

7. Щукин П. В. Определение структуры изображений методами когерентной оптики.— В кн.: Статистические свойства микроструктур. [I Всесоюз. конф.]. М.: ОНТИмикробиопром, 1978, с. 94—95.

*Поступила в редакцию 1 августа 1980 г.;
окончательный вариант — 29 июня 1981 г.*

УДК 681.3.06

Введение. Одним из важных методов ускорения разработки систем автоматизации эксперимента на основе аппаратуры КАМАК является использование унифицированных языков программирования, что позволяет не только сократить сроки подготовки программного обеспечения этих систем, но и обеспечить независимость создаваемых программ от типов используемых крейт-контроллеров. В соответствии с этим комитет ESONE разработал и стандартизовал несколько уровней программных средств для управления аппаратурой КАМАК. В статье описывается реализация подмножества одного из таких языков — КАМАК-языка промежуточного уровня IML [1, 2] для автономного крейт-контроллера JCAM-10 [3], построенного на основе микропроцессора INTEL-8080.

Программирование КАМАК-команд для микропроцессора INTEL-8080. Автономный контроллер крейта JCAM-10 конструктивно выполнен в виде модуля КАМАК со встроенным микропроцессором INTEL-8080. Описание контроллера приведено в [3].

Рассмотрим подробнее устройство КАМАК-страницы. КАМАК-страница представляет собой 1 К 24-разрядных слов. Адреса этих слов памяти по отношению к микропроцессору являются виртуальными и служат лишь для запуска КАМАК-цикла по передаче данных между КАМАК-страницей и регистрами модулей. Микропроцессор имеет доступ к каждому из трех байтов 24-разрядного слова КАМАК-страницы.

Адреса регистров модулей в крейте и КАМАК-функции образуют виртуальную КАМАК-память с адресами C000—FFFF.

Автономный контроллер JCAM-10 имеет 8 уровней прерывания: 4 КАМАК-уровня и 4 системных уровня. Прерывания могут быть маскированы с помощью общей маски, индивидуальной маски на каждый уровень прерывания и КАМАК-команд маскирования L-сигналов.

Программирование КАМАК-команд в автономном контроллере JCAM-10 сводится к использованию стандартных инструкций микропроцессора INTEL-8080, ссылающихся на специальную виртуальную память.

Для выполнения КАМАК-команд управления и проверки статуса модулей (без передачи данных) необходимо однократное обращение к виртуальной КАМАК-памяти, адрес которой соответствует необходимому сочетанию F, N, A. Сигналы Q и X, вырабатываемые модулями, помещаются в рабочий регистр (обычно аккумулятор).

КАМАК-команды чтения или записи требуют (кроме обращения к виртуальной КАМАК-памяти) выполнения инструкции, использующей обращение к КАМАК-странице. При этом осуществляется КАМАК-цикл с указанными ранее F, N, A и передачей данных между регистром модуля и КАМАК-страницей в направлении, определяемом значением функции F.

Описание реализации подмножества языка IML. Описываемое ниже подмножество языка IML призвано обеспечить возможность выполнения одиночных действий и относительную независимость программ от кон-

фигурации аппаратуры КАМАК и типов адресации полей памяти. Поэтому в него включены лишь следующие группы операторов: операторы одиночного действия, операторы управления крейтом в целом и декларативные операторы, обеспечивающие работу операторов одиночного действия.

Остановимся подробнее на реализованных операторах.

1. Оператор описания полей памяти. Оператор имеет вид

LOCM DAT, TA, L

где LOCM — имя оператора, DAT — имя поля памяти, TA — тип адресации, L — длина поля памяти (1, 2 или 3 байта).

Возможны следующие типы адресации: I — непосредственная (DAT определена как константа), DM — прямая адресация памяти, DC — прямая адресация КАМАК-страницы, R — данные в регистре DAT, IDM — косвенная адресация памяти через регистр DAT (в регистре — адрес памяти), IDC — косвенная адресация КАМАК-страницы через регистр DAT (в регистре — адрес памяти в КАМАК-странице).

2. Оператор описания регистров модулей. Оператор имеет вид

LOCD NAME, MOD, N, A

где LOCD — имя оператора, NAME — символическое имя описываемого регистра, MOD — тип определяемого КАМАК-адреса, N — номер станции модуля, A — субадрес определяемого регистра.

Возможны два типа КАМАК-адреса: H — константа (определяется константа с именем NAME и со значением КАМАК-адреса), P — переменная (резервируется поле памяти в 2 байта с адресом NAME, в которое помещается КАМАК-адрес).

