

вентарный номер контролируемого ГПМ, дата и смена работы, код (номер) обрабатываемой детали; 2) данные о программе обработки — размер партии, сменное задание, текущий выпуск на момент запроса; 3) сведения о времени работы по УП и простоях на текущий момент с указанием вида простоя и его длительности; 4) данные о простоях контролируемого ГПМ с начала смены, причины простоя отказавшего устройства и реквизиты исполнителей, устраняющих неисправности.

Накопленные данные могут быть выведены также в виде непрерывной последовательности интервалов работы, исчисляемых как временем, так и числом отработанных циклов и простоев.

Использование подсистемы ОКНИТО в целом и имитационной модели в частности позволит потребителю сократить производственные затраты, повысить качество продукции и получить в конечном счете реальную выгоду.

(Окончание подборки в следующем номере)

МЕТАЛЛОРЕЖУЩИЕ СТАНКИ

Концепция геометрического ISO-процессора для систем ЧПУ

В. Л. Сосонкин, Г.М. Мартинов

В вычислительной технике и технике ЧПУ стремятся к разумному сочетанию программных и аппаратных решений с тем, чтобы найти удовлетворительный компромисс между противоречивыми требованиями гибкости и быстродействия. Наилучшую основу для этого создает модульная архитектура вычислительной или управляющей системы, при которой стандартизованы как отдельные модули, так и их интерфейсы.

В работах [1 и 2] предложен принцип начальной декомпозиции системы ЧПУ на модули, связанный с выделением задач управления (геометрической, логической, технологической, терминальной и др.). Более глубокая декомпозиция требует структуризации самих задач управления. Ниже эта проблема рассмотрена применительно к наиболее сложной, геометрической задаче, так как здесь увеличение резервов гибкости и ресурсов быстродействия позволяет непосредственно улучшить технико-экономические показатели системы ЧПУ в целом.

Для решения указанной проблемы разработана концепция геометрического ISO-процессора. Происхождение этого наименования связано с языком ISO-7 bit управляющих программ (УП), а суть концепции состоит в использовании двух принципов организации программно-аппаратной среды для решения геометрической задачи. Согласно первому принципу ISO-процессор должен быть построен таким образом, чтобы он воспринимал операторы языка ISO-7 bit как

машинные инструкции. Второй принцип заключается в том, что ISO-процессор реализуется на основе подхода, идея которого заимствована из области объектно-ориентированного программирования [3] (далее подход называется объектно-ориентированным).

Архитектура и система команд геометрического ISO-процессора. По своей структуре ISO-процессор (рис. 1) представляет собой многопроцессорную систему, процессоры которой могут быть реализованы аппаратно, программно или с использованием комбинированных решений. Все четыре процессора (интерпретатор, смежный корректор, интерполятор и диспетчер) работают параллельно.

Интерпретация состоит в предварительной подготовке кадров УП (ISO-программы). Ее упорядоченные кадры пребывают в очереди исходных кадров. Интерпретатор обрабатывает первый кадр из этой очереди и переставляет его либо в очередь кадров, подготовленных для интерполяции, либо в очередь кадров для смежной коррекции.

Необходимость в смежной коррекции возникает при расчете эквидистантных контуров и заключается в коррекции существующих кадров (например, при фрезеровании внутреннего контура) или в синтезе дополнительных кадров (например, при фрезеровании наружного контура). Смежный корректор осуществляет коррекцию кадров из своей очереди и передает их в очередь подготовленных кадров. При необходимости он синтезирует дополнительные кадры

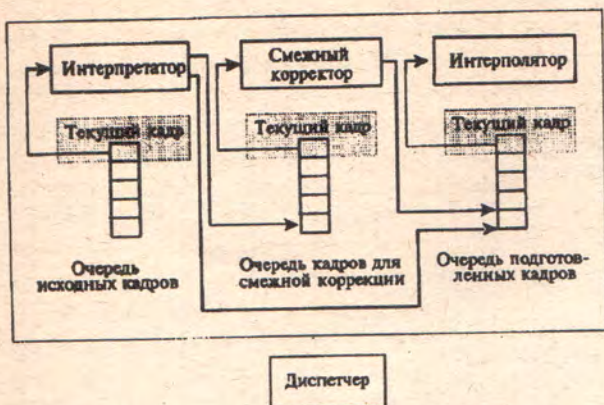


Рис. 1. Структурная схема геометрического ISO-процессора

и вставляет их в ту же очередь подготовленных кадров. Если процесс смежной коррекции выключен, то очередь подготовленных кадров формируется непосредственно в процессе интерпретации.

