

**СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА**

P10-86-712

С.М.Фроликов, М.Н.Шумаков

**ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕССОР,
ЭМУЛИРУЮЩИЙ ПРОМЕЖУТОЧНЫЙ КОД
КОМПИЛЯТОРА ФОРТРАНА**

1986

ВВЕДЕНИЕ

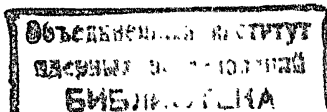
Обработка экспериментальной информации с установок физики высоких энергий требует выполнения большого количества вычислений, основная масса которых приходится на долю программ геометрической реконструкции и отбора событий. Ядро этих программ не занимает большого объема оперативной памяти и слабо использует большинство ресурсов ЭВМ, кроме, разумеется, центрального процессора. Поэтому целесообразным становится использование для первичной обработки информации микропроцессорных устройств, ориентированных на быстрое выполнение вычислительных задач^{1,2}. В дальнейшем такие устройства будем называть вычислительными процессорами /ВП/.

Сформулируем требования, которым должен удовлетворять ВП:

А. Процессор должен выполнять программы, написанные на языке высокого уровня /желательно - Фортране/ и отлаженные на универсальной базовой ЭВМ. Собственных средств программирования и собственных периферийных устройств ВП может не иметь. Поскольку на экспериментальных установках используются чаще всего ЭВМ типа СМ-4, удобно, чтобы процессор исполнял программы, написанные для этих ЭВМ.

Б. Входными данными для программ геометрической реконструкции и предварительного отбора событий являются целые числа в диапазоне от нуля до нескольких тысяч /номера проволок, величины сигналов с кодировщиков и т.д./, выходными - действительные величины /импульсы, координаты и т.д./, имеющие относительную точность не лучше долей процента и меняющиеся в небольшом диапазоне /не более чем в тысячи раз/. В этих задачах редко встречается большое количество итераций, на которых можно получить заметную ошибку вычислений. Поэтому в ВП достаточно ограничиться реализацией операций только с целыми 16-разрядными числами и 32-разрядными числами с плавающей запятой.

В. Требования к быстродействию ВП оценим на примере установок ОИЯИ на У-70 "Позитроний"^{1/5} и "БИС-2"^{1/6}. За один цикл ускорителя /8 секунд/ на магнитную ленту записывается информация, обработка которой на ЭВМ ЕС-1040 занимает для установки "Позитроний" - до 20 и для установки БИС-2 - до 300 секунд. Следовательно, для геометрической реконструкции всех событий с целью отбора перед записью на ленту нужно иметь процессор /или систему процессоров/ с быстродействием соответственно в 2,5 и 40 раз выше, чем быстродействие ЕС-1040.



АНАЛИЗ СПОСОБОВ ПОСТРОЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОРОВ

Отметим прежде всего, что ВП, построенный на базе доступных в настоящее время универсальных микропроцессоров, не позволяет достичь требуемого быстродействия из-за невысокой пропускной способности интерфейса между процессором и памятью, и из-за низкой скорости обработки чисел с плавающей запятой.

Первое ограничение проявляется в том, что даже у наиболее быстро процессора K1810VM86 цикл обращения к памяти занимает не менее 800 нс /4 тактов микропрограммы/, что в несколько раз превышает собственное время цикла микросхем памяти. Причины низкой пропускной способности канала процессор - память заключаются

- в использовании асинхронного протокола, единого для связи с памятью и внешними устройствами,
- в неполной конвейеризации путей пересылки данных,
- в совмещении шины адресов и данных. /Заметим, что в новейших микропроцессорах типа iRX386 или M68020 эти причины устранены или их влияние ослаблено/.

Более удобной элементной базой для построения ВП являются микропроцессорные секции. Их применение предоставляет разработчику большую свободу в выборе различных способов повышения быстродействия. Так, за счет применения быстрых микросхем памяти и синхронного /без анализа готовности/ протокола цикл обмена с ОЗУ удается выполнить за 1 такт микропрограммы. Далее у разработчиков появляется возможность выбрать архитектуру и систему команд, ориентированных на решение нужного класса задач.

Таким образом, в рамках одного уровня технологии производства интегральных схем ВП на секционных микропроцессорах могут иметь существенно более высокое отношение производительность/затраты на изготовление, чем ВП на универсальных микропроцессорах. Очевидной трудностью является отсутствие развитого программного обеспечения, но ее удается обойти, выбрав для ВП систему команд, позволяющую использовать программы, создаваемые на универсальной ЭВМ. Сделать это можно следующими способами:

1. Прямая эмуляция подмножества системы команд универсальной ЭВМ

Выигрыш в отношении производительность/стоимость здесь достигается главным образом за счет упрощения способа взаимодействия процессора с ОЗУ и новой элементной базы.

