исчерпывается. Благодаря вышеперечисленным достоинствам система достаточно универсальна и может быть применена для других классов задач.

ЛИТЕРАТУРА

1. **Баяковский Ю. М., Михайлова Т. Н., Мишакова С. Т.** ГРАФОР — комплекс графических программ на Фортране.— М., 1972. (Препринт/АН СССР, ИПМ; 41).
2. **Математическое** обеспечение графопостроителей. СМОГ, 1 уровень: Инструкция по программированию/Под ред. Ю. А. Кузнецова.— Новосибирск: ВЦ СО АН СССР, 1976.
3. **Энджел Й.** Практическое введение в машинную графику.— М.: Радио и связь, 1984.
4. **Пилюгин В. В., Сумароков Л. Н., Фролов К. В.** Машинная графика и автоматизация научных исследований // Вестн. АН СССР.— 1985.— № 10.
5. **Гилой В.** Интерактивная машинная графика.— М.: Мир, 1981.
6. **Фоли Дж., Вэн Дэм А.** Основы интерактивной машинной графики.— Т. 1—2.— М.: Мир, 1985.
7. **Штатнов Ю. Ю.** Редактирование графической и математической информации на ЭВМ. Институт горного дела СО АН СССР.— Новосибирск, 1981.— Рукопись деп. в ВИНИТИ, № 4391—81.
8. **Штатнов Ю. Ю.** К вопросу о редактировании математической информации с помощью графических методов. Институт горного дела СО АН СССР.— Новосибирск, 1984.— Рукопись деп. в ВИНИТИ, № 6206—84.

Поступила в редакцию 14 ноября 1986 г.

УДК 681.3.06

П. Л. ХРАПКИН

(Новосибирск)

ОБЗОР СРЕДСТВ ПРОГРАММИРОВАНИЯ АППАРАТУРЫ КАМАК

За годы применения стандарта КАМАК в нашей стране сформировался практический набор систем программирования такой аппаратуры. Цель этой статьи — не только рассмотреть многообразие наиболее распространенных средств программирования КАМАК, но и отразить стремление их к унификации, выявить наиболее удобные для массового применения. Предложено типовое решение, выбор которого основан на опыте по применению мини- и микроЭВМ серии СМ и «Электроника», накопленном в Институте ядерной физики СО АН СССР. То же стремление к унификации программных средств систем автоматизации, использующих аппаратуру КАМАК, выражено в ряде обзорных работ [1, 2 и др.].

Уже в самых первых системах автоматизации в ИЯФ СО АН СССР, использующих аппаратуру КАМАК, сложилась практика составления управляющих программ на Фортране. Для работы с конкретными модулями КАМАК используются соответствующие подпрограммы, написанные на языке уровня ассемблера или на Фортране. В последнем случае ассемблер используется лишь для базовых универсальных подпрограмм, ориентированных на работу с любым устройством КАМАК [3].

Несмотря на значительные успехи в области разработки языков программирования, связанные с созданием таких мощных современных языков, как Паскаль, Си и др., наиболее распространен при создании систем автоматизации большого и среднего размеров, казалось бы, давно устаревший Фортран. Конечно, небольшой алгоритм размером в несколько десятков или даже сотен строк можно запрограммировать на ассемблере значительно эффективнее, чем на Фортране, однако пользователь,

не знакомый с ассемблером, обычно тратит на его освоение несколько месяцев, и после этого его программы на ассемблере все еще далеки от совершенства. К преимуществам Фортрана перед другими языками следует отнести и высокую надежность существующих на СМ ЭВМ компиляторов, наличие богатых библиотек для научных расчетов, возможность сравнительно легкого переноса программного обеспечения на другие машины, как правило, имеющие компилятор Фортрана в стандартном программном обеспечении.

Какие еще средства программирования доступны для работы с аппаратурой КАМАК? В соответствии с уровнем и степенью специализации эти средства можно представить в виде своеобразного «спектра», в одном конце которого расположены ассемблер MACRO-11 и другие языки уровня ассемблера, а в другом — специализированные языки и системы.