Функция интерполяции традиционна. Необходимость диспетчеризации вытекает из взаимодействия процессоров в реальном времени и разделения между ними общих системных ресурсов.

При разработке системы команд обратим внимание на две существующие тенденции: согласно первой стремятся к универсальному набору простых команд; вторая же ориентирована на семантический набор специализированных команд.

В первом случае основная проблема состоит в том, чтобы увеличить скорость выполнения наиболее часто встречающихся команд. Для этого привлекается аппаратная часть системы ЧПУ; команды же, используемые реже, реализуются в подпрограммах макроиструкций.

Во втором случае главным является согласование аппаратной части с конкретными языками программирования. Известны примеры, когда архитектура вычислительных машин одной серии зависела от языка и структуры программ. Так, серия GE (Honeywell Multies) соответствовала семантическим моделям программ, написанных на языке PL-1. Модели программ на языке Алгол послужили прототипом архитектуры ЭВМ Burroughs B5500, B6700, ..., B7800. Процессор iAPX-432 предполагает семантические модели структур программ, хотя и не ориентирован на конкретный язык программирования. Особенность архитектуры этого процессора такова, что он поддерживает концепцию объектно-ориентированного программирования [4].

Основные принципы организации процессора iARX-432 и послужили прообразом аппаратной реализации геометрического ISO-процессора. Его машинным языком является система команд, для фор-

мирования которых используются операторы языка ISO-7 bit в их традиционном виде; однако информация УП трактуется иным образом.

Кадр УП на языке ISO-7 bit содержит информацию о требуемых алгоритмах и структурах данных. Алгоритмы представлены подготовительными функциями (G-функциями); структуру данных составляют функции размерных перемещений (X, Y, Z, I, J, K, R), функция подачи (F), а также функция скорости главного движения (S) (последняя в дальнейшем не рассматривается, так как не относится к геометрической задаче). Функции структуры данных можно рассматривать как параметры G-функций, а сами G-функции — как системы команд ISO-процессора.

Все команды разбиты на семь групп в зависимости от функционального назначения. Установлены следующие наименования и назначение групп: Cond (Condition) — организация перехода к следующему кадру; Plan (Plane) — выбор координатной плоскости и переход к относительной системе координат; Dimn (Dimension) — приведение размерности параметров к форме, используемой в алгоритмах интерполяции; Corr (Correction) — расчет эквидистантной траектории; Aclr (Acceleration) — расчеты траекторий на участках разгона и торможения; Del (Time Delay) — задание выдержки времени; Intr (Interpolation) — выбор алгоритма интерполяции.

В каждой группе могут быть выделены несколько подгрупп ортогональных (т.е. взаимоисключающих) G-функций. Пример группирования G-функций для одной из распространенных версий языка ISO-7 bit показан в таблице, где символом * обозначены команды, активизируемые в своей подгруппе по умолчанию, а символ ' указывает на команды, действующие в рамках одного кадра; все остальные обозначения, использованные в таблице, соответствуют языку ISO-7 bit. Активные в данный момент подготовительные функции (команды) образуют G-вектор, размерность которого определяется числом ортогональных подгрупп, а следовательно, зависит от конкретной версии языка.

Возможности интерпретатора. Его структура обусловлена структурой системы команд. Поскольку интерпретация кадра для каждой группы осуществляется независимо, существует семь групповых интерпретаторов.