Однако все производимые над данными действия можно повторить и на процессоре, не имеющем строгой привязки к архитектуре машины - прототипа. В этом случае выигрыш достигается не только за счет аппаратуры, но и за счет системы команд и архитектуры. Рассмотрим такие не прямые способы эмуляции.

2. Эмуляция машинных команд последовательностями микрокоманд

В универсальных ЭВМ каждая машинная команда вызывает выполнение некоторой последовательности действий, задаваемой обычно в виде микропрограммы. Кроме собственно действий над данными выполняются и различные операции служебного характера: выборка команды и параметров, дешифровка команды и т.д.

С помощью несложного транслятора можно представить исходную программу в виде последовательности одних только микроопераций. Количество служебных операций при этом легко сократить до минимума. Так, ненужными становятся выборка и дешифровка команды. Параметры команды /адреса, прямые данные/ указываются в явном виде в теле микрокоманды.

Для выполнения программ, заданных в виде последовательности микроопераций, можно построить процессор с очень простой структурой и высоким быстродействием.

Примером является используемый на многих ускорителях процессор 168/E/1/. 168/E состоит из трех основных частей: ЦПУ на микропроцессорных секциях AM2901, ОЗУ микропрограмм с 24-битным словом и ОЗУ данных с 16-битным словом. Быстродействие 168/E сравнимо с быстродействием ЭВМ IBM 370/168, систему команд которой он эмулирует.

Однако 168/E имеет и очевидные недостатки:

- Каждая команда IBM 370 часто выполняется с помощью нескольких команд 168/E, что приводит к большому расходу ОЗУ микропрограмм.

- Из-за малой длины микропрограммного слова ограничивается возможность повысить быстродействие за счет распараллеливания микроопераций. /Увеличение длины микропрограммного слова привело бы к росту стоимости ОЗУ микропрограмм, не пропорциональному росту быстродействия/.

3. Эмуляция промежуточного кода.

Система команд процессора ФП

С помощью микропроцессорных секций можно эмулировать не только систему команд физически существующей ЭВМ, но и систему команд какого-либо программного объекта. Удобным для нас объектом является промежуточный код /ПК/ компилятора с языка высокого уровня.

Разбиение компиляции на 2 стадии - получение ПК и генерацию машинного кода возникло в результате попыток формализовать построение компиляторов. Оно облегчает, в частности, перенос компилятора с машины на машину, т.к. переделывать приходится только вторую часть компилятора. И наоборот, для построения нового компилятора бывает достаточно создать только его первую часть. Таким образом один и тот же ПК может использоваться в разных

компиляторах, в разных операционных системах и даже на разных машинах. Поэтому процессор, ориентированный на выполнение команд ПК, можно считать в известном смысле универсальным. Машины с близкими к ПК системами команд обсуждаются, например, в [3,4].

Авторами разработан процессор ФП /фортран-процессор/, эмулирующий подмножество системы команд ПК компилятора с Фортрана операционных систем RT-II и RSX-II.

Этот компилятор может вырабатывать объектный код двух типов - прямой /inline/, представляющий собой последовательность машинных операций, и сшитый или нитевидный /threaded/ код, представляющий собой последовательность вызовов библиотечных подпрограмм. Сшитый код можно рассматривать как форму представления ПК, при его использовании программы получают более компактные, но выполняются медленнее.

В табл.1 приводятся прямые и сшитые коды, генерируемые при трансляции выражения $I = J * K + R$.

Таблица 1

Коды, вырабатываемые компилятором с Фортрана-IV
 ОС RT-II в результате компиляции выражения
 $I = J * K + R$

1	2	3	4	5	6
Сшитый код	Слов	Циклов ФП I	Прямой код (FIS)	Слов	Циклов ОЗУ CM
MOI\$MS J	2	3	I MOV J,R1	2	3
MUI\$MS K	2	3	I MUL K,R1	2	3
CFI\$	1	1	I MOV R1,(-SP)	1	2
ADF\$MS R	2	4	I JSR PC, \$CVTIF	2	3+N1
			I MOV R+2,-(SP)	2	4
			I MOV R, -(SP)	2	4
			I FADD SP	1	3
CIF\$	1	1	I JSR PC, \$CVTFI	2	3+N2
MOI\$SM I	2	3	I MOV (SP)+, I	2	4
	10	15		16	29+N1+N2

В столбцах 3 и 6 приведено количество циклов ОЗУ, необходимых для выполнения данного выражения в процессоре ФП и CM-4.

