Программно-аппаратная реализация стека протоколов TCP/IP в процессорах компании Texas Instruments

Игорь ГУК
dsp@scanti.ru
Алексей КУЗМЕНКО
fannast@mail.ru

В настоящее время основным используемым семейством протоколов является стек протоколов TCP/IP (Transmission Control Protocol/Internet Protocol). Это собирательное название для сетевых протоколов разных уровней, используемых в сетях. Термином TCP/IP зачастую обозначают все, что относится к сетевым протоколам TCP и IP, другие протоколы, приложения и даже сетевые среды.

История TCP/IP начинается с того момента, когда Министерство обороны США столкнулось с проблемой объединения большого числа компьютеров с различными операционными системами.

Для описания взаимодействия сетевых устройств была предложена сетевая модель взаимодействия открытых систем (модель взаимодействия открытых систем (ВОС) — англ. Open Systems Interconnection Reference Model-OSI) — абстрактная модель для сетевых коммуникаций. Данная модель реализует уровневый подход к описанию сети. Каждый уровень обслуживает свою часть процесса обмена данными. Благодаря такой структуре совместная работа сетевого оборудования и программного обеспечения становится гораздо проще и понятнее.

За все время существования модели ВОС она не была реализована и, возможно, не будет реализована никогда. Модель слишком сложна, а ее реализация займет слишком много времени. Сегодня используется только некоторое подмножество модели ВОС. Реальные сетевые протоколы вынуждены отклоняться от нее.

Существует несколько мнений о том, как вписать модель TCP/IP в модель ВОС, поскольку уровни в этих моделях не совпадают. Наиболее распространенный подход в согласовании двух моделей представлен в таблице 1.

Структура набора протоколов TCP/IP имеет четыре уровня: уровень сетевого интерфейса (уровень доступа к сети), сетевой уровень (межсетевой уровень), транспортный уровень (уровень взаимодействия сетевых устройств — хостов) и прикладной уровень (уровень обработки).

Уровень доступа к сети управляет обменом данными между сетевыми устройствами в пределах одной сети.

Сетевой уровень — уровень межсетевого взаимодействия; он обеспечивает передачу данных между различными сетями. Сетевой протокол, осуществляющий маршрутизацию, выполняется не только на локальных сетевых устройствах, но и на шлюзах, которые соединяют две сети. Сетевой уровень принимает запрос на посылку пакета от транспортного уровня вместе с указанием адреса получателя. Уровень инкапсулирует пакет в дейтаграмму, заполняет ее заголовок и при необходимости использует алгоритм маршрутизации. На стороне получателя на сетевом уровне удаляется заголовок дейтаграммы и определяется, какой из транспортных протоколов будет обрабатывать пакет.

Транспортный уровень обеспечивает доставку сообщений между двумя хостами. В его задачи входит обнаружение и исправление ошибок передачи пакетов, обеспечение взаимодействия между прикладными программами. Транспортный уровень управляет потоком информации и обеспечивает высокую надежность передачи. Он принимает данные от нескольких прикладных программ и посылает их более низкому уровню. При этом транспортный уровень добавляет каждому пакету дополнительную информацию, в том числе контрольную сумму.

Таблица 1. Соответствие стека протоколов TCP/IP модели ВОС

OSI (ВОС) (7 уровней)	TCP/IP (4 уровня)	Протоколы
Прикладной	Прикладной	FTP (TFTP), HTTP, SNMP, SMTP, MIME, Telnet, DNS, RIP, OSPF, POP3, NNTP, NAT
Представительский		
Сеансовый		
Транспортный	Транспортный	TCP, UDP
Сетевой	Сетевой (межсетевой)	IP, ICMP, IGMP, ARP, RARP, EGP, BGP
Канальный	Уровень доступа к сети	SLIP, PPP
Физический		

Прикладной уровень обеспечивает протоколы всем необходимым для поддержки различных прикладных программ конечного пользователя, таких как пересылка файла или электронная почта. Каждая прикладная программа выбирает тип транспортировки либо последовательность отдельных сообщений, либо их непрерывный поток.

