

## ПАРАЛЛЕЛЬНАЯ ОТКАЗОУСТОЙЧИВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ В ТРАНСПЬЮТЕРНОЙ СИСТЕМЕ МИКРОС-Т

© 1995, М.С. Тарков

*Институт физики полупроводников СО РАН, Новосибирск*

Рассматривается архитектура и программное обеспечение распределительной системы МИКРОС-Т, разработанной для обработки изображений.

### Введение

Цель обработки изображений (ОИ) заключается в том, чтобы из множества объектов, движущихся в наблюдаемой области, выделить объекты, входящие в множество существенных, проанализировать определенные свойства выделенных объектов, а также выдать необходимые сигналы и данные системе управления исполнительными устройствами. Для достижения этой цели следует выполнить обработку изображения и вычисления управляющих воздействий в интервале времени, заданном от момента появления объекта, причем поведение наблюдаемых объектов может быть непредсказуемо. Кроме того, несовершенство датчиков изображения, наличие естественных и искусственных шумов, теней и бликов, относительные перемещения наблюдаемых объектов и средств наблюдения, изменения ракурсов и другие причины усложняют задачу обработки изображения настолько, что требуемая производительность системы ОИ, по некоторым оценкам [1], составляет до  $10^{12}$  операций в секунду. Это обусловило необходимость параллельной реализации процессов обработки.

Немаловажным для систем ОИ, в особенности для необслуживаемых систем бортового исполнения, является требование обеспечения их отказоустойчивого функционирования. Оно может быть удовлетворено в системах с гибкой архитектурой, способных перестраиваться при отказах.

### Вычислительная система МИКРОС-Т

Распределенная система МИКРОС-Т – экспериментальная реализация концепции вычислительных систем (ВС) с программируемой структурой [2-6] и

является продолжением ряда экспериментальных микромашинных систем МИКРОС: МИКРОС (1986 г.), МИКРОС-2 (1992 г.), МИКРОС-Т (1994 г.).

В основу систем ряда МИКРОС положены следующие архитектурные принципы, позволяющие достичь высокой производительности, надежности и живучести:

- параллелизм (процессов управления, доступа к данным и их обработки);
- децентрализация средств управления и распределенность ресурсов;
- программируемость (программная настраиваемость) структуры;
- модульность и конструктивная однородность;
- асинхронность взаимодействий модулей и процессов;
- статическая и динамическая реконфигурируемость системы.

Система с программируемой структурой – совокупность вычислительных модулей – элементарных машин (ЭМ), объединенных программно-настраиваемой сетью связи. Каждая ЭМ ВС семейства МИКРОС включает в себя микроЭВМ и системное устройство, осуществляющее функции программно-управляемого из микроЭВМ модуля распределенного коммутатора сети связи ВС.

Принцип программируемости структуры выражается в программной компоновке связанной совокупности ЭМ, образующих подсистему типа, требуемого для решения задачи. Эффективная организация параллельных вычислений на ВС с программируемой структурой основывается на использовании крупноблочного подхода к распараллеливанию и конвейерно-параллельной асинхронной организации межмодульных обменов.

В качестве ЭМ системы МИКРОС-Т использован высокопроизводительный микропроцессорный элемент – транспьютер. Среди серийно

выпускаемых микропроцессоров транспьютер с наибольшей полнотой обеспечивает реализацию архитектурных принципов ВС с программируемой структурой.

Структурно систему ОИ можно разделить на три составляющие: транспьютерная ВС, средства ввода изображения в ВС, средства вывода и визуализации результатов обработки. Общение пользователя с транспьютерной ВС выполняется с помощью терминала, в качестве которого используется микро ЭВМ IBM PC.

Действующий вариант системы МИКРОС-Т включает в себя восемь ЭМ (транспьютеров) (рис.1).

