EESTI NSV TEADUSTE AKADEEMIA TOIMETISED. XIV KÖIDE
FÜÜSIKA-MATEMAATIKA- JA TEHNIKATEADUSTE SEERIA. 1965, NR. 3

ИЗВЕСТИЯ АКАДЕМИИ НАУК ЭСТОНСКОЙ ССР. ТОМ XIV
СЕРИЯ ФИЗИКО-МАТЕМАТИЧЕСКИХ И ТЕХНИЧЕСКИХ НАУК. 1965, № 3

Х. САЛУМ

ЯЗЫК ДЛЯ ЗАПИСИ И МОДЕЛИРОВАНИЯ РАБОТЫ ЦИФРОВЫХ СТРУКТУРНЫХ СХЕМ (ЦИМОД)

В последнее время уделяется много внимания вопросам автоматизации проектирования цифровых вычислительных машин (ЦВМ), в частности созданию языков и методики моделирования работы как уже созданных, так и еще проектируемых схем [2–7]. Академик В. М. Глушков поставил задачу создания универсального языка для моделирования цифровых схем на имеющихся математических ЦВМ с тем, чтобы избежать необходимости при разработке каждой цифровой схемы создавать и новую моделирующую ее программу. Предлагаемый язык отличается от ранее известных [4, 5] рассмотрением произвольной цифровой схемы как микропрограммно управляемой [1] и ввиду этого может быть более широко использован.

Исходные положения

Произвольная цифровая структурная схема в *статическом режиме* характеризуется:

а) множеством *элементов памяти* $\{T\}$, организованным в одномерные упорядоченные подмножества $\{R\}$, называемые регистрами, и

б) потенциально возможными передачами (и преобразованиями при этом) информации между элементами $\{T_{ij}\}$, т. е. как между подмножествами $\{R\}$, так и внутри последних.

В динамическом режиме все эти потенциальные возможности передачи и преобразования информации реализуются в некотором предопределенном порядке. Сигналы, управляющие отдельными элементарными передачами и преобразованиями информации — *микрооперациями,* — называются *микрокомандами.* Некоторая последовательность микрокоманд образует *микропрограмму.* Интервал времени, в течение которого выполняется одна единственная микрокоманда или их совокупность, называется *микротактом.*

Статический режим структурной схемы записывается на предлагаемом языке ЦИМОД путем задания множества осуществимых в схеме микроопераций.

Динамический режим записывается системой сокращенных дизъюнктивных нормальных форм (с. д. н. ф.), определяющей условия выдачи узлом управления — *датчиком микрокоманд* (*ДМК*) — микрокоманды на выполнение каждой из микроопераций, осуществимой в схеме.

В ходе работы моделирующей программы можно подсчитать *характеристическое число регистра* — список связей регистра, время работы и простоя регистров и узлов, временную диаграмму занятости регистров и другие параметры.

Структура языка

Блок-схема структуры синтаксиса языка ЦИМОД приведена на рис. 1. Металингвистические переменные в схеме (заключенные в прямоугольники) определены до основных символов (заключенных в круги или ова-