3. Оператор одиночного действия без анализа сигналов Q и X. Оператор имеет вид

SA F, NAME [, DAT]

где SA — имя оператора, F — КАМАК-команда, NAME — имя КАМАК-адреса, DAT — имя поля памяти (для команд без передачи данных может быть опущено). Этот оператор выполняется как обращение к КАМАК-адресу NAME с КАМАК-командой F. Данные передаются из поля памяти или в поле памяти с именем DAT. Направление передачи данных определяется значением КАМАК-команды F.

4. Операторы одиночного действия с анализом сигнала Q. Операторы имеют вид

SJNQ F, NAME [, DAT], ADR

SJQ F, NAME [, DAT], ADR

где SJNQ, SJQ — имена операторов, F — КАМАК-команда, DAT — имя поля памяти (необязательный параметр), ADR — адрес команды, которая выполняется следующей за оператором SJNQ(SJQ), если Q = 0 (Q = 1).

5. Операторы одиночного действия с анализом сигнала X. Операторы имеют вид

SJNX F, NAME [, DAT], ADR

SJX F, NAME [, DAT], ADR

где SJNX, SJX — имена операторов, F — КАМАК-команда, DAT — имя поля памяти (необязательный параметр), ADR — адрес команды, которая выполняется следующей за оператором SJNX (SJX), если X = 0 (X = 1).

6. Операторы маскирования и демаскирования прерываний. Операторы имеют вид

SM MASK

RM MASK

где SM, RM — имена операторов разрешения и запрещения прерывания, MASK — константа, определяемая номером станции модуля и положением переключателей в контроллере крейта (имеет размер 1 байт).

7. Операторы управления сигналами I, C, Z и прерываниями. Операторы имеют вид: SI — установка сигнала INHIBIT, RI — сброс сигнала INHIBIT, ZC — выработка сигналов Z и C по крейту, EI — общее разрешение прерываний, DI — общее запрещение прерываний.

Последние два оператора являются командами микропроцессора INTEL-8080.

Описанные выше операторы реализованы в виде макрокоманд на стандартном языке Макроассемблер для микропроцессора INTEL-8080.

Пример программы с использованием языка IML. Составим подпрограмму вывода байта данных на перфолоту. Пусть управление перфолотой осуществляется модулем КАМАК, имеющим следующие КАМАК-команды:

N(i)A(0)F(8) — проверка готовности перфолоты на вывод байта данных (Q = 0 — не готов, Q = 1 — готов);

N(i)A(1)F(8) — проверка наличия ошибки перфолоты (Q = 0 — нет ошибки, Q = 1 — ошибка);

N(i)A(0)F(16) — вывод байта данных на перфолоту.

Тогда один из возможных вариантов подпрограммы вывода байта данных из ячейки памяти с адресом PPBYT на перфолоту на языке Макроассемблер для микропроцессора INTEL-8080 с использованием описанного выше набора макрокоманд, реализующего подмножество языка IML, будет иметь следующий вид:

```

;
;   LOCM PPBYT, DM, 1   ; описание поля памяти
;   LOCD PPA0, H, 8., 0 ; описание регистра A(0)
;   LOCD PPA1, H, 8., 1; ; описание регистра A(1)
;
PPBYT: DB %00          ; байт данных
;
PUNCH: SJQ 8., PPA1, ERROR ; ошибка?
;       SJNQ 8., PPA0, PUNCH ; готов?
;       SA 16., PPA0, PPBYT ; вывод байта данных на перфолоту
;       RET                  ; выход из подпрограммы
ERROR: HLT              ; останов по ошибке

```

Заключение. Применение описанного выше подмножества языка IML облегчает написание, отладку программ и использование их другими программистами, что позволяет сократить сроки проектирования программного обеспечения для управления аппаратурой КАМАК. Необходимо отметить, что создание языка IML для отечественных автономных контроллеров, построенных на базе микропроцессорного набора K580, существенно облегчится, если использовать описанную реализацию.

ЛИТЕРАТУРА

1. The Definition of IML. (A Language for Use in CAMAC Systems).— ESONE [IML]01. Published by the ESONE Committee, October 1974.
2. Казакова Н. А., Панкрац Е. В. Реализация языка промежуточного уровня IML на ЭВМ типа СМ-3.— Автометрия, 1980, № 3.
3. Ипкитюк Н. М. Система КАМАК и микропроцессоры (обзор).— ПТЭ, 1978, № 1.

*Поступила в редакцию 10 апреля 1980 г.;
окончательный вариант — 2 декабря 1980 г.*