В разных версиях языка ISO-7 bit обозначения подготовительных функций и их алгоритмическое наполнение не совпадают, и это вызывает определенные затруднения у разработчиков систем ЧПУ. Особенность предлагаемого подхода состоит в том, что алгоритмы жестко привязаны к конкретным подготовительным функциям, т.е. сама структура системы

Группа	Подгруппа	Функция	Параметры	Семантика
Cond	0	G22	L	Вызов подпрограммы
		G23	L	Условный переход
		G24		Безусловный переход
Plan	0	G17 *		Выбор плоскости XY
		G18		Выбор плоскости ZX
		G19		Выбор плоскости YZ
1	1	G20		Выбор двух осей из шести
		G90 *	X, Y, Z	Выбор абсолютных размеров
		G91	X, Y, Z	Выбор относительных размеров
2	2	G53		Отмена смещения нуля
		G54..G59	X, Y, Z	Вызов смещения нуля
		G93	F	Вызов алгоритма преобразования скорости подачи, заданной величиной, обратной времени отработки кадра
Dimn	0	G94 *	F	Вызов алгоритма преобразования скорости подачи, заданной в мм/мин
		G95	F	Вызов алгоритма преобразования скорости подачи, заданной в мм/об
		G70		Вызов алгоритма преобразования перемещений и подачи, исчисляемых в дюймах и дюйм/мин
1	1	G71		Вызов алгоритма преобразования перемещений и подачи, исчисляемых в миллиметрах и мм/мин
		G0	X, Y, Z	Вариант коррекции для выхода к началу эквидистанты
		G1 *	X, Y, Z	Вариант коррекции для движения по эквидистанте линейного контура
G2	0	G2	X, Y, Z, I, J, K, R	Вариант коррекции для движения по эквидистанте кругового контура по часовой стрелке
		G3	X, Y, Z, I, J, K, R	Вариант коррекции для движения по эквидистанте кругового контура против часовой стрелки
		G40	D	Отмена коррекции траектории фрезы
1	1	G41	D	Коррекция траектории фрезы (инструмент слева от контура)
		G42	D	Коррекция траектории фрезы (инструмент справа от контура)
		G68		Выбор алгоритма для кадра стыковки: обход контура по дуге
2	2	G69		Выбор алгоритма для кадра стыковки: обход контура "углом" с остановкой
		G0	X, Y, Z	Выбор алгоритма расчетов разгона и торможения на участке ускоренного подвода
		G1	X, Y, Z	Выбор алгоритма расчетов разгона и торможения на участке линейной траектории
Aclr	0	G2	X, Y, Z, I, J, K, R	Выбор алгоритма расчетов разгона и торможения на участке круговой траектории по часовой стрелке
		G3	X, Y, Z, I, J, K, R	Выбор алгоритма расчетов разгона и торможения на участке круговой траектории против часовой стрелки
		G8 *		Выбор алгоритма управления разгоном и торможением
1	1	G9		Отмена управления разгоном и торможением
		G4	F	Выдержка времени (с)
Del	0	G104	F	Выдержка времени (об)
		G0	X, Y, Z	Линейная интерполяция на ускоренном ходу
Intr	0	G1	X, Y, Z	Линейная интерполяция
		G2	X, Y, Z, I, J, K, R	Круговая интерполяция, по часовой стрелке
		G3	X, Y, Z, I, J, K, R	Круговая интерполяция, против часовой стрелки

команд определяет ресурсы системы ЧПУ. Такое решение обеспечивает гибкость последней, поскольку ее функциональное назначение и возможности обусловлены множеством выбранных (отобранных) подготавливаемых функций.

В каждом групповом интерпретаторе заложены подгруппы, с которыми он работает. Групповой интерпретатор обращается к конкретной подготавливаемой функции как к соответствующей координате G-вектора и передает ей управление.

Использование подобной схемы означает, что в одной и той же системе ЧПУ могут использоваться различные системы команд. Проблема настройки на систему команд рассматривается ниже.

Сущность объектно-ориентированного подхода. Основными понятиями, поясняющими объектно-ориентированный подход, являются инкапсуляция (encapsulation), наследование (inheritance) и полиморфизм (polymorphism).

Инкапсуляция — это объединение алгоритмов (правил, методов) и данных. Для абстрактного единого компонента, называемого объектом, одновременно рассматриваются некоторая структура данных, определяющая его атрибуты, и некоторая группа функций, характеризующих его поведение.

Наследование — это свойство одних объектов приобретать атрибуты и поведение других объектов без дублирования кодов, реализующих атрибуты и поведение. Таким образом, существуют объекты-предки и объекты-наследники. Тип объекта служит шаблоном для всех порожденных экземпляров этого типа. Порождение новых объектов осуществляется так называемыми конструкторами (constructor), а уничтожение ранее созданных объектов — деструкторами (destructor).

