

Параллельное программирование процессоров ЦОС TMS320C6000 в среде 3L DIAMOND

СРЕДА DIAMOND ФИРМЫ 3L (WWW.3L.COM), ИЗВЕСТНАЯ РАНЕЕ КАК СРЕДА 'PARALLEL C', ЯВЛЯЕТСЯ КОМПАКТНОЙ СРЕДОЙ РАЗРАБОТКИ МУЛЬТИЗАДАЧНЫХ ПРИЛОЖЕНИЙ ДЛЯ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ. НАСТОЯЩАЯ СТАТЬЯ ДАЕТ ОБЩЕЕ ПРЕДСТАВЛЕНИЕ О СРЕДЕ 3L DIAMOND И ПРИНЦИПАХ ЕЁ ПОСТРОЕНИЯ. В КАЧЕСТВЕ ПРИМЕРА АППАРАТНОЙ ПЛАТФОРМЫ РАССМОТРЕНО ФУНКЦИОНИРОВАНИЕ СРЕДЫ 3L DIAMOND НА МУЛЬТИПРОЦЕССОРЕ ЦОС TORNADO-PX64XXQ ФИРМЫ "МИКРОЛАБ СИСТЕМС" (WWW.MLABSYS.COM), КОТОРЫЙ ПОСТРОЕН НА БАЗЕ ЧЕТЫРЕХ ПРОЦЕССОРОВ ЦОС TMS320C6416.

Известным традиционным методом описания мультипроцессорных систем является их интерпретация как множества процессоров, возможно, с некоторыми "мультипроцессорными ухищрениями", причем каждый из процессоров функционирует в своей однопроцессорной среде. Несмотря на то, что такой подход в общем-то правомерен, он, тем не менее, очень далек от "удовлетворительного" и оставляет разработчика наедине с большим числом сложных проблем, таких как загрузка программ, межпроцессорная коммуникация, синхронизация и т.п. Среда параллельного программирования 3L DIAMOND построена на базе принципиально другого подхода, а именно — описания системы как единого целого, и специально создавалась для "структурированной" разработки мультипроцессорного ПО для любого числа процессоров. Мультипроцессор ЦОС TORNADO-PX64xxQ фирмы "МикроЛАБ Системс" является хорошим примером удачного сочетания мультипроцессорных аппаратных средств и концепции среды 3L DIAMOND. Неизбежным результатом непосредственного программирования таких сложных мультипроцессорных устройств явилось бы очень большое время разработки мультипроцессорного ПО в конечных изделиях, однако среда 3L DIAMOND позволяет разработчику практически полностью отключиться от всех деталей и очень быстро получить функционирующее мультипроцессорное и мультизадачное приложение.

Основополагающим принципом построения среды 3L DIAMOND является концепция *последовательных коммуни-*

цирующих процессов (С. А. Р. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985), которая описывает любое программное приложение в терминах независимых *процессов (processes)* (или *задач (tasks)*), которые самосинхронизируются и коммуницируют между собой путем обмена *сообщениями (messages)* через *каналы связи (channels)*. В данном случае, сообщение — это просто блок данных, а канал связи — это механизм однонаправленной передачи данных между двумя задачами. Среда 3L DIAMOND удачно использует эту концепцию для сети из процессоров.

АППАРАТНАЯ МОДЕЛЬ

Среда 3L DIAMOND использует простую аппаратную модель мультипроцессорной аппаратуры, в которой будет функционировать конечное приложение как множество *процессоров*, связанных между собой двунаправленными *проводами (wires)* (или *межпроцессорными связями (interprocessor links)*). В рамках этой концепции всегда должен присутствовать *корневой процессор ('root' processor)*, а все остальные процессоры могут иметь любые уникальные имена. На рис. 1 представлен пример трехпроцессорной системы, состоящей из процессоров с именами "Root", "P2" и "P3", которые попарно соединены с помощью трех связей: две связи между процессорами "Root" и "P2", и ещё одна связь между процессорами "P2" и "P3". Эта аппаратная модель легко масштабируется, и каждые новые дополнительные процессоры лишь привносят новые

дополнительные коммуникационные ресурсы и никак не ограничивают коммуникацию между другими процессорами. Важным фактором является то, что программное приложение, разработанное с помощью среды 3L DIAMOND, пересылая данные по межпроцессорной связи, совершенно не обязано знать, как именно реализована эта связь.