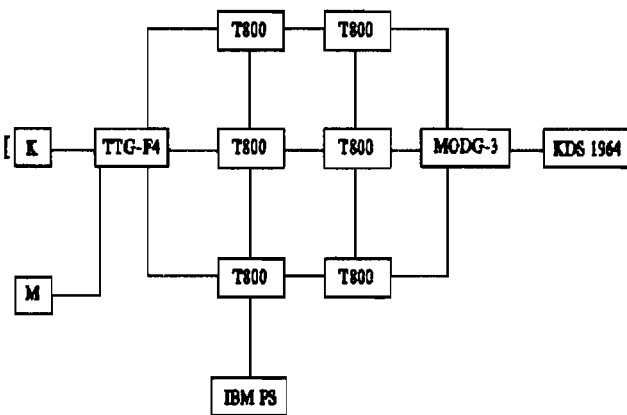


Рис. 1

Изображение вводится в систему с помощью камеры (К) на основе ПЗС матрицы. Визуальный контроль вводимого изображения осуществляется посредством телевизионного монитора (М). Камера и монитор подключены к транспьютерному модулю Фреймграббера (модулю фиксации кадра) TTG-F4, реализующему две функции: ввод изображения в видеопамять и выполнение вычислений. Ввод изображения из фреймграббера в соседние транспьютеры осуществляется параллельно. Обработанное изображение выводится через графический транспьютерный модуль MODG-3 на монитор KDS 1964. Все ЭМ реализованы на транспьютерах типа INMOS T800 и INMOS T805. Каждая ЭМ содержит компоненту децентрализованной распределенной операционной системы (ОС) МИКРОС-Т, способной динамически настраиваться на произвольную топологию ВС. Вследствие этого ВС может: 1) неограниченно наращиваться; 2) функционировать при произвольных кратных отказах ЭМ и линков, если отказы не нарушают связности структуры ВС.

### Операционная система ВС МИКРОС-Т

Широко применяемые в транспьютерных ВС средства параллельного программирования фирмы 3L [7] и операционная система (ОС) Helios [8] имеют ряд недостатков. Одним из наиболее существенных среди них является необходимость предварительного описания конфигурации вычислительной системы и структуры параллельной программы. При конфигурировании параллельной программы процессам и информационным связям между ними ставятся в однозначное соответствие их физические носители: транспьютеры и линии связи. Подобный статический подход к конфигурированию исключает возможность оперативной автоматической перенастройки системы в случае отказов ее элементов.

Реализация режима отказоустойчивой параллельной обработки изображений требует исключения необходимости конфигурирования параллельных программ в явном виде, введения механизмов слежения за состоянием элементов системы, разработки средств реконфигурации при обнаружении отказов.

Децентрализованная распределенная ОС ВС семейства МИКРОС [9] способна функционировать на ВС произвольной конфигурации и создает в каждой ЭМ окружение, позволяющее осуществлять динамическую настройку параллельной программы на произвольную конфигурацию подсистемы. Число ЭМ в подсистеме также может быть произвольным. Децентрализованные процедуры маршрутизации обеспечивают передачу сообщений между любыми ЭМ подсистемы. Указанные свойства ОС ВС МИКРОС являются основой для организации живучего функционирования ВС и отказоустойчивого выполнения параллельных программ.

Операционная система ВС МИКРОС-Т имеет иерархическую структуру (рис.2). Каждый уровень ОС строится на основе предыдущих.

Нижний (нулевой) уровень ОС включает средства инициации ВС, приводящие все ЭМ ВС в исходное состояние и средства управления процессами (ядро ОС) ЭМ. Программные средства управления процессами расширяют возможности аппаратных средств транспьютера. Так, например, если планировщик транспьютера задает только два уровня приоритета процессов, то программные средства позволяют сделать число уровней приоритета произвольным. Для организации взаимодействий между процессами введена расширенная концепция семафора [10], которая позволяет связать операции над значениями

