ЗО

УДК 004.27: 004.052.3

А.С. Окунев, Е.С. Градов, Н.Н. Левченко
Институт проблем информатики РАН, г. Москва, Россия
oku@bur.oivta.ru, gradov@bur.oivta.ru, nick@bur.oivta.ru

Исследование вариантов обеспечения отказоустойчивого функционирования вычислительной системы с автоматическим распределением ресурсов

В данной статье дано краткое описание вычислительной системы с автоматическим распределением ресурсов. Анализируется необходимость отказоустойчивого функционирования системы. Предложены различные варианты аппаратной реализации механизмов восстановления вычислительных процессов после сбоя или отказа в исполнительных устройствах вычислительной системы с автоматическим распределением ресурсов.

Введение

В настоящее время в Институте проблем информатики Российской академии наук (ИПИ РАН) разрабатывается вычислительная система с автоматическим распределением ресурсов (ВСАРР). Введение в эту вычислительную систему некоторых новых аппаратных решений дает возможность использовать все преимущества, а также устранить большинство недостатков архитектур, использующих модель вычислений, управляемых потоком данных, снижая требования к объему памяти, сложности отдельных компонентов и стоимости системы и позволяя практически полностью исключить программиста из процесса распределения ресурсов вычислительной системы. Блок-схема ВСАРР представлена на рис. 1.

Принципы функционирования ВСАРР

Кратко описывая работу ВСАРР, напомним, что программу потока данных можно представить в виде виртуального графа вычислений. Граф состоит из вершин-узлов и ребер-указателей, по которым перемещаются данные к следующему узлу. В узле графа выполняется программа обработки поступивших на него операндов и контекста. Единицами информации, циркулирующими в такой вычислительной системе, являются токен и пара. Токены представляют собой структуру, которая содержит: передаваемое данное, ключ, а также служебные признаки и атрибуты. Токены перемещаются по ветвям графа. На вход программы узла графа поступает для обработки в исполнительном устройстве (ИУ) пара, содержащая, как правило, два значения данных. В результате обработки пары токены-результаты либо передаются на входы модулей ассоциативной памяти (МАП) для продолжения работы программы, либо передаются в хост-машину в виде окончательных результатов решения задачи. Подробное описание работы вычислительной системы приведено в [1], [2].

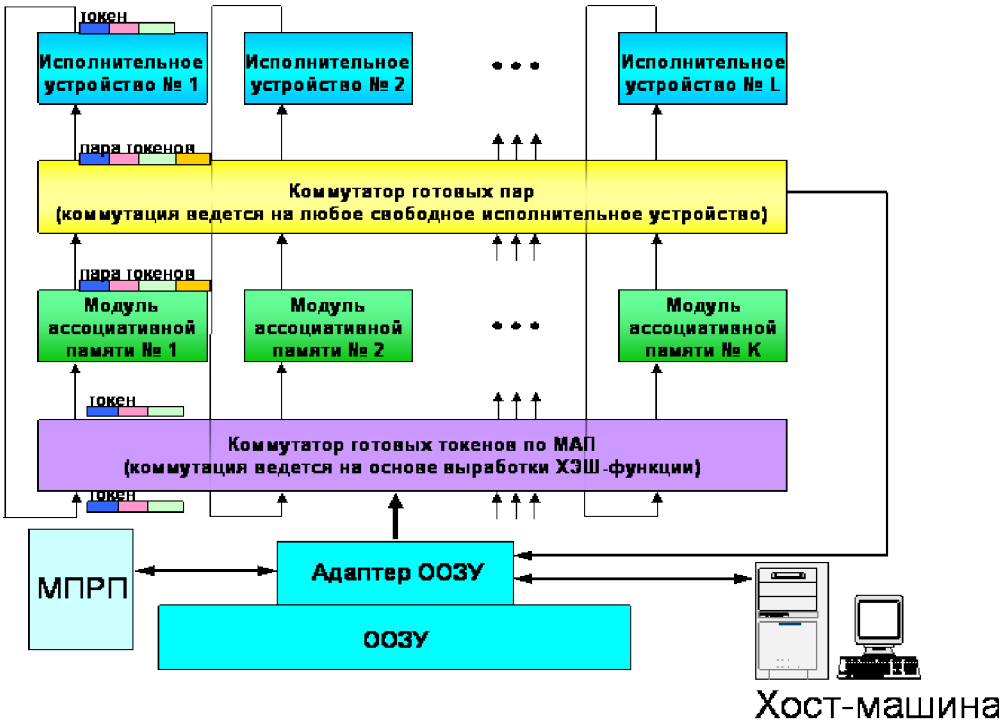

Рисунок 1 – Блок-схема ВСАРР[1]

