

90-515

Л Я



сообщения
объединенного
института
ядерных
исследований
дубна

P10-90-515

В.И.Приходько, Я.Судек

ЦИФРОВЫЕ СИГНАЛЬНЫЕ ПРОЦЕССОРЫ

1990

Введение

В обзоре рассматриваются техника цифровой обработки сигналов и структура сигнальных процессоров, формулируются требования к ним и к их блокам, показаны подходы к программному обеспечению, технологии и способам оценки производительности цифровых сигнальных процессоров, дается краткий анализ развития этой техники и приведены примеры ее применения. В конце приводится список литературы с краткими заметками о содержании статей.

Цифровые сигналы

В тексте для "цифровой обработки сигнала" будем использовать название "dsp" (digital signal processing). Для "цифрового сигнального процессора" будем употреблять название "DSP" (Digital Signal Processor). Название ICH-DSP будем употреблять в случае, если речь пойдет об однокристалльных DSP; и BB-DSP - в случае DSP, созданного из блоков.

DSP обрабатывают цифровые сигналы, т.е. сигналы, дискретные по амплитуде и во времени. Они представляют собой последовательность двоичных чисел.

Цифровые сигналы могут быть:

- искусственного происхождения (генерирует их, например, компьютер);
- аналогового происхождения.

Цифровой сигнал аналогового происхождения возникает при выборке и оцифровке входного аналогового сигнала. Основным требованием к выборке является выполнение теоремы Шеннона-Котельникова, которая гласит, что любой аналоговый сигнал с полосой частоты, ограниченной частотой F_1 , можно однозначно заменить последовательностью дискретных значений, полученных выборкой из непрерывного сигнала в дискретные моменты времени с частотой F_s . Если выполнено требование

$$F_s > 2 \cdot F_1, \quad (1)$$

то из таких отсчетов можно однозначно восстановить аналоговый сигнал.

Чтобы не допустить искажения спектра, необходимо ограничить полосу частоты входного сигнала фильтром (antialiasing filter) и выполнить таким образом требование (1).

При оцифровке амплитуды сигнала в дискретные моменты времени возникает цифровой сигнал, который дискретен и по амплитуде. Такой сигнал можно обработать при помощи DSP. Если нужно выполнить преобразование обработанного сигнала обратно в аналоговый сигнал, необходимо после выходного ЦАП включить "реконструкционный" фильтр, который удаляет частоты, не соответствующие реальному сигналу и являющиеся следствием оцифровки.

Что такое цифровая обработка сигнала (dsp)?

Это применение математического алгоритма к обработке цифрового сигнала, полученного, как правило, из АЦ-преобразователя. Алгоритм в большинстве случаев периодически повторяется. Инструкции, которые реализуют алгоритм, вызывают из памяти данные и коэффициенты и затем обрабатывают их. Далее выходной цифровой сигнал обрабатывается в цифровой форме или преобразуется снова в аналоговый сигнал. Общая схема dsp представлена на рис.1.

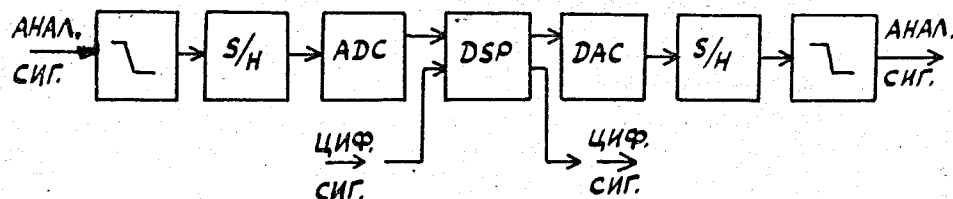


Рис.1. Общая схема dsp.

Когда возникла необходимость dsp?

В некоторых случаях обрабатывать сигнал аналоговыми методами или очень трудно, или невозможно, например, в случае обработки двумерных изображений. Ранее такие задачи решались классическими методами вычислительной техники на универсальных компьютерах. Обработка происходила в большинстве случаев вне реального времени (off-line).

Затем для некоторых сложных задач (в телекоммуникации, научных исследованиях, военной технике) были созданы на базе интегральных схем (ИС) малой (SSI) и средней (MSI) степени интеграции блоки сигнальных процессоров, которые работали в реальном времени.

Появление микропроцессоров (МП) в 70-х годах расширило сферу применения dsp (off-line). Для DSP быстро возник огромный рынок - телекоммуникация. В 1979г. появился первый специализированный DSP (Intel i2920), позже другие DSP. С сегодняшними DSP их даже трудно сравнить (i2920 имел 18000 транзисторов, а современный TMS320C50 уже 1000000), но они открыли дверь для применения методов цифровой обработки в задачах, для решения которых до тех пор применялась только аналоговая техника (например, в телекоммуникации начали применять DSP вместо аналоговых фильтров).

Где используется dsp?

Сейчас рынок DSP составляет ежегодно около миллиарда долларов и продолжает расти примерно на 30% в год. Область применения DSP распространяется от игрушек до космической техники. Можно указать следующие области и задачи, где использование DSP уже сейчас дает значительный эффект:

- цифровая фильтрация в реальном времени,
- спектральный анализ сигналов в реальном времени (FFT),
- анализ и синтез речи,
- управление при помощи голоса,
- звукотехника (CD, эквалайзеры),
- передача данных и компрессия данных,
- шифраторы и дешифраторы в сетях связи,
- цифровая обработка биосигналов в медицине,
- цифровая обработка сигналов в геофизике,
- корреляционный анализ,
- цифровая обработка изображений, в том числе и в коммерческой видеотехнике,
- компьютерная графика,
- сонары,
- обработка данных в радиолокации и другие применения в военной технике,
- сбор и обработка данных в научных экспериментах,
- электронное оборудование автомобилей,
- робототехника,
- контроллеры дисков и другие.

Какие алгоритмы применяются в dsp?

Подавляющее большинство применений dsp-техники основано на алгоритмах цифровой фильтрации (Finite Impulse Response (FIR), Infinite Impulse Response (IIR) и другие), дискретных преобразований Фурье (FFT) и операциях с матрицами (свертка и др.). В случае применения DSP "не по прямому назначению", например в системах сбора и обработки экспериментальных данных, используются разнообразные алгоритмы любой степени сложности.

Как быстро должен работать DSP ?

Большинство задач, которые решает DSP, являются задачами реального времени. Приблизительные характеристики потоков данных показаны на рис.2.

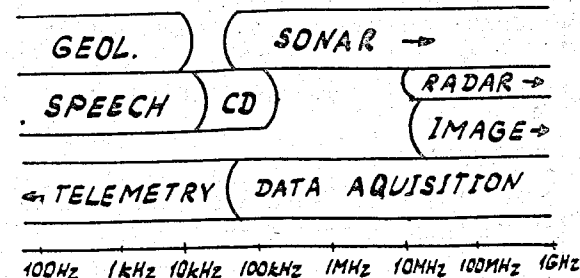


Рис.2. Приблизительные потоки данных в задачах реального времени.

Пример задачи для DSP

Одной из самых распространенных задач для DSP является фильтрация в реальном времени (on-line). Часто употребляется трансверсальный фильтр FIR, который описывается уравнением

$$Y_n = \sum_{k=0}^M A_k \cdot X_{n-k} \quad (2)$$

где $\{X_n\}$ и $\{Y_n\}$ - входная и выходная последовательности сигналов,
 A_k - коэффициенты фильтра,
 M - степень фильтра.

FIR - фильтр символически показан на рис.3.

Для вычисления одного выходного отсчета Y_n нужно запомнить $M+1$ последних входных отсчетов X_n и $M+1$ коэффициентов A_k , а также сделать $M+1$ умножений и $M+1$ суммирований, т.е. $2M+2$ арифметических операций. Если частота выборки и оцифровки $F=8кГц$ и если хотим создать FIR - фильтр 10-й степени ($M=10$) (и это не очень строгие требования в системах связи), тогда для вычисления одного выходного отсчета останется $1/F=125мкс$. Кроме вычисления, надо еще обеспечить ввод коэффициентов фильтра A_k , ввод данных X_n и вывод результата.

Микропроцессор i8080 выполняет одно умножение 16×16 бит за $500мкс$, а с сопроцессором 8231 за $31мкс$. Для МП i8086 это время

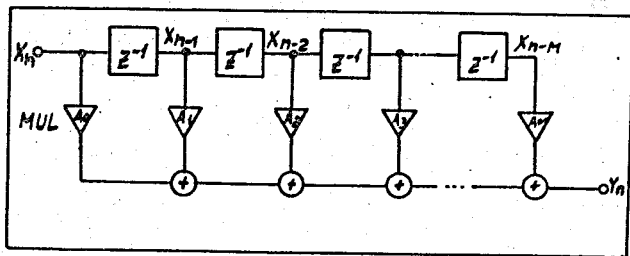


Рис.3. FIR - фильтр.

составляет $32мкс$ (и $17мкс$ с сопроцессором 8087). Ясно, что стандартный МП не способен в реальном времени решить эту задачу.

Процессор i2920 - самый первый и наиболее медленный из DSP - вычисляет всю задачу, т.е. выполняет 22 математические операции и I/O операции в течение $36мкс$.

Ниже приведено время решения этой задачи для некоторых современных DSP:

NEC mPD7720 (1981г.): $3,3мкс/16бит$ FXP (Fixed Point)
 AD ADSP-2100 (1986г.): $2,5мкс/16бит$ FXP
 NEC mPD77230 (1987г.): $1,8мкс/32бит$ FLP (Floating Point)
 TI TMS320C30 (1987г.): $0,6мкс/32бит$ FLP.

Какие выводы можно сделать из этих результатов?

Конечно, реализация описанного фильтра не является единственным критерием для сравнения DSP, но этот тип задачи

чрезвычайно распространен и ее анализ нам покажет, как должна выглядеть схема, выполняющая dsp.

