

Таков перечень модулей, разработанных, изготовленных и смонтированных в систему. Благодаря широко внедренной унификации, значительно облегчающей проектирование, в течение 1972—1973 гг. планируется расширение серии до 30—40 наименований.

Поступила в редакцию 31 октября 1972 г.

УДК 681.325.021 : 0.07

**О. З. ГУСЕВ, Ю. Н. ЗОЛУХИН, З. А. ЛИВШИЦ,
Ю. К. ПОСТОЕНКО, В. С. ЯКУШЕВ**
(Новосибирск)

СПЕЦИФИКА УПРАВЛЕНИЯ В САМАС

1. Стандарт САМАС [1, 2] в той его части, которая регламентирует функциональные возможности систем, является примером применения концепций, возникших в вычислительной технике, к построению измерительных систем. Имеется глубокая аналогия между программой, реализуемой вычислительной машиной, и алгоритмом функционирования системы, выполненной в САМАС; основное отличие заключается в наборах элементарных операций (базисе).

Нужно отметить, что указанные разделы стандарта оставляют разработчику значительную свободу действий. Это, разумеется, удобно; однако его положение из-за отмеченного обстоятельства вполне сравнимо с ситуацией, в которой находится программист, пользующийся ЭВМ без математического обеспечения.

В то же время очевидно, что можно выделить ряд процедур, входящих практически в любой алгоритм управления (по крайней мере, для широкого класса систем), и стандартизовать эти процедуры. Такая мера (аналогичная созданию библиотеки стандартных программ для ЭВМ) существенно упрощает проектирование, повышает его качество и благотворно влияет на системное программное обеспечение.

Следует, видимо, подчеркнуть отношение авторов к этой проблеме: мы не считаем, что стандартизация процедур должна войти в САМАС, но, по нашему мнению, внутреннее «укрупнение» стандарта полезно любой группе разработчиков. Такая работа, в частности, проводится в ИАЭ и СКБ НИИ СО АН СССР.

Некоторые результаты этой деятельности отражены в данной статье, которая в определенном смысле является комментарием к части стандарта САМАС, относящейся к управлению. Для удобства читателя в тексте приводятся необходимые для понимания изложения сведения из [1, 2].

2. Основными функциями управляющего системой алгоритма являются:

занесение начальных условий, определяющих совокупность параметров и режимов работы системы: временное и амплитудное разрешение, другие метрологические характеристики;

исходные коммутации: распределение потоков заявок (маскирование, задание приоритета);

предварительное распределение потоков данных (подготовка массивов для ЭВМ, БЗУ, МОЗУ, НМЛ, регистраторов);
установление порядка обмена и его осуществление (между модулями, каркасами, вертикалью и ЭВМ);
оперативное управление параметрами: изменение частот опроса, количества запросов, пропускаемых в систему;
перераспределение потоков данных (например, при перегрузке канала или БЗУ);
упорядочение массива в ОЗУ ЭВМ при наличии канала прямого доступа.

3. Алгоритмы управления могут быть разделены на три основные группы по характеру используемой для управления информации о системе:

жесткое управление, не зависящее от состояния системы; например, совокупность действий, приводящих систему в исходное состояние или определяющих текущее состояние (аналог — программа без условных переходов);

жесткое управление с привлечением информации о состоянии системы по инициативе управления (аналог — программа, содержащая условные переходы);

гибкое управление, включающее совокупность реакций на запросы периферии и изменение управления по инициативе периферии (аналог — программа, предусматривающая прерывание по запросам от активных внешних устройств и обработку этих запросов); методы программирования и программируемые средства, применяемые в этих трех способах, различны.

Процедуры организации условных переходов и обработки запросов в алгоритме управления являются, на наш взгляд, удобным и важным объектом «внутренней» стандартизации (несколько ниже мы опишем «соглашения», принятые на этот счет в ИАЭ и СКБ НИИ СО АН СССР).

4. Построение системы на принципах программного управления обеспечивает широкие возможности реализации алгоритмов управления на всех уровнях иерархии: ЭВМ — вертикаль — каркас — модуль — датчик (исполнительное устройство).

Обсудим возможности «базиса», предоставляемого стандартом САМАС, для реализации алгоритмов управления, в том числе некоторые нестандартные (но «незапрещенные») приемы.

Основной одноадресный формат команд содержит трехступенчатый адрес CNA и код операции F . Структура адреса соответствует иерархическому принципу построения системы: C — адрес каркаса; N — номер модуля; A — адрес устройства в модуле (субадрес). В простых системах могут использоваться укороченные форматы адресов. Табл. 1 и 2 содержат перечни операций и стандартных адресов CNA .

Прямое программное управление является наиболее простым методом, предполагающим исполнение команды в адресуемом устройстве по сигналам на шинах F . Прямое управление обеспечивается минимальными аппаратными затратами в модуле, в основном на дешифраторы адресов и операций. Причем полная дешифрация необязательна, так как число управляющих функций модуля, как правило, невелико, а кодирование AF избыточно.