Отказоустойчивость системы

Одним из важных моментов, которые нужно учитывать при проектировании любых вычислительных систем (наша система не исключение) является вопрос обеспечения отказоустойчивости. Отказоустойчивость – свойство вычислительной системы правильно функционировать в течение заданного времени в условиях возникновения отказов или сбоев в работе отдельных структурных элементов вычислительной системы.

Важность решения этих вопросов для нашей системы ВСАРР обусловлена в первую очередь тем, что в системе для эффективной работы должно быть реализовано больше количество колец, т.е. много работающих независимо друг от друга структурных элементов системы: ИУ и МАП. Отказ одного из них при отсутствии средств восстановления и реконфигурирования может привести к выходу из строя всей системы. Кроме того, данная высокопроизводительная система разрабатывается для современных фундаментальных задач, которые имеют большое практическое значение. К ним относятся задачи, требующие решения в реальном масштабе времени, задачи ядерной энергетики, климата и им подобные, которые требуют повышенной надежности аппаратуры.

Необходимо отметить, что исполнительные устройства вычислительной системы обезличены (пара может выполняться на любом из свободных ИУ), поэтому при отказе одного из них возникает меньше трудностей для восстановления работы системы по сравнению с возможными отказами модулей АП, которые косвенно адресуются через хеш-функцию.