Широко распространен ассемблер высокого уровня PL-11 [4], разработанный в Европейском центре ядерных исследований (ЦЕРН). Наряду с обычными для ассемблера объектами — регистрами, ячейками и другими — этот язык содержит традиционные для структурных языков высокого уровня конструкции FOR, REPEAT, IF — THEN — ELSE и т. п., значительно облегчающие процесс программирования. Как и всякий ассемблер, язык PL-11 позволяет управлять аппаратурой КАМАК, обращаясь непосредственно к регистрам крейт-контроллера. Кроме того, PL-11 обладает специальными языковыми средствами для работы с КАМАК. Однако, стремясь к стандартизации и упрощению программного обеспечения, в ЦЕРНе отказались от применения этих средств и используют стандартный базовый набор библиотеки Фортрана [5]. С точки зрения применимости для создания систем автоматизации PL-11, занимая промежуточное место между ассемблером и языками высокого уровня, на наш взгляд, так же, как ассемблер, не пригоден для массового применения. С другой стороны, PL-11 не дает решающих преимуществ по сравнению с ассемблером при написании небольших подпрограмм.

В работах [6, 7] и ряде других в качестве основного языка программирования систем автоматизации физического эксперимента используется Паскаль. Обсуждение достоинств и недостатков этого языка выходит за рамки настоящей статьи, отсылаем читателя к [8]. Для программирования аппаратуры используется возможность Паскаля, как и PL-11, обращаться к указанным адресам. Как будет показано ниже, исполнение КАМАК-функций при наличии параллельных процессов должно состоять из нескольких неразрывных операций, выполняемых с высоким приоритетом в слове состояния процессора. Эту всегда одну и ту же последовательность операций целесообразно выделить в специальную процедуру. Соображения, связанные со стандартизацией, делают желательным оформление такой процедуры в соответствии с рекомендациями ESONE [9], а соображения эффективности — реализацию его на ассемблере. Так вновь приходим к идее базового пакета подпрограмм для работы с аппаратурой КАМАК. Ситуация с языком Си аналогична: предлагаемые в [10] средства макрогенерации лишь отчасти способствуют решению проблемы.

В конце 70-х — начале 80-х годов было создано множество специализированных языков для программирования КАМАК [11—13] и др. Зачастую усилия, затраченные на создание программного обеспечения такого рода, не были оправданы. Реализация таких языков в виде интерпретатора приводила к низкому быстродействию и невозможности создания программ достаточно большого объема. Кроме того, обычно собственно взаимодействие с аппаратурой составляет небольшую часть таких программ, основной же объем программного обеспечения — это алгоритмы обработки данных. Специализированные языки могут быть полезны только в методических целях, для обучения.

Другой класс специализированных средств программирования аппаратуры КАМАК — системы автоматизации сбора и накопления данных — позволяет предельно эффективно осуществить сбор информации

странена.

Одним из первых предложений по унифицированным системам программирования КАМАК была разработка комитета ESONE — язык промежуточного уровня IML [16]. Эти предложения реализованы в работе [17] в виде библиотеки макроопределений, т. е. пользователю по-прежнему предлагается работать на ассемблере, «усиленном» удобными средствами взаимодействия с аппаратурой. Такой подход, на наш взгляд, проигрывает в мобильности, наглядности и простоте программного обеспечения по сравнению с Фортраном и зачастую не обеспечивает предельной эффективности ассемблера.

Следующие предложения комитета ESONE — ESONE/RTB/01, 02 и 03 [18] связаны с языком программирования Бейсик, дополненным возможностями ввода-вывода в реальном времени. Ни одна из упомянутых выше систем программирования этого класса как в нашей стране, так и за рубежом не соответствует рекомендациям [18].

Наконец, в 1978 г. были изданы рекомендации ESONE/SR/01 [9], стандартизирующие вид подпрограмм, взаимодействующих с аппаратурой КАМАК, для языков высокого уровня, и в частности для Фортрана. В 1980 г. в ИЯФ СО АН СССР написан пакет подпрограмм, соответствующий рекомендациям ESONE/SR/01 и работающий в ЭВМ «Электроника 60» с операционной системой RSX-11S. Впоследствии пакет был поставлен на ЭВМ СМ-3, СМ-4, СМ-1300 и других, совместимых с ними машинах, в операционных системах RSX-11M, RT-11. Модификация исполнительной системы Фортрана позволила работать с пакетом [3] в режиме «stand alone», т. е. без операционной системы.