В течение времени на рынке присутствовали различные процессоры, которые использовали встроенные контроллеры межпроцессорной связи по принципу "точка-точка". Одними из первых были транспьютеры фирмы INMOS, которые в этом смысле имели "идеальные" встроенные контроллеры межпроцессорной связи с полностью аппаратным управлением, что делало их исключительно простыми и эффективными в применении. Позже, фирма TEXAS INSTRUMENTS выпустила процессоры ЦОС TMS320C4x со встроенными "коммуникационными портами", которые были намного быстрее межпроцессорных связей транспьютеров фирмы INMOS, однако большая часть управления ими должна была осуществляться программно, что делало их значительно более сложными в управлении и применении. Ещё позже, фирма ANALOG DEVICES

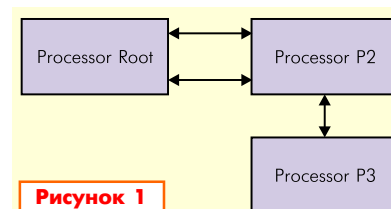


Рисунок 1

Аппаратная модель

выпустила процессоры ЦОС SHARC со встроенными контроллерами межпроцессорной связи, которые практически полностью управлялись из ПО, что приводило к ещё более значительному усложнению в управлении и применении. Наконец, совсем недавно, встроенные контроллеры межпроцессорной связи полностью исчезли с выпуском процессоров ЦОС TMS320C6000 фирмой TEXAS INSTRUMENTS.

В связи с этим, разработчики аппаратуры на процессорах ЦОС TMS320C6000 вынуждены придумывать свои собственные механизмы межпроцессорной коммуникации с использованием ПЛИС, стандартных интерфейсов (Firewire и др.), шин (PCI и др.), заказных БМК и так далее.

Возвращаясь к примеру мультипроцессора TORNADO-PX64xxQ фирмы "МикроЛАБ Системс", среда 3L DIAMOND использует двухуровневый принцип описания аппаратных межпроцессорных связей для этого устройства. Внутри устройства все процессоры ЦОС объединены общей высокоскоростной памятью с индивидуальным доступом, которая используется средой 3L DIAMOND как первый уровень для реализации шести высокоскоростных межпроцессорных связей, связывающих все возможные

пары процессоров внутри одного устройства. При этом, оставшаяся часть общей памяти может быть использована в приложениях пользователя. Реальная скорость передачи данных через эти межпроцессорные связи в среде 3L DIAMOND превышает 160 Мбайт/с при мгновенной скорости внутри блока данных 400 Мбайт/с. В тоже время, "межплатные" межпроцессорные связи для нескольких объединяемых мультипроцессоров TORNADO-PX64xxQ осуществляются через внешние аппаратные последовательные порты (McBSP) процессоров ЦОС TMS320C6000 и соответствуют второму уровню межпроцессорных связей в среде 3L DIAMOND. Такая двухуровневая концепция межпроцессорных каналов среды 3L DIAMOND для мультипроцессора TORNADO-PX64xxQ отлично вписывается в аппаратную модель 3L DIAMOND, так как добавление новых таких устройств в мультипроцессорную систему расширяет суммарную пропускную способность всех межпроцессорных связей при увеличении суммарной производительности процессоров ЦОС. Это коренным образом отличается от аппаратной реализации мультипроцессорной системы с "общей шиной" в качестве единственного канала

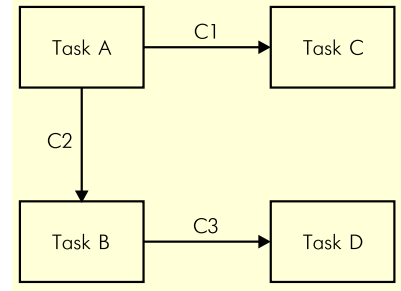


Рисунок 2 Программная модель

межпроцессорных связей, так как добавление новых процессоров в такой системе увеличивает нагрузку на шину и ведет к эффективному снижению пропускной способности каждой межпроцессорной связи.

ПРОГРАММНАЯ МОДЕЛЬ

На рис. 2 представлена простая программная модель, которую среда 3L DIAMOND использует для описания некоторого приложения.

