

С.В.Моргунов

ЭМУЛЯТОР КАМАК-СИСТЕМ В СРЕДЕ ДОС ЕС ЭВМ

(г.Куйбышев)

В настоящее время в связи с построением автоматизированных систем научных исследований (АСНИ) широкое распространение получили измерительно-вычислительные комплексы (ИВК), в состав которых входит управляющая микро-ЭВМ и некоторый набор периферийных устройств сбора, обработки данных, выполненных в стандарте КАМАК [1,2]. Естественно, возникает потребность в создании программного обеспечения (ПО) для подобных ИВК. Однако процесс создания и отладки ПО с использованием микро-ЭВМ ("Электроника-60") и КАМАК-периферии малоэффективен в силу ограниченности средств отладки; снижение времени проектирования и повышение эффективности отладки ПО являются одной из основных задач. Поэтому в последнее время уделяется особое внимание построению цифровых имитационных моделей различных вычислительных средств, используемых в качестве концептуальной основы программно-аппаратной эмуляции.

Использование подобных кросс-систем имеет ряд следующих преимуществ, позволяющих решать вопрос об эффективности создания и отладки ПО [3]:

возможность создания и отладки ПО ИВК при отсутствии аппаратных компонент ИВК (микро-ЭВМ и (или) КАМАК-аппаратуры);

появление качественно нового уровня отладки программ, написанных для микро-ЭВМ и КАМАК-периферии. Использование богатого арсенала средств отладки ЕС ЭВМ;

возможность параллелизации во времени двух процессов;

разработка нового модуля стандарта КАМАК и создание программ работы с ним;

возможность проведения статистических исследований работы ИВК на модели (исследование степени загруженности магистрали КАМАК, частоты обращения к тому или иному модулю, функции, время работы моделирующей аппаратуры и др.).

Этими качествами обладает эмулятор полной конфигурации [4], основными блоками которого являются эмулятор микро-ЭВМ "Электроника-60" и эмулятор КАМАК-системы.

Эмулятор КАМАК-системы *CAMEL* (*CAMac Emulation*) представляет собой реализацию алгоритма идентификации-интерпретации [4], суть которого сводится к следующему: из входного сигнала выделяется код КАМАК-функции, а также параметры, необходимые для обработки этой функции в реальном устройстве. Имитация (интерпретация) сводится к изменению значения соответствующего элемента памяти, представляющего собой модель внутренних регистров КАМАК-модулей. При этом формируются входные и выходные сигналы, характеризующие текущее состояние модуля. В силу выбранной методологии реализации эмулятора, программный блок @IDEN помимо функций "чистой" идентификации входного сигнала выполняет еще и функции управления работой всей системы в целом, являясь, таким образом, некоторым подобием монитора системы. К функциям управления относятся:

1. Настройка *CAMEL* на тот или иной режим работы (режим "чистой" эмуляции: "эмуляция + трассирование"; режим "эмуляция + статистический сбор").

2. Организация циклических операций ввода (вывода и частичной трассировки отлаживаемых программных блоков).

Прежде чем перейти к рассмотрению вопроса о функционировании *CAMEL*, поясним некоторые принятые обозначения:

1. @ <ИМЯ> - идентификатор законченных, автономных программных блоков эмулятора;

2. S_T - текущее состояние эмулятора, S_K - конечное состояние эмулятора. Реально S_T и S_K определяются вне системы *CAMEL* и необходимы для таймирования работы эмулятора.

