

ИСТОРИЯ СОЗДАНИЯ И РАЗВИТИЯ ДССП: ОТ “СЕТУНИ-70” ДО ТРОИЧНОЙ ВИРТУАЛЬНОЙ МАШИНЫ

к.ф.м.н. А.А. Бурцев, к.ф.м.н. С.А. Сидоров
НИИСИ РАН, НИЛ ЭВМ МГУ
burtsev@niisi.msk.ru, sidorov@niisi.msk.ru

Общая характеристика системы

Диалоговая система структурированного программирования (ДССП) [1] была создана в начале 80-х годов XX века в проблемной лаборатории ЭВМ на факультете ВМК МГУ под руководством Н.П.Брусенцова. ДССП была призвана существенно облегчить разработку программного оснащения (ПО) для широкого класса малых цифровых машин – мини- и микрокомпьютеров.

ДССП явилась закономерным результатом цикла работ по усовершенствованию миникомпьютерной архитектуры, который на протяжении ряда лет проводил коллектив сотрудников под руководством Н.П.Брусенцова. Этапы этой работы ознаменовались созданием первой в мире троичной малой цифровой машины (ЦМ) “Сетунь”, затем экспериментальной машины “Сетунь-70”, которая впоследствии (в 1975 г.) была модернизирована для поддержки структурированного программирования на уровне машинных команд [2].

При создании ДССП преследовалось сразу несколько целей. Во-первых, ДССП создавалась в качестве программного имитатора (эмулятора) новой двоичной цифровой минимашины, в которой предполагалось воплотить ценные идеи двухстековой архитектуры и структурированных команд управления, унаследованные от модернизированной троичной ЦМ “Сетунь-70”. В роли такого имитатора или образно говоря виртуального процессора по сути и функционирует внутренний интерпретатор ДССП, являющийся сердцевинной (ядром) системы.

Во-вторых, ДССП создавалась как интегрированная среда разработки программ, легко доступная для освоения широкому классу пользователей микрокомпьютерной техники и персональных компьютеров. Поэтому при создании ДССП была заимствована идея словарной организации системы ФОРТ (Forth) и обеспечен диалоговый характер взаимодействия с пользователем на всех стадиях разработки программы: начиная от её редактирования и вплоть до её окончательной отладки. Такой диалог с пользователем обеспечивается внешним интерпретатором ДССП, функционирующем в режиме исполнения потока слов-команд.

Программа, составленная на входном языке ДССП, сначала преобразовывается во внутреннее представление, для характеристики которого применяется специальный термин – сшитый (threaded) код. Совокупность программных компонент, выполняющих такое преобразование, принято называть компилирующими средствами или просто компилятором системы. Внутренний и внешний интерпретатор, компилятор и словарь ядра, а также встроенные редактор текстов и отладчик вместе составляют базовый комплект ПО системы. На его основе путём наращивания словаря каждый пользователь может создать свой вариант расширения ДССП, отвечающим нуждам его прикладной области.

Программирование в ДССП заключается в наполнении её словаря новыми словами. Всякая операция (процедура), поддерживаемая системой, обозначается отдельным словом (под которым понимается любая последовательность печатных символов, отличных от пробела). Совокупность слов, понимаемых системой, и составляет её язык (словарь). Система расширяется путём определения новых слов, которые сразу же после добавления в словарь могут быть использованы в системе.

Любое действие, совершаемое системой, вызывается словом. Предусмотренные в системе слова исполняют самые разнообразные функции:

- выполняют вычисления (+ - *) или обработку данных (E3 C2 DD) в стеке;
- доступ к памяти (@W !W) или к переменным (! ');
- служат для организации ветвлений (IF+ BR- BRS) и циклов (DW RP);
- обеспечивают построение тел новых слов и добавление их в словарь (: VAR);
- управляют состоянием словаря (FORGET :: CLEAR);
- или даже самим процессом выполнения задания в системе (LOAD SAVE).

ДССП характеризуется двухстековой архитектурой. Стек операндов, над которым действуют все операции по обработке данных, приводит к компактной польской инверсной записи арифметических выражений, а управляющий стек служит для сохранения адресов возвратов из процедур.

Следующие две характерные черты ДССП выгодно отличают её от системы ФОРТ:

- специфическая процедурная версия структурированных команд управления;
- и поддержка пошаговой (step-by-step) нисходящей (top-down) разработки программы.