Полиморфизм — это распространение общих действий (задач) на множество разных объектов. Полиморфизм повышает уровень абстрагирования при описании действий, поскольку собственные действия объектов не рассматриваются. Свойство полиморфизма реализуется через так называемое «позднее связывание» с адресом процедуры в составе объекта, которое осуществляется только в момент непосредственного обращения к ней (т.е. при выполнении УП). Процедуры объекта (т.е. методы), имеющие позднее связывание, называются виртуальными. Позднее связывание поддерживается специальной системной структурой, называемой VMT (Virtual Method Table — таблица виртуальных методов). Через эту таблицу и осуществляется вызов виртуальных методов на этапе выполнения УП.

Идеи объектно-ориентированного подхода в наиболее последовательной форме реализованы в компиляторе Borland Pascal 7.0.

Объектно-ориентированный подход к реализации команд геометрического ISO-процессора. С каждой командой ISO-процессора сопоставлен свой объект. В основе иерархии объектов, соответствующих системе команд ISO-процессора (рис. 2), лежит TObject из стандартной библиотеки пакета Borland Pascal 7.0. Это абстрактный тип объекта, не имеющий экземпляров, но позволяющий порождать объекты и встраивать их в коллекции (динамические массивы) с тем, чтобы производить над ними операции (в том числе и итерационные) как над элементами очереди.

Так, базовым объектом-предком для всех объектов системы команд ISO-процессора служит TCommand (см. рис. 2). Объявление этого объекта наследником TObject показано на рис. 3, а. Здесь буква T является идентификатором типа; поле Name соответствует адресу функции в языке ISO-7 bit. Для инициализации вновь созданного объекта служит конструктор (Init), а для уничтожения объекта после его использования — деструктор (Done).

Исходя из того, что все команды языка ISO-7 bit подразделяются в системе команд ISO-процессора на G-функции и их параметры, построим частные объекты TDataCommand и TGCommand (рис. 3, б и в), порожденные объектом TCommand.

Экземплярами типа объекта TDataCommand служат функции размерных перемещений, подачи и т.д. Все экземпляры выступают как параметры G-функций, и поэтому конкретные экземпляры создаются при каждом появлении соответствующих функций (перемещений, подачи и пр.) в исходном коде УП.

TGCommand является объектом абстрактного типа, который выступает в качестве предка при создании типов объектов для каждой подготавливаемой функции. Все объекты одного типа имеют аналогичные правила поведения. Методы (процедуры) Interpret, Interpolate, Correct относятся к числу виртуальных (virtual), т.е. имеют позднее связывание и определяют поведение объекта при интерпретации, интерполяции и смежной коррекции кадров УП.

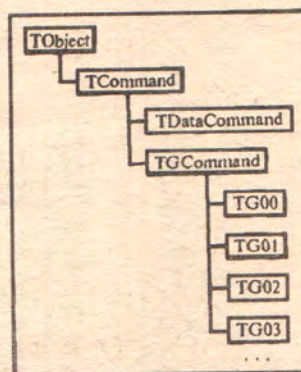


Рис. 2. Иерархия объектов системы команд ISO-процессора

(Продолжение на с.18)


```

TCommand = object(TObject)
Name: PChar;
constructor Init(AName: PChar);
destructor Done; virtual;
end;

```

а)

```

TDataCommand = object(TCommand)
Value: Real; { значение функции }
constructor Init(AName: PChar; AValue: Real);
end;

```

б)

```

TGCommand = object(TCommand)
Number: Byte; { номер G-функции }
GroupIndex: Integer; { идентификатор подгруппы }
constructor Init(AName: PChar; AValue: Real);
procedure Interpret(P: PObject; Block: PIdCollection; var Pr: Integer); virtual;
procedure Interpolate(P: PObject; Block: PIdCollection; var Pr: Integer); virtual;
procedure Correct(P: PObject; Block: PIdCollection; var Pr: Integer); virtual;
end;

```

в)

```

TG22 = object (TGCommand) { вызов подпрограммы }
procedure Interpret(P: PObject; Block: PIdCollection; var Pr: Integer); virtual;
end;