[1] МПРП – микропроцессор регулирования параллелизма

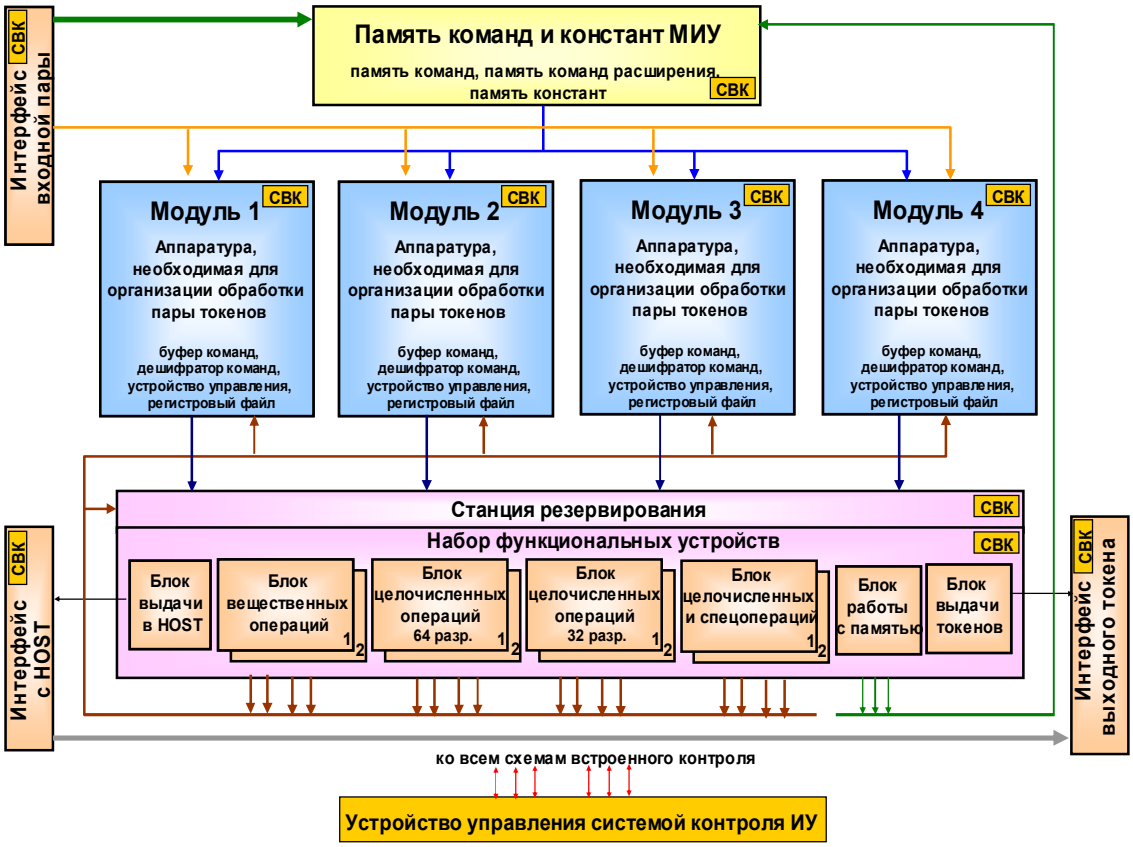

Рисунок 2 – Структура многопоточного исполнительного устройства

Использование принципов потока данных, заложенных в основу вычислительной системы с автоматическим распределением ресурсов (ВСАРР), дает преимущества при организации одновременной многопоточной обработки в одном ИУ, то есть параллельной обработке нескольких пар (нитей) в одном ИУ ВСАРР. Это объясняется тем, что разделение на нити (потоки команд, не взаимодействующие между собой) в архитектуре потока данных происходит автоматически. Разделение на нити аппаратно выполняется ассоциативной памятью системы. В связи с вышеупомянутыми преимуществами одним из наиболее оптимальных вариантов реализации ИУ является многопоточное исполнительное устройство (МИУ) (рис. 2).

МИУ обеспечивает параллельное независимое выполнение 4 потоков команд. Для этого предназначены четыре модуля МИУ. Каждый модуль содержит аппаратуру, необходимую для организации вычислительного процесса: буфер команд, устройство выборки и декодирования команд, регистровый файл и т.д. Эти четыре модуля имеют в совместном использовании общий набор функциональных устройств. Таким образом, при выполнении команды одной из нитей операнды поступают на общие функциональные устройства, обрабатываются и результат возвращается в модуль МИУ, организующий выполнение данной нити. Все четыре модуля МИУ разделяют общую память команд/констант, которая содержит полный набор команд для обработки всех узлов программы. Общим для всех модулей является устройство входного интерфейса пар, которое принимает пары от ассоциативной памяти и распределяет их на обработку между модулями МИУ. Общим является устройство выходного интерфейса токенов, которое обеспечивает сбор готовых результирующих токенов из модулей МИУ и

отправку их на коммутатор модулей ассоциативной памяти. Также общим является устройство управления интерфейсом с хост-машиной, предназначенное для тестирования и отладки МИУ. Подробно особенности функционирования и реализации многопоточных исполнительных устройств ВСАРР описаны в [3].

При выполнении программы узла из исполнительного устройства могут выдаваться токены результата, отсюда возникают следующие проблемы.

В случае возникновения сбоя в МИУ необходимо предотвратить выдачу токенов с искаженной информацией, так как после выдачи такого токена из МИУ во внешнюю среду остановить распространение ошибки по системе уже невозможно. Искаженная информация может находиться как в поле данных токена, так и в поле контекста. Отсюда возможные последствия: токен с искаженным контекстом может провзаимодействовать совсем не с тем токеном, для которого он предназначался или вообще осесть в МАП на неопределенное время. Токен с искаженной информацией в поле данных может спариться правильно, но последующие вычисления будут неверными.

При повторе программы узла, в которой произошел сбой, необходимо предотвратить повторную выдачу уже выданных токенов. Преимущество исследуемой системы заключается в том, что содержимое памятей ИУ (в исследуемой системе это память команд, память команд расширения и память констант) и исходная входная пара сохраняются в ИУ. То есть до успешного завершения выполнения программы узла не происходит перезаписи команд и констант в память, что в дальнейшем значительно облегчает повторное исполнение программы в случае возникновения сбоя или отказа.

При отказе какого-либо из модулей ИУ повторение программы узла необходимо осуществить на исправном модуле, следовательно, необходимо создание механизмов передачи информации, необходимой для восстановления вычислительного процесса, на исправный модуль ИУ.