В данном примере представлены четыре задачи, соединенные тремя каналами связи. Задача "А" может послать данные задаче "С" через канал "С1"

MicroLAB Systems
 127591, Москва, Россия,
 Дубнинская ул., д. 83, оф. 612
 WEB: www.mlabsys.com
 E-mail: info@mlabsys.com
 тел./факс: (095)-900-6208

Комплексные решения ЦОС для процессоров TMS320

TORNADO СИСТЕМЫ ЦОС
MIRAGE ЭМУЛЯТОРЫ!
АЦП/ЦАП, СОПРОЦЕССОРЫ

TORNADO
 системы ЦОС для ПК и автономных приложений на базе TMS320C3x/5000/6000
\$540+
 дочерние модули АЦП/ЦАП, сопроцессоры ЦОС
\$215+

MIRAGE
 эмуляторы JTAG/MPSD для TMS320
\$1000+
TI Code Composer
\$1300 (C3x/C4x)
\$1995 (Studio)

Hypersignal
 САПР алгоритмов и систем ЦОС
\$1995+
QEDesign
 САПР цифровых фильтров
\$1095+
Nucleus PLUS, 3L Diamond
 ОСПБ

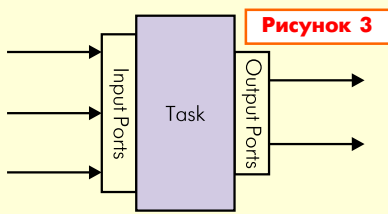


Рисунок 3

Задача и её порты ввода/вывода

КОНФИГУРАЦИЯ

Последним этапом в разработке приложения, работающего под управлением среды 3L DIAMOND, и именно то, что отличает 3L DIAMOND от традиционных мультипроцессорных систем, является *конфигурация* (рис. 4), которая задает соответствие между логической структурой приложения (в терминах задач и каналов связи) и аппаратной моделью (в терминах процессоров и межпроцессорных связей конкретного устройства). Это осуществляется с помощью утилиты *конфигуратора* среды 3L DIAMOND, которая объединяет задачи со всеми программными модулями, необходимыми для запуска и исполнения конкретного приложения. В некотором смысле, *конфигуратор* среды 3L DIAMOND является специализированным мультипроцессорным компоновщиком (*linker*).

Поскольку конфигуратор имеет дело с полным множеством задач, функционирующих в рамках конкретного приложения, то он может точно определить, какие конкретно сервисы среды 3L DIAMOND необходимы для работы данного приложения и, соответственно, подсоединяет только те системные компоненты (драйверы устройств и т.п.), которые будут реально использоваться во время исполнения приложения, для минимизации фактического размера исполняемого кода.

Конфигуратор управляется с помощью *конфигурационного файла*, который содержит текстовое описание структуры конкретного приложения и устройства и их взаимного соответствия. На рис. 5 представлен пример конфигурационного файла для приложения, приведенного на рис. 1 и 2.

В начале конфигурационного файла содержится описание трех процессоров с указанием их уникальных имен и типов. Тип процессора необходим конфигуратору для определения числа встроенных контроллеров межпроцессорной связи (если таковые имеются), карты памяти и т.п. Ключевое слово "WIRE" используется для описания межпроцессорных связей. Так, например, связь "W1" указывает, что межпроцессорная связь 3 процессора "Root" соединяется с межпроцессорной связью 3 процессора "P2". На этом описание аппаратной модели в конфигурационном файле заканчивается.

После описания аппаратной модели в конфигурационном файле следует описание структуры программной модели в терминах задач, каналов и связей между ними. Для каждой задачи указывается число входных и выходных портов и объем необходимой памяти для стека и данных. Каналы описываются с помощью ключевого слова "CONNECT" и включают номера входных и выходных портов конкретных задач. Например, канал "C1" определяется как канал, связывающий первый выходной порт задачи "TaskA" с первым входным портом задачи "TaskC".

Наконец, аппаратная и программная модели ставятся в соответствие друг другу путем распределения четырех задач по трем процессорам.

Настоящий пример показывает, что среда 3L DIAMOND должна организовывать три различных типа каналов. Канал "C2" соединяет две задачи на одном процессоре, канал "C1" соединяет

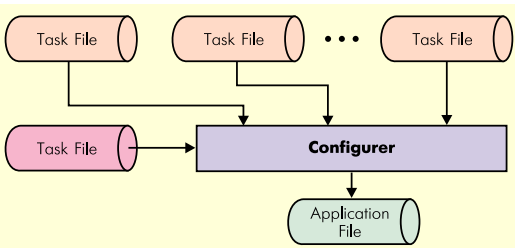


Рисунок 4 Конфигурация

и задаче "B" через канал "C2". Задача "B", в свою очередь, может посылать данные задаче "D", используя канал "C3". В приведенном примере задача "C" не может передавать данные задаче "D", так как между ними нет прямого канала связи.