В ФП реализовано подмножество команд ПК, содержащее 29 операций над данными /16-разрядными целыми, 32-разрядными числами с плавающей запятой/ и 14 команд управления программой /табл.3/. Имеется 5 типов адресации /табл.2/.

Таблица 2

Типы адресации процессора ФП

- I - Операнд находится в слове, следующем за командой.
- M - Следующее слово содержит адрес операнда.
- P - Следующее слово содержит адрес адреса операнда.
- S - Операнд в стеке.
- A - В стеке - адрес операнда.

Таблица 3

Система команд процессора ФП

1. Двуместные операции над целыми и действительными числами

Мнемоника	Адресация				Операция
MOI\$..	I	IA	IM	IP	Пересылка целых чисел
MOF\$..	OS	OA	OM	OP	Пересылка действ.чисел
ADI\$..	SS	SA	SM	SP	Сложение целых чисел
SUI\$..	IS	IA	IM	IP	Вычитание целых чисел
ADF\$..	MS	MA	MM	MP	Сложение действ.чисел
SUF\$..	PS	PA	PM	PP	Вычитание действ.чисел
CMI\$..	SS	SI	SM	SP	Сравнение целых чисел
	IS	II	IM	IP	
CMF\$..	AS	AI	AM	AP	Сравнение действ.чисел
	PS	PI	PM	PP	
MUI\$..		SS	IS		Умножение целых чисел
MUF\$..					Умножение действ.чисел
DII\$..		MS	PS		Деление целых чисел
DIF\$..					Деление действ.чисел
AND\$SS		SS			Побитное "И"
IOR\$SS		SS			Побитное "ИЛИ"
XOR\$SS		SS			Побитное сложение

2. Одноместные операции

ICI\$.		Увеличение целого числа на единицу
DCI\$.	MS	Уменьшение целого числа на единицу
NGI\$.		Изменение знака целого числа
NGF\$.	AP	Изменение знака действ.числа

TSI\$.	S	I	Проверка знака целого числа
TSP\$.	M	P	Проверка знака действ.числа
CPI\$.	M		Скопировать стек в память
CPF\$.	M		/Спасение промежуточного результата/
CFI\$	S		Преобразовать целое число в действ.
CIF\$	S		Преобразовать действ.число в целое

3. Операции работы с массивами

SAI\$..	IM	IP	Помещает в стек адрес элемента массива целых чисел.
SAF\$..	SM	SP	Помещает в стек адрес элемента массива действ. чисел.
SVI\$..	MM	MP	Помещает в стек значение элемента массива целых чисел.
SVF\$..	PP	PP	Помещает в стек значение элемента массива действ. чисел.

Операндами являются индекс элемента массива и адрес первого элемента массива

4. Операции управления программой

BRA\$			Безусловный переход. Адрес в след.слове.
BLE\$	BGE\$	BGT\$	Переходы по условию. Адрес в след.слове.
BNE\$	BEQ\$	BLT\$	
AIF\$			Арифметический IF /трехадресный/
CAL\$			Вызвать подпрограмму, связать фактические и формальные параметры.
RET\$I	RET\$F		Возврат из подпрограммы-функции
RET\$			Возврат из подпрограммы

5. Операции конца цикла

	Переменная цикла	Адрес перехода	Крайнее значение переменной цикла	Шаг
NMI\$..	M	I	I M P	I I
NPI\$..	P			M P

Операции окончания цикла имеют три операнда, если шаг цикла равен единице, и четыре операнда, если шаг отличен от единицы. Например:

команда NMI\$II оканчивает цикл, начинающийся с оператора
D 0 IO I = 2, I, IO

команда NMI\$MM оканчивает цикл, начинающийся с оператора
D 0 IO I = MIN, MAX, STEP

команда NPI\$PP оканчивает цикл, начинающийся с этого же оператора, в том случае, когда MIN, MAX, STEP - формальные параметры подпрограммы.

6. Ассемблерные команды

NOP\$	Пустой оператор
CALL\$	Вызов подпрограммы без передачи параметров
STP\$	Стоп
SHL\$	Сдвиг целого числа влево
SHR\$	Сдвиг целого числа вправо
BYT\$L	Выделение левого байта
BYT\$R	Выделение правого байта
RDE\$	Прочитать в стек слово из расширенной памяти
WRE\$	Записать слово из стека в расширенную память
INE\$	Инкрементировать слово в расширенной памяти

Рассмотрим, что дает нам эмуляция ПК по сравнению с эмуляцией машинных команд.

А. Команды ПК задают действия над данными, которые во многих случаях могут быть выполнены только за несколько машинных команд. Поэтому в процессоре, выполняющем коды ПК, по сравнению с эмулятором машинных команд:

- Имеется больше возможностей для параллельного исполнения отдельных микроопераций.