В случае полного отказа какого-либо из МИУ выполнение программы узла необходимо осуществить на исправном МИУ, следовательно, необходима разработка алгоритмов передачи информации, необходимой для восстановления вычислительного процесса, на исправное МИУ.

Реализация механизма восстановления системы при отказе

В данной статье рассмотрены некоторые механизмы, обеспечивающие восстановление работы вычислительной системы при отказе или сбое многопоточных исполнительных устройств.

Решение об отказе МИУ принимается в трех случаях:

1. Отказ памяти команд/констант МИУ.
2. Отказ всех 4 модулей МИУ, отвечающих за обработку входных пар.
3. Отказ всех функциональных устройств одного типа.

Восстановление системы после отказа с использованием буферных регистров

Первый вариант реализации механизма восстановления системы после сбоя или отказа МИУ заключается в следующем [4].

При отказе МИУ необходимо обеспечить непрерывность выполнения программы. Для этого нужно зафиксировать следующую информацию о программе узла, которая выполнялась на момент отказа или сбоя в МИУ:

– должны быть сохранены все пары (поскольку МИУ является многопоточным, то одновременно в каждом могут обрабатываться до 4 пар), которые были приняты в МИУ для исполнения, но обработка которых не завершилась в результате сбоя или отказа;
– необходимо запомнить все токены, которые выдавались из этого МИУ до момента прерывания и относились к каждой из пар, обрабатываемой в данном МИУ.

В состав аппаратуры включаются два буфера: буфер токенов МИУ, который располагается на выходе МИУ, и буфер пар на входе МИУ (рис. 3).

Пока не закончилась обработка пары, токены из буфера МИУ не передаются через коммутатор токенов в МАПы. В обычном, бессбойном режиме работы МИУ буфер начинает освобождаться только после завершения работы узла (т.е. при завершении обработки пары). И только в этот момент токены, относящиеся к данному узлу, передаются для дальнейшей обработки.

При сбое на одном из направлений МИУ происходит перезапуск соответствующей пары, а токены из части буфера (выходной буфер разделен на четыре секции), связанной с тем направлением, на котором произошел сбой, уничтожаются простым стиранием.

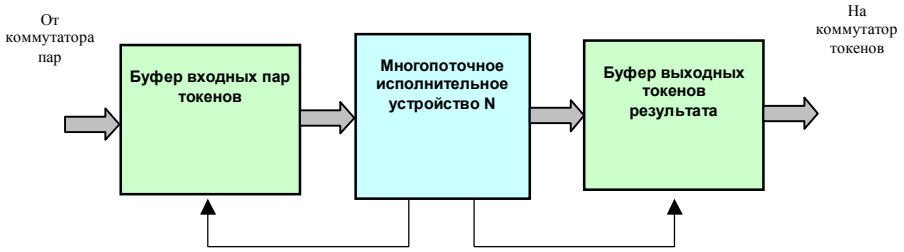

Рисунок 3 – Аппаратура для первого варианта

В режиме восстановления работоспособности МИУ по сбою происходит трехкратный перезапуск пары, вызвавшей сбой, на выполнение. Если перезапуск неуспешный, фиксируется отказ МИУ. При этом в МИУ завершается обработка пар на оставшихся направлениях, после чего это МИУ исключается из работы системы, что и фиксируется в операционной системе. При отказе МИУ в момент запуска механизма восстановления оно закрывается для приема пар от модулей АП, а пары, обработка которых не завершена и не может быть продолжена на этом МИУ, перенаправляются через коммутатор пар на другое свободное МИУ.

Следует отметить, что с помощью выходного буфера токенов МИУ, кроме всего прочего, можно достаточно легко реализовать один из методов аппаратного регулирования параллелизма задач. То есть с помощью управляющих программных конструкций и аппаратной поддержки МПРП можно изменять режим работы буферов токенов: переходить от режима работы в виде очереди на режим стековой обработки токенов, в зависимости от степени загруженности модулей АП.

Данный вариант восстановления вычислительных процессов применим для всех возможных программ узлов, однако он имеет несколько недостатков:

1. Происходит замедление вычислительных процессов в системе, так как пока программа узла не закончится успешно, токены результата не выдаются, а следовательно, другие узлы графа задачи не активируются.
2. Необходим буфер выходных токенов значительного размера.