В данное время развитием семейства протоколов TCP/IP управляет международная общественная организация, именуемая Сообществом Интернета (Internet Society, ISOC). Стандарты для TCP/IP публикуются в сериях документов, называемых RFC (Request for Comments). Хотя Интернет не является собственностью ни одной организации, некоторые из них отвечают за управление им.

Набор протоколов TCP/IP предоставляет пользователям две основные службы, которые применяют прикладные программы: негарантированное дейтаграммное средство доставки пакетов и гарантированное потоковое транспортное средство.

- IP — дейтаграммное средство доставки пакетов (доставка без установления соединения). Определяет маршрут только на основе адресной информации, содержащейся в сообщении. Применяется при передаче небольших объемов информации, например, игры в реальном масштабе времени.

- TCP — гарантированное потоковое транспортное средство (с установлением соединения). При установлении соединения предполагается, что получатель готов к приему данных от конкретного отправителя. Это означает, что все параметры взаимодействия согласованы и у получателя выделены соответствующие ресурсы для обеспечения приема. При этом обеспечивается восстановление данных при ошибках, потере, неправильном порядке приема пакетов.

Протокол IP обеспечивает доставку данных между двумя (или более) компьютерами. Однако на одном узле может функционировать параллельно несколько программ, которым требуется доступ к сети. Следовательно, данные внутри компьютерной системы должны распределяться между программами. Поэтому при передаче данных по сети недостаточно просто адресовать конкретный узел. Необходимо также идентифицировать программу-получателя, что невозможно осуществить средствами сетевого уровня.

Другой серьезной проблемой IP является невозможность передачи больших массивов данных. Протокол IP разбивает передаваемые данные на пакеты, каждый из которых передается в сеть независимо от других. В случае если какие-либо пакеты потерялись, то модуль IP на принимающей стороне не может обнаружить потерю.

Для решения этих проблем и разработаны протоколы транспортного уровня TCP. Идентификация программ в протоколах TCP обеспечивается уникальными числовыми значениями, так называемыми номерами портов. Для каждого протокола существуют стандартные списки соответствия номеров портов и программ (пример на рис. 1).

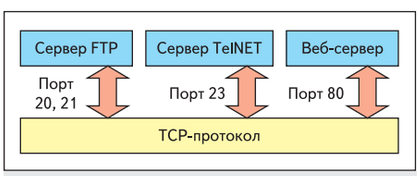

Рис. 1. Пример соответствия портов и программ

Таким образом, протокол сетевого уровня IP и транспортные протоколы TCP реализуют двухуровневую схему адресации: номер порта позволяет однозначно идентифицировать программу в рамках сетевого устройства (компьютера), который, в свою очередь, однозначно определяется IP-адресом. Таким образом, комбинация IP-адреса и номера порта однозначно идентифицирует программу в сети. Такой комбинированный адрес называется сокетом (socket).

Кроме тенденции расширения области применения сетевых технологий, есть еще одно направление развития техники — внедрение цифровых методов обработки различных видов информационных потоков. Эти две тенденции, являясь по сути различными, очень связаны друг с другом. Так как передавать информацию по сети наиболее удобно именно в цифровом виде, а в то же время, очень часто источником информации является аналоговое устройство. Таким образом, с ростом степени проникновения сетевых технологий в сферу жизнедеятельности человека все чаще возникает задача, с одной стороны, обеспечить двунаправленное цифро-аналоговое преобразование исходного и принятого сигналов, а с другой — обеспечить сетевое взаимодействия между двумя хостами. Это в свою очередь обуславливает актуальность построения устройств, обладающих как возможностью сетевого подключения, так и способное производить цифровую обработку исходной аналоговой информации.

Промышленность (к большому сожалению, только западная) очень чутко откликнулась на потребность в таких устройствах и предложила DSP-процессоры с возможностью простой реализации сетевых интерфейсов.

Выбирая цифровой сигнальный процессор, необходимо представлять существующий рынок DSP, знать основных производителей и направления развития создаваемых ими процессоров. В таблице 2 представлен приблизительный перечень основных производителей и их примерная доля рынка производства DSP-процессоров.