В 1983 г. в ИРЭ АН СССР выполнена аналогичная работа: реализован пакет подпрограмм, соответствующий ESONE/SR/01 [17]. От разработки ИЯФ СО АН СССР этот пакет отличает в основном трактовка понятия «программный канал», используемого при блочных передачах. Например, после инициации блочного обмена данными подпрограммой CSUBL возврат из нее происходит до завершения передачи, а определить, завершен ли обмен, можно посредством специальных подпрограмм, проверяющих состояние «программного канала».

Рассмотрим упомянутые выше средства программирования аппаратуры КАМАК с точки зрения применимости для создания больших, средних и малых систем автоматизации. Для этого сопоставим типичный объем программ, использующих эти средства, и характерное быстродействие (рис. 1). В [1] приведены аналогичные данные, но в качестве критериев сопоставления взяты величины информационных потоков (количество передаваемых слов данных) и требуемое быстродействие (1/с). Общий характер распределения в [1] соответствует рис. 1.

Конечно, данные, которые приведены на рис. 1, носят оценочный, приблизительный, характер, но они, как и данные [1], позволяют сделать вывод: Фортран, дополненный базовой библиотекой CAMAC [3], находясь в средней части «спектра» средств программирования аппаратуры, наиболее пригоден для массового, неспецифического применения, при создании программного обеспечения больших и средних систем автоматизации.

Рассмотрим подробнее некоторые особенности базового пакета подпрограмм для работы с КАМАК.

При работе с наиболее распространенными типами контроллеров — СС-11, К-16, К0606 и аналогичными — команда КАМАК выполняется в три этапа [19]: 1) занесение в статусный регистр контроллера F-команды КАМАК; 2) собственно обращение к блоку; 3) чтение статусного регистра контроллера сигналов X и Q. Если передается 24-битовое слово

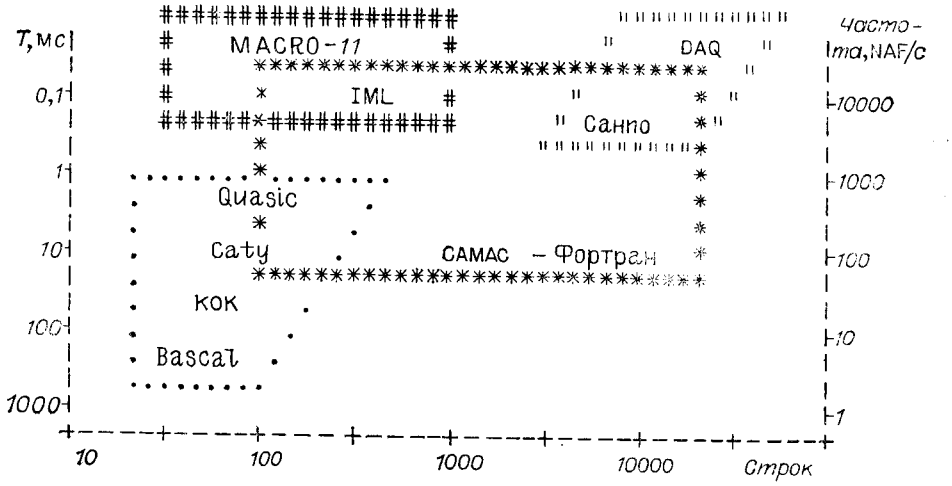

Рис. 1. Типичный объем программ и характерное быстродействие некоторых средств программирования аппаратуры КАМАК

данных, то необходимо еще и чтение регистра старшего байта. Для исключения взаимного влияния параллельных процессов, использующих один и тот же контроллер, необходимо все эти этапы выполнять неразрывно. Поэтому перед занесением F в статусный регистр нужно увеличить приоритет в слове состояния процессора и уменьшить его можно только после считывания X и Q. Это требование возникает не только в многопрограммной операционной системе, но и при наличии любых асинхронных процессов, например, программ обработки прерываний, связанных с запросами L.

Естественно описанные выше этапы оформить в виде отдельной подпрограммы. Так, во многих организациях (например, [20]) появились подпрограммы вида

CALL NAF(K, KAMADR, JF, BUF, IQX).