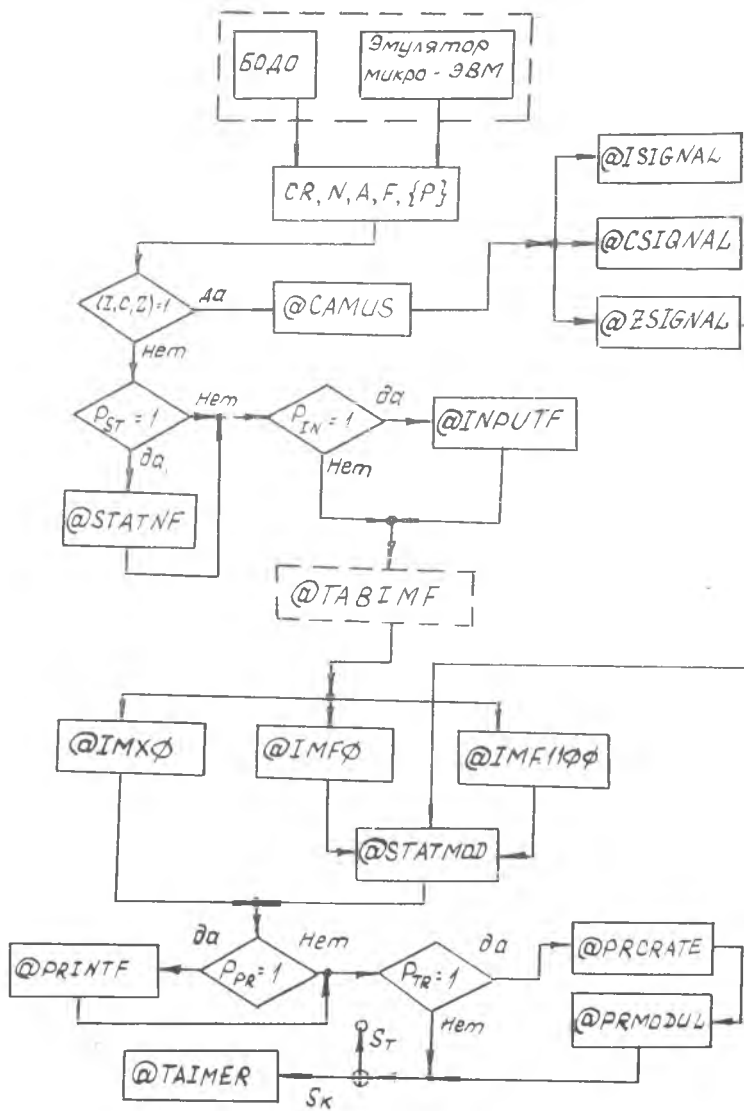
3. $\{P_{ST}, P_{IN}, P_{PR}, P_{TR}\}$ - набор признаков, выставляемых системой обработки директив отладки КАМАК-команд.

4. $\{I_i\}_{i=1, \bar{n}}$ - совокупность входных воздействий. Это информация о регистре управления и состоянии (PUC) и "общей шине" (OШ).

5. Блок @TABIMP представляет собой массив имен программ интерпретации, т.е. не является функциональным программным блоком.

С учетом введенных обозначений рассмотрим работу системы *CAMEL*

(рис.1). На вход системы (вход-блок @IDEN) поступают в форматизованном виде значения PUC и OШ, а также информация из блока обработки директив отладки (БОДО). Блок @IDEN идентифицирует следующие параметры из PUC и OШ: CR - номер крейта, N - номер станции в крейте, F - код КАМАК-операции, A - субадрес, сигналы управления I, C, Z . Если выставлены биты сигналов I, C, Z , то происходит имитация отработки этих сигналов в крейте в соответ-



Р и с. 1. Функциональная схема системы CAMEL.

ствии с алгоритмом обработки каждого из этих сигналов в реальном КАМАК-модуле. Имитация обработки сигналов осуществляется блоками: @CAMUS, @ISIGNAL, @CSIGNAL, @ZSIGNAL, после этого блок @IDEN анализирует параметры CR, N на принадлежность допустимому интервалу $CR \in [1,7], NE \in [1,24]$. В случае нарушения этих условий формируется и выдается на печать сообщение об ошибке и текущем состоянии системы. Эмуляция в этом случае не происходит. Система переходит в состояние *Ступица*.

Если же $CR \in [1,7], NE \in [1,24]$, то блок @IDEN формирует адрес A_{CN} соответствующего элемента массива модели внутренних регистров КАМАК-модуля, а также адрес A_{IN} программы имитации данной функции (по таблице @TABIMF). По вычислительному последнему адресу в таблице находится имя необходимой программы интерпретации и осуществляется обращение к последней.

Программы интерпретации выделены в три группы:

1. Если команда не принадлежит набору команд активизированного модуля, то формируется выходной сигнал $X \neq 0$ и выдается соответствующее сообщение блоком @IMxφ.

2. Если поступившая команда принадлежит набору команд данного модуля и алгоритм обработки этой команды общий для всех модулей КАМАК-системы, то ее имитация производится одной из программ блока @IMFφ.

3. Если имеет место случай 2, но алгоритм обработки команды специфичен для каждого модуля, то ее имитация осуществляется одной из программ блока @IMF11φφ.

Интерпретация сводится к изменению содержимого элемента с адресом массива памяти, являющегося моделью внутренних регистров КАМАК-устройств. А также формируются признаки $X = I$ и пересылаются в РУС.

