

сообщения
объединенного
института
ядерных
исследований
Дубна

1873/2-80

21/4-80
13 - 13031

С.Г.Басиладзе

**REST - ПРОСТЕЙШИЙ ИНТЕРПРЕТАТОР
ДЛЯ 8-РАЗРЯДНЫХ МИКРОПРОЦЕССОРОВ**

1980

В настоящей работе предпринята попытка создания достаточно простого проблемно-ориентированного языка программирования для специалистов, использующих микропроцессоры в составе однокрепных систем КАМАК. Ставилось целью: обеспечить диалоговый режим работы, сочетающийся с возможно большим быстродействием; добиться малости объема памяти, необходимой для использования языка; достичь удобств отладки программ, близких к обеспечиваемым системами с дисковой памятью. Язык базируется на системе команд 8-разрядных микропроцессоров Intel-8080^{/1/} и на командах КАМАК системы MISKA^{/2,3/}.

ОБЩИЕ ПРИНЦИПЫ ОРГАНИЗАЦИИ

В основу построения языка положены следующие принципы:

1. Язык построен по схеме интерпретатора^{/4/}, т.е. каждому оператору программы-источника /далее - текст/ соответствует определенная подпрограмма, хранящаяся в ЗУ микро-ЭВМ. Данная структура /хранение всех подпрограмм в памяти/ обеспечивает, как известно, диалоговый режим работы.

2. В процессе записи текста генерируется объектная программа /далее - программа/, содержащая набор машинных команд CALL...*, а также преобразованные в выбранный формат /см. ниже/ числа, имеющиеся в тексте. Хотя подобная трансляция и не является обязательной в языке-интерпретаторе, это значительно убыстряет исполнение программы, поскольку исключаются потери времени на опознавание символов текста и преобразование чисел из десятичного в выбранный формат /особенно заметные при циклических операциях/. Генерирование объектной программы позволяет обеспечить быстродействие на уровне языков-трансляторов. В то же время она занимает достаточно малый объем, т.к. каждая подпрограмма содержится в памяти только в одном месте, а повторяются лишь команды их вызова.

3. Алгоритм вычислений записывается в обратной бескомочной записи (Polish Mode)**, т.е. сначала вводятся операнды,

* вызывающих подпрограммы, последовательность которых определяется текстом.

** используемой в калькуляторах фирмы Hewlett Packard^{/5/} и в языке ФОРТ^{/4/}.



а затем указывается действие, производимое над ними. При сохранении минимального числа действий, характерного для алгебраических выражений с приведенными подобными членами, такая форма записи освобождает от необходимости лексического анализа выражений /количества и старшинства скобок/, поскольку алгоритм вычислений записывается в виде линейной последовательности действий. Это сильно упрощает язык. Для хранения промежуточных результатов при таком способе вычислений, как известно, необходима стековая память (LIFO)*.

4. Следующей мерой, принятой для упрощения языка, является отказ от буквенных символьных наименований подпрограмм и массивов данных /и, соответственно, программного аппарата их опознавания и распределения памяти/. Метками подпрограмм являются цифровые обозначения, их вызов производится путем косвенной адресации. Поскольку язык предназначен для сравнительно простых информационно-измерительных систем, в которых не возникает больших проблем с распределением памяти, местоположение массивов данных определяется программистом.

5. Для сокращения объема и длины текста каждый оператор записывается одним знаком /буквой/, последовательно можно записывать неограниченное количество операторов. Символы окончания строки (Ret) и перевода строки (L.F.) не являются значащими и не воспринимаются программой.

6. Язык организован таким образом, что позволяет исключить перфоленты при написании программ и их редактировании. Перфолента необходима лишь для начального ввода текста после включения микро-ЭВМ, и вывода его для хранения перед ее выключением. Это достигается использованием монитора и редактора системы MCS-80^{6/}, записанных так же, как и сам интерпретатор, в ПЗУ микро-ЭВМ. Объем интерпретатора составляет всего 3К байт, а монитора и редактора - соответственно 2К и 3К байт; для работы им необходимо также от 500 до 8К байт /при редактировании/ ОЗУ.