- Уменьшается количество циклов, затрачиваемых на выборку команд.

- Обеспечивается компактность кода программы.

Наибольший выигрыш достигается в операциях для работы с элементами и адресами массивов, пересылок вещественных чисел, вызовов подпрограмм и операциях окончания циклов.

Б. В системе команд процессора ФП, в отличие от системы команд СМ ЭВМ, различаются операции со стеком, связанные с хранением адресов возвратов из подпрограмм /стек программ/ от опера-

ций со стеком, связанных с хранением промежуточных результатов вычислений /стек данных/. Со стеком данных процессор должен выполнять большую часть своих операций. Глубина стека данных может быть небольшой, поскольку он очищается по исполнению каждой фортрановской строчки. Поэтому стек для хранения промежуточных результатов вычислений в ФП оказалось возможным выполнить на регистрах микропроцессоров, что значительно подняло скорость вычислений.

В. ФП имеет гораздо меньшее количество команд и типов адресаций, чем СМ-4, что позволило выполнить каждую из команд в виде отдельной микропрограммы, а в качестве кода команды использовать адрес этой микропрограммы. Поэтому в ФП упрощаются схемы, реализующие различные способы адресации, и отсутствует дешифратор команд.

Г. В системе команд ФП не используются признаки ветвлений С и V, а генерация признаков N и Z требуется только в очень ограниченном количестве команд. Это позволило значительно упростить схемы выработки этих признаков.

Таким образом, эмулятор ПК, по сравнению с эмулятором машинных команд, имеет более высокое быстродействие, аппаратно устроен проще и использует более компактный код программы.

Перечисленные преимущества следует считать следствием того, что команды ПК являются командами более высокого уровня, чем машинные инструкции.

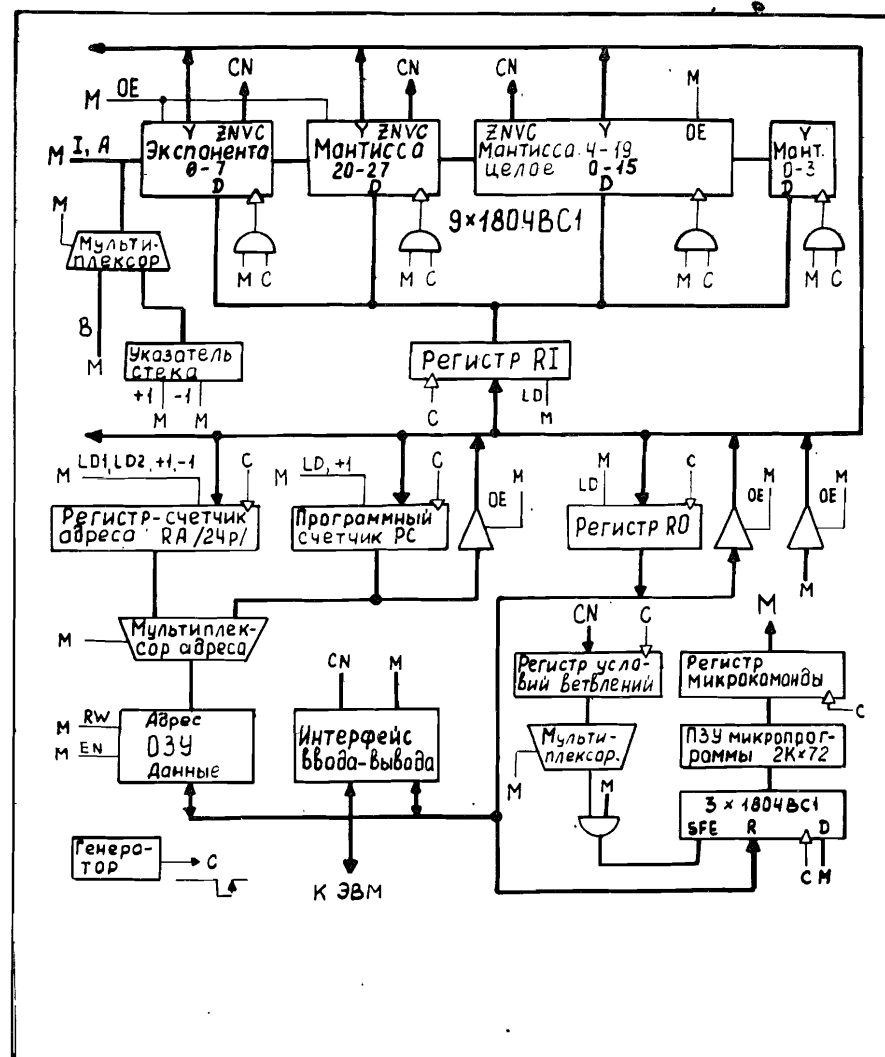
АППАРАТУРА ФОРТРАН-ПРОЦЕССОРА

Процессор построен в основном на микросхемах ТТЛШ серий К1804, К531, К555 и других. Структурная схема ФП приведена на рисунке. Процессор состоит из трех основных узлов:

