

Упорядоченность такой структуры в большей степени зависит от дисперсности материала, нагрева и ускорении порошкового материала плазменной струей, протекания процессов упругого и упруго-пластического деформирования частиц на основе и развитии гетерогенного топохимического взаимодействия на контактных поверхностях.

Несмотря на достаточно многочисленные исследования в области плазменного напыления, полученные результаты характеризуются в основном исследованиями по схеме «режим получения – свойства покрытий». Для постановки задач моделирования формирования покрытия из кластеров определенного размера необходимо конкретизировать диапазоны рассматриваемых параметров процесса, конкретизировать используемые экспериментальные факты и дать соответствующие оценки возможностям реализации тех или иных физических процессов [2,3].

Принципиально важным для понимания возможностей плазменного газо-термического метода в обеспечении эксплуатационных свойств покрытий являются ответы на следующие вопросы. Какова природа и какие факторы обеспечивают прочность закрепления частиц на основе, какими факторами определяется геометрическая форма кластера после его закрепления на основе, каковы возможности процесса в обеспечении прочности закрепления частиц на уже сформировавшихся кластерах структуры, каково влияние остаточных напряжений и деформаций на

гетерогенную прочность формируемого покрытия и т.д.

Проведено математическое моделирование процессов, нагрева и ускорении порошкового материала плазменной струей и процессов упругого и упруго-пластического деформирования частиц при формировании слоя покрытия.

Результаты математического моделирования позволяют при напылении порошков определенного состава и дисперсности определить режимы получения покрытий кластерной структуры.

Проведенные исследования эксплуатационных свойств покрытий показали, что покрытия, полученные из порошков узкой дисперсности с сформированной упорядоченной кластерной структурой обладают более высокими показателями износостойкости, термостойкости и жаростойкости.

СПИСОК ЛИТЕРАТУРЫ

1. Барвинок В.А. Плазма в технологии, надежность, ресурс. М.: Наука и технология, 2005. 456с.
2. В.А. Барвинок, В.И.Богданович, И.А. Докукина и др. Физическое и математическое моделирование динамики движения дисперсных частиц в плазменной струе // Известия Самарского научного центра РАН, Спец. выпуск, Самара. - 2008. - Том 3. –С. 83-96.
3. Докукина И.А. Теоретические исследования формирования мезоструктурноупорядоченных кластерных структур в плазменных покрытиях// Проблемы машиностроения и автоматизации. – М., 2009. - №4. С. 136-141.

ВСТРОЕННЫЕ СРЕДСТВА КОНТРОЛЯ БОРТОВОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ ПОД УПРАВЛЕНИЕМ ОПЕРАЦИОННОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ КАК ИТЕРАТИВНЫЙ АГРЕГИРОВАННЫЙ ОБЪЕКТ

© 2012 Долбня Н.А.

ОАО «Ульяновское конструкторское бюро приборостроения», Ульяновск

INTERNAL CONTROLS BOARD COMPUTER SYSTEM RUNNING REAL

PJSC "Ulyanovsk Instrumental Manufacturing Design Bureau", Ulyanovsk

The article refers to the method of organization of built-in control board computer systems running real time operating system. Built-in controls are considered as an aggregate object is an iterative. This aggregated object is implemented based on the design pattern "iterator". This section describes the advantages and details of the software implementation of this approach.

В большинстве современных авиационных систем, построенных по принципу ИМА (интегрированная модульная авионика), с использованием встраиваемых решений на основе применения ОСРВ (операционных систем реального времени), разработчикам приходится решать круг типовых задач таких, как:

- загрузки прикладных программ во флэш-память главного модуля;
- начальную инициализацию контроллеров главного модуля;
- сбор данных с периферийных по отношению к главному модулю контроллеров;
- обработка прерываний от периферийных контроллеров;
- встроенный контроль узлов как главного модуля, так и периферийных.

Начальная инициализация контроллеров главного модуля выполняется средствами начального загрузчика или, в случае особых требований к времени старта ОСРВ, средствами драйверов ОСРВ. Так как на базе одного процессорного модуля могут выпускаться несколько модификаций вычислителей или индикаторов, а стоимость портирования ОСРВ довольно высока, то ОСРВ является специфичной только для главного процессорного модуля, соответственно, инициализацией периферийных контроллеров занимаются уже прикладные программы посредством обращения к драйверам контроллеров главного модуля.