Производители DSP достаточно четко позиционируют свои изделия для использования в тех или иных приложениях. Это оказывает влияние и на их архитектуру, и на быстродействие, и на оснащение процессора тем или иным набором периферийных модулей. В таблице 3 показано позиционирование производителями своих DSP.

Таблица 2. Основные производители DSP и принадлежащие им доли рынка

Место	Название фирмы	Доля рынка DSP
1	Texas Instruments	54,3%
2	Freescale Semiconductor	14,1%
3	Analog Devices	8,0%
4	Philips Semiconductors	7,5%
5	Agere Systems	7,3%
6	Toshiba	4,9%
7	DSP Group	2,2%
8	NEC Electronics	0,6%
9	Fujitsu	0,4%
10	Intersil	0,3%
11	Другие фирмы	0,5%

При анализе данных, представленных в таблицах 2, 3 обращает на себя внимание продукция компании Texas Instruments (TI). Эта компания обладает более чем половиной рынка сигнальных процессоров и предлагает решения для широкого круга задач.

При выборе устройства, с которым придется работать программистам, есть и еще один очень важный критерий — простота работы с процессором. При работе с DSP компании Texas Instruments используется интегральная среда разработки Code Composer Studio (CCS). Это единая интегрированная среда, которая включает текстовый редактор с подсветкой синтаксиса, Си-компилятор, ассемблер, высокоэффективный оптимизатор кода, компоновщик, загрузчик программного кода. Кроме этого, CCS обеспечивает поддержку аппаратного отладчика, который позволяет получить доступ в режиме реального времени ко всем модулям процессора.

Для реализации сетевых технологий на базе цифрового сигнального процессора необходимо, чтобы выбранный DSP имел соответствующую аппаратную и программную поддержку. Этому условию в полной мере соответствуют процессоры компании TI — TMS320C6455.

Главная особенность процессора TMS320C6455 состоит в наличии встроенного в чип контроллера 10/100/1000 Мбит/с Ethernet MAC (EMAC — Ethernet Media Access Control), являющегося связующим звеном между ядром процессора и сетью. Данный элемент периферии DSP в совокупности с соответствующим программным обеспечением и образует физический уровень (MAC-уровень) Ethernet.

Основная задача этого уровня — на передаче собрать кадр Ethernet из данных, которые пришли с верхних уровней, программно реализованных на процессоре, и отправить его в линию, адресату; на приеме — выбрать адресованный ему кадр из потока данных, разобрать его, а извлеченные данные отправить на обработку в верхние уровни.

Таблица 3. Области применения сигнальных процессоров разных производителей

Функция	Процессор
Обработка видео, видеонаблюдение, цифровые камеры, 3D-графика	**TI:** TMS320C6455, TMS320DM64x/DaVinci, OMAP35xx **Philips:** PNX1300, PNX1500, PNX1700 **Freescale:** MPC52xx
Обработка аудио, распознавание речи, синтез звука	**TI:** TMS320C62xx, TMS320C67xx **Analog Devices:** SHARC
Портативные медиаустройства	**TI:** TMS320C54xx, TMS320C55xx, OMAP35xx **Analog Devices:** семейство Blackfin
Беспроводная связь, телекоммуникации, модемы, сетевые устройства	**TI:** TMS320DM64x/DaVinci, TMS320C64xx, TMS320C64x+, TMS320C54xx, TMS320C55xx **Freescale:** MPC7xxx, MPC86xx, MPC8xx PowerQUICC I, MPC82xx PowerQUICC II, MPC83xx PowerQUICC II Pro, MPC85xx PowerQUICC III **Analog Devices:** Blackfin, TigerSHARC **Philips:** PNX1300
Управление приводами, преобразование мощности, автомобильная электроника, предметы домашнего обихода, офисное оборудование	**TI:** TMS320C28xx, TMS320C24xx **Analog Devices:** ADSP-21xx **Freescale:** MPC55xx, MPC55xx
Медицина, биометрия, измерительные системы	**TI:** TMS320DM64x/DaVinci, TMS320C62xx, TMS320C67xx, TMS320C55xx, TMS320C28xx, OMAP35xx **Analog Devices:** TigerSHARC, SHARC