В ДССП каждая из команд ветвлений вместо перехода по телу сшитого кода (как это обычно делается в системе ФОРТ) по сути осуществляет вызов процедуры слова, выбранного по условию, а каждая команда цикла организует многократный вызов процедуры слова, пока такой вызов допускается условием цикла. Таким образом, в ДССП поддержка структурированного программирования осуществляется непосредственно на уровне сшитого кода аналогично тому, как такая поддержка была обеспечена в ЦМ “Сетунь-70” на уровне машинных команд.

Определение нового слова в ДССП и в языке ФОРТ синтаксически выглядят одинаково:

: **P P1 P2 ... Pn** ; [определение новой процедуры с именем **P**]

Однако ФОРТ допускает подобное определение, только если все слова **P1 P2 ... Pn** уже были определены и стали известны системе. А ДССП допускает, что в этой последовательности слов **P1 P2 ... Pn** могут встречаться в том числе и такие слова, которые еще не были определены и предполагается, что будут определены впоследствии. Именно благодаря такой особенности ДССП и позволяет разрабатывать программу методом нисходящего анализа: сначала определить главное слово, затем слова следующего уровня, из которых оно определяется, и так постепенно спуститься до детального определения самых простых слов.

Первая версия ДССП

Первая версия ДССП (ДССП-НЦ) была создана (1980-1982 гг.) для микрокомпьютеров “Электроника НЦ-03Д” [3,4]. Уже с этой версии в ДССП нашли отражение характерные черты языка ассемблера ЦМ “Сетунь-70” [5]: его структурированные команды управления для организации ветвления (**BRS**) и цикла по условию (**DWON**).

Чтобы обеспечить технологию пошаговой нисходящей разработки программы, для внутреннего представления программы в этой версии ДССП был предложен особый вид сшитого кода, получивший название “сшитый код двойной косвенности”.

ДССП для компьютеров унифицированной архитектуры

На следующем этапе своего развития ДССП создавалась уже для микрокомпьютеров (“Электроника-НЦ80”) так называемой унифицированной архитектуры (PDP-11). Причём наряду с основной версией ДССП-80 [6] в этот период (1982-1984 гг.) разрабатывались также и экспериментальные версии, представляющие собой варианты развития ДССП, ориентированные на построение программ управления периферией (ДССП-ПМ [7]) и операционных систем реального времени (ДССП-РВ [8]). В каждой из этих версий использовался прямой сшитый код для внутреннего представления программы, а основной цикл внутреннего интерпретатора реализовывался одной командой базовой машины (**JSR @R5+**).

В ДССП-80 были предложены новые команды для организации циклов (**RP, DO**) с выходами по условию (**EX-, EX0, EX+**), а также типизированный набор префиксных операций (**! ' ...**) для единообразной работы с переменными и массивами различных типов. В составе базового ПО появился встроенный ассемблер [9], который позволил создавать тела новых слов в коде базовой машины. Была усовершенствована [10] организация словаря (он стал перемещаемым, в нём появились подсловари и секции), позволившая упростить разработку сложных многомодульных программ.

Для автоматизации сборки как самой ДССП, так и законченной прикладной программы были предложены специальные инструментальные средства: компоновщик [11] и целевой компилятор [12]. Причём, они были реализованы на языке самой ДССП.

Хотя экспериментальные версии ДССП-ПМ и ДССП-РВ не получили самостоятельного развития, предложенные в них новые программные механизмы были частично реализованы в

последующих версиях ДССП. Это механизм исключительных ситуаций, сопрограммный механизм, механизм прерываний на уровне сшитого кода, а также разнообразные средства организации и взаимодействия параллельных процессов.

Для успешного освоения студентами и школьниками навыков разработки программ в ДССП были изданы (1987-1988 гг.) учебные пособия и методические указания [13, 14]. Описанный в них язык ДССП (РАЯ) стал фактически стандартом на многие годы.

Чуть позже ДССП была реализована на компьютерах унифицированной архитектуры не только как самостоятельная резидентная система, но и как отдельное приложение, способное наряду с другими задачами функционировать в среде операционной системы РАФОС (RT-11) [15], а также ОСПВ (RSX-11) [16]. В словаре ДССП появились новые команды для взаимодействия со службами используемой ОС, в частности, для работы с файлами.