В то же время в некоторых случаях перечень AF оказывается недостаточным в адресной части или по смыслу разрешенных стандартных и нестандартных операций, в том числе использующих R - или W -шины. Например, для управления модулем быстрого ОЗУ емкостью более 32 ячеек необходимо расширять адресную часть команды. Расширение функций или адресации прямого управления возможно в первую очередь за счет использования модификаторов функции и адресов. При

Таблица 1

Перечень операций САМАС

Группа операций		Действие	Ответ	Использование R-и W-шин
Чтение	0	Чтение регистра I группы	$Q=1$, если имеется регистр по $A(j)$	Используют R-шины
	1	Чтение регистра II группы		
	2	Чтение с гашением регистра I группы		
	3	Чтение дополнительного кода регистра I группы		
	4,6 5,7	Нестандартное Резервное		
Управление	8	Проверка запроса	$Q=1$, если есть запрос по $A(j)$	Не используют R- и W-шины
	9	Гашение регистра I группы		
	10	Гашение запроса		
	11	Гашение регистра II группы		
	12,14 13,15	Нестандартное Резервное		
Занесение	16	Перезапись регистра I группы	$Q=1$, если имеется регистр по $A(j)$	Используют W-шины
	17	Перезапись регистра II группы		
	18	Селективная перезапись регистра I группы		
	19	Селективная перезапись регистра II группы		
	20,22 21,23	Нестандартное Резервное		
Управление	24	Запрет	Q равно признаку по $A(j)$	Не используют R- и W-шины
	25	Приращение предварительно выбранных регистров		
	26	Разрешение		
	27	Проверка состояния		
	28,30	Нестандартное		
	29,31	Резервное		

прямом управлении возможны два способа модификации AF : совместное использование AF , т. е. модификация F при помощи части A или A при помощи части F ; размещение модификаторов в группе W-шин.

В частности, в качестве модификатора адресов в командах чтения — запись используется шина $F1$, позволяющая удвоить число адре-

Таблица 2

Некоторые стандартные и рекомендуемые адреса САМАС

Адрес			Адресуемые устройства	Операция	Примечание	
C	N	A				
0	×	×	Каркас управления вертикалью	×	} Стандарт 4600	
×	0	×				Резерв
1÷7	1÷23	0÷15	Устройства в модулях общего назначения	×		
1÷7	24	—	Предварительно выбранные модули	×		
1÷7	26	—	Все модули каркаса	×		
1÷7	28,30	—	Модуль управления каркасом	×		
×	25,27	×	Резерв	×		
×	29,31	×	Резерв	×		
×	×	8	Регистр характеристики модуля	1, 11, 17, 19		} Рекомендации по комиссии по математическому обеспечению САМАС
×	×	9÷14	Регистры управления и состояния			
×	×	15	Регистр запросов модуля			

сов на чтение — запись за счет разделения совокупности устройств на I и II группы.

Аналогично можно использовать $F2$ и $F4$, если шин $F8$ и $F16$ достаточно для разделения в модуле команд чтения — записи. Нестандартные операции могут модифицироваться субадресами, так как они легко выделяются значениями $F4=1$ и $F8=1$. Однако здесь фантазия разработчиков должна быть ограничена требованием использовать субадреса по прямому назначению, т. е. для адресации устройств, имеющих самостоятельное функциональное назначение.

Модификация команд при помощи W -шин возможна, так как основной формат прямого управления САМАС использует принцип непосредственной адресации данных, т. е. сигналы на W -, A - и F -шинах могут присутствовать одновременно. При этом кажется целесообразным занимать под модификатор старшие 8 разрядов W -шин, обычно не используемых для передачи данных. Запоминания модификатора в модуле при прямом управлении не требуется. В качестве команд, модифицированных по W -шинам, могут быть использованы $AF16$, $AF19$, $AF20$, $AF22$.

Косвенное программное управление предполагает модификацию операций и адресов содержимым специальных регистров,

Таблица 3

Команды коллективного управления

Команда		Адресуемые устройства	Способ адресации
обозначение	действие		
BZ	Установка в исходное состояние	Все модули вертикали, включая контроллеры	Шина $BZ \rightarrow$ шина Z
Z	То же	Все модули каркаса	$B\{C(i)N(28)A(8)F(26)\} \rightarrow$ \rightarrow шина Z
C	Гашение (очистить)	» » »	$B\{C(i)N(28)A(9)F(26)\} \rightarrow$ \rightarrow шина C
I	Запрет	Все модули каркаса, кроме контроллера	Установка $I=B\{C(i)N(30)A(9)F(26)\} \rightarrow$ \rightarrow шина I Сброс $I=B\{C(i)N(30)A(9)F(24)\} \rightarrow$ \rightarrow шина I
BG	Чтение строки требований	Модуль классификатора запросов	Шина BG

размещаемых в модулях по субадресам II группы $A(9 \div 14)$. Данные в эти регистры должны быть предварительно занесены из контроллера.

