
МОДЕЛИРОВАНИЕ ЦИФРОВЫХ СИСТЕМ

А.К. Олефир

I. Постановка задачи и общий метод её решения

В работе [1] было показано, что при разработке сложных цифровых систем, характеризующихся наличием большого числа взаимных связей между подсистемами, возникает ряд вопросов, которые невозможно решить чисто аналитическими методами. На различных уровнях синтеза необходимо точно знать, как согласуется работа отдельных подсистем. Если имеется разработанная схема системы, известны параметры элементов схемы и входные воздействия, задача проверки соответствия схемы её назначению решается имитацией информационных процессов, происходящих в системе. Эта имитация в данной работе называется прямым моделированием, так как моделирующая программа работает непосредственно по закодированной принципиально-монтажной схеме устройства.

К разработке системы прямого моделирования были предъявлены следующие основные требования:

1. обеспечить возможность наращивания элементов в моделируемой системе до 10 тысяч ;
2. в моделируемой системе учитывать наличие переходных процессов;
3. для устройств различной сложности обеспечить одинако-

во простое пользование системой моделирования.

Система прямого моделирования разработана с учетом реализации на ЭВМ. В её состав входит система кодирования и интерпретирующая программа с набором подпрограмм, соответствующих типам функциональных элементов.

Процесс моделирования состоит в вычислении состояния каждого элемента для последовательных дискретных моментов времени (тактов) с учетом входных воздействий и внутренних состояний элементов в предыдущем такте.

В настоящее время известен ряд работ по моделированию дискретных вычислительных устройств с помощью ЭВМ ([2] - [6]). В работах [2], [3] и др. описание схем производится на языке булевых функций, в ряде других работ (например, [4]) применяется язык типа АЛГОЛ. В работе [5] изложен принцип построения системы прямого моделирования. Во всех упомянутых работах переходные процессы не учитываются. В работе [6] описана система моделирования, учитывающая переходные процессы в элементах. Однако наличие входного языка в ряде случаев затрудняет применение этой системы.

В настоящей работе описана система прямого моделирования, учитывающая наличие переходных процессов в устройствах. Она характеризуется следующими особенностями: для моделирования не требуется знания функций, реализуемых устройством; не требуется промежуточного языка, так как интерпретирующая программа работает непосредственно с закодированной схемой; переходные процессы, связанные с передачей информации между элементами системы, могут быть представлены с любой необходимой точностью.

Рассмотрим общую структуру модели цифровой системы. Система разбивается на подсистемы, либо используется существующее разбиение системы на отдельные устройства (в состав каждого из них должно входить не более 200 элементов). Информация о подсистеме в виде перечня элементов и перечня связей между ними хранится во внешних запоминающих устройствах (ВЗУ) и называется блоком. Каждому блоку присваивается свой порядковый номер и каждому элементу в блоке также присваивается свой номер. Таким образом, номер элемента системы состоит из номера элемента в блоке (E_i) и номера блока (B_j). В оперативном запоминающем устройстве (ОЗУ) находится основная программа, которая последовательно вызывает в оперативную па-

мять блоки, вычисляет состояние каждого элемента и записывает блок на прежнее место. При этом каждый элемент блока отмечается новым состоянием ("единица", "ноль" или "переходной процесс").

Таким образом вычисляются состояния элементов каждого блока в каждом такте. Для связи между блоками используется таблица соединений блоков (ТСБ), в которой хранятся коды сигналов, действующих между блоками в данный момент времени. Необходимо отметить, что блоки являются равноправными, и система допускает наращивание, которое ограничено размерами памяти. Имеется лишь один нестандартный блок (нулевой блок - B_0), в котором хранится информация о входных воздействиях. После вызова этого блока основная программа не работает, а работает программа, которая находится в самом нулевом блоке. Эта программа (ИП₀) разносит входные воздействия по входам элементов системы, обращаясь не к самим элементам, находящимся в блоках, а к таблице соединений блоков. Это позволяет существенно сократить время, расходуемое на вызовы блоков из ВЗУ. Основная программа производит вычисления до тех пор, пока в нулевом блоке не будут исчерпаны все входные воздействия. После вычисления состояния системы для последнего такта, отмеченного в нулевом блоке, программа останавливается.