Распространение ДССП на другие компьютерные платформы

С начала 90-х годов ДССП стала активно использоваться в НИИСИ РАН в качестве инструментальной среды программирования для разработки внутреннего программного оснащения [17] ряда компьютерных модулей, создаваемых на основе разных микропроцессоров (MC68020, R3000, Intel80386).

В дальнейшем над развитием ДССП и распространением её на другие платформы трудились как сотрудники НИЛ ЭВМ МГУ вместе со студентами и аспирантами, так и сотрудники НИИСИ РАН [18].

В результате на протяжении ряда лет ДССП была реализована для компьютеров самой разнообразной архитектуры и в различных операционных средах. К середине 90-х годов ДССП функционировала почти на всех компьютерных платформах (PDP-11, Intel8080, Intel80x86, Motorola 68020, VAX, R3000) и в операционных средах (RT-11, RSX-11, UNIX, CP/M, MSDOS, Windows), получивших распространение в нашей стране.

ДССП для 32-разрядных микрокомпьютеров

Наряду с 16-разрядными появились и 32-разрядные версии ДССП (ДССП-32), оперирующие с 32-битными данными, адресами и стеками. Для платформы Intel80386 была создана версия ДССП-32р [19], позволившая под управлением ДОС-экстендера DOS4gw работать (в защищённом режиме) с расширенным объёмом памяти. Тем самым было преодолено досадное ограничение (640Кб) на расширяемость самой системы, препятствующее построению прикладных программ большого размера.

Впоследствии (1997-98 гг.) ядро ДССП-32р подверглось существенной модернизации. Во-первых, в ядро был встроен новый механизм для прерывания на уровне сшитого кода [20]. Этим механизмом обеспечивалось прерывание ДССП-программы не после исполнения очередной команды базовой машины, а лишь после завершения очередной ДССП-команды. Тем самым гарантировалась неделимость исполнения слов-примитивов, т.е. операций, реализованных в коде базовой машины. С таким механизмом стало возможным исполнять в качестве реакции на прерывание любую ДССП-команду, в том числе, возбуждать ситуацию в текущем процессе, переключаться на другой параллельный процесс или посылать сигнал для возобновления задержанного процесса.

Во-вторых, в ядро ДССП был добавлен сопрограммный механизм [21]. И если раньше диспетчеры параллельных процессов в ДССП [22] можно было реализовать только в коде базовой машины (на языке её ассемблера), то новые механизмы (прерываний и сопрограмм) позволили создавать разнообразные мониторы, управляющие развитием параллельных процессов, как программные модули, полностью разработанные на языке самой ДССП.

Мобильная версия ДССП/С

Отдельные версии ДССП для разных платформ были плохо совместимы друг с другом. Их общий недостаток заключался в том, что ядро каждой из них создавалось на языке ассемблера базовой машины, что затрудняло перенос созданного ПО на другие компьютерные платформы. В 1998 г. в НИИСИ РАН по инициативе Шумакова М.Н. была создана мобильная версия ДССП (ДССП/С), ядро которой было написано на языке Си [23].

ДССП/С обеспечивает мобильность получаемого кода. Это означает, что программа, разработанная на языке ДССП в одной среде реализации ДССП/С, может напрямую исполняться в другой среде реализации ДССП/С, не требуя каких-либо изменений ни в исходном тексте самой программы, ни в полученном итоговом исполняемом коде.

ДССП/С использует особый знаковый шитый код с относительной адресацией, специально приспособленный для обеспечения мобильности интерпретатора на языке Си (тело примитивной процедуры представлено в нём номером Си-функции ядра интерпретатора, которая и реализует ее операцию). Благодаря отмеченному свойству переносимости, разработанное на языке Си ядро ДССП, первоначально реализованное для работы в ОС семейства Windows, было затем успешно перенесено и на другие платформы (ОС Solaris для машины Ultra Sparc, MS-DOS с DOS-extender DOS4gw, Linux).

При разработке ядра на языке Си была проведена такая модернизация работы с данными, благодаря которой в ДССП впоследствии были обеспечены возможности построения новых типов [24], аналогичных средствам объектно-ориентированного программирования.