В число специальных регистров, модифицирующих основные команды, входят регистры предварительно выбранных адресов, применение которых позволяет, в частности, осуществить многоадресное обращение к модулям, и регистры управления, в том числе регистры специфических операций и регистры коммутации структуры модуля. Как правило, в таких регистрах должно использоваться позиционное размещение управляющих битов, чтобы исключить дешифрацию кодов.

Помимо косвенных методов программного многоадресного управления, в набор команд САМАС входят также команды коллективного управления с неполной адресацией и безадресные команды. В табл. 3 сведены команды коллективного управления устройствами системы. Эти команды позволяют уменьшить время выполнения процедуры приведения системы в нужное состояние, блокировки отдельных каркасов и чтения упорядоченных запросов каркаса.

Косвенные методы управления позволяют реализовать объединение R - и W -шин для двухадресных передач модификацией нестандартных

команд при помощи содержимого управляющего регистра или использования предварительно заносимых адресов и нестандартных команд. Содержит 1 бит информации о состоянии модуля. Типичны следующие применения:

для $Q=0$

«выключено»;

«ошибка управления»* («чужая» команда) — код операции и субадрес NAF не соответствует ни одному из используемых в модуле;

«ошибка в данных» — принятые в модуль данные содержат ошибки (например, по четности);

«занято» — команда не может быть выполнена, так как модуль занят выполнением предыдущей операции или переполнен (например, ЗУ);

«уточнить состояние» — ответ, требующий некоторых действий управления по уточнению состояния;

для $Q=1$

«хранит запрос» — устройство по субадресу $NA(i)$ в состоянии запроса [операция $F(8)$];

«наличие признака» — устройство по субадресу $NA(i)$ в состоянии 1 [операция $F(27)$];

«нормальное состояние» — стандартный ответ, предполагающий продолжение основной программы управления.

Часть указанных функций Q может объединяться; в этом случае при отсутствии однозначной интерпретации ответов на каждый NAF приходится использовать операцию $F(27)$ для проверки признаков по субадресам или в качестве альтернативного варианта вводить в модуль регистр состояний, программно доступный по команде чтения и одному из субадресов из II группы.

Регистр состояний содержит статусное слово, в котором указанные выше совместимые и несовместимые признаки могут быть размещены позиционно. По существу, чтение статусного слова — параллельная проверка состояния устройства, в отличие от последовательной — при помощи $F(27)$ и субадресов. Аппаратурные затраты при реализации этих способов почти одинаковы, однако при использовании регистра состояний может быть достигнута большая оперативность.

Процедура проверки ответов и (в некоторых случаях) состояний может следовать за каждой командой. При этом команда должна сопровождаться тремя признаками: «требуется ли проверка $Q?$ », «нормальное значение $Q(q)$ », «требуется ли уточнение состояния при $Q \neq q?$ ». Признаки могут быть описаны в программе управления при автономной проверке ответов. На рис. 1 приведен алгоритм проверки ответа на очередную команду (NAF).

Легко видеть, что реализация i -го блока проверки Q может быть громоздкой как в программе ЭВМ, так и при аппаратурной реализации, если не достичь некоторых соглашений о стандартном использовании Q и структуре статусных слов, модуля, каркаса и т. д. Можно предложить

* По рекомендации комиссии по математическому обеспечению САМАС в мае 1971 г. эта функция возложена на шину X , разрешенную к применению вместе с шиной Q .

следующие ограничения: проверка Q обязательна при поступлении любой команды, нормальный ответ: $Q=1$; модуль обязательно содержит регистр состояний, программно доступный по $A(11) F(1)$; при $Q=0$ читается содержимое регистра состояний модуля, к которому была обращена команда; часть стандартных состояний жестко закреплена по разрядам (см., например, табл. 4).

Неиспользуемые разряды SW находятся в состоянии 0. Указанные ограничения позволяют сократить объем математического обеспечения в ЭВМ или существенно упростить аппаратную часть за счет стандартизации реакций на изменения в системе.

7. Стандартом САМАС предусмотрена система активных запросов (L -сигналов), посредством которых устройства, входящие в систему,

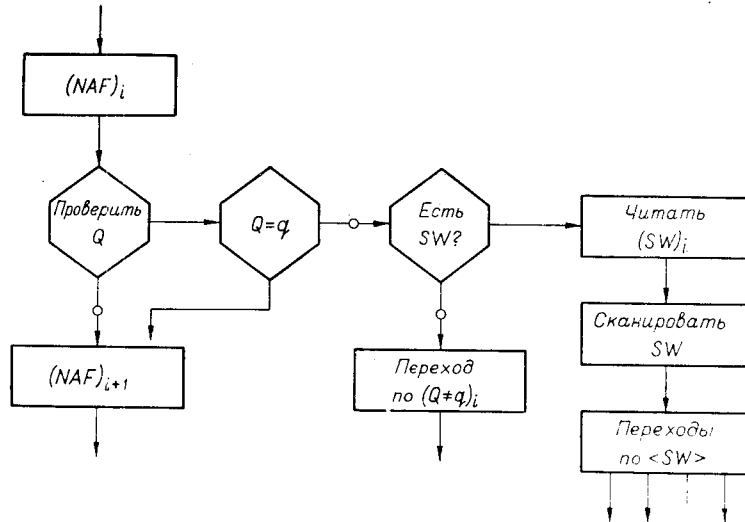


Рис. 1. Алгоритм проверки ответов:
 q — нормальное значение Q ; SW — слово состояния.