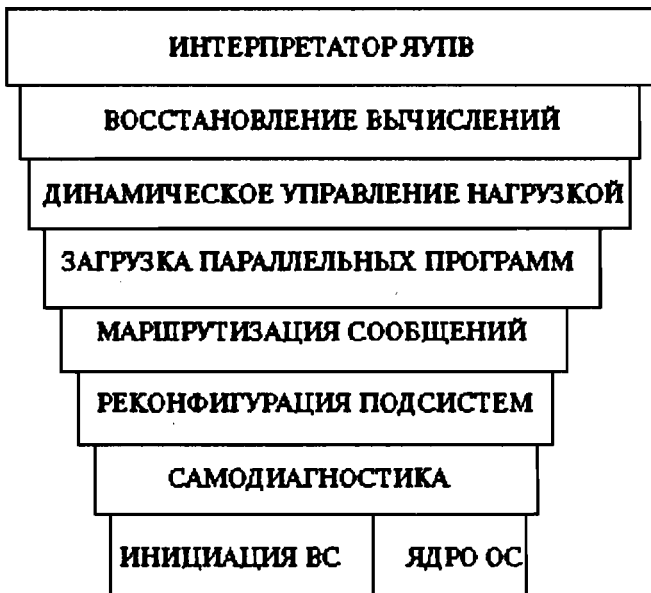


Рис. 2

семафоров с операциями над составными объектами, управляемыми этими семафорами. В результате упрощается структура ядра ОС и сокращается его объем.

Уровень 1 ОС образован средствами самоконтроля и самодиагностики ВС, обеспечивающими определение состояния ВС в условиях кратных отказов. Заключение об исправности или неисправности отдельных ЭМ принимается коллективно всеми ЭМ системы на основе сопоставления результатов взаимопроверки ЭМ [11]. Самоконтроль и самодиагностика реализуются как система параллельных процессов, выполняемая всеми исправными ЭМ системы.

Средства построения и реконфигурации виртуальных подсистем (уровень 2), используя информацию о состоянии ЭМ, полученную средствами самодиагностики, формируют в каждой ЭМ окружение, содержащее описание положения данной ЭМ в системе относительно других ЭМ. Параметры окружения используются средствами маршрутизации сообщений в подсистеме (уровень 3). Уровень 4 ОС составляют средства загрузки параллельных программ в подсистему. Уровни 0, ..., 4 (за исключением программы инициации) входят в состав резидентной части ОС каждой ЭМ.

Средства динамического управления нагрузкой ЭМ (уровень 5), перераспределяют программы и данные между ЭМ подсистемы по завершении ее реконфигурации. На уровне 6 (восстановление вы-

числений) выполняются операции, связанные с перезапуском процесса вычислений с заданной точки возврата. Перезапуск осуществляется после перераспределения программ и данных. Уровни 5 и 6 ОС могут быть включены либо в резидентную часть ОС, либо в загрузочный модуль параллельной программы пользователя в зависимости от выбора методов управления нагрузкой и восстановления вычислений, которые могут быть как универсальными, так и специфичными для конкретной области применения ВС.

Интерпретатор языка управления параллельными вычислениями (уровень 7) по командам с пультавого терминала порождает процессы, осуществляющие: порождение подсистемы требуемого типа ("дерево", "линейка", "кольцо" и т.п.) из требуемого числа работоспособных ЭМ; загрузку параллельной программы в сформированную подсистему и инициацию ее выполнения. Интерпретатор загружается в транспьютер, непосредственно связанный с терминалом.

### Языки параллельного программирования ВС МИКРОС-Т

Языки параллельного программирования Си-С и Фортран-С построены путем расширения традиционных языков Си и Фортран примитивами организации межмашинных взаимодействий (передачи сообщений) и примитивами оценки параметров подсистемы, на которой выполняется параллельная программа. Первые позволяют организовать взаимодействия между любыми ветвями программы, вторые дают возможность использовать параметры подсистемы для адаптации программы к текущей конфигурации последней. Это свойство существенно как с точки зрения простоты организации вычислительного эксперимента, так и с точки зрения организации отказоустойчивых параллельных вычислений. Реализация данных примитивов основывается на средствах ОС ВС МИКРОС.