ДССП в мире науки и современной компьютерной техники

Таким образом, на протяжении почти 20-летнего периода своего развития от версии к версии (ДССП-НЦ ==> ДССП-80 ==> ДССП-32 ==> ДССП-32p ==> ДССП/С) ДССП постоянно совершенствовалась, приобретая новые возможности, присущие современным языкам и системам программирования. В результате ДССП превратилась в среду программирования [25], позволяющую применять широкий спектр современных методов структурированной разработки сложных программных систем, сосредоточив в себе средства для структурированного, модульного, объектно-ориентированного и параллельного программирования, а также средства для структурированной обработки исключительных ситуаций.

Однако, несмотря на отмеченные достоинства, ДССП всё же не получила достаточно широкого распространения на рынке систем программирования для микрокомпьютеров. Надо признать, что её потеснили в этой области интегрированные среды программирования с языками Си и Паскаль. На наш взгляд, причиной тому стала вовсе не ущербность самой ДССП, а вступившие в силу рыночные законы. ДССП проиграла конкурентную борьбу, которую и не могла бы выиграть в сложившихся условиях. Ведь ДССП создавалась, развивалась и предлагалась не как рыночный товар, а как научный продукт, результат ряда научно-исследовательских работ, направленных на усовершенствование архитектуры микрокомпьютеров и их системного ПО.

На протяжении многих лет ДССП использовалась для выполнения курсовых и дипломных работ, а также для разработки полезных программ, имеющих важное научно-практическое значение. Перечислим некоторые из них:

- Программное оснащение автоматизированной системы обучения “Наставник” [26].
- Система автоматизированной подготовки учебных курсов для АСО “Наставник” [27].
- Кросс-система для микропроцессора K580 в ДССП [28].
- Производственная система в ДССП-80 [29].
- Адаптируемая отладочная система разработки микроконтроллерных устройств [30].
- Система внутреннего программного оснащения DPROM [17].
- Система компьютеризации булевой алгебры в ДССП [31].
- DED – экранный отладчик и редактор в стиле “микромир” [33].

В качестве подтверждения высокой научной оценки ДССП ниже приведён список подготовленных кандидатских диссертаций, темы которых были тесно связаны с её созданием и развитием [в скобках указан год защиты диссертации]:

- [1983] Златкус Г.В. Диалоговая Система Структурированного программирования (ДССП) для микрокомпьютеров.
- [1986] Руднев И.А. Исследование путей и реализация на основе ДССП целевой компиляции программ для встраиваемых микропроцессоров.
- [1987] Сидоров С.А. Исследование мобильности ДССП и разработка средств её переноса.

- [1988] Захаров В.Б. Средства разработки программ в ДССП.
- [1991] Грачёв А.Ю. Структурирование данных в диалоговой системе программирования ДССП.
- [1996] Бурцев А.А. Расширение языков программирования средствами обработки исключительных ситуаций.

Результаты научных исследований, связанные с разработкой ДССП, регулярно публиковались в журналах и сборниках трудов, посвящённых компьютерной тематике, а также докладывались на многих международных конференциях (в том числе и по-английски [32-36]).

В 1996 году благодаря усилиям сотрудников НИИСИ РАН Шумакова М.Н. и Локтева Н.М. в Интернете (по адресу: www.dssp.msk.ru) открылся первый сайт, посвященный ДССП.

В настоящее время информацию о ДССП в Интернете можно найти по адресам:

- www.forth.org.ru/~dssp (сюда “переехал” первый ДССП-сайт Шумакова и Локтева);
- www.computer-museum.ru/histsoft/dssp.htm (страничка ДССП на сайте виртуального компьютерного музея).

Как всякий “бесценный” научный продукт, произведённый в нашей стране, ДССП распространялась фактически бесплатно. В советское время она передавалась в эксплуатацию по договорам о научно-техническом сотрудничестве, а с появлением сайта ДССП в Интернете её мог скачать (причём вместе с исходными текстами) любой желающий. Можно сказать, ДССП была открытой системой [18] и распространялась как свободное ПО.

ДССП интенсивно развивалась, пока не возникало проблем с финансовым обеспечением сопутствующих ей направлений научных исследований. Но резкое сокращение научного финансирования привело к вынужденному прекращению ряда работ по дальнейшему развитию ДССП. Может быть, вследствие этого так и не были воплощены в ДССП многие полезные идеи, присущие современным системам программирования. Например, ДССП так и не превратилась в визуальную среду разработки программ с привлекательной графической оболочкой, и до сих пор функционирует как консольное приложение, предлагающее пользователю простой диалог в режиме командной строки.