```

```

TG23 = object (TGCommand) { условный переход }
procedure Interpolate(P: PObject; Block: PIdCollection; var Pr: Integer); virtual;
end;

```

```

TG24 = object (TGCommand) { безусловный переход }
procedure Interpret(P: PObject; Block: PIdCollection; var Pr: Integer); virtual;
end;

```

Рис. 4. Производные от TGCommand в группе Cond G-функций

Рис. 3. Базовый объект-предок (а) и его производные (б и в)

В рассмотренных процедурах в качестве параметра P (типа PObject) выступают объекты соответствующих процессов (интерпретации, интерполяции или смежной коррекции); параметр Block (типа PIdCollection) соответствует обрабатываемому кадру; параметр Pr (типа Integer) используется для обмена информацией между подгруппами системы команд при выборе конкретного алгоритма G-функции.

Методы переопределяются в порождаемых объектах, использующих различные алгоритмы. Так, порожденными объектами TGCommand в группе Cond G-функций являются объекты, показанные на рис. 4. В этих объектах переопределены методы для процессов интерпретации и интерполяции, так как поведение объектов отличается от поведения их предков. Поскольку существенными для этих объектов являются их алгоритмы, не требуется создавать новые экземпляры объектов на каждом появлении соответствующей G-функции в исходном коде УП.

Формат объекта в архитектуре ISO-процессора представлен на рис. 5 двумя компонентами: структурой экземпляра и структурой таблицы VMT, причем число точек входа в последнюю есть общее число используемых методов (процедур).

Схема образования объектов в системе команд ISO-процессора показана на рис. 6. Любые поля данных, унаследованные от типа предка, запоминаются до образования новых полей, заново определенных в порожденном типе.

Кадр УП представляет собой набор функций, имеющий номер, а сама УП в свою очередь является набором кадров, имеющим имя; поэтому кадр и УП можно определить как экземпляры одного типа объекта TIdCollection (рис. 7).

Использование объекта TCollection в качестве предка позволяет включать в кадр команды, сформированные уже после компиляции кода объекта, что создает предпосылки для обновления принятой версии языка ISO-7 bit.

Организация интерпретаторов. Работу конкретных процессоров проследим на примере интерпретаторов. При отработке кадра УП попеременно вызываются интерпретаторы, ориентированные на разные группы команд. При этом предварительное формирование текущего G-вектора осуществляется G-интерпретатором, вызываемым в первую очередь.

В процессе интерпретации происходит переход к локальной системе координат, на которую ориентированы алгоритмы интерполяции. Поэтому построена структура данных TCurrentPoint (рис. 8), где в поле Sr (Source) записаны исходные координаты перед началом отработки кадра, а в поле Eq (Equidistant) — соответствующие координаты эквидистанты.

Далее сформирована объектная модель интерпретатора (рис. 9). В качестве полей данных здесь выступают G-вектор (GVector), запись с исходными коор-

Унаследованные поля данных
Адрес таблицы VMT
Вновь определенные поля данных

а)

Размер экземпляра
Адрес таблицы VMT объекта-предка
Число точек входа в таблицу VMT
Идентификаторы виртуальных методов
Адреса виртуальных методов

б)

Рис. 5. Формат объекта в архитектуре ISO-процессора: а — структура экземпляра объекта; б — структура таблицы VMT

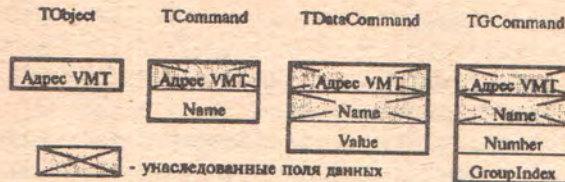


Рис. 6. Схема образования объектов в системе команд ISO-процессора

динатами (CrPt) и указатель на очередь входных кадров (CL). Интерпретаторы и алгоритмы обработки G-функций определены как методы, обеспечивающие подготовку кадра.

Правила интерпретации связывают текущий кадр (Block) с G-функцией, входящей в G-вектор, вызывая соответствующий метод обработки кадра. Задача G-интерпретатора (GIntrpr) состоит в обследовании G-функций кадра для их последующего задания в качестве текущих координат G-вектора.