Так, например, для организации конвейерно-параллельных обменов используются примитивы:

– `send(pole, message, size)` – для передачи массива `message` из `size` байт в соседнюю ЭМ по направлению `pole`;

– `receive(pole, message, size)` – для приема массива `message` из `size` байт из соседней ЭМ по направлению `pole`.

В отличие от примитивов обмена Оккама и подобных ему языков (например, Parallel C фирмы 3L) примитивы `send` и `receive` реализуют межма-

шинные обмены асинхронно, что упрощает программирование параллельных вычислений и уменьшает время пребывания процессов в состоянии ожидания.

Дифференцированные обмены реализуются примитивами:

– send\_diff(number, message, size) – для передачи массива message из size байт в ЭМ с номером number;

– receive\_diff(number, message, size) – для приема массива message из size байт из ЭМ с номером number.

Трансляционные обмены реализуются примитивами:

– send\_all(message, size) – для передачи массива message из size байт во все ЭМ подсистемы;

– receive\_all(number, message, size) – для приема массива message из size байт; number получает значение, равное номеру передающей ЭМ.

Примером примитива, оценивающего параметры подсистемы, может служить sys(E,P) [12], определяющий номер E данной ЭМ в подсистеме и число P машин в подсистеме. Используя параметры E и P, ветвь параллельной программы выделяет соответствующий номеру E фрагмент обрабатываемого программой массива данных. Фрагмент выделяется путем вычисления граничных значений индексов цикла, в котором происходит обработка массива, через размер массива и значения параметров E и P. Другими словами, программа настраивается на число P машин подсистемы, т. е. инвариантна к P.

### Распараллеливание алгоритмов обработки изображений

Целью предварительной ОИ является улучшение качества изображения и сокращение объема информации, представленной на изображении, до требуемого уровня. В результате выполнения предварительной ОИ происходит: 1) удаление импульсных помех (сглаживание изображения); 2) усиление контраста; 3) выделение контуров объектов.

В настоящее время широкое распространение получили локальные алгоритмы предварительной ОИ [13]. В этих алгоритмах в вычислении значения яркости любого пиксела участвует его окрестность, выделяемая пространственным окном, форма которого в общем случае может быть достаточно произвольной, а размеры по сравнению с размерами кадра малы. Пространственная локализация обрабатываемых данных уменьшает объем вычислений и упрощает их распараллеливание.

Кроме того, такие алгоритмы не нуждаются в использовании априорных статистических характеристик неискаженного изображения и шума. Указанные преимущества обусловили выбор локальных алгоритмов для моделирования предварительной ОИ.

Пусть  $F=(f(i,j))$ ,  $i,j=1,2,\dots,N$  – матрица исходного изображения, подвергающегося обработке, а  $G=(g(i,j))$ ,  $i,j=1,2,\dots,N$  – матрица преобразованного изображения. Значительная часть преобразований описывается сверткой  $H \times F$  функции яркости  $f(i,j)$  с совокупностью весовых коэффициентов оператора

$$H = (h(k,l,i,j)), \quad k,l=-M,\dots,0,\dots,M, \quad i,j=1,\dots,N;$$

$$g(i,j) = \sum_{k=-M}^M \sum_{l=-M}^M h(k,l,i,j) * f(k+i,l+j). \quad (1)$$

В (1) используется квадратное окно размером  $((2M+1) \times (2M+1))$ ,  $M=1,2,\dots,M \ll T$ . Весовые коэффициенты окна образуют маску оператора.

Локальность алгоритмов предварительной ОИ обуславливает простоту метода их распараллеливания, суть которого заключается в следующем [14]. Двумерный массив изображения, состоящий из  $N \times N$  пикселей, разбивается по строкам на P равных сегментов, где P – число ЭМ в подсистеме, обрабатывающей изображение. В машине ЭМ<sub>i</sub>,  $i=1,2,\dots,P-2$ , обрабатываются строки массива изображения с номерами  $(i \cdot N/P) - M, \dots, M + (i+1)N/P$ , где M – число, характеризующее размер маски локального оператора, т.е. сегменты массива, обрабатываемые машинами ЭМ<sub>i</sub> и ЭМ<sub>i+1</sub>,  $i=0,1,\dots,P-2$ , частично перекрываются (имеют общие строки). Благодаря такому способу распределения строк массива его сегменты обрабатываются независимо (параллельно).