В среде 3L DIAMOND задача представляет собой законченную полностью скомпонованную (слинкованную) исполнимую программу без внешних ссылок, написанную на языке C, которая запускается с функции *main*. Каждая задача имеет два вектора *портов ввода/вывода*, которые автоматически передаются ей в момент запуска (рис. 3). *Входные порты* определяют список каналов связи, с которых данная задача может читать сообщения, а *выходные порты* определяют список каналов связи, в которые данная задача может отправлять сообщения. Задача не знает (и не должна знать), что подключено с другого конца каждого канала связи, который к ней подсоединен: это могут быть либо другие задачи, либо какие-то аппаратные устройства ввода/вывода. Как только задача запускается, она проявляет себя как "поток" (*thread*), функционирующий под управлением диспетчера задач. Каждый "поток" может динамически порождать новый "поток", и число "потоков", одновременно исполняемых на одном процессоре, ограничено лишь размером памяти, доступной этому процессору. Среда 3L DIAMOND распределяет вычислительную мощность процессора между всеми "потоками", исполняемыми в данном процессоре, в соответствии с их приоритетами.

```

PROCESSOR Root      TYPE=MyType
PROCESSOR P2       TYPE=MyType
PROCESSOR P3       TYPE=MyType

WIRE W1 Root[3]    P2[3]
WIRE W2 Root[4]    P2[4]
WIRE W2 P[4]       P3[4]

TASK TaskA ins=0   outs=2   DATA=30K
TASK TaskB ins=1   outs=1   DATA=25K
TASK TaskC ins=1   outs=0   DATA=50K
TASK TaskD ins=1   outs=0   DATA=30K

CONNECT C1 TaskA[0] TaskC[0]
CONNECT C2 TaskA[1] TaskB[0]
CONNECT C3 TaskB[0] TaskD[0]

PLACE TaskA Root
PLACE TaskB Root
PLACE TaskC P2
PLACE TaskD P3

```

Рисунок 5

Конфигурационный файл

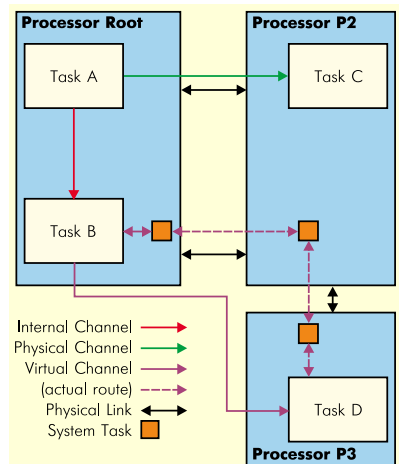


Рисунок 6 Типы каналов

две задачи на двух разных процессорах, которые физически соединены межпроцессорной связью, и канал "С3" соединяет две задачи на двух разных процессорах, которые физически не соединены межпроцессорной связью, но соединены между собой через другой процессор ("P2"). В терминах среды 3L DIAMOND, "С2" называется *внутренним каналом*, "С1" — *физическим каналом*, а "С3" — *виртуальным каналом* (рис. 6). Даже если это приложение использует различные типы каналов связи, конкретные задачи об этом не знают. Так, если бы мы захотели разместить задачу "TaskB" на процессоре "P2", то каналы "С2" и "С3" реально стали бы физическими каналами, однако никакой передачи ПО не потребовалось бы, включая перекомпиляцию и перелинковку. Для модификации приложения достаточно было бы перезапуска только конфигурирования.

Виртуальные каналы связи используются для передачи сообщений между задачами в случае, если нет непосредственной физической межпроцессорной связи между этими процессорами. Конфигуратор автоматически организует такой канал путем включения соответствующей системной задачи и таблицы маршрутизации там, где это необходимо, чтобы передавать сообщения через промежуточные процессоры. Маршрутизация выполняется полностью автоматически без задержек в работе приложения.

Результатом работы конфигурирования является единственный файл приложения в среде 3L DIAMOND, содержащий всю необходимую информацию для загрузки процессоров. Когда этот файл загружается в систему, то задачи и другие программные компоненты автоматически распределяются по конкретным процессорам, после чего происходит запуск приложения. Каждый процессор загружается только теми задачами, которые были явно указаны для выполнения на нём.