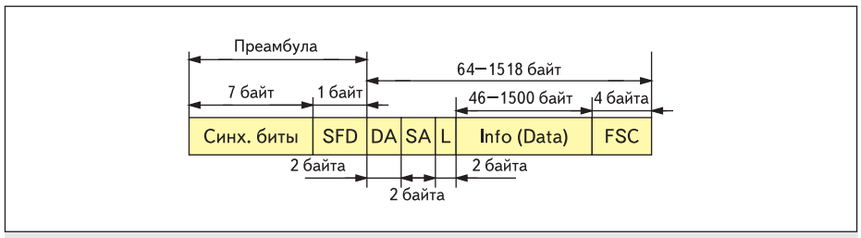

Рис. 2. Структура кадра сети Ethernet

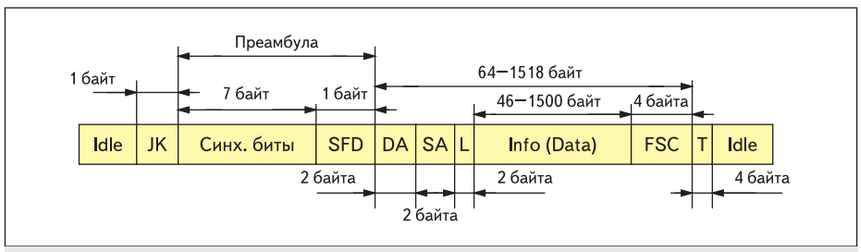

Рис. 3. Структура кадра сети Fast Ethernet

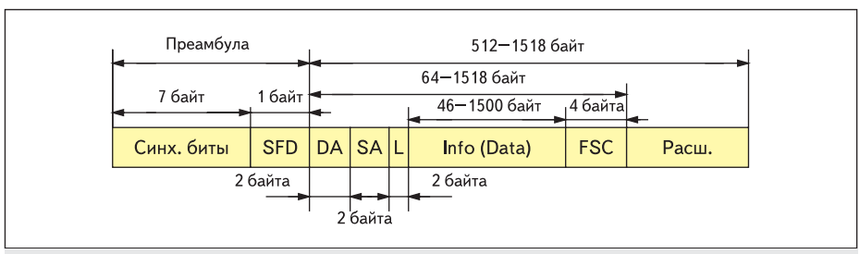

Рис. 4. Структура кадра сети Gigabit Ethernet

Нетрудно заметить, что EMAC способен работать не только с Ethernet, но также с Fast Ethernet и Gigabit Ethernet. Интерфейсы у этих сетей одинаковые — RJ45, одинаков и способ доступа станций к среде передачи, но в структуре кадров у них имеются некоторые различия. Структура кадров представлена на рис. 2–4.

На рис. 2–4 применены следующие обозначения:

- Синхробиты — последовательность 7 байт, каждый из которых равен значению 10101010.
- SFD (Start of Frame Delimiter) — метка начала кадра — 10101011.
- DA (Destination Address) — адрес назначения.
- SA (Source Address) — адрес отправления.
- L (Length) — длина поля данных.
- Info (Data) — поле данных, после передачи данных передаются дополнительные поля для их выравнивания.
- FCS (Frame Check Sequence) — проверочная последовательность кадра, формируется методом CRC-32.
- Idle — отсутствие передачи кадров, передается последовательность единиц.
- JK — символы, означающие начало передачи кадра — J=11000, K=10001 (в Fast Ethernet).
- T — символ, означающий окончание кадра — 01101.
- Расш. (расширение носителя) — не несет информации и применяется для увеличения длительности кадра, которая, в свою очередь, должна превышать время двойного оборота, ограничивающего максимальную длину сети и минимальную длительность кадра. Это необходимо для того, чтобы передающая станция имела возможность гарантированно обнаружить коллизии.