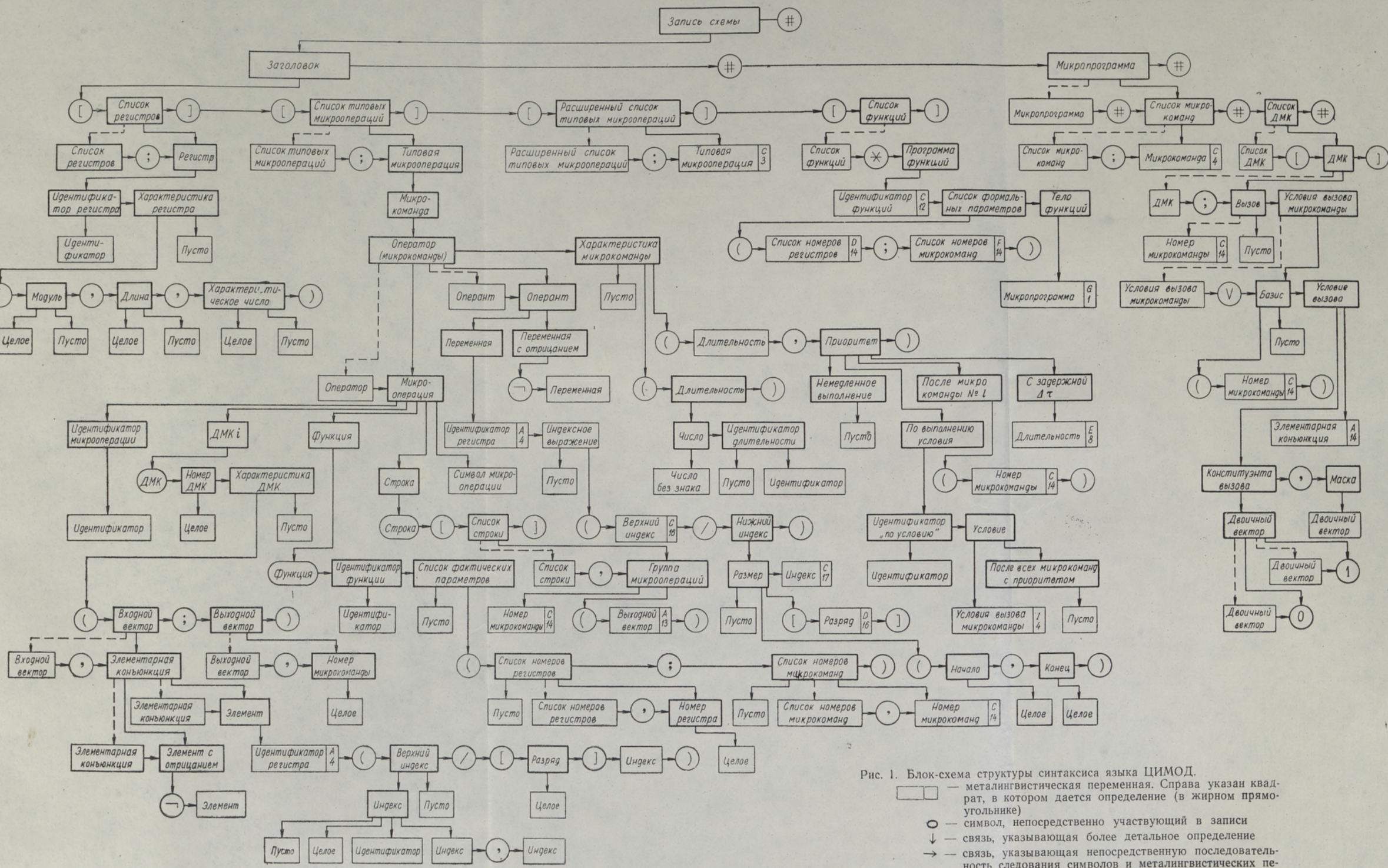


Рис. 1. Блок-схема структуры синтаксиса языка ЦИМОД.

— металингвистическая переменная. Справа указан квадрат, в котором дается определение (в жирном прямоугольнике)

○ — символ, непосредственно участвующий в записи

↓ — связь, указывающая более детальное определение

→ — связь, указывающая непосредственную последовательность следования символов и металингвистических переменных

лы) только в том случае, если они отличаются от принятых в АЛГОЛ-60 [9]. Большей частью же в качестве основных символов языка использованы символы АЛГОЛ-60, иногда в несколько измененном смысле. Металингвистические переменные, которые определяются непосредственно в том месте схемы, где они расположены, заключены в жирные прямоугольники; для остальных же в правом конце прямоугольника указан квадрат схемы, в котором дается определение (если оно расположено дальше соседнего квадрата). Вертикальной стрелкой в схеме соединены определения металингвистических переменных; горизонтальные стрелки указывают последовательность металингвистических переменных и основных символов, входящих в определение.

Основными символами языка ЦИМОД являются буквы, цифры и ограничители. В качестве *букв* допускаются полный латинский и русский алфавиты. Символ пустого знака (отсутствие символа) отдельной буквой или ограничителем не является, но допускается в записи. Следовательно, интервалы в записях идентификаторов, микрокоманд и т. п. не изменяют их смысла, но допускаются для большей наглядности записи. Например, идентификатор *ввод 10тичный* эквивалентен *ввод 10тичный*, эквивалентен *вв од10ти чный* и т. д. *Ограничителями* являются символы различных микроопераций, например, символы $0\rightarrow$ | $1\rightarrow$ | $\rightleftarrows$ | $\leftrightarrows$ | $\leftarrow 0$ | $\leftarrow 1$ | $\downarrow$ | $\uparrow$ | $\vee$ | $\wedge$ | $\neg$ | $+$ | $-$ | $\times$ |: (где | означает металингвистическое «или»), которые обозначают соответственно: сдвиг вправо с заполнением освободившихся разрядов регистра нулем или единицей соответственно; циклический сдвиг вправо и влево; сдвиг влево с заполнением 0 или 1; гашение (запись нуля); возбуждение (запись единицы); дизъюнкция; конъюнкция; инверсия; сложить; вычесть; умножить; разделить. В качестве ограничителей действуют также [| (|) |] | , | ; и некоторые другие символы. Списки букв и микроопераций могут быть произвольно расширены по мере надобности. В качестве логических значений разрешается использовать 1 и 0, что позволяет строить *двоичные векторы* (*1* 11). Символ $\vee$ применяется в двух различных значениях: при записи микрооперации как символ микрооперации типа дизъюнкция, а при записи ДМК — как разделитель. Ввиду непересечения областей применения распознавание значения символа $\vee$ трудностей не представляет.