МИКРОЯДРО ОС

Потоки (*threads*) в среде 3L DIAMOND управляются с помощью компактного *микроядра ОС*, представляющего собой высоко оптимизированный код, осуществляющий управление диспетчеризацией задач, управление прерываниями и управления другими системными ресурсами. Размер микроядра ОС для среды 3L DIAMOND для процессоров

ЦОС TMS320C6000 составляет всего 5,5 Кбайт для кода и 1,5 Кбайт для данных. Другие системные компоненты (драйверы и т.п.) включаются в конечное приложение только тогда, когда это необходимо. В связи с тем, что высокая эффективность исполнительного кода является ключевым условием для процессоров ЦОС, то микроядро ОС для среды 3L DIAMOND было разработано и оптимизировано таким образом, чтобы внести только абсолютный минимум избыточности (эффективной потери производительности) в приложение пользователя, и активизируется только тогда, когда приложение делает непосредственный запрос к нему (за исключение таймера, который активизируется каждую миллисекунду на несколько процессорных тактов).

Все потоки в среде 3L DIAMOND диспетчеризируются с использованием простейшей схемы с "вытеснением" (*pre-emptive*) и "циклическими" (*round-robin*) приоритетами с помощью опционального механизма временного разделения (*time-slicing*). Диспетчеризация осуществляется очень быстро. Так, время полного переключения задач (известного как *время переключения контекста*) для мультипроцессора ЦОС *TORNADO-PX64xxQ* составляет всего 170 нс.

Среда 3L DIAMOND является открытой. Так, стандартное микроядро ОС для среды 3L DIAMOND для процессоров ЦОС TMS320C6000 содержит только минимум необходимого для работы самой среды 3L DIAMOND. Однако, при необходимости пользователь может добавить свои собственные системные модули драйверов устройств ввода/вывода, обработки прерываний и т.п., которые также будут автоматически загружаться конфигурированием среды 3L DIAMOND.

КОММУНИКАЦИЯ С УПРАВЛЯЮЩИМ КОМПЬЮТЕРОМ

Приложения, функционирующие в среде 3L DIAMOND, имеют полный доступ к сервисам на управляющем ПК. Это достигается с помощью приложения *Windows Server*, которое функционирует в среде *Windows*, осуществляет загрузку мультипроцессорных приложений и предоставляет полную поддержку функций ввода/вывода для всех задач приложения. Так, любая задача приложения в среде 3L DIAMOND может использовать функцию *printf* из любого процессора. В этом случае, механизм

виртуальных каналов среды 3L DIAMOND автоматически маршрутизирует запрос ввода/вывода между управляющим ПК и конкретной задачей.

Для мультипроцессора ЦОС *TORNADO-PX64xxQ* связь между управляющим ПК и процессорами ЦОС осуществляется через *HP*-порты процессоров ЦОС. Среда 3L DIAMOND не использует медленный *JTAG*-порт процессоров ЦОС, который используется средой разработки и отладки приложений ЦОС *Code Composer* фирмы *TEXAS INSTRUMENTS*. Очень важно, что встроенные аппаратные приложения, которые не используют в своем составе ПК, могут также разрабатываться и отлаживаться с помощью *Windows Server* среды 3L DIAMOND путем ввода/вывода диагностических сообщений на экран ПК и т.п., но после отладки могут быть перекомпилированы с запрещенными функциями ввода/вывода, чтобы исключить коммуникацию с ПК.

ПРОСТОТА

Среда 3L DIAMOND представляется обманчиво простой, и именно эта простота является главным привлекательным моментом. В то время как другие среды программирования процессоров ЦОС "наращивают слой за слоем" своей логической организации, предоставляя пользователю озадачивающе бесконечный набор сервисов и опций, среда 3L DIAMOND специально использует простейший способ создания приложений при сохранении максимальной эффективности, которая проявляется как в минимизации избыточности системы, так и в минимизации усилий разработчика. При этом, среда 3L DIAMOND "незримо" управляет большинством внутрисистемных задач и другими деталями, слабо влияющими на производительность приложений, предоставляя пользователю возможность сконцентрироваться на разработке конкретного приложения.

Управление размещением памяти является простым примером того, как среда 3L DIAMOND позволяет ускорить разработку приложений. Возвращаясь к рис. 5, можно увидеть, что единственное управление памятью задач осуществляется с помощью ключевого слова "DATA=", указывающего размер стека и области данных для задачи, и Вы нигде не встретите указания конкретного типа памяти и метода его использования. Конфигуратор среды 3L DIAMOND автоматически анализирует тип каждого

#include <stdio.h>

Рисунок 7

```
main()
{
    printf("Hello world\n");
}
```

Исходная программа HelloTask.c

PROCESSOR	Root	PX64xxQ
TASK	HelloTask	data=30K
PLACE	HelloTask	Root

Рисунок 8 Конфигурационный файл Hello.cfg

```
C> c6xc HelloTask.c
C> c6xlink HelloTask
C> config Hello
C> WS3L Hello
```