Для реализации сетевых возможностей в DSP компания компании Texas Instruments предложила реализацию стека протоколов TCP/IP, созданную непосредственно для работы с процессорами TI. Основные компоненты предложенной реализации стека протоколов представлены на рис. 5.

Для создания приложений TCP/IP компаниями TI и Spectrum Digital совместно был представлен совершенно новый набор программных компонентов, получивший название Network Developer Kit (NDK). Основная задача NDK — облегчить разработку сетевых приложений, сделать более наглядным и привычным выполнение всех функций сети.

До сих пор программисты, занимавшиеся сетевыми задачами, вынуждены были вести разработку на физическом уровне. При этом ни о каком исполнении приложений на прикладном уровне не могло быть и речи, приходилось последовательно, шаг за шагом, уровень за уровнем расписывать имеющимися средствами каждый протокол стека. Совершенно очевидно, что эта задача является очень трудоемкой и сложной по исполнению. Стоит также отметить, что к имеющимся средствам можно отнести ассемблер, некоторые команды языка C/C++, поддерживаемые интегрированной средой разработки CCS, а также возможности DSP/BIOS.

Самое главное достоинство NDK заключается в том, что в нем реализованы функции и команды, традиционные для языка C/C++ по части сети. NDK дает возможность созда-

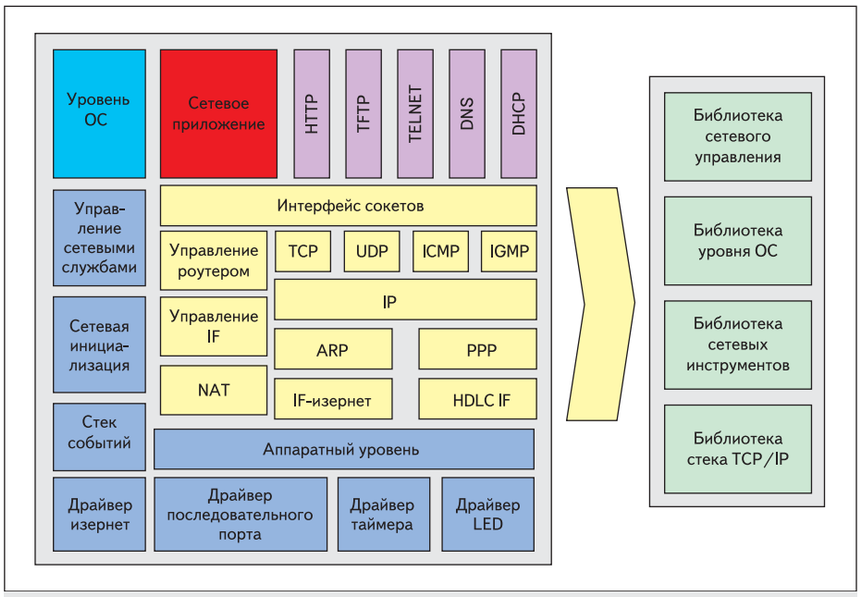

Рис. 5. Компоненты стека протоколов TCP/IP для процессоров компании TI

вать сетевые приложения с помощью сокетов Беркли.

Таким образом, этот набор снимает необходимость работы программиста на самом нижнем уровне, а также позволяет не заботиться о написании всего стека на верхних уровнях в силу того, что представленный в NDK стек TCP/IP является полнофункциональным и отвечает всем требованиям. Программист может теперь выполнять поставленные перед ним задачи на привычном прикладном уровне. Это существенно сокращает время разработки приложений, повышает производительность, а также приводит к наиболее рациональному использованию ресурсов сигнального процессора.

NDK представляет собой набор библиотек, каждая из которых является определенным уровнем абстракции, составляющим часть общей структуры стека TCP/IP:

- STACK.LIB;
- NETTOOL.LIB;
- OS.LIB и MiniPrintf.LIB;
- HAL.LIB;
- NETCTRL.LIB.

Взаимосвязь компонентов NDK представлена на рис. 6.