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ

Формат чисел в языке REST может быть различным при вводе, выводе и в процессе производства вычислений. В последнем случае, т.е. непосредственно в микро-ЭВМ под число отводятся три байта ОЗУ.

При производстве арифметических операций число представляется в двоичной экспоненциальной форме /с плавающей запятой/:

* Отсюда произведено название REST: REverse mode-STack.

два младших байта отведены под мантиссу /старший, 16-й бит которой всегда содержит 1/, а старший байт - под показатель степени, знак числа /8-й бит/ и знак экспоненты /7-й бит - см. рис. 1а/. Для КАМАК-операций и адресов операций "Запись", "Чтение"/с памятью/ число представляется в виде целого /6 младших разрядов третьего байта переводятся сдвигами вправо мантиссы в нули - рис. 1б/. Значащая часть числа выбрана 16-разрядной для того, чтобы обеспечить адресацию к полному возможному полю памяти микро-ЭВМ, а также выдержать наиболее употребляемую длину слова данных КАМАК.

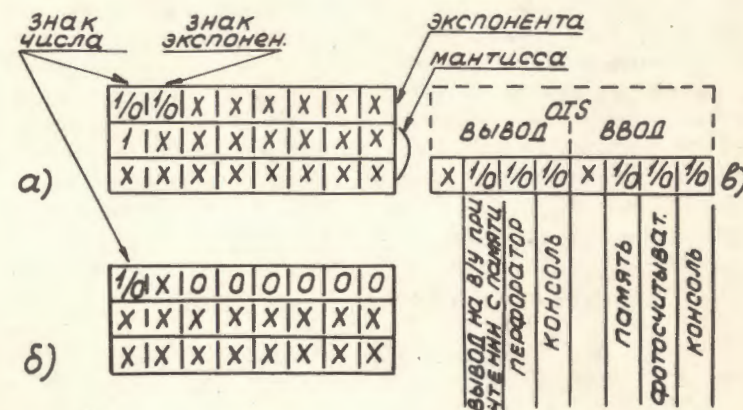


Рис. 1. Формат двоичных нормированных чисел /а/, целых чисел - /б/, структура младшего байта ОИС - /в/.

Ввод числа производится в десятичной экспоненциальной форме. Экспонента /до двух цифр/ отделяется от мантиссы /до пяти цифр/ знаком "\", обозначающим положительную степень десяти, либо знаком "/", обозначающим отрицательную степень десяти. Например: $314/2 = 3,14$, но $314 \setminus 2 = 31400$. Отрицательные числа обозначаются символом "-" /код ASCII 5FH/ для различия в обратной записи со знаком "-", обозначающим вычитание. Вводятся могут также целые десятичные числа, не превышающие $65535 / 2^{16} - 1$, они автоматически переводятся в двоичную экспоненциальную форму. Признаком окончания числа является символ пропуска (Space)*, или любой оператор.

* Записывается только в текст, в программе отсутствует.

При выводе числа всегда представляются в десятичной экспоненциальной форме. Рабочий диапазон чисел в языке составляет от $+5 \cdot 10^{-23}$ до $+5 \cdot 10^{23}$. Относительная ошибка при выполнении одной операции $\sim 2 \cdot 10^{-5}$ /кроме операции возведения в степень/, при вводе-выводе чисел преобразование форматов экспонент дает погрешность до $5 \cdot 10^{-4}$.