- 1) Ясно, что основой работы DSP в FIR - фильтрах (а также в IIR, FFT и т.д.) является быстрое выполнение операций умножения (MUL) и суммирования (ACC). Все DSP (исключая только i2920) имеют поэтому как умножитель, так и сумматор (MAC=Multiplier+ACumulator).
- 2) Производительность DSP повысится в 2 раза, если MAC может осуществлять одновременно умножение и суммирование - в нашем случае, например, умножить $X_n \cdot 2.A2$ и одновременно предыдущий результат $X_{n-1} \cdot A1$ прибавить в сумматоре к последней сумме. Эта возможность есть у всех DSP.
- 3) Дальнейшее ускорение работы DSP получим, если данные, коэффициенты и инструкции программы будут храниться не в общей памяти (как у стандартных МП с архитектурой фон Неймана), а отдельно, и DSP сможет к ним обращаться одновременно (Гарвардская архитектура).
- 4) Часто приходится одновременно обращаться к двум операндам, например, к данным и коэффициентам, поэтому у многих DSP существуют или две отдельные шины для данных, или быстрый доступ к двум банкам памяти данных в одном цикле (мультиплекс). Как исключение, данные можно хранить и в памяти для программы.
- 5) Из рис.3 следует, что при вычислении Y_n DSP использует данные $X_n, X_{n-1}, \dots, X_{n-M}$, а Y_{n+1} - данные $X_{n+1}, X_n, \dots, X_{n-M+1}$. В этом случае выгодно, если генератор адресов (ADRGEN), который управляет чтением/записью данных X_n , обеспечивает автоматический сдвиг при адресации данных без того, чтобы реально сдвигать данные X_n, X_{n-1}, \dots в памяти. Этот режим называется circular buffer и присутствует почти у всех DSP.
- 6) Из характера задач dsp следует частое использование программных циклов и переходов. Это означает, что в DSP должна быть возможность реализации циклов, переходов и вызовов подпрограмм простыми инструкциями, без дополнительных проверок.

Из чего состоит DSP?

На примере FIR проиллюстрированы различные требования к DSP. Блок-схема, представленная на рис.4, показывает, как примерно может выглядеть DSP.

Прежде всего DSP должен иметь умножитель и блок ALU, который работает не только как сумматор, но и выполняет другие команды, типичные для стандартного МП, например, логические операции, сдвиг и т.п. Блок MUL+ALU будем в дальнейшем называть CPU.

CPU должен иметь высокое быстродействие, обеспечивать запись результатов в регистры общего доступа, управлять статусными регистрами и выполнять другие операции. Результаты работы ALU и MUL часто нужно обрабатывать в регистрах сдвига (SHIFTER, BARELL SHIFTER-BS), чтобы предотвратить переполнение, обеспечить округление и т.д.

От скорости, внутренней архитектуры и структуры регистров ALU, MUL и BS зависит производительность DSP в целом.

Вопрос точности вычислений будет детально рассмотрен позже.

DSP должен иметь быстродействующую память достаточного объема для хранения программы (PROG M). Часто необходим (в случае ICH-DSP) доступ к инструкциям во внешней памяти (PROG/DATA BUS). На первых порах при несовершенной технологии

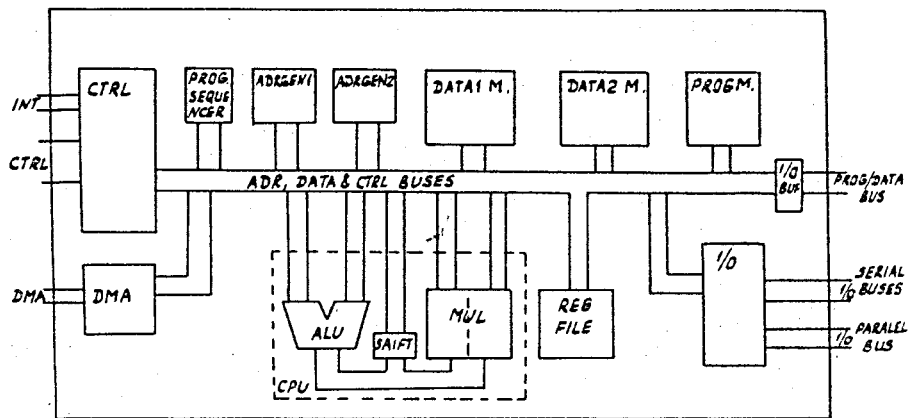


Рис.4. Обобщенная блок-схема DSP.

1CH-DSP использовали или небольшую внутреннюю память для программ, или большую, но за счет упрощения внутренней структуры. Некоторые фирмы решали эту проблему путем исключения всей памяти из чипа DSP и обеспечения доступа к внешней блокам памяти через несколько шин. Кроме того, необходимостью стала быстрая CACHE - память непосредственно на кристалле DSP для самых критичных по времени подпрограмм. Современные 1CH-DSP объединяют преимущества большой внутренней памяти (до нескольких тысяч инструкций), CACHE-памяти и быстрого доступа к внешним инструкциям.

Память для данных (DATA M.) развивалась таким же образом. Сейчас в 1CH-DSP часто используют две внутренние RAM с объемом до нескольких килослов данных и большое адресное пространство для внешних данных. В одном цикле работы DSP можно выполнять и инструкции с тремя операндами. Третий операнд можно брать из разных регистров: или из тех, которые являются частью почти всех блоков, или из так называемого REGISTER FILE - специального блока быстрых регистров с общим доступом.

Программный секвенсер (PROGRAM SEQUENCER) требуется для управления потоком инструкций программы, он должен обеспечить не только нормальный ход выполнения программы (program counter, address pointer), но также переходы, вызов подпрограмм (обслуживание регистров STACK) и прерывания. Кроме высокой скорости, секвенсер должен допускать реализацию циклов в программе без дополнительного обслуживания.

Генератор адресов (ADRGEN) определяет позицию в памяти данных для чтения/записи коэффициентов и данных. Многие алгоритмы нуждаются в очень сложных перемещениях данных, поэтому ADRGEN должен быть и быстрым, и достаточно гибким. Иногда используются два ADRGEN в одном DSP, при этом один из них может выполнять функции программного секвенсера.

Контроллер канала прямого доступа (DMA) нужен для быстрой перекачки данных (иногда и инструкций) между внешней и внутренней памятью. Если DMA контроллер имеет свой ADRGEN, то он может

работать одновременно с CPU и эмулировать DATA CACHE. Это позволяет использовать в качестве внешней памяти данных медленные (но зато дешевые) DRAM. Вопрос использования памяти данных и программ может решаться и по-другому. Некоторые DSP не имеют на чипе никакой памяти, в таком случае должна быть очень быстрая внешняя память, часто с доступом менее 50нс (т.е. дорогая). Существует возможность применения нескольких внешних памяти данных с соответствующим количеством шин. Тогда они могут работать медленнее, но повышается стоимость корпуса DSP. В общем случае невозможно однозначно сказать, какой подход наилучший, так как разные задачи имеют разные требования к объему памяти и скорости доступа к ней.

DSP имеют один или несколько последовательных портов для ввода и вывода данных (SERIAL IN/OUT PORT). Они используются для связи с АЦП, ЦАП, для соединения DSP между собой и т.д. Часто они имеют гибкую настройку формата данных и частоты переноса. Частота задается или внутренним осциллятором, или внешним тактовым сигналом. Связь поддерживается сигналами типа светофор.

Параллельный порт имеет в большинстве случаев полную ширину данных, иногда он мультиплексирован.

Управляющий блок (CTRL) выполняет различные функции, например, генерирование опорной частоты, инициализацию (RESET), обработку прерываний и т.д., но основной функцией являются дешифрация инструкций и координация их выполнения.

Какие бывают форматы данных и типы CPU в DSP?

Данные, которые обрабатывают MUL и ALU (т.е. CPU), имеют разную ширину. Существует DSP, CPU которых обрабатывает данные шириной 8, 16, 18, 22, 24, 32, 64 бита.

Если CPU работает с этими данными исключительно как с целыми числами (integer, fix point), будем называть его целочисленным CPU (FXP CPU).

Если CPU может обрабатывать входные данные как числа в экспоненциальном виде, будем называть его CPU с плавающей запятой (Floating point - FLP CPU). По аналогии 32-разрядный DSP, имеющий CPU с плавающей запятой, будем называть 32 бит FLP DSP, в нем из 32 бит слова данных 24 отводится под мантиссу и 8 - под экспоненту.

Процессоры целочисленные, с плавающей запятой - динамика и точность

Ширина данных и тип CPU вместе определяют точность вычисления и динамический диапазон данных, с которыми может работать CPU.

Динамический диапазон (ДД) практически измеряемых величин при исследовании природных явлений обычно не превышает 145дБ. Работа с синтезированными объектами (машинная графика, моделирование и др.) требует большего ДД.

16 бит FXP CPU может работать с числами от 0 до 65535, ДД в этом случае составляет около 96дБ ($20 \cdot \log 65535$). Но его точность (5 цифр) сильно падает при сигналах, близких нулю.

16 бит FLP CPU с организацией - 11 бит мантисса и 5 бит экспонента - имеет ДД около 190дБ. Его точность только 4 цифры (11 бит), но останется такой же в пределах всего ДД, т.е. и при низких уровнях сигнала.

Общий взгляд на динамику, в зависимости от ширины слова и типа CPU представлен на рис.5.

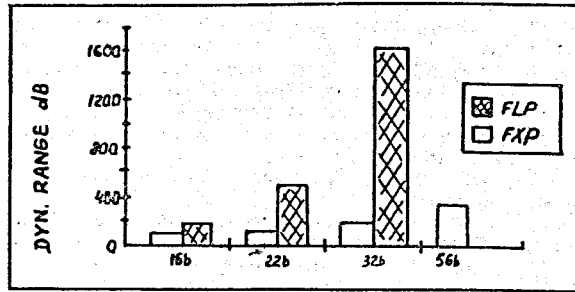


Рис.5. Зависимость динамического диапазона от ширины слова и типа CPU.