NDK связывается с операционной системой и аппаратным обеспечением низкого уровня с помощью различных программных интерфейсов. Операционная система OS представляется в виде уровня абстракции операционной системы (OS.LIB), а требуемое аппаратное обеспечение поддерживается библиотекой уровня абстракции аппаратных средств (HAL.LIB). Эти библиотеки используются для связи стека с DSP/BIOS и периферией системы.

Библиотека STACK.LIB является главной в реализации структуры стека TCP/IP. Она содержит все: от самого верхнего уровня сокетов до нижнего уровня Ethetnet и протокола PPP (Point to Point protocol). Библиотека собрана для использования операционной системы DSP/BIOS и не нуждается в перестроении, когда идет переход с одной платформы на другую. В NDK включены различные сборки данной библиотеки. Различные ее варианты могут включать либо исключать такие особенности, как протокол PPP или PPPoE (PPP over Ethernet), а также протокол NAT (Network Address Translation).

Библиотека NETTOOL.LIB содержит все сетевые службы, основанные на сокетах, которые поддерживает NDK, плюс несколько дополнительных инструментов проектирования для помощи при разработке сетевых приложений. Наиболее часто используемый компонент данной библиотеки — это система конфигурации, основанная на тегах (метках). Система конфигурации управляет практически каждым элементом стека и его службами. Конфигурации могут храниться в энергонезависимой ОЗУ для автоматической загрузки при включении.

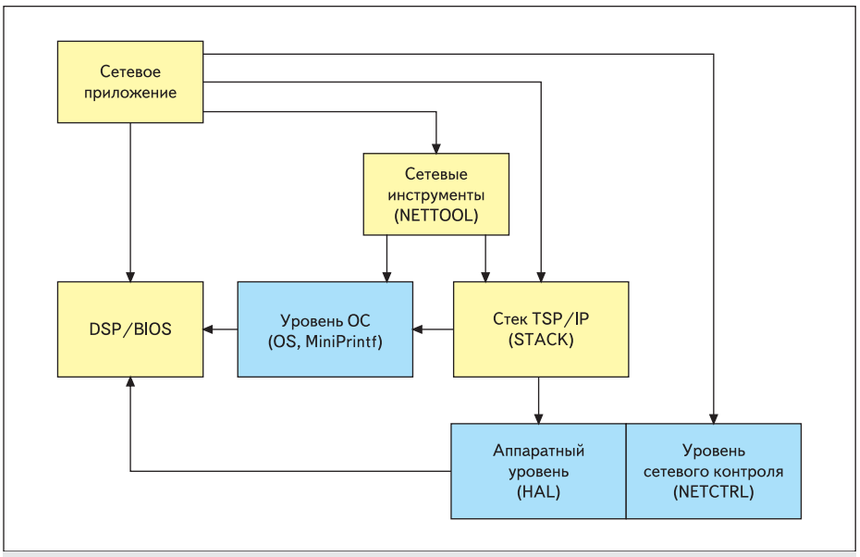

Рис. 6. Взаимосвязь компонентов NDK

Библиотеки OS.LIB и MiniPrintf.LIB. Эти библиотеки формируют небольшой уровень абстракции, который преобразует некоторые вызовы функций OS в вызовы функций DSP/BIOS. Этот уровень абстракции позволяет программисту системы DSP/BIOS настраивать систему NDK на любую OS, основанную на DSP/BIOS. Он включает управление нитью задач, распределение памяти, управление буферами, вывод на экран, запись, секционирование и связь с кэшем.

MiniPrintf.LIB обеспечивает особые, «небольшие» функции вывода, помогающие сохранить маленькую нагрузку на процессор. Они упакованы в отдельную библиотеку, поэтому можно использовать либо полновесные RTS-функции вывода, обеспечиваемые Code Composer Studio, либо маленькие функции, включенные в библиотеку MiniPrintf.LIB.

Библиотека HAL.LIB содержит файлы, которые связывают периферию аппаратного обеспечения с NDK. К этой периферии относятся таймеры, лампочки-индикаторы, устройства Ethernet и последовательные порты. Фактически уровень абстракции аппаратного обеспечения, образуемый этой библиотекой, — это и есть набор драйверов для перечисленных блоков периферии.