Здесь входной параметр K — требуемое количество КАМАК-циклов, после завершения подпрограммы K — количество реально выполненных команд КАМАК; KAMADR — так называемый КАМАК-адрес, содержащий информацию о номере крейта, позиции модуля N и субадресе A (в случае контроллеров крейта типа К-16, СС-11 и аналогичных KAMADR — просто значение адреса на общей шине); JF — выполняемая КАМАК-функция; BUF — адрес массива, куда или откуда, в зависимости от типа КАМАК-функции, пересылается информация; IQX определяется значением X и Q в последнем выполненном КАМАК-цикле, если после очередной команды КАМАК $X=0$ или $Q=0$ происходит немедленный выход из подпрограммы с соответствующим значением IQX и K.

Время выполнения подпрограммы определяется очевидной зависимостью

$$T = T_{\text{пп}} + T_{\text{ос}} + \text{K} * DT,$$

где $T_{\text{пп}}$ — накладные расходы, связанные с передачей и анализом параметров на входе и выходе из подпрограммы; $T_{\text{ос}}$ — затраты времени, определяемые используемой операционной системой; K описан выше; DT — время исполнения единичной команды КАМАК. Для микроЭВМ «Электроника 60» $T_{\text{пп}}$ составляет более 100 мкс, $T_{\text{ос}}$ — нулевое, а DT — несколько десятков микросекунд.

Очевидное достоинство такого подхода — предельная простота и довольно высокая эффективность. Вместе с тем для использования LAM-сигналов необходимы какие-то дополнительные средства. Спецификацией

PDP 11/60	RSX-11M	2,2 мс	
VAX-780	VMS	1,8 мс	
HP 21 MX	RTE	1,0 мс	(многопользовательская ОС)
	BCS	0,12 мс	(однопользовательская ОС)
«Nord-100»		0,20 мс	
		0,07 мс	(привилегированная программа)
«Электроника 60»		0,22 мс	(CSSA)
		0,07 мс	(DT для CSUBC)
		0,15 мс	(ожидание L без ОС)
		1,0 мс	(то же с обращением к RSX-11S)

Рис. 2. Время исполнения команд КАМАК для различных ЭВМ и операционных систем. Для Электроники 60» измерения проводились с пакетом [3]

функционально полного набора таких средств является рекомендация комитета ESONE [9].

Реализации подхода, традиционного для операционных систем RSX и RT, когда взаимодействие с каким-либо устройством или группой однотипных устройств управляется специфической частью операционной системы — драйвером, посвящены работы [5, 21]. КАМАК-драйвер, наряду с обычными функциями (выполнить отдельную команду, прочитать или записать целый блок данных), позволяет разделять станцию КАМАК между несколькими конкурирующими программами, поочередно прикрепляющими и освобождающими ее. Однако, обеспечивая типичный для многопользовательских операционных систем сервис, драйвер вместе с тем существенно увеличивает время, требуемое для выполнения отдельной команды КАМАК, причем чем сложнее операционная система, тем больше требуется времени для осуществления многочисленных проверок (рис. 2). Кроме того, для работы с драйвером в рамках языка высокого уровня все равно требуется пакет подпрограмм, написанных на ассемблере.

В [5] дано время исполнения отдельной команды КАМАК (CSSA) для различных процессоров и операционных систем. Приведем эти данные, дополнив их нашими измерениями.

Рис. 2 подтверждает изложенное выше, когда не требуется контроль операционной системы, например, в случае однопользовательской операционной системы, для привилегированной программы и т. п., взаимодействие с аппаратурой происходит значительно быстрее.

В микроЭВМ «Электроника 60», обслуживающей обычно один-два крейта КАМАК одного экспериментатора, обращение к аппаратуре происходит непосредственно через страницу ввода-вывода, минуя операционную систему, и, следовательно, очень быстро. В пакете [3] пользователю предоставлена возможность выбора режима работы при ожидании запросов L: обращение к операционной системе для запуска следующего по приоритету процесса или циклическое ожидание появления запроса в самой подпрограмме пакета.