сообщают управлению (по собственной инициативе) сведения о своем состоянии. Важно отметить, что роль L -сигналов не ограничивается уведомлением управления об эффектах, возникающих в результате неисправностей или каких-либо внешних причин и приводящих к нарушению нормального функционирования системы. Наиболее существенной является возможность с помощью запросов оперативно вводить в управляющие блоки (контроллеры или ЭВМ) полезную для управления

Таблица 4

Номер разряда	Наименование признака
1	Заблокирован
2	Занят
3	Хранит запрос
4	Ошибка (в данных, в управлении)
5	Переполнение
6	Требуется вмешательство оператора
7	Нестандартное использование
8	Длинное SW : в 9-м — 24-м разрядах или по субадресу $A(12)$

информацию; это во многом избавляет основную программу от необходимости извлекать эту информацию и значительно увеличивает гибкость алгоритма. Указанная интерпретация системы запросов позволяет использовать L -сигналы как важный рабочий инструмент при синтезе алгоритмов. (Ясно, что эффективное применение запросов для нужд управления требует достаточно развитой аппаратуры прерываний.)

Ниже приведен перечень некоторых запросов, типичных для систем сбора и обработки данных (одни из них носят характер «тревог», другие способствуют повышению «расторопности» программы):

«ошибка» — появляется при обнаружении ошибки в данных, поступающих извне;

«переполнение» — возникает при переполнении счетчиков, сумматоров, при выходе аналоговых сигналов за шкалу или уставку, переполнении буферных ЗУ или нарушении границ защиты их адресов и т. д.;

«внимание» — предшествует появлению каких-либо сигналов или данных (например, в синхронном режиме), может переводить систему в режим ожидания какого-либо запроса или состояния;

«синхронизация» — запрос от таймеров, синхронизирующих систему, или требование данных сверху, поступающих в синхронном по отношению к периферийному устройству режиме;

«есть данные» — возникает при появлении извне данных, которые должны быть приняты системой и могут быть искажены последующими;

«готов к приему данных» — сообщение о готовности к получению некоторой информации, в скорейшей обработке которой система «заинтересована»;

«конец работы (готов)» — сигнал об окончании работы внешнего устройства по командам, заданным ранее.

В вопросе о количестве запросов в модуле разумной представляется следующая концепция: разработчик не должен стремиться к минимизации числа допустимых запросов с тем, чтобы любая полезная для управления информация из модуля была программно доступной через систему запросов; в то же время должна быть заложена возмож-

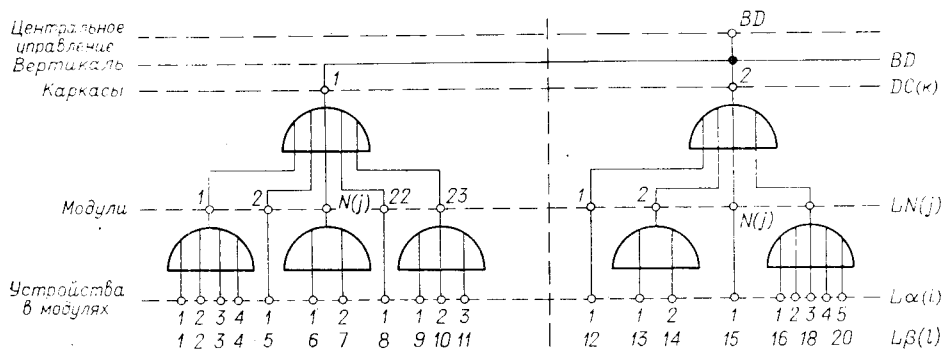


Рис. 2. Иерархия запросов САМАС.

ность подавления и блокирования маловажных для управляющей программы запросов (иначе будет существенно осложнена проблема идентификации запросов, к обсуждению которой мы сейчас переходим).

8. Система обработки запросов, включая аппаратную и программную части, должна обеспечивать выполнение трех основных процедур, предшествующих подпрограмме реакции на запрос: выбор приоритетного запроса, поиск источника запроса, идентификация запроса.

Процедуры эти могут быть частично или полностью совмещены, вплоть до перехода к подпрограмме обслуживания запроса путем прямой или косвенной адресации ее начальной ячейки.