При математических операциях в CPU результат часто намного больше, чем операнды. Поэтому внутренние шины DSP, которые соединяют ALU и MUL, значительно шире, чем данные в памяти, что дает возможность работать с "несимметричными" данными (например, 16-разрядное ALU может иметь один вход для операнда из памяти шириной 16 бит, а другой - шириной 24 бита - для ввода результата из MUL). Широкие внутренние шины в CPU частично предохраняют от переполнения и повышают точность вычислений. Этот подход употребляется как для FXP CPU, так и для FLP CPU.

FXP CPU проще, дешевле и быстрее, чем процессор с FLP CPU, но ему часто грозит опасность переполнения. Программист должен эту опасность ограничивать при помощи масштабирования сдвиговыми регистрами (Barrell Shifter). Некоторые CPU типа FLP имеют внутреннюю архитектуру типа FXP. Экспонента обрабатывается отдельно (BLOCK FLOATING POINT). Это позволяет сохранить высокую динамику FLP и упростить CPU, но требует сложного управления.

Развитие технологии позволяет создавать быстрые "полноценные" FLP CPU на чипе 1CH-DSP и программист может сосредоточиться на алгоритме и не следить за переполнением. В качестве примера можно назвать TMS320C30 - 32 бит FLP DSP (24 бит + 8 бит), имеющий высокую точность вычислений, большой динамический диапазон (около 1600дБ), малое время выполнения типичных операций (например, операция $(A \times B) + C$ выполняется за 60нс), и ширину данных в CPU (так называемый внутренний динамический диапазон) 40 бит.

Существуют разные возможности представления числа с плавающей запятой. Часто употребляется двоичное дополнение, стандарты DEC VAX, IEEE754(.10). Многие FLP DSP работают в своем специальном формате данных, но имеют инструкции для преобразования данных разных форматов при вводе/выводе.

Высокая точность и большой динамический диапазон FLP CPU необходимы для качественной обработки акустических сигналов, в телекоммуникации, моделировании и многих других задачах. Вместе с тем существуют области применения, где высокая точность не нужна (например, при обработке видеосигнала достаточно 8 бит), но часто требуется высокая скорость обработки. В этих случаях лучше использовать несколько дешевых процессоров или специальные архитектуры.

Любой МП может реализовать dsp алгоритм, если для этого у него будет достаточно времени. Но DSP применяются чаще всего для решения задач реального времени.

Их можно определить пятью условиями, которые вытекают из высказанного выше. Чем лучше выполняет DSP эти требования, тем больше он отличается от стандартного МП.

- 1) Быстрая и гибкая арифметика. Возможность осуществлять в одной инструкции такие операции, как умножение, умножение с суммированием, сдвиг, стандартная арифметика и логические операции. Кроме того, CPU должен справляться с любой последовательностью вычислений.
- 2) Повышенный внутренний динамический диапазон для операций умножения и суммирования (см. раздел о точности).
- 3) Одновременный доступ к двум операндам для операций типа $(A \times B) + C$ (см. пример фильтра FIR).
- 4) Наличие циклических буферов или на внутренней, или на внешней памяти данных для значительного упрощения выполнения некоторых алгоритмов и ускорения вычислений (см. пример фильтра FIR).
- 5) Автоматические программные циклы и переходы.

Как реализовать DSP?

Пять упомянутых пунктов и общие представления о том, что должно входить в состав DSP, допускают несколько возможностей реализации системы dsp.

Во второй половине 70-х годов начали появляться первые специализированные микросхемы, выполняющие функции отдельных блоков, например, умножители, регистровые файлы, генераторы адресов и т.д. Постепенно разными фирмами были созданы комплекты ИС, из которых можно было строить ВВ-DSP.

В 1979 г. появился первый DSP на одном чипе. Сейчас существует большое количество разных ВВ-DSP и 1CH-DSP.

Первые 1CH-DSP обладали меньшей вычислительной мощностью и производительностью, чем ВВ-DSP. Современные 1CH-DSP по сравнению с теми ВВ-DSP ушли далеко вперед. Но "блочный" подход развивался дальше. Кроме ИС, представляющих собой функциональные блоки для dsp (как, например, ИС умножителя, ИС ALU и т.д.), появились также ИС, содержащие в себе все блоки для FIR фильтра 10-й степени или для вычисления "ядра" FFT (butterfly) и др. Эти блоки были специализированы для определенного алгоритма и не имели признаков универсального DSP.

Позже возникло третье направление 1CH-DSP, в котором применены новые технологические возможности VLSI и новый подход - ASIC (Application Specific IC). Появились новые типы микросхем:

- 1) Специальные заказные ИС для конкретных целей.
- 2) Полузаказные ИС, в которых используется "ядро" стандартного DSP и на тот же чип добавляются нужные памяти и интерфейсные блоки.
- 3) Полузаказные ИС, в которых основные блоки (сумматоры, умножители и т.д.) интегрируются в одном чипе вместе с "матрицей вентиляей" (gate array, sea of gates). Примером является продукция фирм LSI-Logic, Plessey.

Второй подход употребляется очень широко. Как "ядро" используется, например, отличный 16 бит FXP DSP фирмы Texas Instruments TMS320C10. На этой основе созданы чипы TMS320C14 для HD-контроллера и TMS320C17 для телекоммуникации.

Этот подход обходится сравнительно дешево, причем у пользователя остаются все те удобные средства для разработки программ, которые используются для "ядра", т.е. 320C10. В качестве ядра применяется также 24 бит FLP DSP 86220 (Fujitsu). Блочный подход по всем критериям (цена, конструкция, потребляемая мощность, объем) уступает "интегральному", но в некоторых случаях является единственным выходом.

1CH-DSP всегда были на нижнем пределе требований, но это понятие является сейчас очень относительным и возможности 1CH-DSP огромные.

Общие тенденции развития DSP показаны на рис.6.

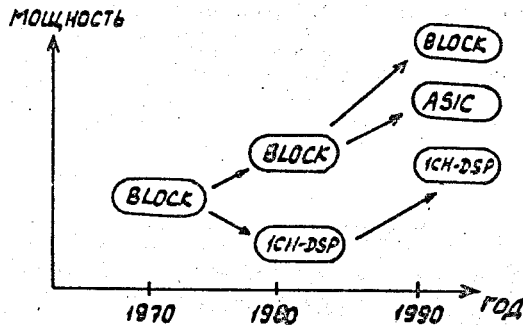


Рис.6. Тенденция роста производительности DSP.

Процессоры специальной архитектуры

В этой части коротко изложены основные понятия, касающиеся нестандартных архитектур процессоров. Некоторые из этих подходов используются или могут использоваться и в технике DSP.

Pipeline (конвейер)

Один из методов повышения производительности процессора состоит в том, что процессор осуществляет несколько операций одновременно. Это напоминает изготовление автомашин конвейерным способом. Этот метод в разных вариантах применяется во всех DSP, но, конечно, не может обеспечить выполнения всех задач dsp.

Параллельная обработка

Производительность процессора можно повысить за счет одновременного решения нескольких задач. В случае одного процессора он переключается между несколькими задачами. В многопроцессорной системе задачи делятся на несколько частей и каждый процессор выполняет часть задачи.

Array processor

Параллельный процессор, предназначенный для быстрой обработки больших блоков данных, которые необходимо сначала организовать в стандартную структуру, чаще всего в матрицу.

Systolic array

Это тоже тип параллельных процессоров. Несколько идентичных процессоров взаимодействуют таким образом, что соседние процессоры соединены друг с другом несколькими шинами ввода/вывода и все процессоры соединены общей шиной с управляющим процессором и общей памятью. Все процессоры одновременно выполняют одну инструкцию с разными данными. Эта система часто называется single-instruction, multiple-data-stream (SIMD).

Data-flow архитектура

- является одной из самых совершенных версий параллельной обработки. Она использует организацию multiple-instruction, multiple-data-stream (MIMD). Несколько процессоров, каждый из которых обрабатывает независимо разные данные, сотрудничают в синхронной системе. Процессоры передают друг другу результаты своей работы вместе со следующей инструкцией. Слова данных "текут" через систему вместе со своими инструкциями до тех пор, пока все операции не выполнятся и окончательный результат не поступит на выход.

Программное обеспечение и средства для разработки

В начале 80-х годов существовали прогнозы, что в начале 90-х годов рынок DSP будет составлять около 1 млрд. долларов. Позже оказалось, что применения DSP не получили массового характера, поэтому цены не падали достаточно быстро и многие разработчики выбрали классический подход.

Причины были разные: DSP-техника была новой, схемотехника непривычной и сложной, практически не существовали стандарты, совместимость новых и старых типов DSP была низкой, конкретные применения требовали разработки математических методов, а программирование было архаичным.

Сейчас предсказания уже оправдываются и интерес к DSP растет. Именно трудности в середине 80-х годов заставили изготовителей поменять подход к пользователю.

Сначала предполагалось, что основным орудием для программирования DSP будет ассемблер. Изготовитель продавал DSP-чип, модуль для разработки (evaluation kit) и ассемблер. Этого хватало для простых алгоритмов и маленьких программ. Сохранить высокую производительность DSP - значит, прежде всего, хорошо использовать возможности параллелизма и конвейеризации. Поэтому фирмы боялись компиляторов языков высокого уровня, первые образцы которых имели эффективность около 30%. Но необходимость широко поддерживать применение DSP привела к появлению компиляторов языка "C", которые при использовании средств оптимизации достигают 90% эффективности. Рынок средств для разработки - аппаратные модули, эмуляторы, симуляторы - заняли "третьи" поставщики, которые с изготовителями чипов только сотрудничают.

Программное обеспечение ориентировано, как правило, на язык "C" и позволяет в критических местах программы использовать код ассемблера.