Оперант есть переменная (с отрицанием или без него), обозначающая регистр (или часть его), участвующий в микрооперации. Все переменные языка суть типа Boolean, поэтому понятие типа используется в семантике для микрооперации. Целое и десятичное число используются только в значении «без знака».

Для максимального приближения записи операнта к привычной форме, его индексы разделены на нижний и верхний, причем оба могут быть составными. Верхний индекс используется для дополнительной характеристики регистра (его связей с другими регистрами и т. п.), а в нижнем индексе в самом начале выделено место, ограниченное скобками, для указания размеров той части регистра, которая участвует в микрооперации.* Например, запись $A(/(24,12)) \Rightarrow$ ЗУ (СР/(12,24)) обозначает микрокоманду «*записать содержимое разрядов с* 12 *по* 24 *регистра А в* ЗУ *по адресу в регистре* СР, *перевернув их порядок на обратный*». Для публи-

* В круглых скобках в нижнем индексе дается длина этой части регистра — номера ее начального и конечного разрядов. Если в круглых скобках дается только одно целое, то оно обозначает конец участка регистра с началом с первого разряда, т. е. длину участка. Одно целое в квадратных скобках обозначает конкретный разряд регистра, т. е. один элемент памяти. Отсутствие скобки в нижнем индексе показывает, что в микрооперации участвует весь регистр с обычной последовательностью разрядов.

кации допускается использование записи $A_{(24,12)} \Rightarrow ЗУ^{СР}_{(12,24)}$ вместо приведенной выше линейной формы. При этом в обозначении одного разряда регистра квадратные скобки могут быть опущены.

Во избежание неоднозначной дешифровки одинаково начерченных букв русского и латинского алфавита введено семантическое ограничение, запрещающее использовать при записи идентификаторов буквы одного алфавита непосредственно рядом с буквами другого, если между ними нет цифр.

Микропрограмма состоит из предложений двух типов:

1) описывающих микрокоманды и

2) описывающих условия выполнения отдельных микрокоманд.

Первые записываются в виде *списков микрокоманд*, вторые — в виде *списков датчиков микрокоманд* (*списков ДМК*). *Список микрокоманд* состоит из двух частей. Первая часть списка микрокоманд — *список типовых микроопераций* — состоит из списка микроопераций с формальными параметрами, а вторая — собственно список микрокоманд — есть список списков фактических параметров, засылаемых при обращении к процедурам. Список типовых микроопераций задается в *заголовке* записи схемы и изменяется только при необходимости добавления новых интерпретирующих подпрограмм, ранее не участвовавших в моделировании (например, при переходе к другой схеме или к другому уровню моделирования этой же схемы), или исключения уже не нужных подпрограмм. Практически этот список является заранее заданным и специально в микропрограмме не описывается. Там задается только собственно список микрокоманд, который и будет рассматриваться впредь, если специально не будет оговорено.

Список микрокоманд состоит из строк вида <*оператор микрокоманды*> <*характеристика микрокоманды*> (например, $B_{(13,24)} + C_{(36,25)} \Rightarrow C_{(25,36)}$ (5 *мксек*)).

Строки списка микрокоманд отделяются друг от друга точкой с запятой. Приведенная выше микрокоманда требует: «*сложить информацию, находящуюся в разрядах с* 13 *по* 24 *регистра B, с информацией в разрядах с* 25 *по* 36 *регистра C, предварительно изменив порядок последней на обратный, и записать полученный результат в разряды с 25 по 36 регистра C в том же порядке, как было в регистре «B»*. Микрокоманда выполняется сразу по поступлению, и ее длительность равна 5 *мксек*. Каждой микрокоманде в списке присваивается свой порядковый номер, начиная с единицы.

Записанный в характеристике микрокоманды *приоритет* обозначает дополнительные условия ее выполнения по отношению к другим микрокомандам, заданным к выполнению в этом же такте моделирования. Задание приоритета позволяет при составлении списка микрокоманд не учитывать взаимную связанность последних во времени.