Необходимо отметить, что интерпретационная часть системы CAMEL (как функции, так и сигналов управления) построена по модульному принципу, т.е. программы интерпретации команд и сигналов собираются из отдельных, автономных программ блока @STATMOD. Это является существенным преимуществом системы CAMEL, так как из сравнительно небольшого по объему блока @STATMOD можно конструировать множество программ интерпретации широкого спектра команд.

Мы рассмотрим случай функционирования CAMEL, когда $\{P_i\}_{i=1, N} \equiv \phi$, т.е. при отсутствии директив пользователя. Этот режим мы назвали режимом "чистой" эмуляции. Однако в большинстве случаев процесс

"чистой" эмуляции сопровождается статистическими исследованиями, которые задаются с помощью директив пользователем, что позволяет повысить наглядность моделирования, эффективность как самих отлаживаемых программ, так и всего процесса отладки в целом.

БОДО на основе анализа директив отладки формирует ряд следующих признаков, которые поступают на вход системы *CAMEL* и анализируются на этапе идентификации:

P_{ST} - признак сбора статистических данных. В случае его единичного значения после идентификации параметров происходит обращение к блоку *@STATNP*, который организует сбор статистики по количеству обращений к данному модулю;

P_{IN}, P_{PR} - признаки циклического ввода (вывода информации при отладке программ). В случае их единичного значения происходит циклический ввод данных и (или) частичная трассировка состояния моделей, которые указаны в директивах отладки;

P_{TR} - признак полной трассировки. В случае его единичного значения производится распечатка состояния модели магистрали и модуля, к которому пришло обращение.

После окончания работы эмулятора *CAMEL* происходит выдача времени эмуляции КАМАК-команд (блок *@TIMER*), а также, если это необходимо, таблиц статистики. Необходимо отметить, что система написана на языке ассемблера ДОС ЕС ЭВМ.

Описанная первая версия эмулятора моделирует работу двух крейтов КАМАК. В дальнейших модификациях системы предусматривается возможность автоматической настройки системы на необходимое количество крейтов и их архитектуру, а также построение модели внешней среды, что существенно расширит применение предлагаемой системы моделирования.

Л и т е р а т у р а

1. Колпаков И.Ф., Смирнов В.А., Хмелевский Е.В. Современное состояние системы КАМАК.- Приборы и техника эксперимента, 1975, № 9, с.7-16.

2. Малые ЭВМ и их применение /Под ред. Б.Н.Наумова.-М.:Статистика, 1980.- 231с.

3. Супник Р.М. Отладка в режиме моделирования.-В кн.:Средства отладки больших систем.-М.:Статистика, 1977, с.107-124.

4. Коварцев А.Н., Кораблин М.А., Шамашов М.А. Имитационное моделирование систем автоматизации эксперимента с использованием мультиторов полной конфигурации.—УСИС, 1979, № 4, с.124—127.

УДК 681.142.2

Л.М.Нусинзон

ОРГАНИЗАЦИЯ ДРАЙВЕРА УПРАВЛЕНИЯ АСИНХРОННОЙ РАБОТОЙ ПРИБОРОВ ЧЕРЕЗ ГРУППОВОЙ КОНТРОЛЛЕР

(г. Свердловск)

При создании средств прикладного программного обеспечения для автоматизированных систем экспериментальных исследований, когда приборы и другие компоненты системы (модули) объединены приборным интерфейсом через групповой контроллер с машинным интерфейсом [1], оказывается удобным считать приборные модули или их совокупности как дополнительные пронумерованные внешние устройства, подключенные к ЭВМ. Для выполнения этими приборными модулями некоторых макроопераций необходим реентерабельный драйвер (процедура), который может взять на себя все функции по управлению асинхронной работой этих модулей на протяжении всей макрооперации: от настройки модулей до ввода или вывода данных в память ЭВМ или на модуль. Для каждой группы приборных модулей, алгоритм управления которыми одинаков, составляется ряд программных обрабатывающих блоков, которые включаются в состав драйвера.

Программное управление модулем или группой модулей при выполнении какой-либо макрооперации можно представить в виде последовательности команд. Время выполнения приборным модулем каждой из команд различно и может быть соизмеримо или меньше времени реакции операционной системы $t_{пр}$ на внешнее прерывание, либо быть значительно большим. Поэтому команды целесообразно разбить на ряд последовательных групп. Каждая группа заканчивается командой, время выполнения которой $t_i > n t_{пр}$, $n \geq 5$.

Время t_i может определяться временем срабатывания или быстродействием модулей и приборного интерфейса. Например, временем, необходимым для коммутации заданного канала коммутатором или аналого-цифрового преобразования. Операции, связанные с передачей