Не останавливаясь на содержательной стороне методов, заметим, что обращение к алгоритмическим процессам с помощью самих методов делает систему команд независимой от конкретной реализации процессов в системе ЧПУ.

Принципы программно-аппаратной реализации ISO-процессора. При выполнении УП кадры в коде языка ISO-7 bit, трансформируются в коллекции объектов следующим образом. На базе функций структур данных создаются экземпляры объекта PDataCommand, а подготовительные функции заменяются указателями соответствующих функций в системе команд ISO-процессора. Переопределенный таким образом кадр поступает во входную очередь интерпретатора, после чего его обработка и общее функционирование системы ЧПУ поддерживается как на аппаратном, так и на программном уровне.

Основу ISO-процессора составляет объектно-ориентированная архитектура, как и в процессоре iAXP-432. При реализации ISO-процессора следует выделить два блока: блок аппаратной поддержки объектов и программный блок, загружаемый в ОЗУ (ППЗУ) (рис. 10).

Блок аппаратной поддержки осуществляет поддержку абстрактных данных на трех уровнях. Первый уровень, играющий роль менеджера объектов, инициализирует типы объектов, создает и уничтожает их экземпляры и управляет вызовом виртуальных методов. На этом уровне реализуются, в частности, следу-

```
TIdCollection = object(TCollection)
  Id: PChar; {идентификатор}
  constructor Init(AId: PChar);
  destructor Done; virtual;
end;
```

Рис. 7. Экземпляры типа TIdCollection

```
TPoint_3D = record {тип записи для точки}
  X, Y, Z: Real; {в пространстве}
end;

TCurrentPoint = record {тип записи для исход-}
  Sr, Eq: TPoint_3D; {ных и эквидистантных}
  end; {координат}
```

Рис. 8. Структуры данных TPoint и TCurrentPoint

ющие функции: диспетчеризация виртуальных методов; определение типа объекта; распределение и освобождение динамических переменных типа объекта; проверка правильности выделения объема памяти под объект при вызове его конструктора.

На втором уровне реализованы базовые объекты, которые лежат в основе иерархий (см., например, рис. 2), поскольку методы этих типов используются наиболее часто. Созданный компилятором код методов базового объекта записывается в ППЗУ вместе с таблицей VMT, в которой точки входа соответствуют адресам методов. К базовым объектам относятся, например, TCollection (с его методами итерационного поиска, сортировки, обращения к элементам), TStream (с его методами записи элементов в поток и считывания из потока), TDataCommand, TGCommand. Создание и функционирование экземпляров базовых объектов, а также выполнение их методов, поддерживаются, как уже отмечалось, первым уровнем.

На третьем уровне реализованы системные объекты (такие, как процессы и порты диспетчеризации), осуществляющие аппаратную поддержку диспетчерских функций операционной системы реального времени (ОСРВ). На этом же уровне находится модуль аппаратно реализованных стандартных алгоритмов (например, интерполяции). Он обеспечивает выполнение G-функций при интерполяции, интерпретации и смежной коррекции кадров УП.

Программный блок включает в себя модуль системы команд и модуль основных процессов ISO-процессора. Система команд определяется принятой версией языка ISO-7 bit, а к основным процессам относятся интерполяция, интерпретация и смежная коррекция.

```
TIntrpr = object(TObject)
  CrPt: TCurrentPoint;
  GVector, CL: PCollection;
  constructor Init;
  destructor Done; virtual;
  procedure GIntrpr(Block: PIdCollection);
  virtual;
  procedure Cond(var Block: PIdCollection);
  virtual; {групповой интерпретатор Cond}
  (...) {другие групповые интерпретаторы}
end;
```