Назначение процедуры выбора приоритетного запроса — выделить из множества запросов единственный (или группу запросов, имеющих единый смысл), подлежащий дальнейшей обработке. Остальные запросы теряются или ожидают своей очереди. Приоритет может быть как установлен после поиска всех источников запросов и анализа сложившейся ситуации, так и задан порядком поиска: от более приоритетного к менее приоритетному. Заметим, что в настоящее время мы не видим ни методов, ни приемлемых аппаратных или программных

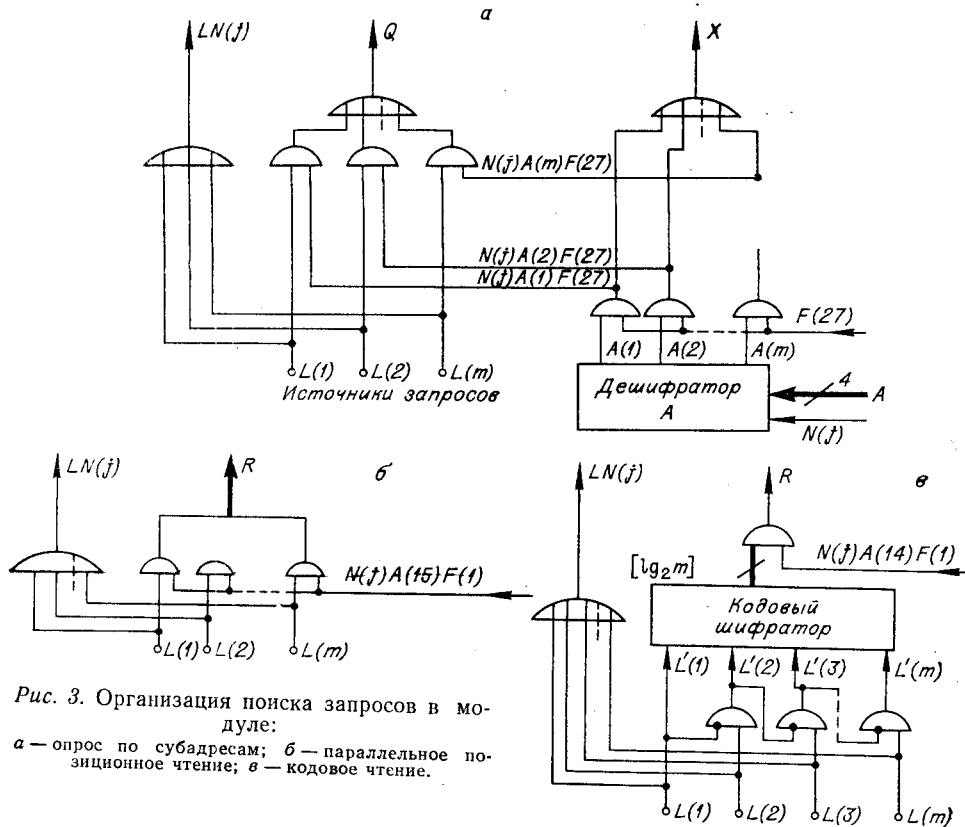


Рис. 3. Организация поиска запросов в модуле:
 а — опрос по субадресам; б — параллельное позиционное чтение; в — кодовое чтение.

средств, позволяющих оперативно реализовать анализ сколько-нибудь сложной ситуации с запросами и перераспределить приоритет, поэтому в дальнейшем изложении предполагаем, что вопрос о распределении приоритетов решается жесткими, как правило, аппаратными методами. Пользователю системы может быть предоставлена возможность изредка менять приоритеты (например, перепайкой или набором штекерных соединений).

Необходимость процедуры поиска запроса — следствие неоднозначности и неопределенности коллективных запросов верхних ступеней иерархии запросов. На рис. 2 изображена иерархическая структура коллективных запросов САМАС. $L_{\alpha}(i)$ — запросы нижнего уровня (от устройств в модуле); $LN(j)$ — запросы модулей; $Dc(k)$ — запрос каркаса; BD — запрос вертикали; $C(k)N(j)\alpha(i)$ — естественный (иерархический) адрес запросов; $\beta(b)$ — порядковый номер запроса. Рисунок иллюстрирует избыточность естественного адреса и независимость порядкового номера от естественного. Фактическое назначение процедуры поиска — выделить адреса запросов CN_{α} при наличии одного из коллективных запросов. Возможности реализации тех или иных методов поиска определяются организацией аппаратной части системы запросов на всех ступенях.

САМАС предоставляет три возможности организации поиска запроса нижней ступени: проверку запроса по субадресам (сканирование по субадресам), параллельное позиционное чтение запросов, параллельное кодовое чтение запросов.

На рис. 3 приведены структурные схемы, соответствующие этим вариантам. Легко видеть, что аппаратные затраты варианта б — на-

меньшие, вариантов a и b — соизмеримы. Отметим также, что вариант a должен совмещать поиск с процедурой приоритета, как указано выше, путем задания последовательности поиска; вариант b реализует схемный приоритет при помощи запрета запросов старших номеров младшими; вариант b предполагает организацию приоритета на одной из верхних ступеней, т. е. допускает некоторую гибкость в организации всей структуры в отличие от a .