ИНТЕРПРЕТИРУЕМАЯ СТРУКТУРА

В процессе работы язык занимает с 256**/100H - в шестнадцатиричной записи/ по 400 /190H/ ячейки ОЗУ микро-ЭВМ, интерпретируя с их помощью вычислительную структуру, показанную на рис. 2. Центральной ее частью является т.н. базовый регистр - BR /3 байта, начиная с 300 - 12CH/. В этот регистр вводятся /выводятся/ числа из программы, из памяти, либо из внешних устройств. Имеется также стековая память из 30 чисел - DS (Data Stack). При вводе числа в BR число, ранее в нем содержавшееся, автоматически переводится в DS; при выводе производится обратное перемещение. Если в программе содержится оператор преобразования числа /нахождения логарифма, синуса и т.п./, то операция производится всегда над числами в BR.

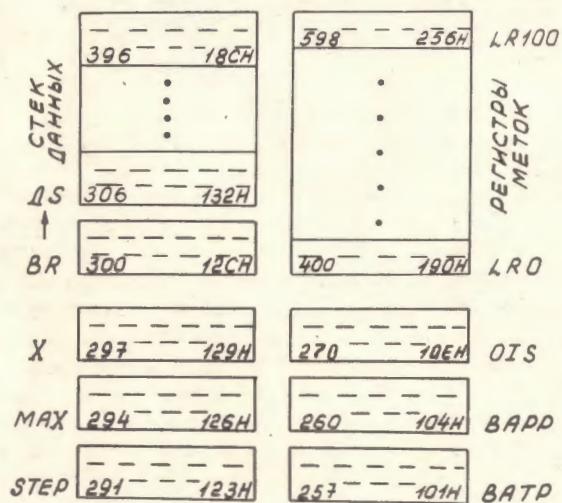


Рис. 2. Структура, интерпретируемая с помощью языка REST.

** В ячейках до 255 (FFH) располагается стек микропроцессора /8/.

Если производится действие с двумя числами /деление, умножение и т.п./, то первое всегда берется из DS, а второе из BR, результат заносится в BR, а указатель стека уменьшается на одно число. При операциях, имеющих некоторое количество численных параметров /вызов подпрограмм, КАМАК-операции и т.п./, эти параметры предварительно должны быть в определенном порядке занесены в BR, откуда они /за исключением последнего перед оператором/ перемещаются в DS /в процессе ввода чисел/ и будут вычеркнуты в процессе выполнения операции.

Для хранения до 100 цифровых меток /наименований/ подпрограмм выделена специальная зона с 400 по 599 ячейки - по два байта на адрес. В коротких программах часть этой зоны может быть использована для записи данных.

Имеется также несколько вспомогательных регистров /в ячейках 256÷300/:

- а/ X, MAX, STEP, CLP - для организации циклов;
- б/ OIS (Output-Input Status) - для организации ввода-вывода текстов и результатов на различные внешние устройства;
- в/ BAPP, BATR - для хранения начальных адресов текста и программы;
- г/ CAT, CAP - для хранения текущих адресов текста и программы при записи и трансляции;
- д/ ST1÷ST5 - для хранения промежуточных результатов при выполнении некоторых операций.

ВВОД ТЕКСТА, ТРАНСЛЯЦИЯ ПРОГРАММЫ

Ввод /чтение/ текста возможен с различных внешних устройств, а также непосредственно из памяти микро-ЭВМ /см. рис. 3/. В первом случае специально выделен режим CIN-ввода текста с клавиатуры консольного устройства /телетайп, дисплей/. Стартовый адрес его 2990H, состояние OIS=11H. Этот режим используется в процессе диалога. Режим "чтения" и трансляции текста непосредственно из памяти - MIN /стартовый адрес 2993H, OIS=14H /необходим после редактирования текста или при повторном запуске программы. В этих режимах текст и программа занимают указанное для них положение /от 000H и 460H - соответственно, см. рис. 3/.