ДССП для троичной виртуальной машины

В настоящее время, когда двоичные цифровые машины приблизились к потолку своих потенциальных возможностей, назрела необходимость построения троичного процессора на современной элементной базе, а также разработки программного оснащения для него. Поэтому в НИЛ ЭВМ МГУ была форсирована работа по созданию имитационной модели троичного процессора, а также кросс-системы разработки троичных программ для него. В качестве основного языка в этой системе разработки предполагалось реализовать (кроме языка ассемблера) троичный вариант языка ДССП (ДССП-Т).

В качестве имитационной модели троичного процессора был предложен (в 2010 г.) вариант реализации программного комплекса (на языке Си), получивший название ТВМ – троичная виртуальная машина. ТВМ запускается как консольное приложение, принимает для исполнения файл закодированной троичной программы и имитирует эффект её исполнения. При этом ТВМ позволяет не только проследить (в замедленном режиме) все стадии процесса исполнения троичной программы, но и управлять ходом этого процесса в режиме диалога (например, прогонять программу по частям, приостанавливая её в нужных точках для просмотра содержимого регистров и ячеек памяти). В качестве инструментального средства подготовки закодированной троичной программы вместе с ТВМ был разработан и ассемблер создаваемого троичного процессора.

При разработке архитектуры ТВМ пришлось принимать множество решений – от представления троичной информации и формата машинного слова до состава команд и принципов взаимодействия с двоичной периферией. При неограниченных возможностях программной реализации основным упорядочивающим фактором стало желание воплотить такой микропроцессор в виде реальной микросхемы в ближайшем будущем.

В двоичных машинах троичный элемент данных – трит – можно представлять парой битов (p,n), значения которых: $(1, 0)$, $(0, 0)$, $(0, 1)$ трактовать соответственно как троичные

значения: **1, 0, -1**. Трайты (из 9 тритов) и троичные машинные слова (из 3-х трайтов или 27-и тритов) можно представлять уже парой битовых векторов ($p[s]$, $n[s]$) (размером $s=9$ для трайта и $s=27$ для троичного слова), которым в двоичных машинах могут соответствовать s -разрядные двоичные слова. На уровне языка ассемблера ТВМ (и далее на уровне языков более высокого уровня) в программе будут представлены уже только троичные данные. Такой подход позволяет ориентироваться при реализации троичного процессора на существующую двоичную элементную базу (в ожидании лучших времён, когда будут созданы реальные троичные элементы и троичные САПР).

На протяжении многих лет ДССП использовалась в качестве выгодной альтернативы программированию на языке ассемблера. Чтобы обеспечить возможность разработки программ для ТВМ на языке ДССП-Т, была предложена реализация (на языке Си) кросс-компилятора с этого языка на язык ассемблера ТВМ. Вместе с кросс-компилятором было разработано ассемблерное ядро, библиотека стандартных модулей и командный интерпретатор для прогона получаемых ДССП-программ в режиме диалога. Все вместе эти программные средства составляют единую кросс-систему разработки программ для троичной виртуальной машины, получившую название ДССП-ТВМ.

Язык ДССП-Т имеет ряд особенностей, которыми он отличается от входных языков различных версий ДССП, реализованных для разнообразных двоичных машин. Основу языка ДССП-Т составляет неполное подмножество базового языка ДССП [13], в котором отсутствуют типичные операции двоичной арифметики, логики и доступа к памяти по двоичным адресам – вместо них предлагаются соответствующие операции, присущие троичной логике и троичной симметричной арифметике. Поскольку язык ДССП-Т ориентирован на обработку не в режиме интерпретации, а в режиме компиляции, в нём отсутствуют средства для построения новых компилируемых слов, действующих в режиме компоновки программы.

При разработке языка ДССП-Т предполагалось воплотить в нём лучшие черты всех предыдущих версий ДССП. Поэтому в него добавлен ряд полезных выразительных средств, которые уже давно хорошо зарекомендовали себя в различных версиях ДССП.

