. 2–7.

10. Котов В.Е. Сети Петри. М.: Наука, Гл. ред. физ.-мат. лит., 1984. 160 с.

Получено 26.04.12.

R.V. Bredikhin, Post-Graduate Student, Southwest State University (Kursk)

I.V. Zotov, Doctor of Sciences, Professor, Southwest State University (Kursk)
(e-mail: zotovigor@yandex.ru)

A DISTRIBUTED ITERATIVE BARRIER SYNCHRONIZATION PROCEDURE FOR 2D-MESH MULTICOMPUTER-ON-A-CHIP SYSTEMS

A hardware-level procedure representing the barrier synchronization in iterative parallel programs performed by 2D-mesh multicomputer-on-a-chip systems is presented, the correctness of dynamic behavior of the procedure is investigated based on the Petri nets theory.

Key words: multicomputer-on-a-chip, VLSI, parallel processes, barrier synchronization, Petri nets.

УДК 004.932

П.С. Козлов, аспирант, Юго-Западный государственный университет (Курск)
(e-mail: ps.kozlov@gmail.com)

М.И. Труфанов, доцент, Юго-Западный государственный университет (Курск)
(e-mail: temp1202@mail.ru)

СПЕЦИАЛИЗИРОВАННОЕ ОПТИКО-ЭЛЕКТРОННОЕ УСТРОЙСТВО ВЫЧИСЛЕНИЯ ПОВЕРХНОСТНОГО НАТЯЖЕНИЯ ЖИДКОСТИ

Представлена структурно-функциональная организация и принципы функционирования оптико-электронного устройства для вычисления поверхностного натяжения жидкости посредством анализа формы сформированного в исследуемой жидкости пузырька. Получены экспериментальные результаты, отражающие время обработки данных и относительную ошибку вычисления поверхностного натяжения жидкости.

Ключевые слова: распознавание изображений, поверхностное натяжение, форма, оптические искажения, калибровка.

Наблюдающееся в настоящее время революционное развитие современных вычислительных средств обработки цифровых данных обуславливает их внедрение в области промышленности, где традиционно их использование не было пре-

дусмотрено, что позволяет в конечном итоге не только исключить или значительно снизить человеческий труд, но и многократно повысить эффективность технологических процессов, определяе-