Имеется также "свободный вход" /стартовый адрес 2996H/. В этом режиме программист сам предварительно задает места расположения текста и программы и режим ввода-вывода. Если "1" содержится /рис. 1в/:

- а/ во 2 бите OIS, то ввод производится с перфоленты;
- б/ в 5 бите OIS, то вывод производится на консоль;

в/ в 6 бите OIS, то вывод производится на перфоленту;
 г/ в 7 бите OIS, то производится вывод текста на консоль
 или перфоратор при "чтении" его из памяти /OIS соответ-
 ственно 54H, либо 64H/.

После инициации в стек процессора /рис. 3/ заносится адрес
 возврата интерпретатора в исходное положение, куда он вер-
 нется после исполнения вводимой программы. Сигналом готовно-
 сти интерпретатора к вводу нового текста является вывод зна-
 ка "!" на консоль.

Адрес 29B5H является точкой вызова интерпретатора как не-
 кой подпрограммы, который можно производить из любой внешней
 программы.

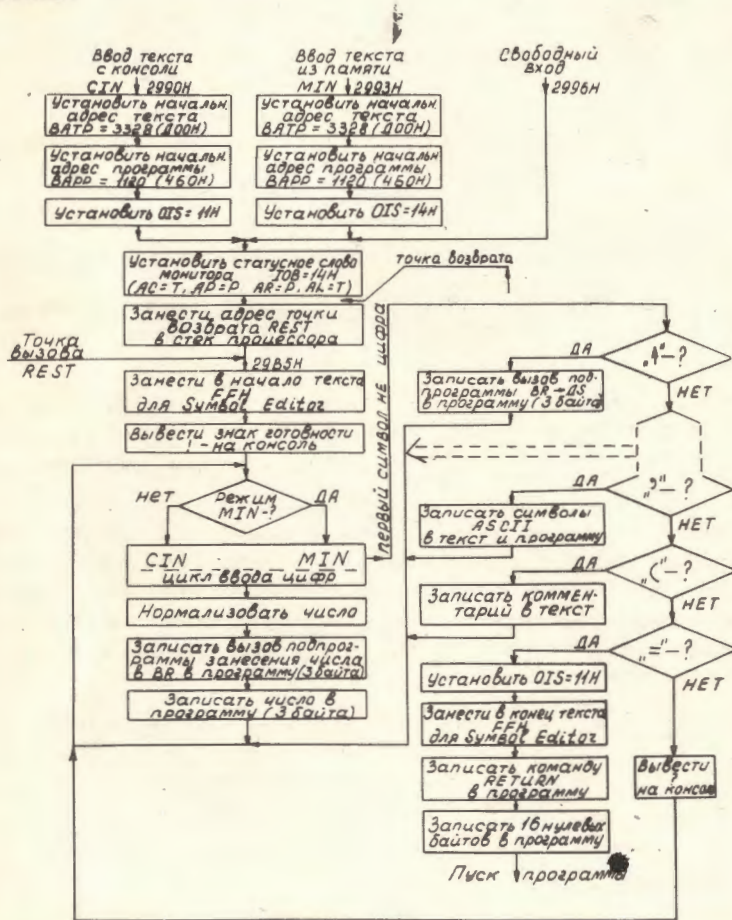


Рис. 3. Алгоритм действия управляющей программы языка REST.

Программа ввода чисел и операторов циклическая /см.рис. 3/.
 В процессе ввода производится запись чисел и операторов в
 текст /"незаконные" символы отбрасываются со знаком "?".
 Одновременно производится запись команды CALL... (CD...) - вы-
 зова соответствующей подпрограммы в программу. Команда вызо-
 ва занимает 3 байта; если в текст было введено число, то
 к команде CD-75-24/H/ добавляются еще три байта, содержащие
 это число в нормированном виде. В программу также заносятся
 коды символов ASCII, которые необходимо вывести на внешнее
 устройство в процессе ее исполнения.

Признаком окончания текста и командой перехода к исполне-
 нию программы является знак "=". При появлении этого знака
 OIS -байт устанавливается /рис. 1в/ в положение ввода-вы-
 вода с консоли, в программу записывается команда возврата
 /C9/ в исходную точку и далее 16 нулевых байтов /для визуаль-
 ного выделения программы/. В начало и конец текста вводятся
 коды FF(H), необходимые для распознавания текста редактором
 /см. ниже/.