Разработка внутреннего представления, интерпретатора и компилятора ДССП-программ всегда подчинялась принципу “одно слово текста переводится в одно слово кода”. Следуя этому принципу, удалось разработать такую систему команд ТВМ, которая позволяет ей напрямую исполнять программный код, являющийся внутренним представлением ДССП-программы. Так что теперь можно сказать, что формируемый ДССП-компилятором сшитый код станет действительно процедурным кодом, подлежащим непосредственному исполнению аппаратно реализованной машиной.

Библиография

1. Брусенцов Н.П., Златкус Г.В., Руднев И.А. ДССП - диалоговая система структурированного программирования. // Программное оснащение микрокомпьютеров. М.: Изд-во МГУ, 1982, с.11-40.
2. Брусенцов Н.П., Рамиль Альварес Х. Структурированное программирование на малой цифровой машине. // Вычислительная техника и вопросы кибернетики. Вып. 15. М.: Изд-во МГУ, 1978, с.3-8.
3. Златкус Г.В. Принципы построения системы структурированного программирования для микрокомпьютеров. // Вычислительная техника и вопросы кибернетики. Вып. 19. М.: Изд-во МГУ, 1982, с.73-76.
4. Захаров В.Б., Златкус Г.В., Руднев И.А., Сидоров С.А. Реализация диалоговой системы структурированного программирования на микрокомпьютере “Электроника НЦ-03Д”. // Архитектура и программное оснащение цифровых систем. М.: Изд-во МГУ, 1984, с.10-17.
5. Златкус Г.В. Ассемблер малой ЦМ “Сетунь-70”. // Вычислительная техника и вопросы кибернетики. Вып. 17. М.: Изд-во МГУ, 1981, с.26-34.
6. Брусенцов Н.П., Захаров В.Б., Руднев И.А., Сидоров С.А. Диалоговая система структурированного программирования ДССП-80. // Диалоговые микрокомпьютерные системы. М.: Изд-во МГУ, 1986, с.3-21.

7. Бурцев А.А. Периферийный монитор – развитие архитектуры ввода/вывода ДССП. // Диалоговые микрокомпьютерные системы. М.: Изд-во МГУ, 1986, с.42-51.
8. Борисов А.В. Диалоговая система структурированного программирования в реальном времени – ДССП-РВ. // Диалоговые микрокомпьютерные системы. М.: Изд-во МГУ, 1986, с.51-62.
9. Чанышев Н.А. Язык встроенного ассемблера для ДССП-80. // Диалоговые микрокомпьютерные системы. М.: Изд-во МГУ, 1986, с.21-27.
10. Захаров В.Б., Мельникова Е.А. Организация словаря в ДССП-80. // Диалоговые микрокомпьютерные системы. М.: Изд-во МГУ, 1986, с.38-42.
11. Сидоров С.А. Исследование переносимости ДССП. // Диалоговые микрокомпьютерные системы. М.: Изд-во МГУ, 1986, с.30-37.
12. Руднев И.А. Целевая компиляция ДССП-программ. // Диалоговые микрокомпьютерные системы. М.: Изд-во МГУ, 1986, с.28-30.
13. Брусенцов Н.П., Захаров В.Б., Руднев С.А., Сидоров С.А., Чанышев Н.А. Развиваемый адаптивный язык РАЯ диалоговой системы программирования ДССП. М.: Изд-во Моск. Ун-та, 1987 г. – 80 с.
14. Брусенцов Н.П., Захаров В.Б., Руднев С.А., Сидоров С.А., Чанышев Н.А. Методические указания по программированию в ДССП. Ротапринт НИВЦ МГУ, 1988 г. – 40 с.
15. Чанышев Н.А. ДССП на персональных компьютерах в операционной среде RT-11. // Программное оснащение персональных компьютеров. М.: Изд-во Моск. ун-та, 1990, с.28-32.
16. Чанышев Н.А. ДССП в операционной среде ОС-РВ. // Математические проблемы информатики. М.: Изд-во Ленингр. ун-та, 1987, с.40-44.
17. Лякина Е.Н., Пшеничный К.А., Сидоров С.А., Шумаков М.Н. Система внутреннего программного оснащения DPROM. // Вопросы кибернетики. Сб. статей под ред. В.Б.Бетелина. М., 1999. с.77-86.
18. Сидоров С.А., Шумаков М.Н. ДССП как открытая система. // Дискретные модели. Анализ, синтез и оптимизация. Спб.: СпбГУ, 1998. с.191-201.
19. Вольнец Р.В. Реализация ДССП для микропроцессоров Intel 80386. // Дискретные модели. Анализ, синтез и оптимизация. Спб.: СпбГУ, 1998. с.201-227.
20. Бурцев А.А. Механизм прерываний для интерпретатора сшитого кода. // Вопросы кибернетики. Сб. статей под ред. В.Б.Бетелина. М., 2005. с.177-191.
21. Бурцев А.А., Шумаков М.Н. Сопрограммный механизм в ДССП как основа для построения мониторов параллельных процессов. // Вопросы кибернетики. Сб. статей под ред. В.Б.Бетелина. М., 1999. с.45-63.
22. Вольнец Р. Диспетчер параллельных процессов в ДССП-32р. // Интегрированная система обучения, конструирования программ и разработки дидактических материалов. М.: Изд-во факультета ВМиК МГУ, 1996. с.70-80.
23. Бурцев А.А., Франтов Д.В., Шумаков М.Н. Разработка интерпретатора сшитого кода на языке Си. // Вопросы кибернетики. Сб. статей под ред. В.Б.Бетелина. М., 1999. с.64-76.
24. Бурцев А.А., Рамиль Альварес Х. Средства объектно-ориентированного программирования в ДССП. // Программные системы и инструменты. Тематический сборник №4, М.: Изд-во факультета ВМК МГУ, 2003. с.166-175.
25. Бурцев А.А. ДССП – среда структурированной разработки программ как сложных систем. // Вторая Международная конференция “Системный анализ и информационные технологии” САИТ-2007 (10-14 сентября 2007 г., Обнинск, Россия): Труды конференции. Изд-во ЛКИ, 2007. т. 2. с.190-194.
26. Брусенцов Н.П., Маслов С.П., Рамиль Альварес Х. Микрокомпьютерная система обучения “Наставник”. –М.: Наука, Гл. ред. физ.-мат. лит., 1990. гл.4. с.115-145.
27. Лозовая Е.А. Система автоматизированной подготовки учебных курсов для МСО “Наставник”. // Дискретные системы и их программное обеспечение. Л.: Изд-во Ленингр. Ун-та, 1990. с.149-154.
28. Сидоров С.А. Кросс-система для микропроцессора К580 в ДССП. // Автоматизация разработки и моделирование вычислительных и микропроцессорных систем: Материалы семинара. М., 1983. с.119-122.