ДМК, в общем случае, является *дешифратором* [3], входы которого отождествлены с некоторым подмножеством выходов элементов памяти, а каждый выход соответствует одной конкретной микрокоманде из списка микрокоманд. Возбуждение выхода соответствует необходимости осуществить соответствующую этому выходу микрокоманду. В физической схеме это соответствует подаче командного сигнала на соответствующую командную шину. *Выходной вектор* (*B* 13) — это *двоичный вектор* (*I* 11), каждый бит которого однозначно соответствует одной микрокоманде в списке микрокоманд. Выходной вектор может быть записан как последовательность номеров микрокоманд, отделенных друг от друга запятой. Выходные векторы всех ДМК, использующих один и тот же список микро-

команд, имеют одинаковую длину, т. е. эти генераторы микрокоманд как бы используют один и тот же комплект командных шин.

Каждая сокращенная дизъюнктивная нормальная форма (с.д.н.ф.) как принятая форма записи условий вызова микрокоманды соответствует возбуждению одной из командных шин на выходе ДМК. В случае записи с.д.н.ф. в виде *пар двоичных векторов* (*I* 10) нужно в записи ДМК по крайней мере один раз указать *базис* (*I* 6) — список соответствия входных щин ДМК выходам элементов памяти, — задаваемый в виде *элементарной конъюнкции* (*A* 14). Базис отделяется от условия вызова заключением в круглые скобки. Нули в двоичном векторе *маска* (*I*10) означают, что соответствующие переменные во входном векторе ДМК не учитываются при проверке логических условий — сравнении входного вектора с записью ДМК.*

Перед первым базисом условий вызова микрокоманды или на его месте может быть записан *вызов* (*I* 4) — заключенный в круглые скобки номер микрокоманды, которому соответствуют условия вызова. Отсутствие вызова означает, что условия соответствуют следующей по порядку микрокоманде. Неоднозначности распознавания не возникает, так как базис всегда является элементарной конъюнкцией, составленной из идентификаторов элементов памяти, а вызов — всегда целое. Явная запись вызова позволяет сократить запись ДМК за счет невыписывания тождественно нулевых элементов выходного вектора.

Длина регистра (*B* 6) в единицах исчисления указывает количество разрядов в нем. *Характеристическое число регистра* (*B* 6) получается путем присвоения каждой микрокоманде в расширенном списке типовых микрокоманд веса на основании степени 2 (т. е. вес микрокоманды i равен $p_i = 2^{i-1}$) и суммированием весов всех микрокоманд, вызывающих микрооперации с участием данного регистра. Характеристическое число и расширенный список типовых микрокоманд могут быть заданы в самом начале моделирования или же получены в результате его. В первом случае производится дополнительный контроль правильности записи.

Микрокоманды в языке ЦИМОД делятся на 4 типа:

1) микрокоманда на выполнение одной микрооперации (описана выше);

2) микрокоманды типа ДМК соответствуют переходу по метке в АЛГОЛ-60. Содержанием микрокоманды этого типа является проверка условий разветвления микропрограммы:

3) микрокоманды типа *строка* позволяют не записывать в виде различных тактов ДМК безусловные последовательности микрокоманд. Например, асинхронная микропрограмма на рис. 2 может быть записана как «строка [5, 3, (7, 8, 9), 4]», но микропрограмму на рис. 3, ввиду наличия в ней точек

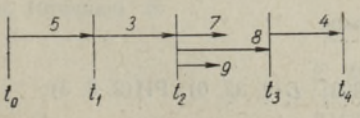


Рис. 2.

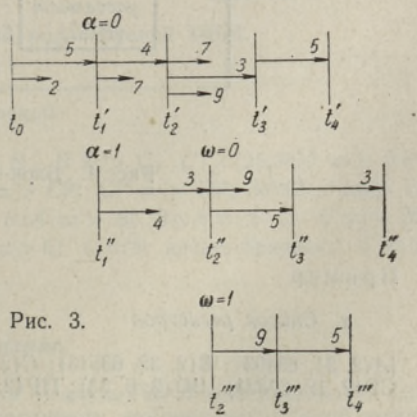


Рис. 3.

* Метод маски предложен автору С. Д. Михновским в частной беседе в 1963 г.

условного выбора (моменты условного времени $t_1' = t_1''$ и $t_2'' = t_2'''$), невозможно представить в виде одной строки, и для указанных контрольных точек [2] обязательно требуется запись условий вызова;

4) микрокоманды типа *функция* соответствуют процедурам в АЛГОЛ-60. В качестве *фактических параметров* (*D* 12) задается *список номеров регистров* (*D* 14) (перечень номеров последних), с которыми функция оперирует, и *список номеров микрокоманд* (*F* 14) (место функции в микропрограмме). По своей структуре функция соответствует *микропрограмме* (*G* 1). Это позволяет при многократном повторении одинаковых участков схемы (или микропрограммы) не повторять их записи.