Программы, написанные только на ассемблере, дают для конкретного DSP лучшие результаты и в "High End" - применениях используются и сейчас (так же как в аппаратуре используются дорогие ВВ-DSP). Но их разработка требует больших затрат времени и средств. Программа, написанная на удобном языке, будет работать медленнее, но отношение цена/мощность будет все равно

лучше, потому что в стоимость системы с DSP необходимо включить и разработку программного обеспечения.

Возник и другой подход к разработке программного обеспечения - применение "экспертных систем". Это интерактивные системы, ориентированные, например, на цифровые фильтры; в них пользователь "рисует" характеристику фильтра на экране PC, определяет свою аппаратную конфигурацию DSP и получает программу, которую после моделирования можно сразу "прожечь" в DSP.

Как среда для разработки используется прежде всего IBM PC, но часто применяются также VAX, Macintosh, Sun и др. Модули DSP для этих компьютеров годятся не только для "первых шагов" освоения DSP, но и для решения серьезных задач в реальных применениях техники dsp (фирмы Spectral Inovation, AT&T, Spectrum Signal Processing и др.).

Самый современный подход - это система SPOX фирмы Spectrum Microsystems. Это операционная система реального времени для DSP фирмы TI типа TMS320C30 с широкими возможностями.

Можно сказать, что средства для разработки аппаратуры и программ для DSP играют сейчас, как минимум, такую же роль, как чипы DSP.

Как оценивать и выбирать DSP?

Производительность более-менее универсального процессора можно характеризовать:

- скоростью обработки инструкций (в миллионах инструкций в секунду - MIPS),
- скоростью выполнения математических операций в миллионах в секунду (в MOPS для FXP DSP и в MFLOPS для FLP DSP).

Эти две характеристики - MIPS, MOPS (MFLOPS) - отличают DSP не только от классических МП, но и от арифметических сопроцессоров, процессоров с архитектурой RISC и транспьютеров.

DSP являются специальными процессорами, которые предназначены прежде всего для циклического выполнения определенных математических операций (например, типа $(Ax+B)+C$).

Современные лучшие 1CH-DSP достигают 33 MFLOPS и 16,5 MIPS (TMS320C30) и BB-DSP -100 MFLOPS. При этом не говорится о специализированных DSP-процессорах, созданных для конкретной задачи, или о мультипроцессорных системах.

Надо понимать, что для одного выходного (обработанного) отсчета DSP должен сделать примерно 10-100 математических операций. Тогда входной поток данных для TMS320C30 составляет максимум 330к - 3,3М отсчетов/с, и максимальная полоса частот входного аналогового сигнала - только 165кГц-1,65МГц!

Это объясняет гонку за количеством MFLOPS всеми доступными средствами - т.е. применением и специальных мультипроцессорных архитектур, и процессоров специального назначения.

Стандартная программа для управления и вычисления (т.е. не для dsp) использует смесь инструкций, в которой редко встречаются циклические вычисления одного типа друг за другом. Высокая производительность определяется в этом случае, в основном, скоростью обработки общих инструкций (MIPS). Поэтому современные МП имеют производительность в MIPS больше, чем DSP, но их "математическая" производительность относительно низка. Если используется больше математики, можно применить сопроцессор, у которого меньше MFLOPS, чем в DSP, но зато он быстро выполняет некоторые функции, которые DSP делает медленно, потому что в dsp-алгоритмах эти операции не встречаются (например, трансцендентные функции).

RISC - процессоры и транспьютеры имеют очень высокую скорость потока инструкций и данных, огромное адресное пространство и возможность (транспьютеры) простого и эффективного соединения процессоров, но их скорость в MFLOPS в лучшем случае средняя (9 MFLOPS - RISC Motorola MC88000). RISC - процессоры могут заменить DSP там, где требуется сложное управление, много инструкций для "манипуляции", но меньше циклов dsp (робототехника, лазерные принтеры, графические ускорители).

Транспьютеры могут идеально сотрудничать с DSP в мультипроцессорных системах, где объединяется высокая вычислительная мощность DSP с совершенной системой передачи данных, которой владеет транспьютер.

При оценке DSP и возможном сравнении производительность в MIPS и MFLOPS является только одним из критериев. Всегда важно учитывать, какие алгоритмы будут использоваться в конкретной задаче, затем определить, какую производительность обеспечит данный DSP для этого алгоритма и насколько просто данный алгоритм можно реализовать в этом DSP.

Для фильтров и других "одномерных" применений часто в качестве критерия оценки производительности берется реализация FIR, IIR- фильтров и FFT. Для двумерных применений такими критериями являются прежде всего FFT и операции с матрицами.

Выбирая DSP, необходимо также следить за доступностью средств для разработки, простотой подключения к окружающей среде, точностью и динамическим диапазоном CPU, необходимостью использования внешних памятей и т.д.

Важной является и цена. Иногда новый тип DSP в 2-3 раза мощнее, чем старый DSP, но при этом в 10-20 раз дороже.

Надо также знать, позволит ли данный DSP дальнейшее развитие нашей аппаратуры (резервы для расширения).

Наконец, надо сохранять определенную долю консерватизма, чтобы не поддаться цветным рекламным бесконечного числа фирм, предлагающих нам свою продукцию на самых выгодных условиях.

Технология

В начальной фазе развития DSP высокую скорость могла обеспечить только биполярная технология. Мощность быстрого умножителя составляла 1-3Вт, ECL - до 9Вт. Создавать из таких элементов 1CH-DSP было невозможно не только из-за большой рассеиваемой мощности, но также и в связи с технологическими ограничениями размеров структур. Со временем эти параметры улучшились, и для специальных применений употребляют биполярные ИС и сейчас (BB-DSP), в первую очередь, из-за их скорости (умножение 16x16 бит выполняется за 5нс). Но усовершенствовались также и MOS-технология. В первых 1CH-DSP использовалась технология NMOS, например, в процессоре i2920 - (6мкм NMOS, 18000 транзисторов, 0,8Вт). В настоящее время для 1CH-DSP применяется исключительно CMOS-технология. Ярким примером такого DSP является TMS320C50 (0,8мкм CMOS, 1000000 транзисторов, 1Вт).

Ожидается также внедрение технологии на основе арсенида галлия (GaAs) и BiCMOS.

Сигнальный процессор TMS 320C30

Этот 32бит FLP 1CH-DSP фирмы Texas Instruments является одним из самых современных, совершенных и сложных DSP. Он появился на рынке в 1988 году. В нем используется 1мкм CMOS технология, цикл длится 60нс, рассеиваемая мощность - 1Вт,

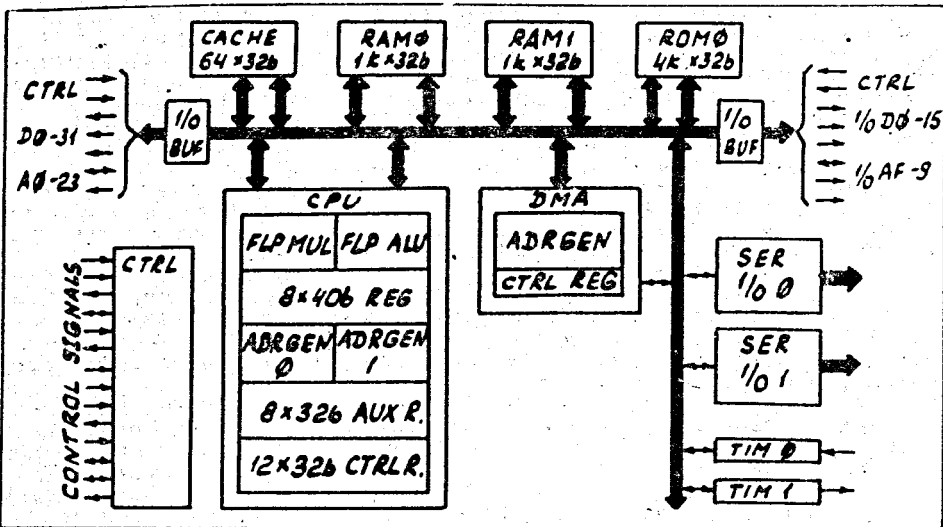


Рис.7. Архитектура процессора TMS320C30.

скорость-16,5 MIPS и 33 MFLOPS. Его архитектура представлена на рис.7. Этот DSP имеет 32-разрядную шину данных и 24-разрядную шину адресов (адресное пространство 16М слов). Каждая из двух RAM имеет "двойной" доступ, т.е. в одном цикле можно обратиться к двум операндам.

Блок CPU содержит MUL, работающий или как FXP 24x24->32 бит, или как FLP 32x32->40 бит, а также 32-разрядный кольцевой сдвигатель BS, 32 бит FXP/FLP ALU, 2 адресных генератора и 28 регистров. Большое число регистров ускоряет работу CPU и упрощает программирование. Восемь регистров шириной 40 бит предназначены для FLP вычислений, следующие 8 вспомогательных регистров AUX R. используются для промежуточного хранения результатов и для работы генераторов адресов. ADRGEN могут работать одновременно, т.е. в одном цикле имеется доступ к двум позициям RAM или ROM. Можно использовать сложные способы адресации, например, циклический буфер.

Двенадцать регистров CTRL R. применяются для различных управляющих функций, таких как управление стеками, хранение статуса процессора, управление циклами, прерывания и т.д.

CACHE-память 64x32 бит ускоряет выполнение критических по времени подпрограмм.

Блок DMA имеет свои отдельные шины и свой генератор адресов. При работе с данным DSP может использоваться медленные внешние устройства и памяти. В этом случае DMA-контроллер эмулирует функцию DATA CACHE. Например, DMA - контроллер может читать данные с обоих параллельных портов и записывать их в блок RAM1. Одновременно CPU может обрабатывать данные из блока RAM0. Потом функции RAM0 и RAM1 меняются.