- центральное процессорное устройство /ЦПУ/, содержащее АЛУ, рабочие регистры и стек данных;
- ОЗУ и система ввода-вывода;
- микроконтроллер с ПЗУ микропрограмм.

1. Центральное процессорное устройство

Основу ЦПУ составляют 9 микросхем секционированных микропроцессоров К1804ВС1 /АМ2901/, две из которых образуют 8-битный процессор для операций с экспонентой и семь других - 28-битный процессор для операций с мантиссой числа с плавающей запятой. Знаки действительных чисел обрабатываются в отдельном логическом устройстве. Четыре микросхемы процессора мантиссы предназначены также для выполнения операций над целыми 16-разрядными числами.



На все микросхемы поступает общий управляющий код микрокоманд. Для исполнения операции отдельно над экспонентой, мантиссой или целым числом тактирующий импульс подается только на микросхемы нужной группы. Возможно также исполнение операции сразу над всем числом с плавающей запятой /например, при пересылках/ или только с его старшей частью /экспонента + восемь старших разрядов мантиссы/, или с младшими разрядами мантиссы.

В ЦПУ используется внутренний 36-разрядный формат представления действительного числа, более удобный для обработки, чем стандартный 32-разрядный формат хранения в ЭВМ PDP-11. Во внут-

ренем: формате мантисса имеет дополнительные младшие разряды для округления, дополнительный старший разряд переполнения и разряд с восстановленной скрытой единицей. Преобразование форматов производится аппаратно при выполнении операций обмена между ЦПУ и ОЗУ процессора.

Все операции по обработке чисел с плавающей запятой производятся на АЛУ и регистрах микропроцессоров /в том числе операция сборки числа из двух 16-разрядных слов при чтении из памяти и операция обратной разборки при записи в память/. Поэтому, хотя ЦПУ оперирует 36-разрядными числами, все внешние по отношению к микросхемам К1804ВС1 шины данных являются 16-разрядными.

Среди других особенностей ЦПУ отметим повышение в три раза тактовой частоты ЦПУ при выполнении сдвигов мантиссы в операциях выравнивания порядков и нормализации. Для этого мантисса размещается в Q-регистре микросхем 1804ВС1, сдвиги которого можно производить гораздо быстрее, чем через АЛУ. Подсчет числа сдвигов ведется во внешнем счетчике, функции которого исполняет регистр RI /см.рисунок/.

2. Стек адресов возврата и стек данных

Как уже отмечалось выше, система команд ФП позволяет отделить стек программ от стека данных. Стек программ расположен в основной памяти ФП. В качестве указателя стека используется один из регистров общего назначения ЦПУ.

В качестве стека данных используются 8 регистров общего назначения ЦПУ. При обращении к стеку данных адрес регистров ЦПУ /порт В/ берется из реверсивного счетчика - указателя стека данных. Любой операнд, независимо от типа /целый или вещественный/, занимает в стеке данных один регистр.

3. Внутренняя шина

Все пересылки данных между ЦПУ и системой памяти и ввода-вывода производятся через одну 16-разрядную внутреннюю шину. Информация на шину может поступать из младшей части ЦПУ /при пересылке целых чисел или младших разрядов вещественных/, из старшей части ЦПУ /старшие разряды вещественных чисел/, из ОЗУ процессора, из микрокоманды /прямые данные/ и др. Приемниками могут быть входной регистр RI ЦПУ, программный счетчик РС и регистры адреса и данных ОЗУ процессора.

4. Память данных и программ

Источником адреса памяти могут служить программный счетчик РС /в операциях выборки команд/ или регистр-счетчик адреса RA. Во время записи данные берутся из регистра вывода R0. Во время чтения данные передаются на внутреннюю шину или в регистр команды микроконтроллера, если выбирается код команды.

Операции обращения к памяти и операции в ЦПУ могут выполняться одновременно за счет использования конвейерных регистров в канале передачи адреса и данных к памяти. Функции конвейерных регистров исполняют регистры R0, RA и РС.