Рассмотренные варианты организации поиска запросов нижнего уровня применимы на всех верхних ступенях независимо и в различных комбинациях. В качестве примера приведем два алгоритма поиска запросов: рис. 4 — последовательный алгоритм поиска; рис. 5 — параллельный кодový. В алгоритмах использованы стандартные адреса табл. 2. Операция формирования команды сводится к замене части кодового слова команды, т. е. «сшиванию» кодовых слоев, или приращению адреса. Отметим, что простое сшивание кодов при чтении очередного возможно лишь при выполнении следующих условий: полного единообразия размещения запросов в разрядах слова; унификации субадресов кодовых или позиционных регистров; однозначного соответствия

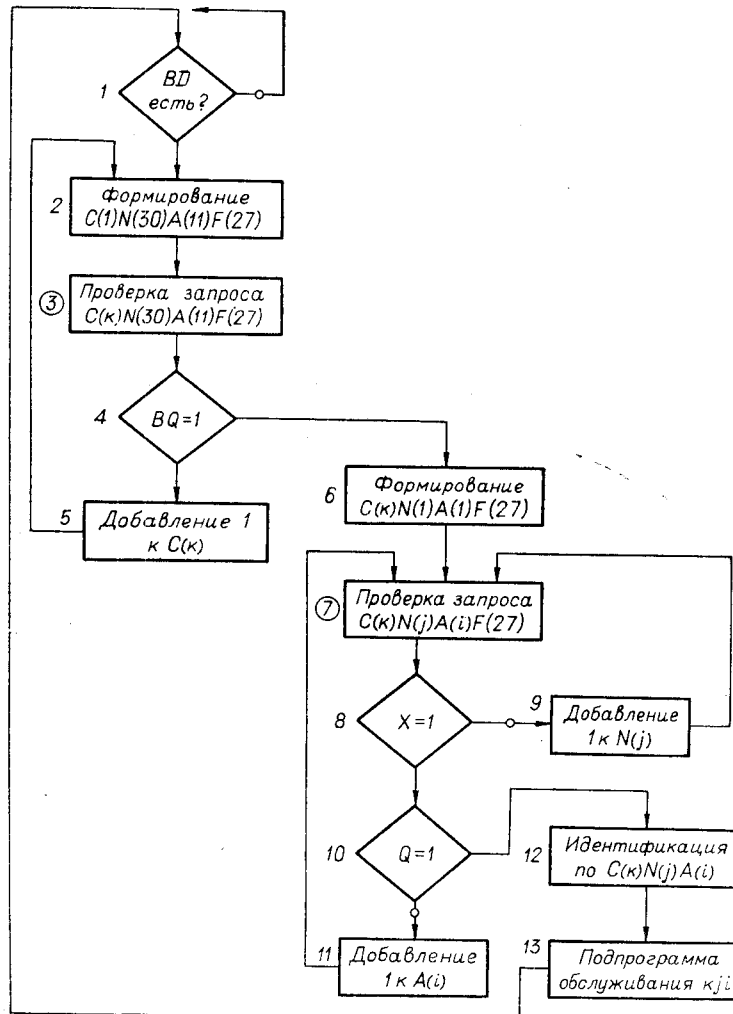


Рис. 4. Алгоритм последовательного поиска запроса.

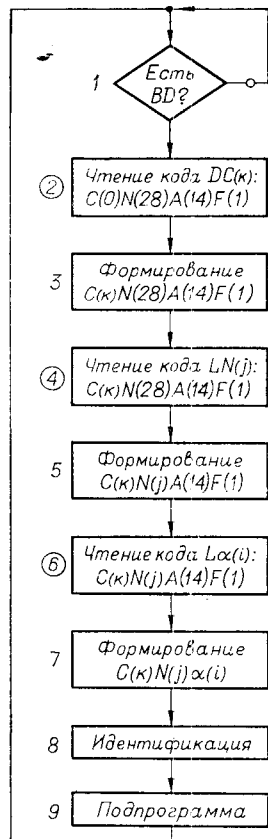


Рис. 5. Алгоритм кодового поиска запроса.

предпочтительно позиционное представление запросов. На рис. 6 приведена структура модуля, специализированного для поиска позиционных запросов. Алгоритм работы схемы идентичен приведенному на рис. 5, за исключением адресов позиционных регистров в контроллере каркаса и модулях. Отметим, что централизация поиска в контроллере вертикали не всегда целесообразна, особенно в тех случаях, когда каркасы имеют автономное управление. При этом часть запросов обрабатывается специализированным контроллером каркаса. Там же следует организовывать поиск всех запросов до уровня *NA*. Подготовленный контроллером приоритетный адрес может поступать в контроллер вертикали за один цикл САМАС вместе с кодовым номером каркаса. Подобная организация системы запросов несколько увеличивает объем оборудования в сравнении, например, со структурой, изображенной на рис. 3, однако увеличивает оперативность системы запросов, избавляя контроллер вертикали или процессор от излишних операций.