В качестве примера будет рассмотрен поток информации в малой УЦВМ подобной М-3. Блок-схема моделируемой ЦВМ приведена на рис. 4, а таблица кодов операции на рис. 5. Для облегчения понимания примера курсивом даны дополнительные объяснения, выходящие за пределы стандартной записи на языке ЦИМОД.

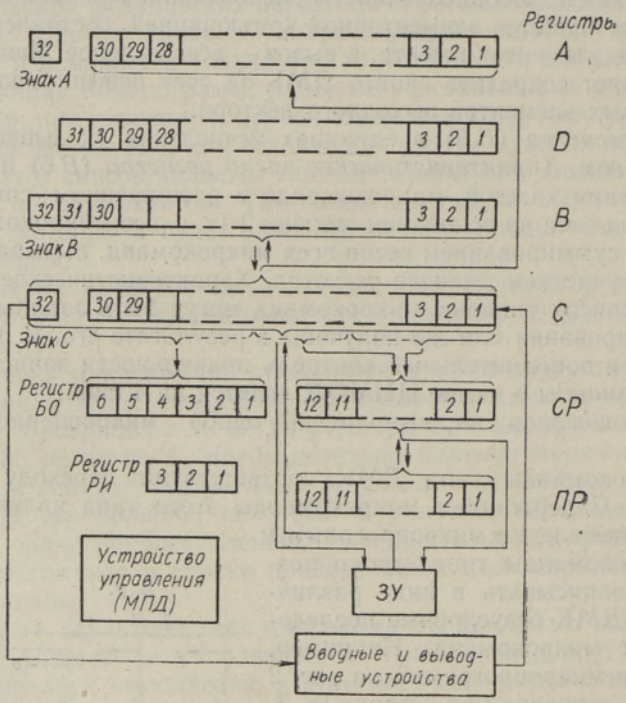


Рис. 4. Блок-схема моделируемой ЦВМ.

Пример

1. Заголовок

а. Список регистров

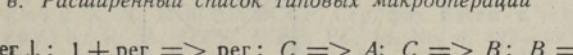
[*A*(2, 31, 63493); *B*(2, 32, 63513); *C*(2, 31, 2032893); *D*(2, 32, 0); РИ(2, 3, 3); СР(2, 12, 67521); БО(2, 6, 33); ПР(2, 12, 770)]

б. Список типовых микроопераций

[per ↓; 1 + per => per; per1 => per2]