ОПЕРАТОРЫ

Перемещение данных. Числа, содержащиеся в программе, в про-
 цессе ее исполнения вводятся в BR /число, бывшее там ранее,
 переводится в DS* /.

↑-246BH.** Данный оператор добавляет в стек данных число,
 равное содержимому в BR. Такое копирование необходимо, на-
 пример, перед ссылкой числа в память или перед выводом на
 внешнее устройство, если мы хотим, чтобы оно осталось в BR.

W-2489H. Выводит число из DS в три байта памяти, адрес
 младшего из которых содержится в BR. Например: 314/2 1000W
 есть команда занести число 3,14 по 1000 адресу. Оба числа
 /данное и адрес/ вычеркиваются после исполнения команды, а в
 BR вводится следующее число из DS, т.е. стек данных в итоге
 уменьшается на два числа.

R-249EH. Чтение в BR числа из памяти по адресу, содержа-
 щемуся в BR до этой операции. Например: 1000R - будет коман-
 дой возвращения числа 3,14 в BR /в памяти оно сохранится/.
 Стек данных при этой операции не изменяется.

* Здесь и далее имеется в виду вершина стека данных.

** Адрес данной подпрограммы.

O-24B8H. Вывод числа из BR на внешнее устройство /определяемое кодом в OIS /. После вывода в BR переводится число из DS.

I-24A5H. Ввод числа в BR из внешнего устройства /определяемого кодом в OIS /. Число, содержащееся в BR ранее, переводится в DS.

X-28A3H. В BR заносится число, равное содержимому регистра X /содержимое X остается неизменным/. Предыдущее число из BR переводится в DS.

АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ

+ -2552H. Число в DS складывается алгебраически с числом в BR. Например, в программе: $_{314/2} \ 271/2 + 0 = \ 04298/04$!

- -2578H. Вычитание из числа в DS числа в BR.

. -2585H. Умножение числа в DS на число в BR.

: -25A9H. Деление числа в DS на число в BR.

& -25D2H. Побитное логическое умножение мантиссы числа в DS на мантиссу числа в BR.

Во всех перечисленных выше операциях результат заносится в BR, а DS уменьшается на одно число. При операции & экспонента числа в DS переносится в BR.

E - 263IH. Нахождение экспоненты - e^{BR} , результат заносится в BR, $DS = const$.

L - 26BDH. Нахождение натурального логарифма BR, результат в BR, $DS = const$.

* -2761H. Операция возведения в степень - DS^{BR} , результат в BR; DS уменьшается на одно число. Эта операция производится с помощью двух предыдущих, поэтому при возведении в целую степень для повышения точности лучше пользоваться операцией умножения.

N - 28B7H. Перевод нормированного числа в BR в представление в виде целого /рис. 1а,б/.

ЛОГИЧЕСКИЕ ОПЕРАЦИИ

400 ÷ 588A. Цифровая метка /символический адрес/ подпрограммы, или точки в программе. Адрес в программе следующего за символом A оператора, или числа в тексте заносится в процессе трансляции в регистр зоны меток /см. рис. 2/, определяемый

числом в тексте перед A. Ни сам оператор A, ни определяющее регистр число не заносятся в программу, а только в текст.

@ -27E6H. Переход /прыжок/ по одному из трех адресов, определяемому знаком числа, которое было в BR до ввода параметров /адресов/ данной операции. Адреса вводятся в виде чисел перед символом @. Поэтому непосредственно перед выполнением данной операции число, бывшее в BR, перемещается на третье место от вершины стека данных /после ввода адресов/. В процессе выполнения описываемой операции адреса выводятся из BR и DS, в результате чего предшествовавшее им число вновь оказывается в BR.