Обращение к памяти производится без анализа готовности и занимает один такт микропрограммы. Длительность такта, при использовании микросхем памяти с временем цикла меньше 140 нс, определяется только задержками в ЦПУ. В ФП используются микросхемы статического ОЗУ К132РУ5Б /4Кбит/ и TMS4167 /16К/ с временем цикла 120 нс.

Реализованные в процессоре команды промежуточного кода транслятора позволяют прямо адресовать ОЗУ объемом до 32Кслов. Для работы с большими массивами данных имеется группа ассемблерных команд, которые формируют 24-разрядный адрес ОЗУ и позволяют адресовать до 8Мслов памяти данных.

5. Интерфейс с ЭВМ

ЭВМ управляет процессором ФП посредством приказов, список которых включает приказы пересылки массива из ЭВМ в память ФП и из памяти ФП в ЭВМ, исполнения программы, передачи в ЭВМ информации об ошибках и другие.

Связь между ЭВМ и ФП осуществляется через 16-разрядный параллельный интерфейс, который используется для передачи и приказов и данных. В режиме пересылки массива ФП может передавать или принимать информацию через параллельный интерфейс со скоростью до 3Мбайт в секунду.

6. Микроконтроллер и ПЗУ микропрограмм

Длина микрокоманды составляет 72 разряда. Память микропрограмм имеет объем 2Кслов и содержит 9 микросхем ПЗУ К556Р17.

Микроконтроллер выполнен на основе микросхем К1804ВУ1 /АМ2909/. Используется схема выборки микрокоманд с двумя конвейерными регистрами: регистром микрокоманды и регистром условий ветвлений в ЦПУ/10/.

7. Конструктив

Опытный образец ФП содержит 134 микросхемы /без микросхем ОЗУ/ и выполнен на макетной плате Фастбас. В дальнейшем предполагается разместить процессор в двойном модуле КАМАК и в одиночном модуле - ОЗУ для него.

8. Оценка быстродействия

Длительность такта микропрограммы составляет 200 нс. Приведем времена исполнения некоторых команд:

ADI\$SS	/сложение целых; операнды в стеке/	0,6 мкс
ADI\$MM	/сложение целых, операнды в памяти/	1,2 мкс
MUI\$SS	/умножение целых/	4,4 мкс /среднее/
DII\$SS	/деление целых/	4,9 мкс /среднее/
ADF\$SS и SUF\$SS	/сложение и вычитание вещ. чисел/	3,4-6,6 мкс
MUF\$SS	/умножение вещественных/	6,5 мкс /среднее/
DIF\$SS	/деление вещественных/	7,8 мкс /среднее/
BGE\$	/переход по условию/	0,5 мкс /среднее/
CAL\$	/вызов подпрограммы, 2 формальных параметра/	3,8 мкс
NMI\$1M	/оператор конца цикла/	2,2 мкс

Быстродействие ФП, измеренное на программе BENCHMARK/7/ /без спецфункций/, составляет ~ 1 млн операций в секунду.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ФП

П.0. ФП разработано в рамках Диалоговой системы структурированного программирования /ДССП/8/. В его состав входят:

1. Утилиты для работы в рамках ДССП с файлами RT-11.
2. Микроассемблер ФП и разработанные с его помощью микропрограммы.
3. Мониторная программа.
4. Ассемблер ФП.
5. Программа RTFP, перерабатывающая исполнительный файл ОС RT-11 в исполнительный файл ФП.
6. Отладчик.

Рассмотрим эти программы подробнее:

1. Утилиты для обмена файлами между ДССП и RT-11 составляют словарь директив /DIR, COPY, DUMP, TYPE/, аналогичных соответствующим директивам RT-11. /Основу этих утилит составляют процедуры, разработанные в проблемной лаборатории ЭВМ МГУ./

2. Микроассемблер ФП описан в/9/. Объем микропрограммной памяти составляет 2К 72-битных слов, из них в настоящее время используется чуть больше половины. Оставшаяся часть предполагается использовать для расширения системы команд ФП в соответствии с требованиями практики.

3. Мониторная программа позволяет использовать терминал ЭВМ в качестве пульта ФП. Эта программа взаимодействует с интерфейсной микропрограммой в ФП, которая организует через параллельный интерфейс с ЭВМ чтение и запись ячеек памяти ФП, запуск программы и т.д. Мониторная программа выполняет те же команды, которые используются в ЭВМ Электроника-60 в режиме пульта терминала. Отличие состоит лишь в том, что вместо команд работы с регистрами используются команды работы со стеком и имеются команды для работы с 32-битными действительными числами.