I цифра КОПа / II цифра КОПа	...0	...1	...2	...3	...4	...5	...6	...7
0...	00 $'A_2+'A_1 \Rightarrow A_2$ $\Rightarrow B$	01 $'A_2-'A_1 \Rightarrow A_2$ $\Rightarrow B$	02 $'A_2:'A_1 \Rightarrow A_2$ $\Rightarrow B$	03 $'A_2\times'A_1 \Rightarrow A_2$ $\Rightarrow B$	04 Останов	05 $'A_1 \Rightarrow A_2$ $\Rightarrow B$	06 $'A_2 \wedge 'A_1 \Rightarrow A_2$ $\Rightarrow B$	07 Ввод 8ричный $\Rightarrow A_2$ $\Rightarrow B$
1...	10 $'A_2+'A_1 \Rightarrow B$	11 $'A_2-'A_1 \Rightarrow B$	12 $'A_2:'A_1 \Rightarrow B$	13 $'A_2\times'A_1 \Rightarrow B$	14 Останов	15 Останов	16 $'A_2 \wedge 'A_1 \Rightarrow B$	17 Останов
2...	20 $'B+'A_1 \Rightarrow A_2$ $\Rightarrow B$	21 $'B-'A_1 \Rightarrow A_2$ $\Rightarrow B$	22 $'B:'A_1 \Rightarrow A_2$ $\Rightarrow B$	23 $'B\times'A_1 \Rightarrow A_2$ $\Rightarrow B$	24 ПУ A_1 $'B \Rightarrow A_2$	25 $'B \Rightarrow A_2$	26 $'B \wedge 'A_1 \Rightarrow A_2$ $\Rightarrow B$	27 Ввод 10тичный $\Rightarrow A_2$ $\Rightarrow B$
3...	30 $'B+'A_1 \Rightarrow B$	31 $'B-'A_1 \Rightarrow B$	32 $'B:'A_1 \Rightarrow B$	33 $'B\times'A_1 \Rightarrow B$	34 УПУ по A_1 при $'B<0$ A_2 $'B\geqslant 0$	35 Останов	36 $'B \wedge 'A_1 \Rightarrow B$	37 Останов
4...	40 $'A_2+'A_1 \Rightarrow A_2$ $\Rightarrow B$ Печать 8ричная	41 $'A_2-'A_1 \Rightarrow A_2$ $\Rightarrow B$ Печать 8ричная	42 $'A_2:'A_1 \Rightarrow A_2$ $\Rightarrow B$ Печать 8ричная	43 $'A_2\times'A_1 \Rightarrow A_2$ $\Rightarrow B$ Печать 8ричная	44 Останов	45 $'A_1 \Rightarrow A_2$ $\Rightarrow B$ Печать 8ричная	46 $'A_2 \wedge 'A_1 \Rightarrow A_2$ $\Rightarrow B$ Печать 8ричная	47 $A_1 \Rightarrow$ Печать 10тичная
5...	50 $\lvert'A_2\rvert+\lvert'A_1\rvert \Rightarrow A_2$ $\Rightarrow B$	51 $\lvert'A_2\rvert-\lvert'A_1\rvert \Rightarrow A_2$ $\Rightarrow B$	52 $\lvert'A_2\rvert:\lvert'A_1\rvert \Rightarrow A_2$ $\Rightarrow B$	53 $\lvert'A_2\rvert\times\lvert'A_1\rvert \Rightarrow A_2$ $\Rightarrow B$	54 Останов	55 Останов	56 Останов	57 Останов
6...	60 $'B+'A_1 \Rightarrow A_2$ $\Rightarrow B$ Печать 8ричная	61 $'B-'A_1 \Rightarrow A_2$ $\Rightarrow B$ Печать 8ричная	62 $'B:'A_1 \Rightarrow A_2$ $\Rightarrow B$ Печать 8ричная	63 $'B\times'A_1 \Rightarrow A_2$ $\Rightarrow B$ Печать 8ричная	64 ПУ A_1 $'B \Rightarrow A_2$ Печать 8ричная	65 $'B \Rightarrow A_2$ $\Rightarrow B$ Печать 8ричная	66 $'B \wedge 'A_1 \Rightarrow A_2$ $\Rightarrow B$ Печать 8ричная	67 $B \Rightarrow$ Печать 10тичная
7...	70 $\lvert'B\rvert+\lvert'A_1\rvert \Rightarrow A_2$ $\Rightarrow B$	71 $\lvert'B\rvert-\lvert'A_1\rvert \Rightarrow A_2$ $\Rightarrow B$	72 $\lvert'B\rvert:\lvert'A_1\rvert \Rightarrow A_2$ $\Rightarrow B$	73 $\lvert'B\rvert\times\lvert'A_1\rvert \Rightarrow A_2$ $\Rightarrow B$	74 Останов .	75 Останов	76 Останов	77 Останов

Расположение команды в регистре С



A - № ячейки
$'A$ - содержимое ячейки №A
ПУ - (безусловная) передача управления
УПУ - условная передача управления

Рис. 5. Таблица кодов операций моделируемой ЦВМ.

в. Расширенный список типовых микроопераций

[рег $\downarrow$; $1+$рег $\Rightarrow$ рег; $C \Rightarrow A$; $C \Rightarrow B$; $B \Rightarrow C$; $C(/(25,30)) \Rightarrow$ БО; $C(/(13,24)) \Rightarrow$ СР; $C(/(1,12)) \Rightarrow$ СР; ПР $\Rightarrow$ СР; СР $\Rightarrow$ ПР; ЗУ(СР/)$\Rightarrow C$; $B+A \Rightarrow B$; $B-A \Rightarrow B$; $B\times A \Rightarrow B$; $B{:}A \Rightarrow B$; $B \wedge A \Rightarrow C$; $C \Rightarrow$ ЗУ СР/); ввод 8ричный $\Rightarrow C$; ввод 10тичный $\Rightarrow C$; $C \Rightarrow$ печать 8ричная; $C \Rightarrow$ печать 10тичная] #

2. Микропрограмма

а. Список микрокоманд для ДМК1 — ДМК4 — цикла РИ. В ходе работы РИ происходит выборка из ЗУ и расшифровка макрокоманды, выборка необходимой информации и расстановка ее в регистрах, подготовка ЦВМ к следующему такту работы и т. д. (рис. 6).