Фирма TI выпускает две версии этого DSP. Одна из них используется как микроконтроллер (84 pin PCC-plastic chip carrier) и не имеет выходов всех шин. Вторая (полная) версия обеспечивает доступ к внешним инструкциям, на корпусе имеются выводы для шин адресов (24) и данных(32), шин ввода/вывода (данных (16), адресов (10)), прерывания (4), последовательных портов (2), таймеров (2) и других сигналов управления.

Два последовательных порта полностью независимы друг от друга и могут передавать данные в формате 8,16,24 и 32 бита. Синхронизация обменов - внутренняя или внешняя, скорость передачи - до 8Мбит/с. Наличие последовательных портов с возможностью подтверждения передачи, двух специальных инструкций и сигналов позволяет достаточно просто соединить несколько DSP в единую синхронную мультипроцессорную систему.

TMS320C30 представляет собой пример модульного подхода к конструкции: все элементы DSP являются по существу блоками ввода/вывода, подключенными к общей внутренней шине. Каждый блок имеет свои регистры, которые адресуются на этой шине как память. Элементы образуют библиотеку модулей, которая позволяет просто создавать заказные версии DSP.

TI разработал для этого DSP разнообразные программные средства, предназначенные для разработки математического обеспечения dsp на компьютерах IBM PC и DEC VAX. Разработчики могут использовать язык "C", подпрограммы на ассемблере, стандартные библиотеки и т.п.

Совместимость серии TMS320 позволяет применить для C30 программу, написанную, например, для C10.

Инструкции C30 можно разделить на следующие категории:

- чтение и запись,
- 2-операндная арифметика и логика,
- 3-операндная арифметика и логика,
- параллельные операции,
- арифметика и логика с записью результатов в память,
- управление программой,
- операции взаимодействия нескольких DSP (interlocked op.).

Иллюстрацией высокой степени параллелизма является, например, команда MPLY3, в которой в одном цикле выполняется FLP-умножение с 3 операндами и FLP-суммирование, также с 3 операндами.

Представляет интерес инструкция "перехода с задержкой" (delayed branch). Если применить ее на 3 шага раньше, чем переход на самом деле нужен, сохранится выгодность структуры "пайплайн".

В 1988 году стоимость образцов TMS320C30 составляла 1300\$, в 1990 году ожидается снижение до 100\$.

Некоторые применения DSP

Предыдущие разделы посвящены, в основном, технике сигнальных процессоров, в них практически не затрагивается обширный круг вопросов, связанных с алгоритмами цифровой обработки сигналов и применением DSP в различных областях науки и техники.

В обзоре не ставилась цель дать рецепт решения каких - либо определенных задач при помощи DSP. Да это, за редкими исключениями, и невозможно, поскольку каждая прикладная задача имеет свою специфику, которая отличает ее от многих других, даже сходных. Эта специфика выражается в конкретных требованиях к системе dsp (потоки данных, алгоритм обработки, требуемая точность, время решения и т.п.). Мы хотели лишь обратить внимание специалистов на существование такого мощного инструмента, как цифровые сигнальные процессоры, кратко описать историю и тенденции их развития и дать представление о характеристиках и возможностях DSP. Вместе с тем есть основания говорить об общих методических подходах к постановке и решению некоторых классов задач. К числу таких задач, представляющих интерес для специалистов, занимающихся автоматизацией физических

исследований, относятся задачи создания систем триггирования, сбора и обработки данных, получаемых в экспериментах на ускорителях.

В современных спектрометрах физики высоких энергий потоки данных могут составлять несколько Гбайт/с. В будущих экспериментах на коллайдерах УНК (Серпухов), LHC (ЦЕРН) и SSC (США) потоки возрастут более чем на порядок. Прямое накопление таких объемов данных практически невозможно, да и нецелесообразно, так как полезные события в этих потоках составляют тысячные доли процента и даже меньше. Триггерные системы экспериментальных установок должны осуществлять в реальном времени предварительный анализ и отбор событий, удовлетворяющих определенным критериям. В зависимости от физической задачи в качестве критериев могут использоваться множественность события, величина поперечного импульса, потеря энергии в калориметре, расстояние от точки распада до вершины события и т.п.

В системах триггирования используются несколько уровней обработки данных, обычно три. Разделение на уровни является условным, границы между ними размыты и распределение функций между 1 и 2 или 2 и 3 уровнями зависит, кроме прочего, от элементной базы и архитектуры триггера. На каждом уровне последовательно сокращается поток данных, но, соответственно

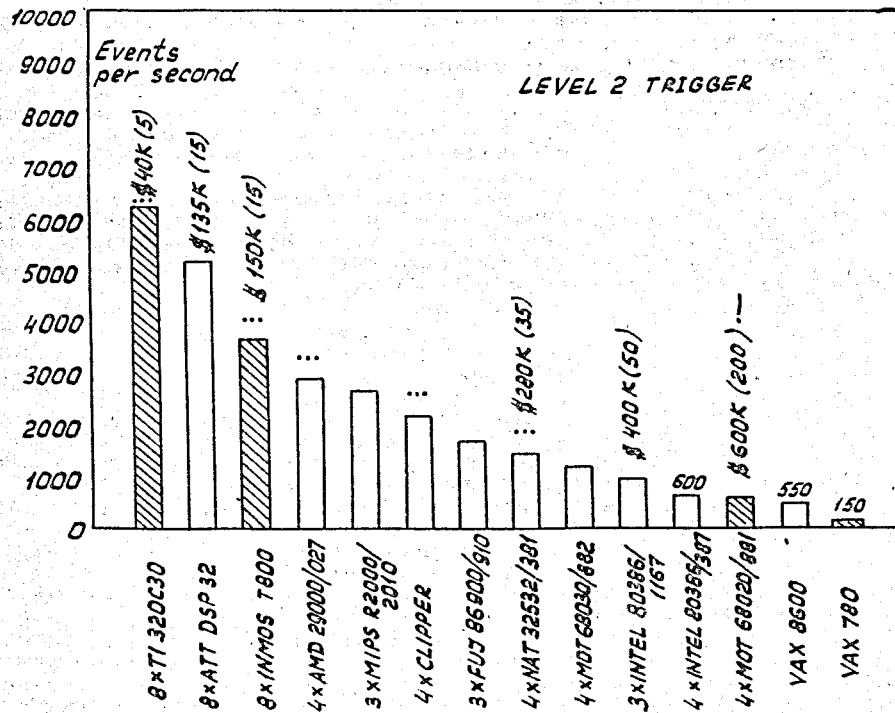


Рис. 8. Зависимость производительности триггерной системы от типа процессоров.

возрастают сложность алгоритмов и время, затрачиваемое на обработку кандидатов в события. Не вдаваясь в детали, отметим, что второй уровень часто является критическим в смысле осуществимости эксперимента или эффективности его выполнения. Именно второй уровень, для которого характерны входной поток $10^4 - 10^5$ событий/с, коэффициент сжатия 100, объем данных 10-100 Кбайт/событие и время решения 10 мкс, является наиболее подходящим для применения DSP, поскольку для отбора полезной информации на этом уровне требуются быстродействующие, высокопроизводительные и относительно дешевые процессоры, легко объединяемые в параллельные вычислительные системы.

Предпосылки к параллельному выполнению операций по обработке данных заложены в самой природе получаемой в физических экспериментах информации:

- события статистически независимы,
- имеется параллелизм внутри события (треки могут восстанавливаться параллельно, и даже внутри одного трека можно параллельно обрабатывать отдельные сегменты);
- процесс обработки легко разбивается на подзадачи.

В SLAC для эксперимента E781 было проведено сравнение различных типов процессоров для триггера второго уровня. Результаты сравнения показаны на рис. 8. Из рисунка видно, что наибольшая производительность триггерной системы при минимальных затратах достигается путем использования сигнальных процессоров 320C30. Эти же процессоры предполагается использовать в триггере третьего уровня. Всего для реализации обоих уровней триггера требуется 5 процессорных плат в стандарте FASTBUS (по 8 DSP на каждой плате) общей стоимостью 40 тысяч \$, в то время как при использовании, например, универсальных МП Моторола 68020 с сопроцессором 68831 потребовалось бы 200 плат (4 процессора на плате, 12 крейтов VME) стоимостью 600 тысяч \$.

На рис. 9 приведена упрощенная схема быстрого цифрового параллельного вычислительного модуля (FDPP), разработанного в ЦЕРНе, в первую очередь, для решения задач реального времени в триггерных системах и системах сбора данных [37].

Модуль содержит два процессора: транспьютер T800 (25 МГц) и сигнальный процессор ATT DSP32C с производительностью 25 MFLOPS. Оба процессора имеют доступ к двум банкам статической памяти емкостью 128 Кбайт (30 нс), которые могут переключаться транспьютером между T800 и DSP32C в течение 100нс. Входные данные после оцифровки в ADC поступают в память FIFO (2048 шестнадцатизрядных слов) с максимальной скоростью 50 Мбайт/с. Оттуда они могут быть записаны в любую область памяти адресного пространства DSP через DMA со скоростью 10 Мбайт/с без приостановки программ, выполняемых процессором. Транспьютер в FDPP имеет локальную динамическую память емкостью 8 Мбайт для хранения программ и данных.

Связь транспьютера с DSP32C осуществляется тремя путями:

- короткие сообщения передаются через "почтовые ящики" (на рисунке не показаны) с использованием нулевого линка транспьютера, в этом случае присутствие данных во входном буфере почтового ящика генерирует прерывание DSP;
- большие блоки данных записываются транспьютером в одну из двухпортовых статических памяти X или Y;
- каналы ввода/вывода DSP используются для мониторинга работы DSP32C и для загрузки программ и констант из T800.

Модуль FDPP смонтирован на стандартной печатной плате, предназначенной для размещения четырех блоков TRAM. Такое решение позволяет подключить FDPP к любой транспьютерной системе,

базирующейся на TRAM, с возможностью использования существующих средств разработки программного обеспечения (Occam 2 toolset) и операционных систем (Helios, UNIX и др.).