4. Для процессоров, имеющих несложную систему команд, с помощью ДССП легко построить простой и достаточно эффективный язык ассемблирования/9/. Ассемблер ФП состоит из:

- словаря операций - 230 переменных, имена которых совпадают с именами команд ФП, а значения - с кодами этих команд;
 - переменной AF - адреса ОЗУ ФП;
 - процедуры ";", заносщей содержимое стека ДССП в область памяти ФП, начинающуюся с адреса AF.

Пример программы на ассемблере ФП приведен ниже.

```

100 ! ISUM
102 ! I /Описание переменных и их места в
        / памяти ФП/
VAR D05 /Описание метки/
200 ! AF /Стартовый адрес программы/
MOI$1M I ;; /D05 I = 1,7/
        AF ! D05 /Определяется значение метки/
ADI$MM I ISUM ;; /ISUM = ISUM+I/
NMI$1I I 7 D05 ;; /5 CONTINUE/

```

5. Программа RTFP. Для получения исполнительного файла процессора ФП достаточно в исполнительном файле, полученном после трансляции и компоновки фортрановской программы, вместо адресов библиотечных подпрограмм подставить коды соответствующих команд ФП. Прямые данные, адреса переменных и программ остаются без изменения.

Заметим, что ФП может выполнять программы существенно большего размера, чем ЭВМ СМ-4 с тем же объемом оперативной памяти, т.к. области библиотечных подпрограмм, резидентной части операционной системы и внешних устройств /в сумме ~ 20Кбайт/ остаются свободными.

6. Отладчик позволяет запускать программу с произвольного адреса, устанавливать точки контрольного останова, менять значения переменных и вносить изменения в код программы, причем для задания кода команды можно использовать мнемоники ассемблера ФП.

ПРОГРАММИРОВАНИЕ ВВОДА-ВЫВОДА

Обмен между ЭВМ и ФП осуществляется массивами 16-битных слов по инициативе ЭВМ. Однако с точки зрения программиста-пользователя имеется два варианта программирования операций ввода-вывода:

А. В фортрановской программе, исполняемой в ФП. В этом случае после запуска программы в ФП ЭВМ работает как процессор ввода-вывода. Команды ввода-вывода /табл.4/ ФП выполняет точно так же, как и команду STP\$ /СТОП/ - сообщает о благополучном окончании работы и адрес останова. ЭВМ читает код команды, по которой произошел останов, и, если это команда ввода-вывода, выполняет требуемые операции обмена данными с ФП и своими внешними устройствами, после чего запускает ФП с адреса, следующего после коман-

ILR\$	Указывает номер логического устройства ввода
ILW\$	Указывает номер логического устройства вывода
TVS\$	Вывод строки символов
TVI\$	Ввод или вывод целого числа в свободном формате
TVF\$	Ввод или вывод действ. числа в свободном формате
EOL\$	Конец строки

ды ввода-вывода. Данный способ обмена удобен для использования в программах разового пользования.

Б. С помощью процедур отладчика ФП. В этом случае программист, пользуясь информацией из выдаваемой компоновщиком RT-11 карты распределения памяти, создает с помощью отладчика ФП специальные процедуры ввода-вывода. Этот способ более трудоемок, но более удобен при выполнении расчетов по программам многоразового пользования и при использовании ФП в составе системы сбора данных, т.к. экономит память ФП и позволяет организовать диалог с оператором с помощью легко изменяемых директив ДССП, а не фиксированных вопросов-ответов, заданных в программе на Фортране.

ОГРАНИЧЕНИЯ ПО СРАВНЕНИЮ С ФОРТРАНОМ-IV ОС RT-11

1. Возможны только 16-битные числа типа INTEGER и 32-битные числа типа REAL /с плавающей запятой/. Работы с байтами и битами возможны только с помощью подпрограмм на ассемблере ФП.
2. Вложенность скобок в арифметических выражениях не должна превышать семи. Вложенность скобок в операторе CALL не должна превышать семи - для первого параметра, шести - для второго, пяти - для третьего и т.д. Подпрограмма не может иметь больше 7 формальных параметров. Это ограничение связано с тем, что под стек выделено только восемь регистров.
3. Имеющееся в настоящее время программное обеспечение допускает использование только свободного формата ввода-вывода и операторов ACCEPT и TYPE.

ЗАКЛЮЧЕНИЕ

Быстродействие ФП согласно программе WHETSTONE BENCH MARK^{17/} TEST составляет около 1 млн. операций в секунду при цикле процессора 180 нс. Быстродействие ЕС-1040, согласно этой программе, составило 330 тыс., СМ-4 - 80 тыс. операций в секунду.