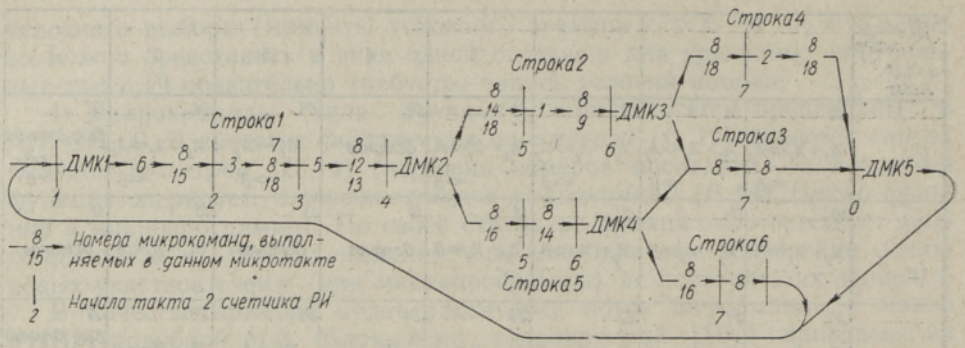


Рис. 6. Цикл работы РИ моделируемой ЦВМ.

$A\downarrow$; $B\downarrow$; $C\downarrow$; РИ$\downarrow$; БО$\downarrow$; СР$\downarrow$; $1+$ПР $\Rightarrow$ ПР; $1+$РИ $\Rightarrow$ РИ; $C \Rightarrow A$; $C \Rightarrow B$; $B \Rightarrow C$; $C(/(25{,}30)) \Rightarrow$ БО; $C(/(13{,}24)) \Rightarrow CP$; $C(/(1{,}12) \Rightarrow$ СР; ПР $\Rightarrow$ СР; СР $\Rightarrow$ ПР; стоп; ЗУ(СР/) $\Rightarrow C$; ДМК1; ДМК2; ДМК3; ДМК4; ДМК5: строка1 [6, (8, 15), 3, (8, 7, 18), 5, (13, 12, 8), 20]; строка2 [(8, 14, 18) 1, (8, 9), 21]; строка3 [8, 8, 23]; строка4 [(8, 18), 2, (8, 10), 23]; строка5 [(8, 16), (8, 14), 22]; строка 6 [(8, 16), 8, 19] #

б. ДМК1. Общая часть для всех макрокоманд. Выполняется всегда при обращении к ней.

[(24)]

в. ДМК2. Определяет является ли команда передачей управления.

[(25) $БО_1 \vee БО_2 \vee \neg БО_3$; (28) $\neg БО_1 \neg БО_2\ БО_3$]

г. ДМК3. Определяет необходимость чтения второго адреса.

[(27) $\neg БО_5$; (26) $БО_5$]

д. ДМК4. Определяет необходимость передачи второго адреса из регистра СР в регистр ПР.

[(26) $В_{32}$; (29) $\neg В_{32}$] #

е. Список микрокоманд ДМК5 — цикла работы МПД. Описывает работу АУ и обращение к внешним устройствам (рис. 7).

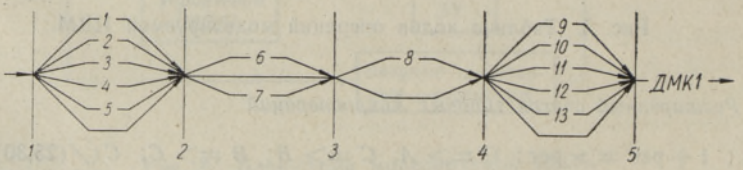


Рис. 7. Цикл работы МПД моделируемой ЦВМ.

$В+А \Rightarrow В$; $В-А \Rightarrow В$; $В \times А \Rightarrow В$; $В:А \Rightarrow В$; $В \wedge А \Rightarrow С$; $В \Rightarrow С$; $С \Rightarrow В$; $С \Rightarrow$ ЗУ(СР/); ввод 8ричный $\Rightarrow С$; ввод 10тичный $\Rightarrow С$; $С \Rightarrow$ вывод 8ричный; $С \Rightarrow$ вывод 10тичный; стоп; ДМК1 #

ж. ДМК5.

Такт 1. Выбор и проведение арифметической операции. Проверка на останов на каждом такте ДМК.

[1 ¬ БО$_1$ ¬ БО$_2$ ¬ БО$_3$; 1БО$_1$ ¬ БО$_2$ ¬ БО$_3$; 1БО$_1$БО$_2$ ¬ БО$_3$; 1 ¬ БО$_1$ БО$_2$ ¬ БО$_3$; 1 ¬ БО$_1$ БО$_2$ БО$_3$; (13) ¬ БО$_1$ ¬ БО$_2$БО$_3$ ¬ БО$_5$ ∨ ¬ БО$_1$ ¬ БО$_2$БО$_3$БО$_4$БО$_6$ ∨ БО$_1$БО$_3$БО$_4$;

Такт 2. Выбор направления дублирования результата операции.