Контурное изображение используется для идентификации на нем объектов в соответствии с заданными эталонами. Обнаружение объекта происходит независимо от его ориентации и масштаба посредством быстрого обобщенного преобразования Хафа (БОПХ) [15]. Сопоставление кадра с эталоном происходит многократно при заданных дискретах изменения угла поворота и масштаба эталона.

Вычисления распараллеливаются по углу поворота эталона, [16] т.е. ЭМ<sub>i</sub>,  $i=0,1,\dots,P-1$ , вычисляет БОПХ в интервале углов  $[i \cdot 2\pi/P, (i+1) \cdot 2\pi/P]$  [11]. При этом в каждой ЭМ обрабатывается весь заданный интервал изменения масштаба эталона.

### Организация отказоустойчивой обработки изображений

Суть предлагаемого подхода к организации отказоустойчивой обработки изображений (ОИ) заключается в использовании системы взаимодействующих процессов со структурой типа "дерево", предложенной в [3] для решения некоторых задач линейной алгебры. Обоснованием такой организации может служить следующее: 1) виртуальная подсистема типа "дерево" может быть реализована на произвольном связном подмножестве ЭМ; 2) децентрализованный алгоритм построения дерева, покрывающего ВС, прост и не является переборным, что существенно при формировании и реконфигурации подсистем в реальном масштабе времени. Дерево удобно также использовать при организации трансляционных ("один – всем") и коллекторных ("все – одному") схем межтранспьютерных обменов информацией, характерных для операций обработки и параллельного ввода/вывода изображений.

Межмашинные взаимодействия в системе ОИ представлены в виде четырех подсистем типа "дерево": 1) подсистема управления состоянием ВС, предназначенная для моделирования отказов ЭМ и межмашинных линий связи и имеющая корнем ЭМ, непосредственно связанную с микроЭВМ IBM PC, используемой в качестве интеллектуального терминала; 2) подсистема обработки, изоморфная подсистеме управления; 3) подсистема ввода, имеющая корнем фрейм-граббер TTG-F4; 4) подсистема вывода, использующая коллекторный тип обмена и имеющая корнем транспьютерный модуль MODG-3.

Обнаружение отказа (соседней ЭМ или линии связи с ней) происходит при попытке передать пакет сообщения. Если подтверждение о приеме пакета не получено за заданный интервал времени, линия связи считается отказавшей. Отслеживание времени передачи пакета осуществляется специальным ("сторожевым") процессом. Любой отказ приводит к инициации процесса реконфигурации подсистем. На время реконфигурации процессы обработки изображений и маршрутизации сообщений блокируются. По окончании реконфигурации блокировка указанных процессов снимается и каждая ветвь параллельной программы обработки изображений настраивается на номер  $E$  содержащей ее ЭМ и число  $P$  ЭМ в реконфигурированной подсистеме ввода, обработки или вывода изображений. На рис.3а представлено нисходящее дерево подсистемы ввода до возникновения отказа

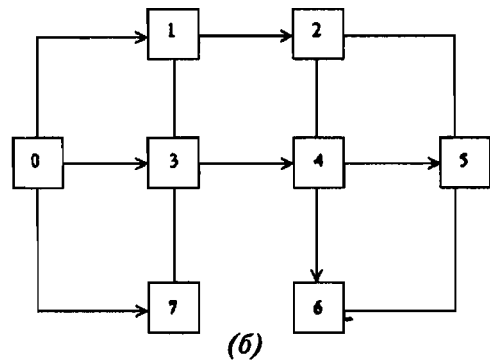
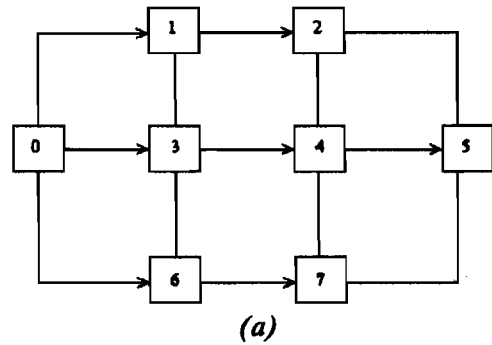