Итак, для программ, написанных на Фортране, быстродействие СМ-4 и ФП отличается в 12 раз. Если же сравнить время выполне-

ния "архитектурно одинаковых", т.е. выполняющих одинаковые действия и требующих одинакового количества циклов обращения к памяти операций СМ-4 и ФП:

ФП		СМ-4	
ADI\$MM	1,2 мкс;	ADD	M1, M2 6,5 мкс;
MUI\$SS	4,4 мкс;	MUL	10,8 мкс;
MUF\$SS	6,5 мкс;	FMUL	34 мкс,

то оно отличается не более чем в 4-7 раз. Следовательно, за счет архитектуры и системы команд достигается примерно двухкратное повышение быстродействия для программ на Фортране.

В целом метод эмуляции промежуточного кода позволил построить достаточно быстродействующий, удобно программируемый и относительно простой процессор. По сравнению с другими эмуляторами^{12/}, построенными на похожей элементной базе, ФП имеет существенно более компактный код программ и примерно одинаковое быстродействие.

Авторы выражают благодарность В.В.Вицеву, обратившему наше внимание на возможность эмуляции промежуточного кода, Т.С.Григашвили и Л.Л.Неменову - за поддержку данной работы, Н.П.Брусенцову, В.Б.Захарову, И.А.Рудневу, С.А.Сидорову - за предоставленную ими возможность использовать новейшие версии и утилиты ДССП, Ю.А.Иванову и А.В.Рыжову - за помощь в изготовлении процессора. Авторы благодарят также С.Г.Бобкова, А.С.Яковлева и Н.Г.Минаева, проявивших постоянный интерес к данной работе и содействовавших ее выполнению.

ЛИТЕРАТУРА

1. Paul F.Kunz. The LASS Hardware Processor, SLAC-PUB-1733, 1976; Nucl.Inst.and Meth, 1976, 135, p.435.
2. Verkerk C. Use of Intelligent Devices in High-Energy Experiments. Proceeding of 1980 CERN School of Computing, Jadwisin, Poland, 1981, CERN, 81-03.
3. Wirth N. The Personal Computer LILITH, Eigenossische Technische Hochschule, Zurich, Institute für Informatik, 1981.
4. Брусенцов Н.П. Стековые машины с изменяемой адресностью команд. Вычислительная техника и вопросы кибернетики, МГУ, М., 1977, в.13.
5. Карпунин В.В. и др. В сб.: XII Международный симпозиум по ядерной электронике, ОИЯИ, Д13-85-793, Дубна, 1985, с.5.
6. Алеев А.Н. и др. ОИЯИ, 1-80-644, Дубна, 1980.

7. Curnow H.J., Witchmann B.A. Computer Journal, 1976, v.19, p.1.
8. Брусенцов Н.П. Микрокомпьютеры. "Наука", М., 1985.
9. Фроликов С.М., Шумаков М.Н. ОИЯИ, P11-86-76, Дубна, 1986, с.291.
10. Bipolar Microprocessor Logic and Interface, Data Book, AMD, 1985.

Фроликов С.М., Шумаков М.Н.

P10-86-712

Вычислительный процессор, эмулирующий
промежуточный код компилятора Фортрана

Описывается аппаратура и программное обеспечение быстрого вычислительного процессора ФП /фортран-процессор/, машинным языком которого является промежуточный код, вырабатываемый компилятором с Фортрана операционных систем RT-II и RSX-II. Быстродействие процессора на вычислительных задачах, использующих 16-битные целые и 32-битные действительные числа примерно в 12 раз выше, чем быстродействие СМ-4. Приблизительно шестикратное повышение быстродействия достигнуто за счет более быстрой аппаратуры и приблизительно двухкратное – за счет более эффективной системы команд и архитектуры. Процессор предназначен для использования в системах приема и отбора информации с установок физики высоких энергий и может быть также использован в качестве сопроцессора к ЭВМ типа СМ-4, Электроника-60.

Работа выполнена в Серпуховском научно-экспериментальном отделе ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1986

Перевод О.С.Виноградовой

Frolikov S.M., Shumakov M.N.

P10-86-712

Hardware Processor Emulating the Intermediate
Code of Fortran Compiler

Hardware and software of fast Fortran processor (FP) are described. Its machine language is an intermediate code of OS RT-II and RSX-II Fortran compiler. The time of execution of Fortran programs with 16-bit integer and 32-bit real is by a factor of 12 better than that of SM-4 computer. Approximately 6-fold increase in speed of response is achieved due to faster hardware and the 2-fold is consequence of more effective system architecture and instruction list. The processor is intended for the use in data acquisition systems of high energy physics setups and could be used as a co-processor to SM-4, Elektronika-60 type computers.

The investigation has been performed at the Serpukhov Scientific-Experimental Department, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1986

Рукопись поступила в издательский отдел
28 октября 1986 года.