Процедура поиска заканчивается определением «точного» адреса источника запроса. Задача идентификации — выяснение адреса (прямого или косвенного) подпрограммы реакции на этот запрос. Использование для ее решения результатов, полученных процедурой поиска (т. е., например, *CNA*), связано с рядом трудностей. Разъясним их вкратце на примере случая, когда *CNA* программно преобразуется в ЭВМ в адрес подпрограммы реакции. Обычно количество модулей в различных каркасах, входящих в систему, неодинаково; различно и количество запросов от модулей. Поэтому адрес подпрограммы реакции

адресации регистров и нумерации запросов. Нарушение хотя бы одного из указанных условий делает невозможным реализацию обоих алгоритмов, так как на каждой операции алгоритма потребуется использование таблиц соответствия (номеров запросов — разрядам; разрядов — субадресам и субадресов — номерам); в то же время их принятие не влечет никаких дополнительных аппаратных затрат. Приведенные соображения еще раз подчеркивают полезность дополнительных «внутренних» регламентов, систематизирующих аппаратные реализации применительно к используемым методам управления.

Отметим некоторые особенности алгоритмов. В алгоритме рис. 4 команды 3 и 7, а в алгоритме рис. 5 команды 2, 4, 6 выполняются в цикле САМАС, т. е. занимают линию связи, остальные операции — внутренние операции процессора или автономного устройства управления. Время последовательного поиска зависит как от числа источников запросов имеющихся в системе, так и от номера активного приоритетного запроса и при значительном числе запросов может быть чрезмерно большим. При параллельном поиске, кодовом или позиционном с аппаратурной организацией приоритета, время поиска фиксированно (в приведенном примере три цикла САМАС). Аппаратурные затраты при полностью кодовом поиске, включая нижний уровень системы, могут быть велики, однако при централизованной организации приоритета и кодирования аппаратурные затраты меньше, чем в любых других методах. При этом на нижнем уровне

не удается вычислить; его приходится определять путем последовательного просмотра таблиц соответствия между «системными» и «машинными» адресами запросов, а это связано со значительными (по сравнению с системным тактом) затратами машинного времени и, следовательно, потерей оперативности. Вообще говоря, можно было бы вычислять по CNA адрес подпрограммы даже при неравномерности распределения запросов (ориентируясь на максимально допустимое число модулей в каркасе и максимально допустимое количество запросов в модуле). Это, однако, привело бы к существенному разрастанию объема таблиц соответствия, что, видимо, нежелательно, если учесть, что в качестве управляющих, как правило, используются мини-ЭВМ.

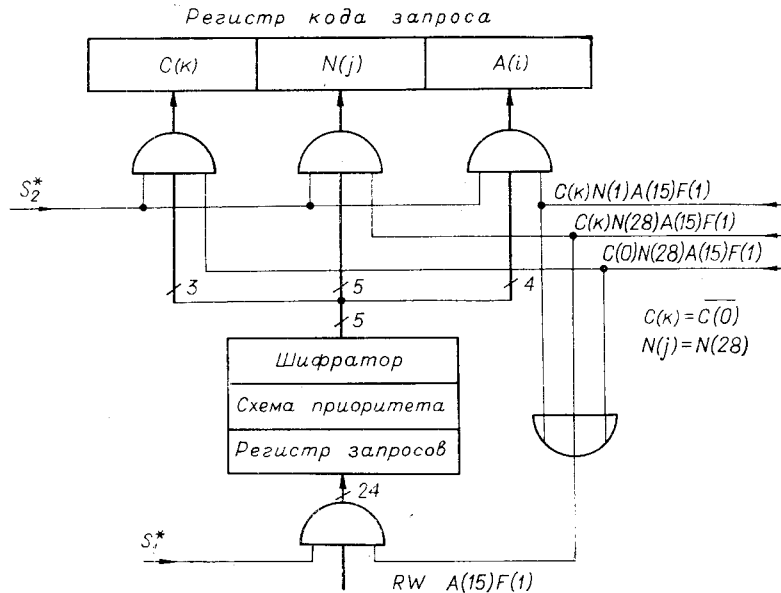


Рис. 6. Структура схемы параллельного поиска запросов в контроллере вертикали.

Таким образом, совместить процедуры поиска и идентификации запроса возможно, например, используя специальные регистры в модуле (или каркасе) для хранения базовых адресов подпрограмм. После окончания процедуры поиска (определения номеров каркаса, модуля и источника запроса) считывается базовый адрес модуля (каркаса) и модифицируется затем номером источника запроса. Модифицированный адрес может представлять собой косвенный адрес начальной ячейки требуемой подпрограммы реакции на запрос.

Итак, введение некоторых дополнительных аппаратных затрат позволяет упростить и ускорить процедуру идентификации.

Изложенные соображения по поводу специфики управления в стандарте САМАС положены в основу «типовой» схемы управления модулем, принятой при разработке систем сбора данных в СКБ НИИ и ИАЭ СО АН СССР. Перечень стандартных устройств и регистров модуля с субадресами и операциями приведен в табл. 5.

9. В заключение обсудим некоторые вопросы организации управления в рамках предложенной схемы.

1. Триггер запрета — разрешения коллективного L -сигнала модуля является одновременно первым триггером статусного регистра — «заблокировки»; обязателен в модуле, вырабатывающем хотя бы один запрос, разрешает поступление L -сигнала на соответствующую шину.