Восстановление системы после отказа с использованием счетчика выданных токенов

Второй вариант реализации механизма восстановления системы при отказе ИУ применим только в случае, если выдаваемые токены будут детерминированными в соответствии с алгоритмом выполняемой программы узла. То есть если будет точно известно, что при повторе программы узла выдаваться будут такие же токены результата.

В этом случае при повторе программы узла, в которой произошел сбой, необходимо предотвратить повторную выдачу уже выданных токенов, для этого необходимо ввести специальный счетчик корректных токенов, выданных во время выполнения программы узла.

Алгоритм работы счетчика выданных токенов изображен на рис. 4. После пересылки на исправное оборудование всех необходимых данных производится его запуск. Работа будет вестись в *режиме обработки аварийной пары*, который отличается от нормального режима тем, что при попытке выдачи токена в течение выполнения программы узла эта выдача будет блокироваться, а из значения счетчика выданных токенов будет вычитаться 1. И так до тех пор, пока это значение не станет равным 0, после этого выдача токенов и работа в целом будет проводиться в обычном режиме. При такой реализации восстановления системы нет необходимости в наличии буфера выданных токенов результата на выходе ИУ.

В случае полного отказа МИУ возникает задача передачи необходимой информации на другое, исправное, МИУ для восстановления вычислительного процесса. Следует учитывать, что в случае отказа МИУ для передачи такой информации необходимо, чтобы система контроля, выходной интерфейс токена и блок выдачи токена МИУ оставались исправными.

После принятия решения об отказе МИУ, устройство управления системой контроля осуществляет следующие действия (блок-схема алгоритма изображена на рис. 5):

1. Вычисления, производимые по программам узлов, останавливаются на всех четырех направлениях. Все команды, поступившие в исполнительный конвейер, аннулируются. Таким образом, за счет применения архитектурных решений в МИУ реализуется точное прерывание в случае возникновения отказа МИУ.

2. Устройство управления системой контроля (УУСК) изолирует МИУ путем отключения входного интерфейса пары. Теперь на вход МИУ пары из буфера-распределителя готовых пар приниматься не будут. Так же блокируется выдача любой информации из МИУ через выходной интерфейс токена, кроме токенов, сформированных УУСК.

3. Поскольку в МИУ на четырех направлениях вычислений одновременно может обрабатывать до 4 пар, необходимо передать информацию для повторения 4 программ узлов на исправные МИУ. УУСК МИУ формирует 8 аварийных токенов из 4 исходных пар, резервные копии которых хранятся на регистрах пар УУСК с момента прихода на входной регистр пары до успешного завершения соответствующих им программ узлов.