Рисунок 9 Команды создания и запуска приложения

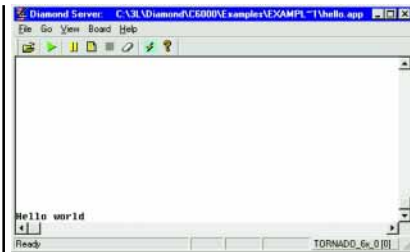
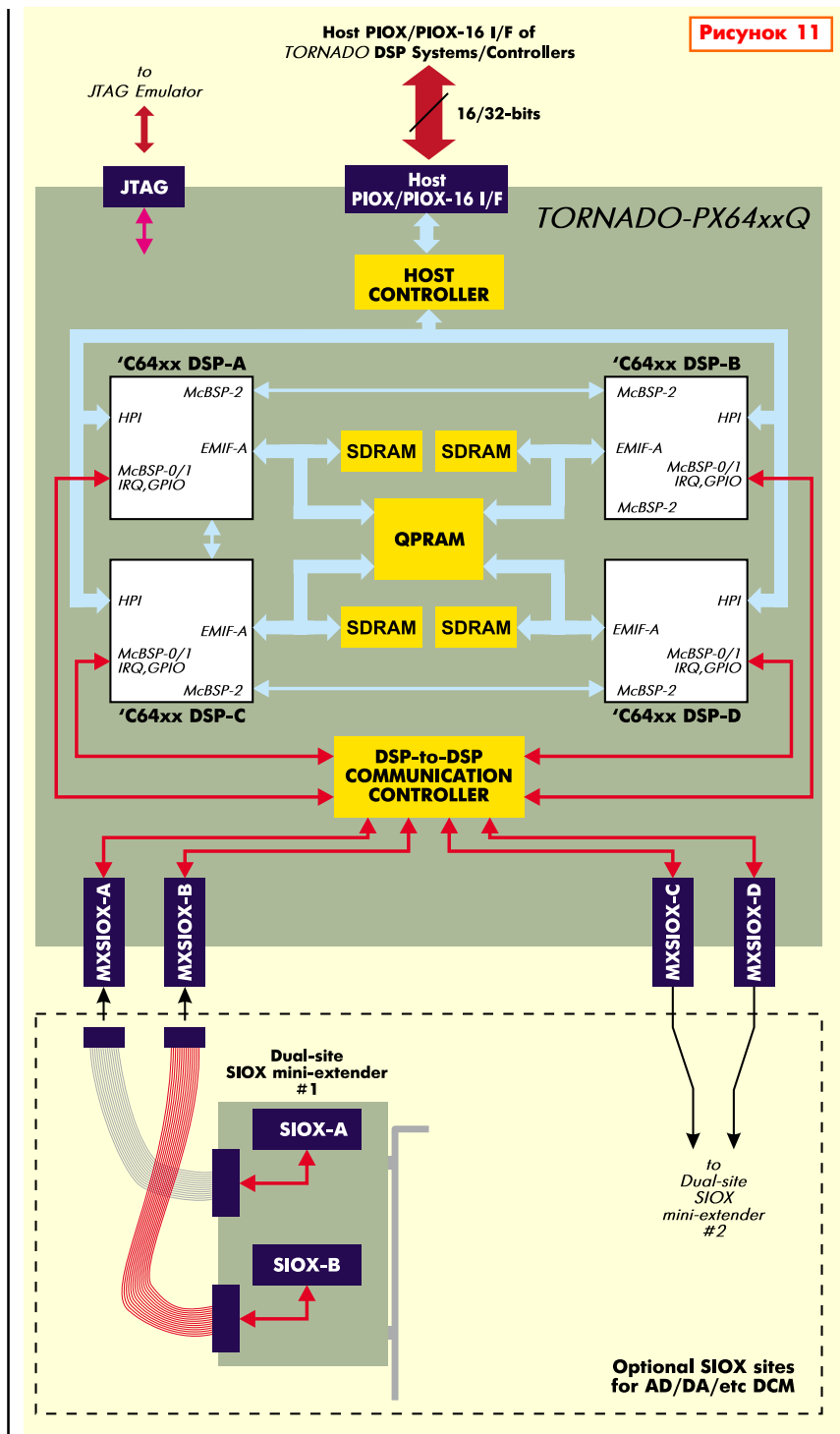


Рисунок 10 Окно Windows Server

процессора в системе, определяет размеры памяти и размещает секции кода и данных. Как только отладка логической части приложения будет закончена и наступит этап оптимизации, среда 3L DIAMOND позволяет пользователю осуществлять полный контроль над распределением памяти, который может включать в себя как простое приоритетное размещение секций кода/данных, так и явное указание адреса памяти для критических секций. Кроме того, конфигуратор среды 3L DIAMOND автоматически выделяет память для секций, которые явно не были указаны разработчиком. Интересно отметить, что начинающие пользователи среды 3L DIAMOND обычно предполагают, что им необходим максимально возможный контроль над распределением памяти, однако, в конце концов, используют самый простейший, обнаружив, что конфигуратор делает это сам практически идеально.