Рис. 9. Модель интерпретатора

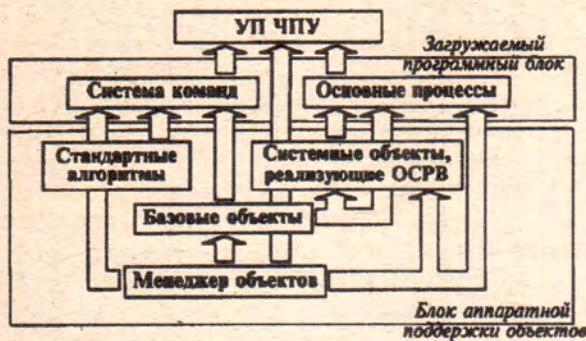


Рис. 10. Схема программно-аппаратной реализации ISO-процессора

Выводы

1. Построение системы ЧПУ по типу геометрического ISO-процессора открывает дополнительные резервы одновременного повышения быстродействия и гибкости. Первое обеспечивается параллельной обработкой процессов и глубокой аппаратной поддержкой системы команд. Второе обусловлено высоким уровнем декомпозиции геометрической задачи ЧПУ на

основе объектно-ориентированного подхода (для разработчиков систем ЧПУ) и возможностью загрузки нового файла системы команд (для квалифицированного конечного пользователя).

2. Открытая модульная архитектура геометрического ISO-процессора создает широкие возможности для дальнейшей эволюции систем ЧПУ.

3. Программно-аппаратная подготовка кадров УП гарантирует непрерывность обработки особо сложных поверхностей.

4. Повышение удельного веса аппаратных решений способствует возрастанию одновременно быстродействия и надежности системы ЧПУ.

Список литературы

1. Сосонкин В.Л. Задачи числового программного управления и их архитектурная реализация в устройствах ЧПУ // Станки и инструмент. — 1988. — № 10. — С. 39-41.
2. Сосонкин В.Л. Концепция системы ЧПУ на основе персонального компьютера (PCNC) // Станки и инструмент. — 1990. — № 11. — С. 9-14.
3. Вуч Г. Объектно-ориентированное программирование с примерами применения: Пер. с англ. — М.: Конкорд, 1992. — 519 с.
4. Органик Э. Организация системы Интел 432: Пер. с англ. — М.: Мир, 1987. — 446 с.

Точность прецизионной токарной обработки при минимизации влияния тепловых возмущений

А.А. Игнатьев, В.А. Добряков

Одной из основных задач, которую необходимо решать при прецизионной токарной обработке деталей с размерами не более 50-70 мм и точностью 2-3-го качества, является минимизация последствий тепловых процессов, возникающих под влиянием как внутренних, так и внешних (по отношению к станку) возмущений. Это связано в том, что погрешность обработки, обусловленная тепловыми деформациями элементов станка, может стать соизмеримой с допусками (2-5 мкм) на размеры обрабатываемых деталей и, следовательно, от нее в значительной мере зависит достигаемая точность деталей [1 и 2].

Один из путей обеспечения требуемой точности обработки на прецизионном токарном станке — стабилизация теплового поля последнего, позволяющая поддерживать размеры обрабатываемых деталей в заданных пределах. В данной статье рассматривается стабилизация, осуществляемая путем выбора соответствующего теплового режима основных формообразующих элементов станка.

Объектом исследований был прецизионный токарный ГПМ мод. ТПАРМ-100М [3]. Если область устойчивого функционирования исследуемого ГПМ

определена по динамическим характеристикам формообразующих элементов и выполнены мероприятия, позволяющие минимизировать уровень колебаний при резании (например, произведена динамическая балансировка шпинделя и оптимизирован технологический режим [3]), то детали из алюминиевых и медных сплавов после обработки имеют некруглость менее 0,5 мкм, волнистость менее 0,1 мкм и шероховатость обработанной поверхности $R_a = 0,04+0,06$ мкм.

Поскольку указанные значения погрешностей в 4-10 раз меньше допусков на размеры, то при оценке точности обработки их можно не учитывать. Тогда суммарная погрешность прецизионной токарной обработки $\Delta \Sigma = \Delta_{и} + \Delta_{т.д.} + \Delta_c + \Delta_{р.г.}$, где $\Delta_{и}$ — погрешность, возникающая в результате износа инструмента; $\Delta_{т.д.}$ — погрешность, обусловленная тепловыми деформациями от воздействия внешних и внутренних тепловых возмущений; Δ_c — погрешность позиционирования суппорта; $\Delta_{р.г.}$ — то же резцовой головки.

Исследования [2] показали, что при изготовлении деталей на исследуемом ГПМ резцами, оснащенными