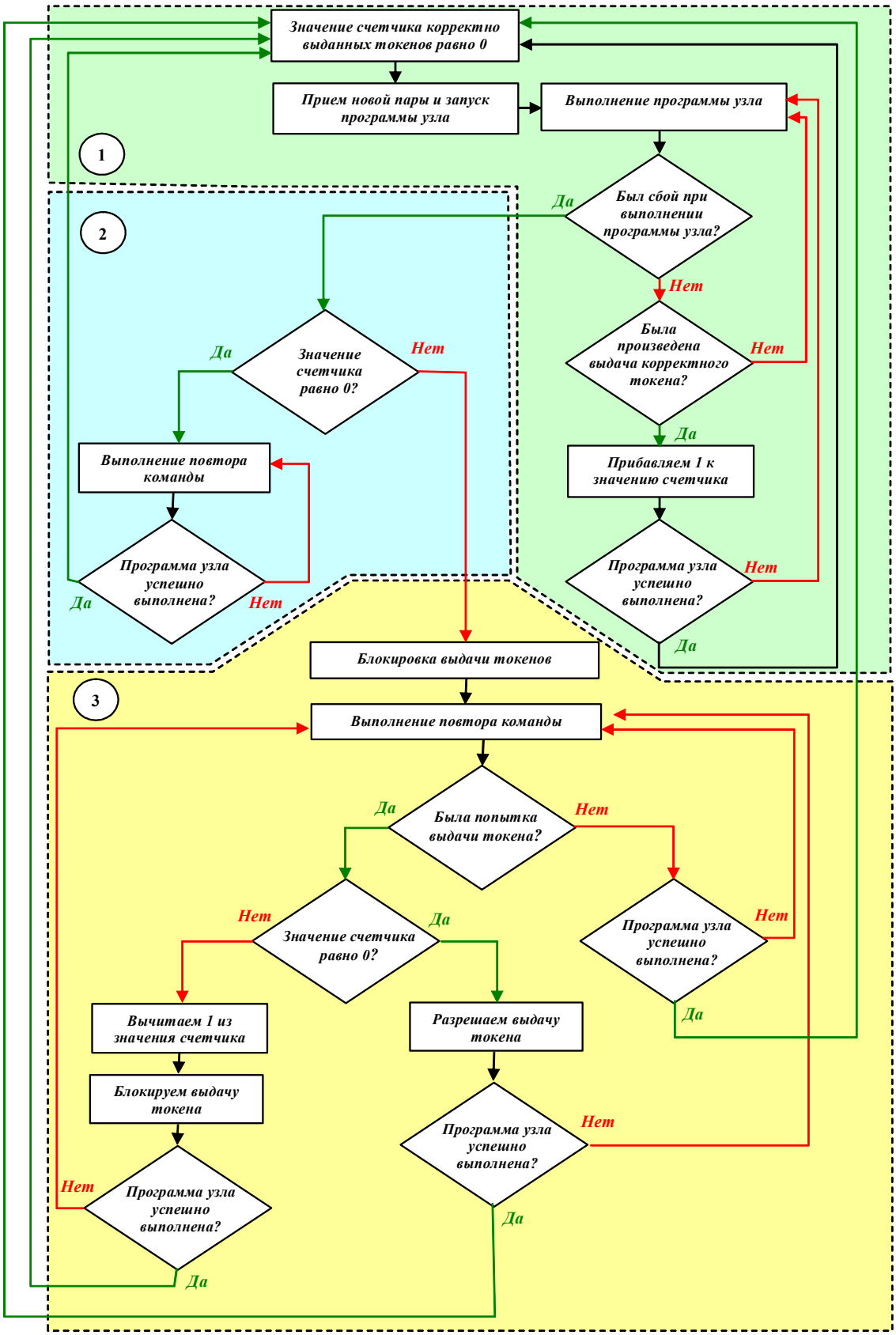

Рисунок 4 – Алгоритм работы счетчика выданных токенов

4. Таким образом, для обоих аварийных токенов необходимо поставить один и тот же номер модуля АП (в котором была сформирована пара) и установить признак принудительной посылки в МАП с фиксированным номером. О процессах формирования и форматах аварийных токенов и аварийной пары подробнее говорится в [5].