Несколько модулей FDPP могут быть легко объединены в сеть или вручную посредством соответствующего соединения транспьютерных линков, или через программно управляемый коммутатор линков IMSC004, что позволяет динамически изменять конфигурацию системы и наращивать ее производительность.

Программное обеспечение для FDPP разрабатывается и отлаживается на IBM PC и затем загружается через транспьютерную сеть в память T800 и DSP32C каждого модуля.

Главная часть операционной системы находится в транспьютерной сети и может управлять работой каждого DSP32C через соответствующий транспьютер. В DSP имеется небольшая ОС, которая через почтовый ящик получает задания от T800 на обработку данных из FIFO (on-line) или из двухпортовой памяти (off-line). Пока DSP обрабатывает данные из одного блока памяти, транспьютер загружает другой блок новыми данными.

Интеграция большой производительности DSP32C и коммуникационных возможностей T800 позволяет решать такие проблемы, с которыми ни один из них не может справиться в одиночку.

Следует отметить, что DSP используется не как сопроцессор для выполнения специфических инструкций, а как мощный вычислитель, реализующий сложные алгоритмы, например, манипуляции с матрицами; нахождение пиков, кластеров и треков; фильтрация и быстрые преобразования Фурье. Ожидается, что FDPP найдет широкое применение не только в триггерных системах второго и третьего уровня, но также в задачах, где требуется применение параллельной архитектуры и типичных dsp-алгоритмов (распознавание образов, обработка изображений, решение систем линейных уравнений и др.).

В несколько ином сочетании используются DSP и транспьютеры в системах сбора данных и триггирования в эксперименте ZEUS (ДЭЗИ, Гамбург) [38]. Здесь DSP применяется, в основном, в триггере первого уровня для выполнения операций "подавления нулей", получения информации о времени дрейфа и площади импульсов, предварительной обработки данных для триггера второго уровня.

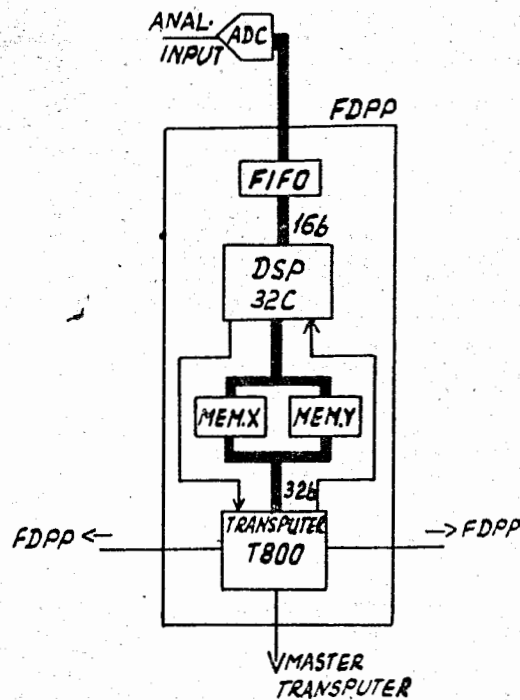


Рис.9. Блок-схема FDPP.

Применение DSP в триггерных системах обсуждается также в ряде других работ (см., например, сборник [39], а также препринт [40] и список литературы к нему).

Сильное впечатление оставляет описанная в работе [41] система сбора и обработки данных, предлагаемая для экспериментов на коллайдерах LHC и SSC (рис.10). Эта система концентрирует в себе максимально доступную на сегодня вычислительную мощность в сочетании с наибольшей скоростью передачи информации, простотой управления и доступа к обрабатываемым данным.

В систему входят контроллер крейта (CC) на базе транспьютера T414 и микропроцессоров Интел 1860 и 1960CA, а также блоки считывания (RC), каждый из которых содержит сигнальный процессор ADSP2101. На DSP возлагаются функции отбраковки шумовых и параметризации полезных импульсов, нахождения сегментов треков и сжатия данных. Эти операции ADSP2101 выполняет очень быстро (например, для нахождения амплитуды и ширины импульса требуется 4,5 мкс, точного определения фронтов - 2,6 мкс, вычисления площади - 2,2 мкс).

Высокопроизводительный микропроцессор 1860 (~100 MIPS) используется для сшивания сегментов треков. К этой работе может подключаться также 1960 (~66 MIPS), однако, основное его назначение - обеспечение быстрой передачи данных между крейтами (скорость передачи 32-разрядных слов через DMA может достигать 132 Мбайт/с).

Применение двухпортовых памяти для межпроцессорных связей до предела упрощает интерфейсную логику. Транспьютер является единственным управляющим процессором в системе. Операционная система, обеспечивающая загрузку программ, считывание данных из RC, тестирование блоков и управление крейтом, целиком написана на языке OCCAM и эффективно поддерживает одновременную работу всех процессоров.

Из приведенных примеров видна тенденция объединения универсальных процессоров и DSP для решения этого класса задач. По-видимому, эта тенденция получит законченное воплощение

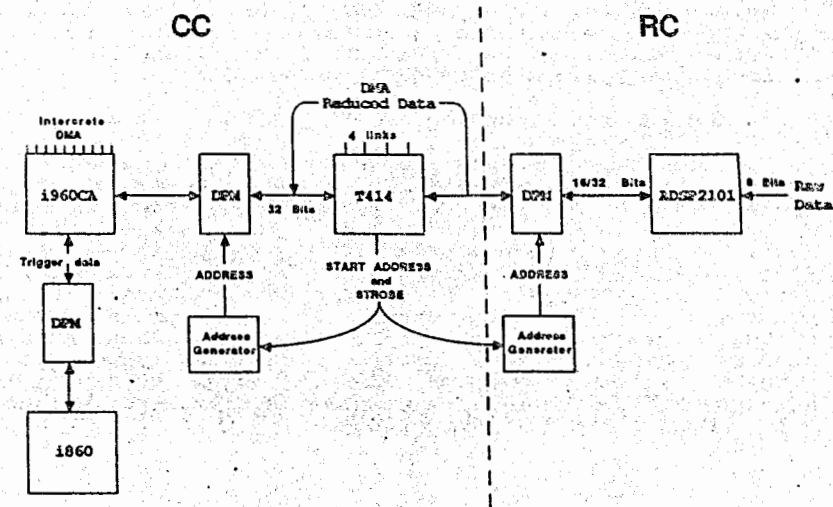


Рис.10. Функциональная схема системы сбора и обработки данных [41].

в разрабатываемом фирмой Интел новом микропроцессоре 80786. Этот МП [42], выполняемый по технологии BiCMOS на кристалле площадью 1 кв. дюйм, будет содержать 50 млн. транзисторов. Основные характеристики 80786:

рабочая частота - 200 МГц;
ширина шины данных - 64 бита;
производительность - 1000 MIPS.

Кроме четырех суперскалярных процессоров с FPU, КЭШ - памяти, контроллера шины, блоков управления памятью, логики для тестирования и самодиагностики, блоков ввода-вывода, он будет иметь встроенный DSP, содержащий около 2 млн. транзисторов.

Трудно переоценить возможности такого "микропроцессора" !

Обзор ИС для dsp

Таблица дает краткую информацию о некоторых 1СН-DSP. В таблице использованы некоторые сокращения:

100 PGA = корпус типа "Pin Grid Array", 100 ножек;
28 DIP = корпус DIP, 28 ножек;
84 PLCC = корпус "Plastic Leaded Chip Carrier", 84 pin;
MAC = умножитель + аккумулятор;
FXP-166 = Fixed point DSP, длина слова данных в ЗУ 16 бит;
16x16->26 = внутренняя длина слова данных (результат работы MAC) составляет 26 бит;
FLP-326-24E8 = Floating point DSP, 32-разрядный, 24 бит мантиса, 8 бит экспонента;
ЗУ: D (P, C) 128x16 = память для данных (программ, констант), 128 слов, разрядность 16 бит;
64FIR-9,5мкс = вычисление одного выходного отсчета фильтра FIR 64-й степени длится 9,5мкс;
1k-CFFT = быстрое преобразование Фурье из 1024 комплексных входных отсчетов.

Отдельно описан интересный DSP IQ-MAC и показаны примеры некоторых чипов специального назначения. Затем следует обзор 16- и 32-разрядных блоков MAC для FXP BB-DSP и для FLP BB-DSP. В конце приведены примеры наборов FLP процессоров.

UT69532 IQ MAC (United Technology (UTMC), 1989). 32 бит FLP, 100 MFLOPS. 1,2мкм CMOS, 2Вт., кор. 280 pin! Тактовая частота 15MHz, несколько операций выполняется в одном цикле. Форматы данных: стандарт IEEE754.10. FLP математика 32 бит (реальные) и 64 бит (компл.) и др. Поддерживает применение ЗУ типа FIFO и RAM. 2x32 бит FLP ALU, 2x32 бит FLP MUL, 23x32 бит MUX, 9 регистров и 1000 бит pipeline регистров! I/O: 2xPAR 32 бит IN, 2xPAR 32 бит OUT, 2xPAR 32 бит IN/OUT порты. 1k-C-FFT-0,7мс. DSP просто объединяются - 10 чипов выполняют 1k-C-FFT за 64мкс! 111 инструкций высокого уровня, например, для FFT butterfly, для фильтров, самотестирование. Одна инстр. выполняет умножение двух реальных матриц 10x10 за 34мкс. Фирма продает также вставные в IBM PC платы для разработки и отладки аппаратуры и SW. Цена в 1989г. около 2000\$.