Необходимо также отметить особенность программной HAL-поддержки NDK, относящуюся к выполнению задачи непосредственно на DSP C6455.

Для разработки сетевых приложений на процессоре TMS320C6455 специалисты создали два особых драйвера: Ethernet (hal_eth_c6455.lib) и user LED (hal_userled_c6455.lib). Они содержат конфигурацию, предназначенную для работы именно с данным ЦПОС, и рекомендуются к использованию при работе с C6455 для того, чтобы наиболее точно выстроить систему и освободить программиста от необходимости воссоздавать нужную конфигурацию из стандартных драйверов.

Библиотеку NETCTRL.LIB (Network Control library) можно считать центром стека. Она управляет взаимодействием между TCP/IP и внешним миром. Из всех модулей стека этот является самым важным для работы NDK. В его обязанности входит:

- Инициализация NDK и драйверов устройств низкого уровня.
- Загрузка и поддержание конфигурации системы.
- Организация связи с драйверами устройств и установление порядка драйверов для вызова в NDK.
- Выгрузка конфигурации системы и очистка драйверов при выходе.

Кроме структурного скелета стека TCP/IP, базирующегося на только что описанных библиотеках и их взаимосвязях, для обеспечения работы с сетью необходимы конкретные программы, приложения, которые и будут выполнять поставленные задачи. Для того чтобы разработчик мог заниматься созданием этих приложений, его необходимо снабдить соответствующим набором команд и функций. Эти команды находятся в наборе заголовочных файлов, подключаемых к создаваемому проекту, главным из которых является NETMAIN.H. Как уже сказано, данные функции и команды являются традиционными для использования сокетов Беркли при создании соединений. Они практически полностью совпадают с аналогами в среде разработки Visual Studio C++, довольно понятны и просты в применении. Например:

- socket() — создание сокета;
- listen() — прослушивание сети, ожидание соединения;
- connect() — установление соединения;
- accept() — прием запроса на установление соединения.

Проекты, созданные при помощи NDK, предполагают использование DSP/BIOS. Конфигурация системы выполняется практически в автоматическом режиме; она может быть создана с помощью готовых tci-файлов, содержащихся в CCS, причем настроена сразу же под конкретную плату, на которой ведется разработка; ее также может настраивать программист, но уже не только в текстовом редакторе, но и в графическом, что дает наибольшую наглядность. Файл конфигурации, созданный DSP/BIOS, также берет на себя такое ответственное задание, как распределение памяти.

Использование DSP/BIOS подразумевает работу с так называемыми нитями (threads), которые делятся на три класса:

- HWI — hardware interrupt — аппаратные прерывания;
- SWI — software interrupt — программные прерывания;
- TASK — задачи.

Эти нити различаются по степени приоритетности выполнения: сначала выполняются HWI, затем SWI и наконец TASK. Внутри класса также существует несколько ступеней приоритета, опираясь на которые, разработчик задает событие для выполнения требуемого действия. При помощи графических и тестовых средств DSP/BIOS можно наблюдать за процессом выполнения нитей в реальном времени.

Еще одним достоинством DSP/BIOS является наличие своих собственных функций и команд — например, команд вывода, управления, диагностики, не затрудняющих работу процессора, а зачастую и разгружающих его. В качестве примера приведем функцию вывода:

- стандартная функция вывода на экран — printf() — выполняется в течение 34 000 циклов, что без особой надобности существенно замедляет работу DSP;
- функция DSP/BIOS вывода на экран (только не в окно Stdout, а собственное окно объекта регистрации LOG) — LOGprintf() — выполняется всего лишь за 32 цикла.

Эти функции не входят в окончательный вариант разрабатываемого кода, но значительно облегчают отладку.

Многие специалисты, занимающиеся программированием ЦСП, отмечают стройность, четкость и качественное выполнение программных кодов, созданных с помощью DSP/BIOS.

Таков краткий обзор возможностей сигнальных процессоров компании TI для реализации сетевых задач. В следующей статье рассмотрим пример использования NDK на базе одной из стандартных отладочных плат, выпускаемых компанией Spectrum Digital.