(6)2 ¬ БО$_3$ ∨ 2 ¬ БО$_2$БО$_3$ ¬ БО$_4$БО$_5$ ∨ 2БО$_1$БО$_2$БО$_3$ ¬ БО$_4$БО$_5$БО$_6$; 2 ¬ БО$_1$БО$_2$БО$_3$ ∨ 2БО$_1$ ¬ БО$_2$БО$_3$ ¬ БО$_4$ ¬ БО$_5$ ∨ 2БО$_1$БО$_2$БО$_3$ ¬ БО$_4$БО$_6$ ∨ 2БО$_1$БО$_2$БО$_3$ ¬ БО$_4$ ¬ БО$_5$;

Такт 3. Запись результата в ЗУ.

(8)3 ¬ БО$_1$ ∨ 3БО$_6$;

Такт 4. Выбор внешних устройств.

(9)4БО$_1$БО$_2$БО$_3$ ¬ БО$_4$ ¬ БО$_5$ ¬ БО$_6$; 4БО$_1$БО$_2$БО$_3$ ¬ БО$_4$БО$_5$ ¬ БО$_6$; (12)4БО$_1$БО$_2$БО$_3$ ¬ БО$_4$ ¬ БО$_5$БО$_6$ ∨ 4БО$_1$БО$_2$БО$_3$ ¬ БО$_4$БО$_5$БО$_6$; (11) 4 ¬ БО$_3$ ¬ БО$_4$БО$_5$ ∨ 4 ¬ БО$_1$ БО$_2$ ¬ БО$_4$ ¬ БО$_5$БО$_6$ ∨ 4БО$_1$ ¬ БО$_2$ ¬ БО$_4$ ¬ БО$_5$БО$_6$ ∨ 4 ¬ БО$_2$ ¬ БО$_4$БО$_5$БО$_6$ ∨ ¬ БО$_1$ ¬ БО$_4$БО$_5$БО$_6$;

Такт 5. Переход к новому циклу РИ.

(14) 5] # #

Примечание. Практика показала целесообразность разрешить использовать в ДМК и строке вместо номеров микрокоманд их идентификаторы в таком виде, как они имеются в списке микрокоманд. Обе формы записи равноправны и могут быть использованы одновременно (но не в одной и той же микрокоманде или одном и том же вызове). Например, строка 3 [8, 8, 23] может быть записана так же, как строка 3 [1 + РИ = РИ; 1 + РИ ⇒ РИ, ДМК5].

Заключение

Проведен опыт моделирования записанного на языке ЦИМОД потока информации вышеприведенного примера. Интерпретирующая программа для М-20 содержала 2000 ячеек плюс 200 рабочих ячеек и констант. Запись структурной схемы ЦВМ занимала 240 ячеек. Время моделирования выполнения одной команды составляло в среднем 1,5 *сек.*

ЛИТЕРАТУРА

1. Глушков В. М., Введение в кибернетику, Киев, 1964.
2. Михновский С. Д., Цифровое моделирование при синтезе ЦВМ, Материалы семинаров по теоретическим вопросам кибернетики, Вопросы теории ЦВМ, вып. 5, Киев, 1963.
3. Салум Х. Л., Изв. АН ЭССР, Сер. физ.-матем. и техн. наук, № 4, 344—358 (1964).
4. Proctor R. M., Trans. IEEE, **EC. 13,** No. 4, 422—430 (1964).
5. Schläeppi H. P., Trans. IEEE, **EC. 13,** No. 4, 439—448 (1964).
6. Grasselli A., Automazione e. strumentazione, **11,** No. 3, 123—125 (1963).
7. Letellier G., Rev. franc. de traitement de l'information, Nr. 2, 113—123 (1963).
8. Comfort W. T., Highly parallel Machines, В кн.: 1962 workshop on computer organization, London, 1964.
9. Агеев М. И., Основы алгоритмического языка АЛГОЛ-60, М., 1964.

Институт кибернетики
Академии наук Эстонской ССР

Поступила в редакцию
12/III 1965

H. SALUM

NUMBRILISTE STRUKTUURSKEEMIDE KIRJELDAMISE JA MODELLEERIMISE KEEL

Artiklis antakse ülaltähendatud keele lühike semantiline kirjeldus. Süntaksit kirjeldatakse struktuurse blokkskeemi abil. Näitena on kirjeldatud arvutile M-3 lähedase skeemi töö.

H. SALUM

A LANGUAGE FOR DESCRIBING AND MODELLING DIGITAL STRUCTURAL SCHEMES

A short semantical description of the language is given. The syntax is described by a structural diagram. A description of a computer similar to the M-3 is given as an example.