Таблица

DSP	112920	1PД7720	TMS32010	1MB764	TMS32020	TMS320C25
Фирма	INTEL	INEC	TI	FUJITSU	TI	TI
Год	11979	11980	1982	11983	11985	11985
Тип DSP	FXP-256	FXP-166	FXP-166	FXP-166	FXP-166	FXP-166
Цикл [нс]	1400	250	200	100	200	100
Технология	1NМОS 6мкм	1NМОS 3мкм	1NМОS 3мкм	CMOS 3мкм	1NМОS 2,7мкм	1NМОS 1,8мкм
Чип [мм2]	130	130	44	190		
Кол. транз.	118000	140000	55000	91000		150000
Исходность	10,8ВТ	0,9ВТ	0,9ВТ	0,3ВТ	1,2ВТ	0,5ВТ
Корпус	128DIP	128DIP	140DIP	88PGA	168PGA	68PLCL
MAC	только ALU	16x16->32	16x16->32	16x16->26	16x16->32	16x16->32
RAM внутр.	D:40x24	D:128x16	D:144x16	D:2x128x16	D:288x16	D:288x16
ROM внутр.	F:192x24	F:512x23	P/D:1,5k x16	P/D:1k x24	P/D:256x16	P/D:256x16
ЗУ внеш.	нет	нет	F/D:4k x16	F:1k x24 D:1k x16	нет	F/D:4k x16
Пар. в/в	нет *	1x8 бит	1	1k x16 бит	16x16бит	16x16бит
Посл. в/в	нет	1	нет	нет	1	1
Пример	141IR-30мкс 120FIR-36мкс	164FIR-9,5мкс 121IR-2,25мкс	101IR-17,4мкс 64CFFT-0,6мс	101IR-4мкс 32FIR-4,3мкс		
Примечание	ЦАП/АЦП (9разр.)		TMS320C10:	DMA		
	* Цифр. в/в		CMOS версия,			
	через ЦАП		цикл 125нс			

Таблица - продолжение

DSP	LM32900	ADSP-2100	DSP56000	mPD77220	mPD77230	MSM6992
ФИРМА	INS	AN. DEV.	MOTOROLA	NEC	NEC	OKI
ГОД	1986	1986	1986	1986	1986	1986
ТИП DSP	FXP-166	FXP-166	FXP-246	FXP-246	FLP326-24E8	FLP226-16E6
ЦИКЛ [нс]	100	125	97,5	100	150	100
ТЕХНОЛОГИЯ	CMOS 2мкм	CMOS	CMOS	CMOS 1,8мкм	CMOS 1,75мкм	CMOS 2мкм
ЧИП [мм ²]						
КОЛ. ТРАНЗ.				200000	370000	125000
МОЩНОСТЬ	0,5Вт	0,6Вт		0,7Вт	1,7Вт	
КОРПУС	172PGA	100PGA	88PLCC	68PGA	68PGA	132PGA
MAC	16x16->32	16x16->40	24x24->56	24x24->48	32x32->47E8	22x22->16E6
RAM ВНУТР.	нет	P:16x24	D:2x256x24	D:512x24	D:512x32	D:2x128x22
			C:16x24	C:16x24		
ROM ВНУТР.	нет	нет	P:2x256x24	P:4k x24	P:2k x32	P:1k x32
			P:2x1k x24	C:1k x32	C:1k x32	
ЗУ ВНЕШ.	D:2x64k x16	P/D:32kx24	D:128k x24	D:4k x24	D:4k x32	D:64k x32
	P:64k x28	D:16k x16	P:64k x24	P/D:4k x24	P/D:4k x32	P:64k x32
ПАР. В/В	(1x16бит *)		1	1	1x16бит	(1x22бит *)
ПОСЛ. В/В	1	нет	2	1	1	нет
ПРИМЕР	256FIR-37кГц	1k-CFFT-7мс	1kCFFT-5мс		1k-CFFT-11мс	1k-CFFT-6мс
	1kCFFT-14мс	64FIR-8мс			дел.-5мс	дел.-4мс
				для дешевой реализации	корень-6мс	
ПРИМЕЧАНИЕ	(*) через память FIFO (16 слов)			алгоритмов	13,4 MFLOPS	10MIPS
				проверены на 77230	DMA нет.	320\$/шт-86г
					В 86г.100%*	(*)DMA 10МГц
					10000шт.	

Таблица - продолжение

DSP	MB86220	MB86232	DSP32	DSP32C	TMS320C30	TMS320C50
ФИРМА	FUJITSU	FUJITSU	AT&T	AT&T	TI	TI
ГОД	1987	1987	1986	1987	1988	1990
ТИП DSP	FLP246-18E6	FLP326-24E8	FLP326-24E8	FLP326-24E8	FLP326-24E8	FLP326-24E8
ЦИКЛ [нс]	80	75	80	62,5	60	35
ТЕХНОЛОГИЯ	CMOS	CMOS 1,3мкм	NMOS 1,5мкм	CMOS 0,75мкм	CMOS 1мкм	CMOS 0,7мкм
ЧИП [мм ²]		182	88			
КОЛ. ТРАНЗ.		450000	155000			1000000
МОЩНОСТЬ		1,2Вт	2,7Вт			
КОРПУС		208PGA	40DIP/100PGA	133PGA		
MAC		32x32->32E8	32x32->32E8	32x32->32E8	32x32->32E8	24x24->32E8
RAM ВНУТР.	D:2 блоки	D:512x32	D:2x512x32	или P/D:	D:2x	D:8,7k x32
			P/D:512x32	3x512x32	P:64x32	
ROM ВНУТР.	P:2k x30	P:1k x32	нет	или:P/D:		P/D:2k x32
				2x512x32		
ЗУ ВНЕШ.	P:64k x30	D:1M x32	P/D:64k x32	P/D:16M x32	P/D:16M x32	
		P:64k x32				
ПАР. В/В	1		1x8бит	1x16бит	1k x 16бит	
ПОСЛ. В/В	1	2	1 (12,5МГц)	1 (16МГц)	2 (8МГц)	
ПРИМЕР						
	12,5 MFLOPS					
	DMA					
	Ядро 86220					
ПРИМЕЧАНИЕ	часто явл.	6,7 MFLOPS	12,5 MFLOPS	16 MFLOPS	16,5 MIPS	28,6 MIPS
	основой	250\$	Адр. простр.	325\$ (1987)	33 MFLOPS	
	зак. DSP.		доступно			
	Цена 30\$!		по байтам			

Некоторые специальные чипы

TDC1028 (TRW). ИС для FIR фильтров 8 степени. Содержит 8 MUL, 8 ACC, 8 регистров для коэффициентов, память для 8 последних входных отсчетов. Ширина данных 4 бита. ИС можно просто объединять как по длине слова, так и по длине фильтра. Макс. частота потока данных 20MHz. Эта ИС является хорошим примером "sistolic-array"- архитектуры.

TMC2220 (TRW). Цифровой коррелятор с длиной слова 4 бита, легко наращиваемый.

MPD7281 (NEC, 1983). ИС "data flow" архитектуры.

IMS A110 (INMOS, 1988). 400000тр., CMOS, 2Вт, кор.100 PGA, 53мм2. Хороший пример параллельной организации. На чипе 21 MAC и 3 сдвиговых регистра (1120 steps), возможна конфигурация как 21 steps pipeline или 3x7 MAC array. Длина слова 8 бит, входная частота данных до 20MHz. Макс. производительность 100 MOPS. Предназначен для обработки видеосигнала. ИС можно очень просто соединить и повышать таким образом производительность процессора. Цена 200\$ в партии более 100шт.

ZR33481 (Zoran) - Digital Filter Processor. Содержит 8 MUL. Специализирован для обработки изображений. Модуль RTC150 фирмы Imaging Technology (для VME) с указанным процессором делает свертку с ядром 3x3 или 4x4 для картинки 512x512 в реальное время (340 MOPS).

PDSP16401 (Plessey) - 2D 15 Mpixels/s edge detector. Содержит 4x2D FIR фильтры и 2 компаратора.

GAP (фирма NCR) - GAP-чип содержит 72 ALU, каждый шириной 1 бит и 120 бит RAM. GAP-чип является примером SIMD -архитектуры. Все ALU делают ту же самую операцию, но с разными данными. GAP работает сравнительно медленно, но если он применяется для данных, которые можно организовать как поле (array), то путем объединения нескольких чипов достигается высокая производительность.

Обзор блоков для dsp

Умножители и сумматоры (MAC) на одном кристалле, целочисленные:

16 BIT:

TRW: TMC2210 - 65нс, 400мВ.
Waferscale Integration: WS5910 - 30нс (используется также как модуль для ASIC).
Logic Devices: LMA1010/1043 - 45нс.
LSI Logic: L64010 - 40нс.
Bipolar Integrated Technology (BIT): В 3011, 2011 - 16нс.

32 BIT:

LSI Logic: L64032. В одном цикле (80-100нс) 32 бит MAC операция, в 4-6 циклах - 64 бит MAC операция. Кор. 132 PGA, 1Вт.
TI: SN74ACT8836: 32x32 - 67 бит MAC/35нс, 64x64 - 128 бит MAC: 7 циклов. Кор. 156 pin.
AMD: 29C323 - 32x32 - 67 бит MAC операция 80 (55)нс, 64x64 /7 циклов. Кор. 169 pin, 25mA.
Micro Integration: набор M132AC/M16MP - 10 шагов pipeline - 1xMAC операция/20нс.

Weitek: XL8137 - 32 бит ALU, MUL, делитель, 32x32 бит регистровый файл, 32 бит BS; Цикл 80, 100, 120нс. Может работать и с XL8136 (32 бит программный секвенсер).

Однокристалльные MAC с плавающей запятой

AMD: 29C325 - 24x24 - 36 бит MAC, IEEE стандарт., 70нс в биполярной версии, 100-120нс в CMOS (но 1/4 потр. мощности).

LSI Logic: L64132 - 32 бит FLPP/55мкс. 30 MFLOPS.

Weitek: WTL3332 - 25 MFLOPS, цикл 80,100,120нс.

AMD: 29C237 - 32x32 - 64 бит MAC. 10 MFLOPS для (AxV)+C.