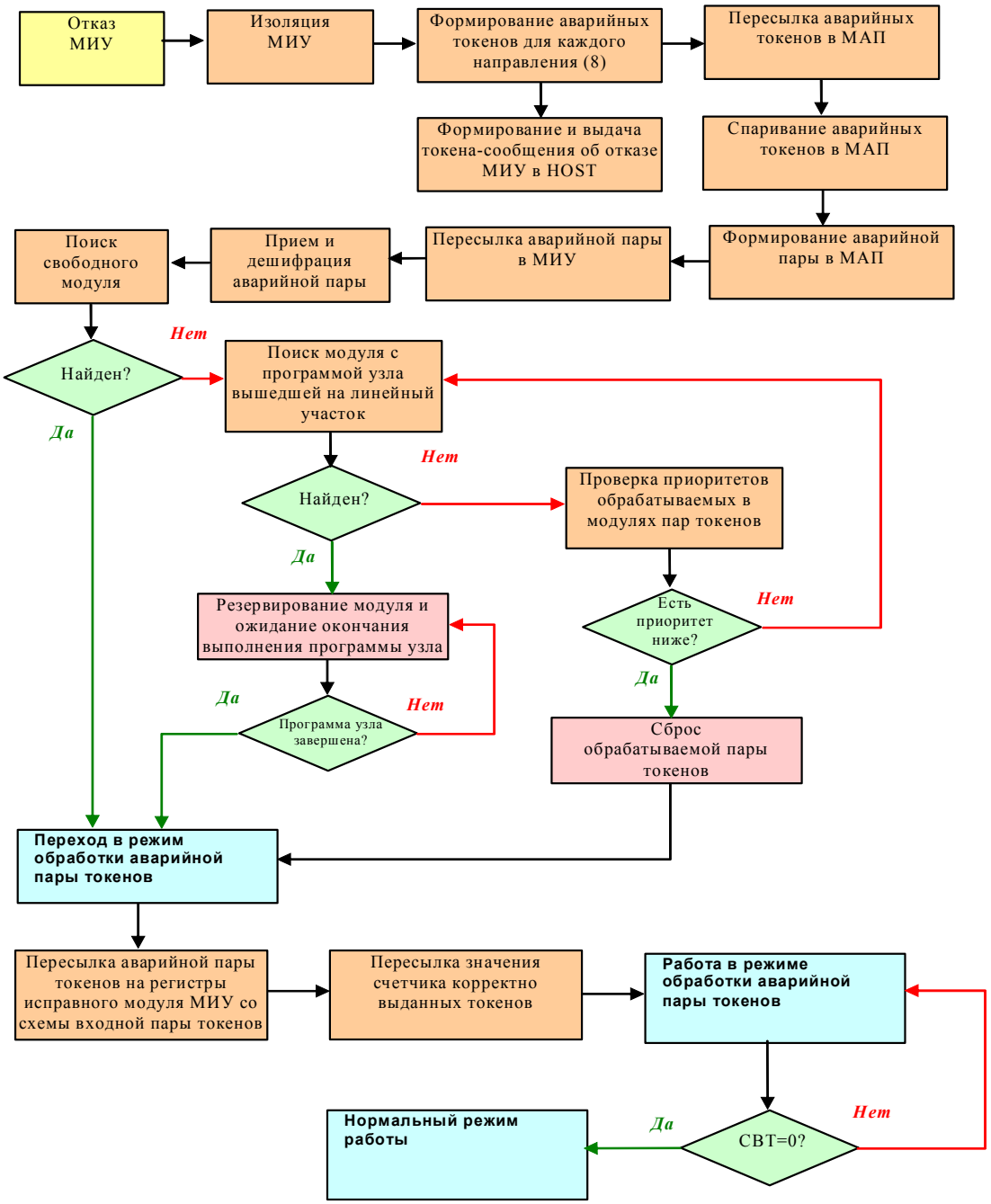

Рисунок 5 – Алгоритм передачи информации, необходимой для восстановления вычислительного процесса, на исправное МИУ в случае возникновения отказа одного из МИУ

5. Аварийные токены пересылаются в МАП, при этом благодаря установке для обоих аварийных токенов каждого направления одного и того же номера модуля АП и установке признака принудительной посылки в МАП с фиксированным номером оба аварийных токена попадают в один и тот же МАП. Первые три поля в ключе каждого токена при поиске в МАП будут закрыты маской, поэтому совпадений ключей по ним обнаружено не будет. Четвертое поле содержит ключ аварийного токена, представляющий собой комбинацию из номера МИУ, где произошел отказ (Nмиу), и номера направления вычислений, в котором обрабатывалась исходная пара (Nнапр). Именно это поле является аргументом поиска в ассоциативной памяти парного аварийного токена из этого же направления вычислений этого же отказавшего МИУ.

6. При обнаружении совпадения ключей двух аварийных токенов, из токенов извлекаются соответствующие поля и формируется аварийная пара.

7. При приходе аварийной пары в исправное МИУ включается специальный режим обработки аварийной пары:

– осуществляется поиск свободного модуля ИУ. Далее свободый исправный модуль принимает на свои регистры пары аварийную пару, которая обрабатывалась в отказавшем МИУ;

– в случае отсутствия свободного модуля в МИУ проводятся мероприятия, аналогичные предпринимаемым при поиске модуля МИУ для обработки пары при отказе модуля МИУ;

– далее начинается выполнение программы узла, при выполнении которой отказало исходное МИУ в режиме обработки аварийной пары, то есть с учетом работы счетчика корректных выданных токенов, алгоритм функционирования которого был описан выше.