Независимость отдельных задач приложения приводит к простому изменению топологии системы. Добавление или удаление процессоров и соответствующая модификация приложения осуществляются очень просто и эффективно, в отличие от других сред, для ко-



Блок-схема мультипроцессора ЦОС TORNADO-PX64xxQ

торых изменение топологии системы и структуры памяти автоматически приводят к изменению исходного текста приложения.

На рис. 7 и 8 приведен пример, иллюстрирующий простоту процесса разработки приложений для среды 3L DIAMOND. В качестве примера используется простейшее приложение, которое выводит строку текста из одного из про-

цессоров мультипроцессора ЦОС ЦОС TORNADO-PX64xxQ.

На рис. 9 приведен список команд, необходимых для компиляции, компоновки, конфигурирования и запуска приложения из командной строки Windows, а на рис. 10 приведено окно Windows Server среды 3L DIAMOND, которое отображает результат работы приложения.

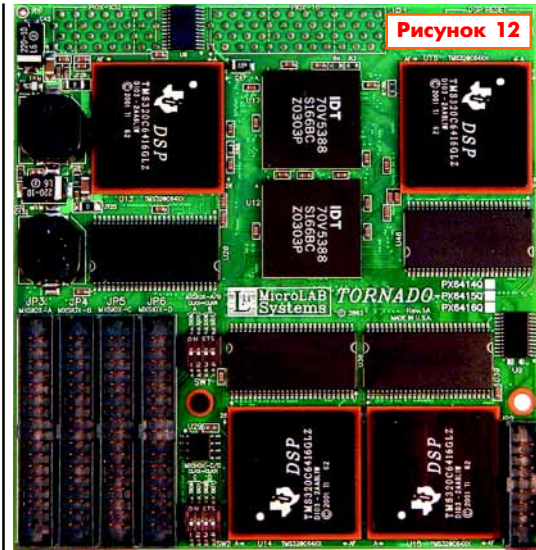


Рисунок 12

Мультипроцессор ЦОС TORNADO-PX64xxQ

ОБЛАСТИ ПРИМЕНЕНИЯ

Среда 3L DIAMOND успешно используется в мире уже в течение 16 лет в различных областях применения процессоров ЦОС: восстановление старых кинолент, стабилизация телескопов, подавление шума, подземные радары, сонары, управление оборудованием

на лесопилке (обработка изображений от нескольких рентгеновских камер для управления распилочным станком) и многое др. Типовые системы используют от одного до четырех процессоров ЦОС, однако имеется также и опыт применения среды 3L DIAMOND на сети из более чем 1000 процессоров.

Интеграция среды 3L DIAMOND и мультипроцессора ЦОС TORNADO-PX64xxQ фирмы "МикроЛАБ Системс" позволяет существенно упростить и ускорить разработку и обеспечить переносимость кода для мультипроцессорных приложений ЦОС. Очень важно отметить, что архитектура мультипроцессора ЦОС TORNADO-PX64xxQ и функционирующая на нём среда 3L DIAMOND отвечают требованиям производительности большинства мультипроцессорных приложений ЦОС и одновременно позволяют просто и эффективно наращивать количество устройств для приложений, требующих супервысокой производительности.

МУЛЬТИПРОЦЕССОР ЦОС TORNADO-PX64XXQ

Мультипроцессор ЦОС TORNADO-PX64xxQ фирмы "МикроЛАБ Системс" (рис. 11 и 12) построен на основе четырех процессоров ЦОС TMS320C64xx (C6414/C6415/C6416) фирмы TEXAS INSTRUMENTS и имеет суммарную производительность ЦОС 23,040 MIPS.

Мультипроцессор ЦОС TORNADO-PX64xxQ имеет два типа межпроцессорных связей (общая память и последовательные каналы), позволяет объединение неограниченного числа таких же и совместимых с ним устройств (сопроцессоры цифрового радиоприема и др.) в единую мультипроцессорную систему, а также обеспечивает возможности ввода/вывода сигналов в реальном времени с внешних модулей АЦП/ЦАП и различных специализированных устройств. Детальное описание можно найти на www.mlabsys.com/t_copr.htm.