Сбор данных от периферийных контроллеров также является прерогативой прикладных программ.

Обработка прерываний выполняется также средствами ОСРВ и прикладных

программ в зависимости от того, что является источником прерываний – контроллер главного модуля или периферийный.

Задача же организации встроенного контроля вычислителя и всей системы, в состав которой он входит, с одной стороны зависит от конкретного исполнения вычислителя, с другой же стороны имеет много общего во всех его исполнениях.

С точки зрения разработчика программного обеспечения для встраиваемой ОСРВ весь модуль с периферийными устройствами представляет собой древовидную модель, корневым элементом которого является центральный процессор главного процессорного модуля, дочерними элементами которого являются контроллеры конечных устройств.

В качестве интерфейсного контроллера главного модуля может выступать любой контроллер интерфейсной мультиплексированной шины, например LocalPlusBus, на которой расположены все конечные устройства вычислителя или индикатора. В общем случае как число интерфейсных шин, так и глубина ветвления описанной древовидной модели может отличаться. Важно лишь то, что драйверы интерфейсных шин реализованы как часть ОСРВ (согласно общепринятым требованиям к ОСРВ это должен быть статически слинкованный с ядром код) и формат обращения к этим драйверам специфичен только для ОСРВ. Специфика же обращения непосредственно к конечным устройствам вычислителя заключена уже в прикладной программе, максимально переносимой на уровне исходных кодов.

Стартовый контроль отвечает за корректность функционирования внутренних и оконечных устройств процессорного модуля при их инициализации. Специфика используемой ОСРВ диктует необходимость размещения кода, отвечающего за стартовый контроль центрального процессора, оперативной и флэш-памяти, в начальный загрузчик. Остальная часть стартового контроля может располагаться в исполняемом коде прикладной программы.

Текущий контроль выполняется параллельно основной прикладной программе. Он может быть вынесен в отдельную программу и даже в одну из отдельных виртуальных машин, механизм которых поддерживается многими современными ОСРВ такими, как LynxOS-178, VxWorks и др. С другой стороны, если текущий контроль и прикладная программа, несущая основную функциональную нагрузку, обладают одинаковым уровнем критичности и текущий контроль выполняется итеративно через довольно большие промежутки времени, вполне допустимым считается организация текущего контроля в виртуальном адресном пространстве основной программы, то есть, в виде одного с ней процесса.

Расширенный контроль выполняется автономно, при остановленной или завершенной основной программе. В таких условиях возможности расширенного контроля ничем не ограничены.

Таким образом, размещение исполняемого кода встроенного контроля и исполняемого кода библиотеки для доступа к оконечным устройствам вычислителя в

едином программном модуле (не смотря на то, что в общем случае архитектурно это разные компоненты) является вполне оправданным.

Основное свойство встроенных средств контроля – это его итеративный характер. Действия при контроле каждого устройства можно легко разбить на несколько шагов так, чтобы выполнение каждого шага занимало достаточно мало времени.

Как только такое разбиение на этапе проектирования программного обеспечения вычислителя будет выполнено, весь контроль как компонент начинает представлять собой агрегированный объект с массивом структур, содержащих унифицированное представление о каждом устройстве, и унифицированным набором точек входа.

Такая унифицированная организация средств контроля практически не сказывается на производительности программного обеспечения вычислителя, так как реализация инициализатора и итератора по-прежнему не является полностью универсальной с точки зрения исполняемого кода.

Независимость от аппаратной составляющей вычислителя позволяет сделать его даже частью ОСРВ и придать ему статус RSC (ReusableSoftwareComponent) после процесса сертификации первой же версии программного обеспечения вычислителя, что существенно сократит временные затраты на проектирование и разработку программного обеспечения в последующих версиях.

АВИОНИКА МАГИСТРАЛЬНОГО САМОЛЁТА ТУ-204СМ

© 2012 Должиков В.А.

Ульяновский филиал конструкторского бюро ОАО “Туполев”, Ульяновск

AVIONICS OF AIRLINER Tu-204CM

© 2012 Dolzhikov V.A.

The airliner Tu-204CM is a modification the airliner Tu-204-100E. This aircraft has the redesign avionics to operatedifferent system of airplane. The modifications an aircraft's avionics