TI: SN74ACT8837 - 64 бит! 20 MFLOPS. Содержит MAC, ALU, регистры, мультиплексор. Кор. 201 PGA.

Weitek: 3364, 3164 - разница только во внешних шинах и корпусе. 20MFLOPS/100нс, двойная точность, 1,5мкм CMOS.

GE Intersil: ISP9326 - 32 бит MAC, 16 MFLOPS. 2x32 бит IN шины, 1x32 бит OUT шина. 1,25мкм CMOS, 100000тр., кор.144PGA, 1Вт.

TRW: TMS3202-FLP-процессор \ поддерживают военный стандарт
Raytheon: Rayfar MAC / FLP: MIL 1750A

Наборы FLP процессоров

Weitek: 2364/2365: 64бит, 32 MFLOPS, заменяют процессор IBM 370, двойная точность 128 бит (16MFLOPS). Цикл 60-70нс. 1,5мкм CMOS, кор. 144 PGA, 1Вт. Стоимость комплекта - около 1000\$.

IDT: IDT 721264/721265: 32 бит 33,4 MFLOPS, 64 бит - 25 MFLOPS. Цикл 30нс.

UTMC: UT 1732/1733. 10 MFLOPS в конвейерном режиме. Поддерживает FLP стандарты IEEE, MIL 1750A, DEC VAX.

AD: ADSP-3212/3222: 40 MFLOPS, деление с двойной точностью 600нс.

BIT: В 3110/3120 или В 2110/2120 - ALU: 32 бит / 40 MFLOPS или 100 MIPS для целочисленных операций 32 и 64 бита. MUL- 28 MFLOPS. Другие функции: BOOLEAN, SHIFT, INVERS, ROOT, DIVIDE, 3x36 бит I/O port.

Заключение

В обзоре дан анализ развития техники DSP и ее сегодняшнее состояние. Показано, что, кроме классических применений для цифровой обработки сигналов, DSP являются мощным инструментом для создания систем триггирования, сбора и обработки данных в экспериментах физики высоких энергий.

Можно ожидать, что дальнейшее развитие техники dsp будет идти в направлении использования новых технологий (GaAs, BiCMOS, субмикронные структуры), интеграции DSP с универсальными микропроцессорами и применения новых архитектурных решений.

Литература

1. K.Marriu. DSP: A technology in search of application. CD 15-11-86; Vol125, No21, pp. 59-77.-Обзорная статья.
2. DSP Uproots Traditional Analog Jobs. ED 28-9-89; Vol137, No20, pp. 49-58.-Обзорная статья.
3. W.Andrewss. Distinction blur between DSP solutions. CD 1-5-89; Vol128, No9, pp.86-99. -Обзорная статья.
4. L.Gunn. At 100 MFLOPS, the fastest DSP chip ever!

ED 13-10-88; Vol36, No23, pp.73-76. — Описание 32 бит FLP MAC UT69532.

5. D.Bursky. Operating systems on DSP streamlines programming. ED 13-10-88; Vol36, No22.—Описание ОС "SPOX" для TMS 320C30.
6. M.Leonard. Digital Signal Processors. ED 13-10-88; Vol36, No22, pp.161-166.—Обзорная статья.
7. J.Novelino. PCs take the reins as DSP design Tools. ED 13-07-89; Vol37, No15, pp.47-52. — Средства для разработки DSP.
8. D.Bursky. Tackle real-time DSP tasks with CMOS chip set. ED 31-03-88; Vol36, No8, pp.45-49.
9. M.Leonard. Floating-point math chips for DSP boast throughput records. Там же, pp.53-55. — Описание MAC DSP9326 (INTERSIL), WTL 2364/2365 и WTL 3164/3364 (Weitek).
10. M.Leonard. Building-block chips are busy widening DSP horizons.—Там же, pp.68-77.
11. R.Cox. FFT-based filter design boosts radar system's process. Там же, pp.81-84.—Спец. FFT (1К C-FFT-0,5мс).
12. J.Roesgen, S.Tung. Moving memory off chip, DSP microprocessor squeezes in more computational power. ED 20-02-86, Vol34, No4, pp.131-143. — Описание ADSP-2100 (16бит FXP).
13. M.Schwartz et al. Signal processor's multiple memory buses shuffle data swiftly. Там же, pp.147-153.—Описание LM32900 (16бит FXP DSP фирмы National Semiconductors Corp.).
14. B.Eichen et al. Floating-point math integrated on chip makes DSP IC a standout.—Там же, pp.159-165. Описание 32бит FLP DSP mPD77230 фирмы NEC.
15. K.Lamb. CMOS DSP building blocks adjust precision dynamically. ED 30-10-86; Vol34, No25, pp.96-103. — Описание набора чипов Plessey PDSP1600 (16 FXP).
16. R.E.Owen. ECL AC slashes system processing time by 80%. Там же, pp.108-115.—Описание ECL MAC B3011 фирмы Bipolar Int. Technology (MAC меньше 10нс).
17. J.A.Cales et al. DSP chip trio links high-level hardware and software. Там же, pp.119-124.—Описание серии Accel 8000 (Weitek).
18. R.Simar Jr., Mike Hames. CMOS DSP chip packs punch of a supercomputer. ED 19-03-87; Vol35, No7, pp.103-110.—Описание TMS320C30 фирмы TI (32бит FLP).
19. TMS320C50 (Technology newsletter). ED 25-05-89; Vol37, No11, p.23. — Краткая заметка.
20. D.Dunnion, M.Stropoli. Design a Hard-disk controller with DSP techniques. ED 22-09-88; Vol36, No21, pp.117-121. HD контроллер на основе TMS320C10.
21. DSP chip processes video signals in real time. ED 11-08-88; Vol36, No18, p.141. — Применение INMOS A110 DSP для обработки изображений.
22. W.P.Hays et al. A 32-bit VLSI Digital Signal Processor. IEEE J.Solid State Circuits, Vol.SC-20, N5, October 1985, pp.998-1004.—Описание DSP-32 (32бит FLP, AT&T).
23. N.Ichiura et al. DSP IC cranks out 22-bit numbers at 20 MFLOPS. ED 30-10-86; Vol34, N25, pp.131-138.—Описание 22 бит (16Е6) FLP OKI M6992 1-CH DSP.
24. T.King-Smith, H.Yassaie. Cascadable DSP chip attacks tasks with parallel punches. ED 30-10-86; Vol43, No25, pp.145-150. Описание чипа A-100 фирмы INMOS.
25. T.Suzuki, K.Ueda. Fixed-point DSP chip offer a low-cost alternative. ED 30-10-86; Vol34, No25, pp.157-160. —Описание MN190x (16бит FXP 1CH-DSP фирмы Matsushita El.).
26. D.Bursky. Digital-signal-processing chips move off the designer's wishlist and into everyday use. ED 17-05-84; Vol32, No10, pp.99-122. — Обзорная статья о нестандартных процессорах.

27. D.Garde, J.Oxaal. CMOS signal processors push to highest throughput of all. 16-bit-slice family creates ultrafast digital signal processor. Там же, pp.135-144.—Набор CMOS ИС фирмы AD: ADSP-1401/1410/1201/1101.
28. B.Windsor, J.Wilson. Arithmetic duo excels in computing floating-point products. Там же, pp.144-151. — Описание ADSP-3210, 3220. Описание стандарта IEEE-7547.
29. T.Dintersmith, J.Nuttall. Putting the chips through their paces in three systems. Там же, pp.151-168.—Применение ИС, описанных в [27, 28].
30. W.Meshack. Data-flow IC makes short work of tough processing chores. Там же, pp.191-206.—Описание data-flow чипа PD7281 фирмы NEC.
31. J.H.Dedrick. Multiport register file simplifies and speeds digital signal processing. Там же, pp.213-222. —Описание register-file LRF08 (Logic Devices) и его применение.
32. G.Ramachandran, S.Jujii. Fast signal processor comes rich with memory, I/O lines on CMOS chip. Там же, pp.227-238. Описание 16бит FXP 1-CH-DSP MB 8764 фирмы Fujitsu и его применение.
33. R.Hester. Analog chip set smoothes one digital signal processing. Там же, pp.243-249.—Описание ЦАП и АЦП фирмы TI для TMS320.
34. R.W.Schafer. The Math behind the MIPS: DSP basics. ED 08-09-88; Vol36, No20, pp.105-109. — Описание некоторых алгоритмов для фильтров и пример для TMS320C25.
35. A.Daly. Selection method steers designers to best DSP solution. CD 01-03-88; Vol25, No5, pp.53-57.—Методы выбора DSP.
36. Ярославский Л.П. Введение в цифровую обработку изображений. Москва, Советское радио, 1979.
37. D.Crosetto. Fast Digital Parallel Processing module FDPP. CERN - DD/89-33, Geneva, 1989.
38. S.Quinton et al.. Data Acquisition for the ZEUS Central Tracking Detector. CERN Transputer Workshop, Geneva, 2 March 1990.
39. E.Fernandez, G.Jarlskod (eds). Proceedings, ECFA study week on instrumentation technology at high-luminosity hadron colliders, Barcelona, 14-21 Sep. 1989.
40. M.Johnson, A.Lankford. Triggering, front - end electronics, and data acquisition for high - rate beauty experiments. SLAC - pub - 4611, Apr. 1988.
41. R.Belushevic, G.Nixon, D.Shaw. An 80 Mbytes/s Data Transfer and Processing Systems. CERN Transputer Workshop, Geneva, 2 March 1990 (full paper to be submitted to NIM).
42. A.Marchioro. The Microprocessor of the Year 2000 - VLSI technology and architectures in the LHC era. CERN, EF Seminar, Geneva, 26 March 1990.
43. Оппенгейм А.В., Шафер Р.В. Цифровая обработка сигналов. Москва, Связь, 1979. Перевод с английского. — Систематическое изложение математических основ dsp.

Рукопись поступила в издательский отдел
13 ноября 1990 года.