Переход по первому адресу происходит, если содержимое BR меньше нуля, по второму - если оно равно нулю, по третьему - если оно больше нуля.

В языке применено три типа адресации: относительная, косвенная и прямая - в зависимости от величины адреса. Если адрес меньше 300, то он воспринимается как указатель смещения /в байтах/ от адреса в программе оператора или числа, следующего в тексте за @ /относительная адресация/. Например, если $BR < 0$, то команда: $\emptyset \ 400 \ 650 @$ реализует переход к точке, следующей сразу за @ /нулевое смещение/. Если величина адреса находится в пределах от 300 до 600, то адресация будет косвенной - через содержимое соответствующего этому адресу регистра меток. Например, при $BR = 0$ команда: $\emptyset \ 400 \ 650 @$ реализует переход на цифровую метку $400A$. И, наконец, если величина адреса более 600, адресация будет прямой. Например, при $BR > 0$, команда: $\emptyset \ 400 \ 650 @$ реализует переход в точку с абсолютным адресом 650.

Безусловный переход осуществляется, если все три числовых параметра @ одинаковы. Например: $402 \ 11 @$ есть переход к символическому адресу $402A$.

\$ -27ECH. Вызов подпрограммы по одному из трех адресов, в зависимости от знака числа, бывшего в BR до перехода к данной операции. Принцип адресации идентичен @.

; -27FH. Возврат из подпрограммы.

< -27F3H. Начало циклической операции, инкрементной величиной которой является X. Перед данным оператором необходимо ввести три числовых параметра: минимальную и максимальную величины X, а также шаг его приращения в цикле. В начале данной операции в указатель цикла CLP (120H) заносится 1 /целочисленное представление/, которая заменяется нулем, когда $X > MAX$.

> -287FH. Возврат в цикле. Если $CLP = \emptyset$, начинается исполнение программы, следующей за данным оператором.

Пример циклической операции:

$$\int_0^{1/3} x^2 dx = 33572/05.$$

Отметим, что циклы, так же как и подпрограммы, могут быть вложенными /при переходе к циклу предыдущие величины X, MAX, STEP, CLP заносятся в стек процессора и выводятся из него по окончании цикла/.

ОПЕРАТОРЫ K, M, ' ', ()

K - 28EDH. Универсальная операция КАМАК. Операции типа "Чтение" выполняются по суммарному сигналу L с крейта, остальные исполняются без ожидания. Оператор требует предварительного введения двух числовых параметров:

a/ 16N + A, где N - номер станции, A - субадрес;

b/ F - функция КАМАК.

Эти параметры вычеркиваются из DS и BR в процессе выполнения операции. При операциях типа "Чтение" считанное 16-рядное число записывается в BR в нормированном виде. При операциях типа "Запись" из BR выводится число, представляемое предварительно в виде целого. Время исполнения универсальной КАМАК-операции ~ 4 мс, оно определяется, в основном, преобразованиями форматов чисел.

M. Выход в программу монитор; содержимое всех регистров остается неизменным. Данная операция весьма удобна при отладке программ, для анализа содержимого BR, DS и других регистров.

' '. Между этими операторами в тексте заключаются символы ASCII, которые необходимо вывести на внешнее устройство /заголовки, обрамление таблиц и т.п./. Длина данной команды в программе равна 2 + число символов, включая ' '.

(). Скобки применяются для внесения комментариев в текст. В программе ни они, ни комментарий не содержатся.

РЕДАКТИРОВАНИЕ

Оператор #. С помощью данного оператора можно в процессе записи текста производить стирание неограниченного количества символов, либо чисел и соответствующих им вызовов подпрограмм в транслируемой программе. Оператор стирает один нецифровой символ из текста и три байта из программы, либо целиком число из текста и шесть байтов из программы. В процессе записи комментариев и выводимых символов ASCII стираются только символы /либо числа/ в тексте. Поэтому после стирания выводимых символов ASCII /включая операторы ' '/ необходимо про-

вести повторную трансляцию текста (G2993), поскольку программа не будет соответствовать исправленному тексту. Трансляцию необходимо повторить и после стирания операторов Space, (,) - по той же причине.