8. После формирования и передачи всех 8 аварийных токенов УУСК ИУ формирует специальный токен-сообщение для HOST-машины. Формат этого токена идентичен формату обычного токена. В его поле данных содержится значение регистра неисправностей ИУ. В данном случае, кроме всей прочей необходимой информации там будет указан номер отказавшего МИУ и сведения о характере отказа. Это необходимо для сбора статистики на HOST-машине о состоянии устройств системы. Токен-сообщение передается на коммутатор токенов, а оттуда пересылается транзитом через МАП на HOST-машину.

Выводы

При возникновении сбоя внутри модуля МИУ производится повторение операций в модуле МИУ, что восстанавливает вычислительный процесс, практически не прерывая его, в результате чего падение производительности МИУ минимально.

В случае отказа какого-либо из МИУ выполнение программы узла производится на исправном МИУ с использованием разработанных алгоритмов передачи информации, необходимой для восстановления вычислительного процесса, на исправное МИУ.

Преимущество ВСАРР заключается в том, что содержимое памятей ИУ (в исследуемой системе – это память команд, память команд расширения и память констант) и исходная входная пара сохраняются в ИУ до успешного завершения выполнения программы узла, это значительно облегчает повторное исполнение программы в случае возникновения сбоя или отказа. Фактически необходимой и достаточной информацией для осуществления повторения программы узла, при выполнении которого произошел отказ какого-либо устройства МИУ или всего МИУ в целом, является исходная пара.

Для восстановления вычислительных процессов достаточно переслать эту информацию на исправный модуль МИУ или на исправное МИУ (в случае полного отказа исходного) и запустить выполнение программы узла в режиме обработки аварийной пары.

Литература

1. Бурцев В.С. Новые принципы организации вычислительных процессов высокого параллелизма // Мат-лы Междунар. конф. «Интеллектуальные и многопроцессорные системы – 2003». – Т. 1. – Таганрог: Изд-во ТРТУ. – 2003.
2. Бурцев В.С. Выбор новой системы организации выполнения высокопараллельных вычислительных процессов, примеры возможных архитектурных решений построения суперЭВМ // Параллелизм вычислительных процессов и развитие архитектуры суперЭВМ. – М.: 1997.
3. Янкевич Е.А. Особенности функционирования и реализации исполнительных устройств вычислительной системы с автоматическим распределением ресурсов // Мат-лы Междунар. конф. «Интеллектуальные и многопроцессорные системы – 2003». – Т. 1. – Таганрог: Изд-во ТРТУ. – 2003. – С. 206-208.
4. Окунев А.С., Левченко Н.Н. Некоторые вопросы обеспечения отказоустойчивости и реконфигурации в вычислительной системе с автоматическим распределением ресурсов // Труды Второй Всероссийской научной конф. «Методы и средства обработки информации – 2005». – М.: Московский государственный университет им. М.В. Ломоносова. – 2005 (в печати).
5. Градов Е.С. Исследование вариантов реализации структуры отказоустойчивого исполнительного устройства вычислительной системы с автоматическим распределением ресурсов // УДК 681.31. – Деп. в ВИНИТИ 31.03.2004, № 537 – В2004. 14 с.

А.С. Окунєв, Є.С. Градов, М.М. Левченко

Дослідження варіантів забезпечення відмовостійкого функціонування обчислювальної системи з автоматичним розподілом ресурсів

У даній статті даний короткий опис обчислювальної системи з автоматичним розподілом ресурсів. Аналізується необхідність відмовостійкого функціонування системи. Запропоновані різні варіанти апаратної реалізації механізмів відновлення обчислювальних процесів після перебою або відмови у виконавчих пристроях обчислювальної системи з автоматичним розподілом ресурсів.

A.S. Orunev, E.S. Gradov, N.N. Levchenko

Research on Ensuring Variants of the Fault-safe Functionning of Computing System with Automatical Resources' Distribution

In the given article the brief description of the computing system with automatic distribution of resources is given. The necessity of fault-safe functioning of system is analyzed. Various variants of hardware realization of mechanisms of restoration of computing processes after failure or refusal in executive devices of the computing system with automatic distribution of resources are offered.

Статья поступила в редакцию 19.07.2005.