Если в системе автоматизации используется более мощная ЭВМ — СМ-4 или аналогичная, взаимодействие с аппаратурой также осуществляется непосредственно через страницу ввода-вывода, для доступа к которой в ОС RSX-11M обычно выделяется специальный раздел IOPAGE. При этом ошибки в программе пользователя могут привести к порче содержимого регистров внешних устройств и ненадежной работе системы в целом. В достаточно сложных системах автоматизации поэтому целесообразно функции взаимодействия с аппаратурой возложить на специально выделенную микроЭВМ, подключенную к СМ-4 линией связи. Это не только защитит систему от ошибок в программах, но и обеспечит высокую эффективность, хотя и усложнит программное обеспечение.

Отметим некоторые дополнительные возможности, возникающие при работе с пакетом [3].

Использование одного и того же пакета большим количеством пользователей выгодно не только потому, что позволяет легко осуществить

обмен созданным ими программным обеспечением. Не менее важен и тот факт, что смена типа контроллера не требует почти никаких изменений в программах. Так, в ИЯФ СО АН СССР практически одновременно сосуществовали контроллеры М-400, К16, К0606 и многие программы перенастраивались на конкретный тип контроллера прямо во время исполнения. Здесь можно говорить о том, что пакет «экранирует» от программиста сложность взаимодействия с контроллером крейта.

Даже в небольших системах автоматизации часто требуется в процессе эксплуатации или отладки отключать или менять КАМАК-адреса отдельных блоков. Предлагаемые в ряде работ [6, 15 и др.] средства генерации программного обеспечения призваны решить эту задачу, однако специфический язык, отсутствие необходимой документации привели КАМАК-адреса будут использоваться несколькими программами, то создается специальный файл, содержащий значения КАМАК-адресов, и ряд программ, считывая этот файл, получает необходимую информацию. Кроме того, пакет [3] предусматривает возможность программного отключения модулей КАМАК: в том случае, когда КАМАК-адрес положителен, т. е. знаковый бит нулевой, все подпрограммы пакета возвращают управление в вызвавшую программу, не производя при этом никаких обращений к аппаратуре. Это позволяет, не прерывая хода эксперимента, ничего не изменяя в текстах программ, динамически переконфигурировать аппаратуру, заменять или отключать отдельные модули КАМАК.

Обычно программное обеспечение систем автоматизации отлаживают в той конфигурации, для которой оно предназначено, «настройка» такого рода наиболее надежна. Однако иногда приходится отлаживать программы, взаимодействующие с каким-либо блоком КАМАК в тот момент, когда он по каким-то причинам отсутствует. Более того, инструментальная ЭВМ может вообще не иметь ни одного крейта КАМАК. В этой ситуации легко подменить базовые подпрограммы пакета на «заглушки», написанные на Фортране. Такая методика вместе с описанным выше программным «отключением» отдельных блоков помогает в отладке основного алгоритма.

В заключение повторим наш вывод о практической применимости языка программирования Фортран и библиотеки базовых подпрограмм — реализации рекомендаций комитета ESONE — для решения широкого спектра задач автоматизации научных исследований.

ЛИТЕРАТУРА

1. **Велихов Е. П., Выставкин А. Н.** Проблемы развития работ по автоматизации научных исследований // Упр. сист. и маш.— 1984.— № 4.
2. **Выставкин А. Н., Олейников А. Я., Панкрац Е. В.** Принципы унификации средств программирования КАМАК-систем.— М., 1985. (Препринт/АН СССР, ИРЭ; 3 (421)).
3. **Вьюшин О. В., Храпкин П. Л.** Пакет стандартных подпрограмм для работы с аппаратурой КАМАК // Автометрия.— 1983.— № 4.
4. **Russel P. D.** PL-11: a Programming Language for the PDP-11 Computer with Addendum.— Geneive: CERN, 1978.
5. **Standart** CAMAC Subroutines for on-Line Computers at CERN // IEEE Trans. on Nucl. Sci.— 1985.— V. NS-28.— N 5.
6. **Балука Г., Саламатин И. М.** Динамически формируемая программная система автоматизации экспериментов // Труды XIX Всесоюз. школы по автоматизации научных исследований.— Новосибирск: ИАиЭ СО АН СССР, 1985.
7. **Балагуров А. М., Миронова Г. М., Островной А. И.** Программное обеспечение системы накопления информации дифрактометра ДН-2 на импульсном реакторе ИБР-2.— Дубна, 1984. (Препринт/ОИЯИ; Р10-84-440).
8. **Hertrich F.** FORTRAN can beat PASCAL for Control Applications // Electronic Design.— 1985.— P. 135—141.