Рис.3

линка между ЭМ<sub>6</sub> и ЭМ<sub>7</sub> (корневая ЭМ<sub>0</sub> – фреймграббер). Стрелками показаны направления передачи фрагментов кадра между ЭМ. После реконфигурации подсистема ввода имеет вид рис.3б. В реконфигурированной подсистеме используется новый линк (между ЭМ<sub>4</sub> и ЭМ<sub>6</sub>).

Обработка изображения состоит из трех этапов: 1) ввод кадра, 2) предварительная обработка и вывод обработанного кадра, 3) селекция и идентификация искомого объекта и вывод его координат (рис.4). Каждый из этих этапов предваряется настройкой программы на соответствующую подсистему. Информацию о номере  $E$  данной ЭМ и числе  $P$  машин в подсистеме цвета (типа)  $C$  программа получает посредством примитива  $csys(C,E,P)$ . Отказ соседней ЭМ или линии связи с ней программа определяет по значению флага  $F$ , устанавливаемого примитивами межмашинного обмена данными. Эти примитивы аналогичны вышеуказанным, но используют дополнительные параметры  $C$  и  $F$ . Например, примитивы  $csend\_diff(F,C,I,M,S)$  и  $creceive\_diff(F,C,I,M,S)$  используются для передачи и приема массива  $M$  размера  $S$  в/из ЭМ с номером  $I$  в подсистеме  $C$ . При  $F>0$  происходит возврат к началу выполнения этапа и посредством  $csys(F,C,E,P)$  иницируется

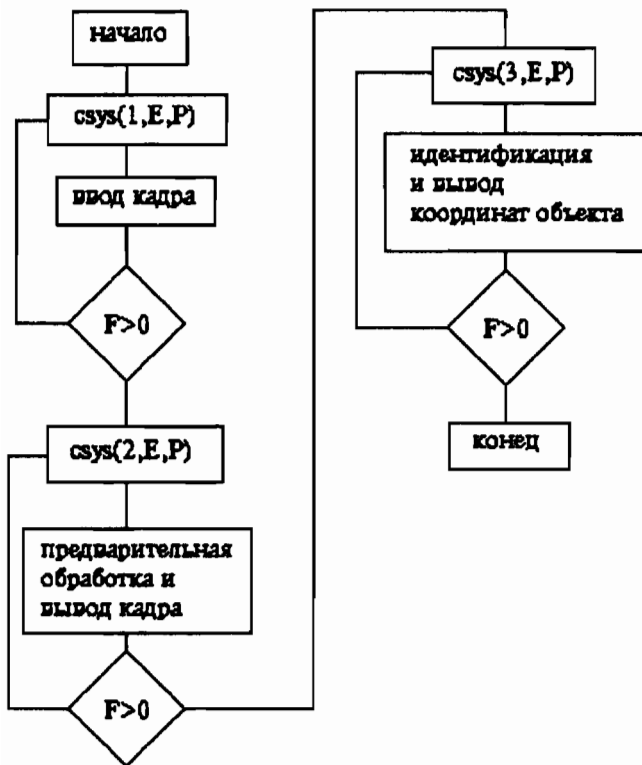


Рис. 4

реконфигурация подсистемы. Распространение по системе сигнала об обнаруженном отказе реализуется волновым алгоритмом. По окончании реконфигурации  $F$  обнуляется, заново определяются параметры  $E$  и  $P$  и вычисления возобновляются. При этом система переходит к обработке следующего кадра.