29. Грачев А.Ю. Продукционная система в ДССП-80. // Программное оснащение персональных компьютеров. М.: изд-во МГУ, 1990, с. 51-63.
30. Лобанов В.И., Локтев Н.М. Адаптируемая отладочная система разработки микроконтроллерных устройств (МКУ) в среде ДССП. (<http://trinary.ru/kb/b072e4dd-defd-46a7-ae2e-7180d31eb98c>).
31. Владимирова Ю.С. Конструктивная реализация булевой алгебры. // Интегрированная система обучения, конструирования программ и разработки дидактических материалов. М.: Изд-во факультета ВМиК МГУ, 1996. с.44-69.
32. Sidorov S.A., Shumakov M.N. DSSP and Forth. Compare analysis. // EuroForth'96 conference proceedings. St.Peterburg, Russia, 1996. (11 pp).
33. Frantov D.V., Shumakov M.N. DED (DSSP Editor + Debugger). // EuroForth'97 conference proceedings. Oxford, England, 1997. p.145-153.
34. Sidorov S.A. Data in DSSP – prefix access in postfix language. // EuroForth'97 conference proceedings. Oxford, England, 1997. p.35-41.
35. Sidorov S.A. Top-down thinking and top-down writing in DSSP. // EuroForth'98 conference proceedings. Schloss Dagstuhl, Germany, 1998 (5 pp.).
36. Burtsev A.A. Interrupt mechanism for threaded code interpreter. // EuroForth'99 conference proceedings. St.Peterburg, Russia, 1999. (8 pp).

Доложено на 2-ой международной конференции “Развитие вычислительной техники и её программного обеспечения в России и странах бывшего СССР”, проходившей в г. Великий Новгород с 12 по 16 сентября 2011 г.

Опубликовано: Труды SORUCOM-2011. Вторая Международная конференция “ Развитие вычислительной техники и её программного обеспечения в России и странах бывшего СССР” (12-16 сентября 2011 г., г. Великий Новгород, Россия) с. 83-88.