Использование редактора. Весьма широкие возможности редактирования полностью записанного /содержащего знак "="/ текста предоставляет программа-редактор ^{16/}. Вызов ее осуществляется с помощью следующих команд монитора:

M2B10, 37FF, 10 (Ret)

GC77 (Ret)

Редактор модифицирован таким образом, что "читает" текст непосредственно из памяти. Длина текста не должна превышать 150 строчек, а среднее количество знаков не более 25 на строчку, поскольку редактор рассчитан на использование ОЗУ емкостью 8К. Все команды в модифицированном редакторе являются доступными, за исключением A и Z. Эти две команды возможны лишь при чтении с ленты, т.е. при обращении к редактору с помощью обычной команды монитора - G10.

После редактирования текста необходимо провести его трансляцию.

ЗАКЛЮЧЕНИЕ

Для специальных применений имеется дополнение к языку REST объемом ~1К, содержащее еще 4 подпрограммы. Первый оператор - H позволяет производить гистограммирование данных с произвольной станции КАМАК в выбираемой зоне памяти. Практический смысл введения этой операции состоит в значительном сокращении длительности цикла гистограммирования /~150 мкс/, по сравнению с тем же циклом, записанным на интерпретаторе. Второй оператор - D позволяет выводить содержимое зоны памяти на XY-осциллограф /через сдвоенный ЦАП/, что дает возможность наблюдения гистограмм, графиков функций и т.п. Два оператора - S и T служат для нахождения синуса и тангенса числа в BR. Последний оператор - Y является резервным /для включения наиболее употребимой подпрограммы пользователя/, он обеспечивает вызов по адресу ICCOH.

На рис. 4 приведен пример программы на языке REST, осуществляющей проверку памяти микро-ЭВМ с помощью шахматных кодов. Границы проверяемого участка памяти задаются с консоли, между записью и считыванием из проверяемых ячеек имеется цикл ожидания - для нейтрализации эффектов динамического запоминания информации на затворных емкостях МДП - транзисторов в ячейках памяти.

```

400--@
402A ' TEST MEMORY, MIN=';
404A ' MAX=';
406A ' OK ' ;
408A '
';
410A 408--XNO'-5555H';
412A 408--XNO'-AAAAH';
414A ;
416A 0 5 1--;(WAITING 20 MS)
400A
408--402--I404--I2(INPUT PARAMETERS OF CYCLE)
<21845N(5555H)--XW65535(ALL 1)416--XR&-410 414 410W(DELETE 5555H)
43690(AAAAH)--XW65535 416--XR&-412 414 412W(DELETE AAAAH)>406--=

```

ПРИМЕЧАНИЕ

ВЫВОД НА CONSUL - ЗНАК - СООТВЕТСТВУЕТ ↑ ,
 ЗНАК □ СООТВЕТСТВУЕТ ↘ .

Рис. 4. Пример программы на языке REST, производящей проверку ОЗУ микро-ЭВМ.

В заключение автор выражает благодарность Л.Реттельбушу за помощь в работе и В.А.Смирнову за полезные обсуждения.

ЛИТЕРАТУРА

1. Intel-8080 Assembly Language Programming Manual, Intel Corporation, Santa Clara, USA, 1976.
2. Немеш Т. ОИЯИ, 10-12106, Дубна, 1979.
3. Немеш Т. и др. ОИЯИ, 10-12077, Дубна, 1979.
4. Posa J.G. Electronics, 1979, No.2, p.105.
5. Hewlett-Packard Journal, USA, November, 1975, 2.
6. MCS-80 User Manual, Intel Corporation, Santa Clara, USA, 1976.

Рукопись поступила в издательский отдел
 26 декабря 1979 года.