Таблица 5

Устройство	Действие	Команда			Использование шин			
		адресуемая		неадресуемая C или Z	W	R	S1	S2
		A	F					
Триггер запрета — разрешения коллективного запроса модуля	Запрет <i>L</i>	0	24	—	—	—	—	+
	Разрешение <i>L</i>	—	—	<i>C</i>	—	—	+	—
		0	26	—	—	—	—	+
Регистр характеристики модуля	Чтение	8	1	—	—	+	—	—
	Занесение	8	17	—	+	—	—	—
Регистр управления	Чтение	9	1	—	—	+	—	—
	Занесение	9	17	—	+	—	+	—
	Установка исходного состояния	—	—	<i>Z</i>	—	—	+	—
	Сброс	—	—	<i>C</i>	—	—	+	—
Регистр состояния (статусный регистр)	Чтение	11	1	—	—	+	—	—
	Установка исходного состояния	—	—	<i>Z</i>	—	—	+	—
	Сброс	—	—	<i>C</i>	—	—	+	—
Регистр маски запросов	Чтение	13	1	—	—	+	—	—
	Занесение	13	17	—	+	—	+	—
	Установка всех «1»	—	—	<i>Z</i>	—	—	+	—
	Сброс	—	—	<i>C</i>	—	—	+	—
Регистр запросов	Чтение	15	1	—	—	+	—	—
	Селективное чтение (через маску)	15	4	—	—	+	—	—
	Селективное гашение	15	20	—	+	—	—	+
	Установка исходного состояния — сброс всех запросов	—	—	<i>Z</i>	—	—	+	—
	Сброс	—	—	<i>C</i>	—	—	+	—

2. Регистр управления реализует специфичное управление в модуле, в том числе хранит команды или их модификаторы, предварительно заносимые по *W*-шинам, и выполняет коммутации в модуле. Применение необязательно.

3. Регистр состояния (статусный регистр) применяется, если модуль генерирует *L*-сигналы, и имеет нетривиальный статус.

4. Регистр маски запросов используется для разрешения (единица в маске) выхода отдельных запросов по разрядам на *L*-шину модуля, обязателен, если число источников запросов больше 1.

5. Регистр запросов содержит все запросы модуля (кроме коллективного) по разрядам, в порядке приоритета младших разрядов. Для управления используются две нестандартные команды: *F(4)* — селективное чтение (по аналогии с *F18*, *F19*) через маску запросов; *F20* — селективное гашение отдельных разрядов регистра в соответствии с единицами на *W*-шинах.

6. Регистр характеристики может содержать данные двух типов (не зависящие от содержимого остальных регистров): о структуре модуля, в том числе о наличии указанных выше устройств; о положении модуля в системной или программной иерархии (например, базовые

константы адреса подпрограмм реакции на запросы модуля). Содержимое регистра не должно влиять на остальные функции модуля.

Отметим дополнительно, что при использовании регистров, указанных выше, желательно ориентироваться на младшие 8 разрядов W - или R -шин. Указанные в табл. 5 AF должны использоваться только по прямому назначению. Субадреса A (10, 12, 14) с операциями II группы резервированы под расширение статуса и управления.

ЛИТЕРАТУРА

1. EUR 4100e. CAMAC. A Modular Instrumentation System for Data Handling. Esone Committee, 1969.
2. EUR 4600e. CAMAC. Organisation of Multi-crate Systems. Esone Committee, 1972.

Поступила в редакцию 31 октября 1972 г.

УДК 631.291.27

Г. Ю. АВЕРБУХ, Ю. Л. РОЗОВ, И. Б. ЧЕЛПАНОВ

(Ленинград)

О ПОГРЕШНОСТИ ИЗМЕРЕНИЯ МАКСИМАЛЬНЫХ ЗНАЧЕНИЙ СТАЦИОНАРНОГО СЛУЧАЙНОГО ПРОЦЕССА ДИСКРЕТНЫМИ МЕТОДАМИ

I. Рассматривается задача оценки точности определения максимальных (минимальных) значений исследуемого сигнала при дискретном методе измерения. Подобная задача возникает в тех случаях, когда регистрации самого процесса не требуется, интерес для исследователя представляют лишь экстремальные значения сигнала, а применение непрерывных методов невозможно в силу каких-либо обстоятельств (в частности, подобная ситуация имеет место при регистрации информации, поступающей от нескольких датчиков на одно измерительное устройство, к которому эти датчики подсоединяются поочередно посредством коммутатора).

Определение экстремальных значений непрерывного процесса по дискретным данным, естественно, приводит к появлению амплитудных и фазовых погрешностей, т. е. погрешностей определения ординаты и абсциссы «пикового» значения.

В ряде случаев важна фиксация экстремальных значений, а интервалы между экстремумами интереса не представляют. Такое положение, в частности, имеет место, когда записи служат исходным материалом для расчета систем на надежность, если появление отказов вызывается выбросами воздействий.

Для нахождения экстремальных значений могут быть использованы как простейшие алгоритмы, основанные только на выборе значений из совокупности в соответствии с определенными логическими условиями, так и более сложные, в которых, помимо направленной выборки, используется также более или менее сложная локальная интерполяция процессов.