9. **Subroutines** for CAMAC ESONE/SR/01: ESONE Comitee, 1978.
10. **Будячевский И. А.** Расширение возможностей программирования на языке С // Труды XIX Всесоюз. школы по автоматизации научных исследований.— Новосибирск: ИАиЭ СО АН СССР, 1985.
11. **Виноградов В. И., Росляков А. Д.** Язык реального времени BASCAL для программирования систем КАМАК // Управ. сист. и маш.— 1978.— № 6.
12. **Бредихин С. В., Песляк П. М.** CATY-M: система для программирования аппаратуры КАМАК. Модифицированный вариант.— Новосибирск, 1981. (Препринт/АН СССР, Сиб. отд-ние, ИАиЭ; № 166).
13. **Подольский Л. В.** Система QUASIC для программирования на мини-ЭВМ.— Пущино, 1980. (Препринт/АН СССР, НЦБИ; № 4).
14. **Hardekopf R. A., Poore R. V., Sunier J. M.** Event analysis language for high-speed data acquisition // IEEE Trans. on Nucl. Sci.— 1981.— V. NS-28, N 5.— P. 3853.
15. **Яновский Г. Я.** Модульная инструментальная система программирования экспериментов САНПО-3.— Новосибирск, 1985. (Препринт/АН СССР, Сиб. отд-ние, ИАиЭ; № 267).
16. **The Definition** of IML, a Language for Use in CAMAC Systems. ESONE/IML/01: ESONE Secretariat, 1974.
17. **Выставкин Е. Н., Олейников А. Я., Панкрац Е. В.** Реализация унифицированных средств программирования КАМАК-систем на ЭВМ серии СМ-4 («Электроника 60»).— М., 1985.— (Препринт/АН СССР, ИРЭ; № 4).
18. **Real-Time** BASIC for CAMAC ESONE/RTB/02: ESONE Comitee, 1976.
19. **CC11 CAMAC** Crate-PDP-11 Interface Type 116: CERN — NP CAMAC, Note 43—00, 1972.
20. **Ступин Ю. В.** Методы автоматизации физических экспериментов и установок на основе ЭВМ.— М.: Энергоатомиздат, 1983.
21. **Кузьменко В. Г., Осипов В. А., Щелканов Ю. В.** Программное обеспечение для работы с аппаратурой СУММА в операционных системах RSX-11M/S.— Серпухов, 1984. (Препринт/ИФВЭ; № 84-161).

Поступила в редакцию 16 декабря 1985 г.

УДК 621.317.791

Э.-А. К. БАГДАНСКИС, В. Б. КВЯДАРАС
(Вильнюс)

ИЗМЕРЕНИЕ ДИНАМИЧЕСКИХ ПАРАМЕТРОВ ЦАП

Одной из наиболее сложных задач метрологического обеспечения производства сверхбыстродействующих интегральных ЦАП является измерение их динамических параметров. Это связано с регистрацией коротких интервалов времени (порядка единиц или десятков наносекунд) при низких уровнях отсчета (порядка единиц милливольт для 6—8-разрядных ЦАП). Классические методы измерения временных интервалов наносекундной длительности [1] из-за сравнительно больших погрешностей измерения не применимы. Поэтому для решения задачи в данном случае предложено сочетать точное преобразование временного масштаба исследуемого сигнала [2—4] с новым способом измерения [5, 6]. Однако анализ погрешностей измерения таких измерителей не производился.

В измерителе, структурная схема которого показана на рис. 1, формирователь импульсов начинает работать от выходного сигнала первого кварцевого генератора КГ1 с частотой повторения F_c, а сигналом второго кварцевого генератора КГ2 запускается формирователь строб-импульсов с частотой повторения $F_c + \Delta F$, причем $F_c \gg \Delta F$.

В связи с тем, что частота следования строб-импульсов выбрана больше частоты повторения исследуемых сигналов, преобразование временного масштаба последних происходит по обратной шкале времени. Это позволяет существенно упростить схему обработки измерительной информации, так как время установления сигнала будет измеряться от момента первого выхода мгновенного значения измеряемого сигнала $U_{ЦАП}$ из зоны, образованной уровнями отсчета, до момента времени, когда