### Заключение

Описаны архитектура и децентрализованная ОС распределенной живучей транспьютерной системы ОИ МИКРОС-Т. Разработаны языки, позволяющие создавать отказоустойчивые параллельные программы, инвариантные к числу исправных ЭМ в системе. Представлены методы, обеспечивающие эффективное распараллеливание локальных алгоритмов предварительной ОИ и алгоритмов идентификации объектов по геометрическим признакам. Предложен подход к организации отказоустойчивой ОИ в живучих ВС. Предложенный подход апробирован на действующем макете системы ОИ из восьми транспьютерных модулей и показал устойчивость параллельных программ ОИ к отказам произвольных транспьютеров и межтранс-

пьютерных линий связи, не нарушающим связность структуры системы. Установлена практически линейная зависимость ускорения вычислений от числа исправных транспьютеров.

### Литература

1. *Weems C. C.* Architectural Requirements of Image Understanding with Respect to Parallel Processing // Proc. of IEEE.-v.79,N4. -1991.- P.537-547.
2. *Димитриев Ю.К., Хорошевский В.Г.* Вычислительные системы из мини-ЭВМ. - М.: Радио и связь, 1982.- 304 с.
3. *Корнеев В.В.* Архитектура вычислительных систем с программируемой структурой.- Новосибирск: Наука.-1985.-196 с.
4. *Хорошевский В.Г.* Инженерный анализ функционирования вычислительных машин и систем.- М.: Радио и связь, 1987.- 256 с.
5. *Хорошевский В.Г., Димитриев Ю.К., Задорожный А.Ф., Тарков М.С.* Вычислительная система с программируемой структурой МИКРОС-2 // Электронное моделирование.-1995.-N1.
6. *Хорошевский В.Г., Димитриев Ю.К., Задорожный А.Ф., Мелентьев В.А., Тарков М.С.* Распределенная система обработки изображений // Материалы международной научно-технической конференции "Информатика и проблемы телекоммуникаций", т.1.- Новосибирск, 1995.- с.120-122.
7. 3L Parallel C. User Guide (v.2.0).3L Ltd, Livingston, 1988.
8. The HELIOS operating system. Prentice-Hall, London, 1989.
9. *Корнеев В.В., Тарков М.С.* Операционная система микромашинной системы с программируемой структурой МИКРОС// Микропроцессорные средства и системы.-1988.-N4. с.41-44.
10. *Корнеев В.В., Монахов О.Г., Тарков М.С.* Ядро операционной системы ЭМ вычислительной системы с программируемой структурой// Вычислительные системы.-Новосибирск, 1981.— Вып.90. с.22-42.
11. *Димитриев Ю.К.* Самодиагностика модульных вычислительных систем.-Новосибирск: ВО "Наука", 1993.-293 с.
12. *Тарков М.С.* Об организации параллельных вычислений в вычислительной системе с программируемой структурой МИКРОС// Организация вычислений на супер-ЭВМ.- Деп. в ВИНТИ 25.03.87, N2184-887. с.32-38
13. *Бакут П.А., Колмогоров Г.С.* Сегментация изображений: методы выделения границ областей // Зарубежная электроника.-1987.-т.10 -С.25-47.

14. *Мелентьев В.А., Тарков М.С. и др.* Моделирование параллельной обработки изображений на базе вычислительной системы МИКРОС-2// Электронное моделирование. 1995, N1.

15. *Jeng S.-C., Tsai W.-H.* Fast Generalized Hough Transform//Pattern Recognition Letters.-1990.-N11.- P.725-733.

16. *Тарков М.С., Снижневская Е. П.* Параллельная обработка изображений на транспьютерной системе с использованием преобразования Хафа//Материалы международной научно-технической конференции "Информатика и проблемы телекоммуникаций", т.1.- Новосибирск, 1995.- С.132-134.

## FAULT-TOLERANT PARALLEL IMAGE PROCESSING IN THE MIKROS-T TRANSPUTER SYSTEM

M.S. Tarkov

The paper describes the architecture and software of the distribution system MIKROS-T designed for image processing.