С вопросами по среде 3L DIAMOND обращайтесь на фирму "МикроЛАБ Системс" — авторизованному дистрибьютору фирмы 3L в России. Тел./факс: +7 (095) 900-6208; e-mail: info@mlabsys.com.

ARGUSSOFT
ДЕПАРТАМЕНТ МИКРОЭЛЕКТРОНИКИ

ЗАО "АРГУССОФТ Компани"

ОФИЦИАЛЬНЫЙ
ДИСТРИБЬЮТОР
Ф И Р М :

ATMEL

AVR-микроконтроллер со встроенным контроллером ЖКИ ATmega169/169L/169V

Основные технические характеристики

ATmega169 - микропотребляющий 8-разрядный КМОП-микроконтроллер с контроллером ЖКИ на кристалле и типовым потреблением не более 40 мкА в активном режиме на частоте 32,768 кГц при напряжении питания 1,8 В.



Для отладки систем на базе микросхемы ATmega169 предлагается стартовый набор ATSTK502, содержащий микросхему ATmega169 и ЖК-индикатор. В комплект также входит компакт-диск с примерами программ.

- Тактовая частота 0...16 МГц (0...8 МГц для версии L и 0...1 МГц для версии V)
- Встроенный RC-генератор 1/2/4/8 МГц с возможностью калибровки
- 4 внешних источника тактирования
- 130 команд (большинство команд выполняется за 62,5 наносекунды)
- Аппаратный умножитель (время выполнения команды умножения 8 x 8 - 125 наносекунд)
- 16 Кбайт загружаемой в системе Flash-памяти (не менее 10000 циклов перезаписи)
- 512 байт ЭСПЗУ (не менее 100 000 циклов перезаписи)
- 1 Кб ОЗУ
- 32 рабочих регистра
- 53 программируемых линии ввода/вывода
- SPI-интерфейс для внутрисхемного программирования
- Драйвер ЖК-индикатора 25 x 4 сегмента
- Два 8-битных таймера/счетчика с раздельными делителями и режимами сравнения
- Один 16-битный таймер/счетчик с режимами сравнения и захвата
- Таймер часов реального времени с отдельным внешним тактовым генератором
- Четыре канала ШИМ
- Программируемый сторожевой таймер со встроенным генератором
- Полнодуплексный последовательный порт UART/USART
- Интерфейс SPI с режимом Master/Slave
- Интерфейс USI с режимами SPIMaster/Slave и I2C Master/Slave
- Встроенная схема сброса при подаче напряжения питания и схема слежения за питанием
- Аналоговый компаратор
- 8-канальный 10-битный АЦП, время преобразования 65 - 270 мкс
 - внутренний источник опорного напряжения 1,1 В
- JTAG-порт для подключения внутрисхемного эмулятора ATJTAGICE
- Типовой потребляемый ток в активном режиме:
 - 1 МГц, 1,8 В - 400 мкА
 - 32 кГц, 2,7 В - 30 мкА (с учетом тока генератора)
 - 32 кГц, 1,8 В - 20 мкА (с учетом тока генератора)
 - 32 кГц, 1,8 В - 40 мкА (с учетом тока генератора и ЖКИ)
- Пять программируемых режимов пониженного энергопотребления
 - потребляемый ток в режиме Power-down - 0,5 мкА (при 1,8 В)
- Диапазон напряжения питания от 1,8 до 5,5 В (для ATmega169V)
- Высоконадежная защита от несанкционированного копирования содержимого ПЗУ
- 64-выводной корпус TQFP и MLF

ANALOG DEVICES

ATMEL

TRACO POWER

BOYDENS

CLARE

SRC DEVICES

IXYS

tyco Electronics

BRAUN

Communication GmbH

HANTRONIX

Honeywell

Москва
Проспект Мира, 95
Тел.: (095) 217-2487/2519/2505
Факс: (095) 216-6642
E-mail: components@argussoft.ru

Санкт-Петербург
ул. Бабюшкина, д.3, оф.315
Тел./факс: (812)567-1849
E-mail: spb@argussoft.ru

Новосибирск
ул. Советская, д.65
Тел./факс: (3832) 22-40-31
E-mail: nsk@argussoft.ru

Екатеринбург
ул. Первомайская, д.104,
оф. 206/5
Тел./факс: (3432) 78-32-42, 78-32-41
E-mail: ural@argussoft.ru

Интернет: <http://components.argussoft.ru> <http://atmel.argussoft.ru>