Рассмотрим моделирование отдельной подсистемы.

2. Моделирование подсистемы

Основная идея моделирования состоит в том, чтобы представить процесс функционирования подсистемы во времени в виде последовательности элементарных действий, выполняемых функциональными элементами подсистемы; функционирование элементов жестко связано с программным имитатором времени (счетчиком тактов), единица приращения которого - такт - является единой для всех элементов подсистемы (и системы). Работа каждого элемента моделируется отдельной подпрограммой. Подпрограмма (ПП) перерабатывает информацию, поступившую на входы элемента в начале такта, и выдает информацию в конце его.

Рассмотрим кодирование подсистемы. Обычно принципиально-монтажная схема цифрового устройства содержит информацию о типах элементов (с их порядковыми номерами), а также о соединениях между элементами.

В закодированном виде блок состоит из двух таблиц: таблицы элементов (ТЭ) и таблицы соединений элементов (ТСЭ).

ТЭ представляет собой перечисление элементов подсистемы; каждому элементу отводится одно машинное слово:

Разряды слова	0	I	2-6	7-18	19	20	21-27	28-39
Кодируемая информация	-	S	-	ρ	$d_{\text{вых}}$	ε	t_i	i

где i - номер элемента в подсистеме;

ρ - тип элемента;

ε - информация о состоянии элемента (устойчивое состояние - $\varepsilon = 0$, переходный процесс - $\varepsilon = 1$);

t_i - содержимое счетчика переходного процесса;

S - сигнал на выходе элемента;

$d_{\text{вых}}$ - отметка о принадлежности элемента к выходу.

Номера, соответствующие функциональным элементам, совпадают с номерами ячеек памяти машины, в которых размещена ТЭ. Этот приём даёт возможность кодировать соединения между элементами в виде связей между ячейками ОЗУ. При этом все входы в функциональный элемент являются равноправными.

Передача информации $E_{a_2} \rightarrow E_{a_1}$ в подсистеме кодируется следующим образом:

Разряды слова	0-6	7-18	19-27	28-39
Кодируемая информация	-	a_1	-	a_2

Массив таких слов составляет ТСЭ. На порядок слов в ТСЭ не накладывается никаких ограничений.

Входные воздействия задаются таблицей входных временных диаграмм (ВД). Каждое слово ВД содержит номер такта, при ко-

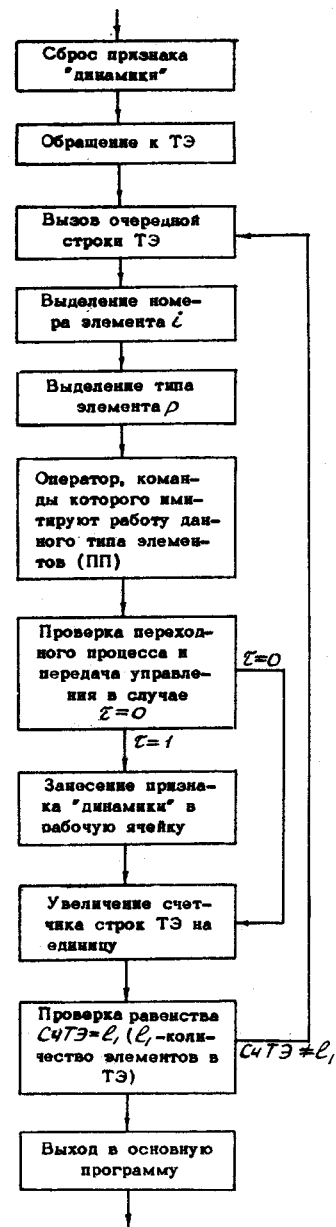


Рис. 1. Блок-схема интерпретирующей программы.

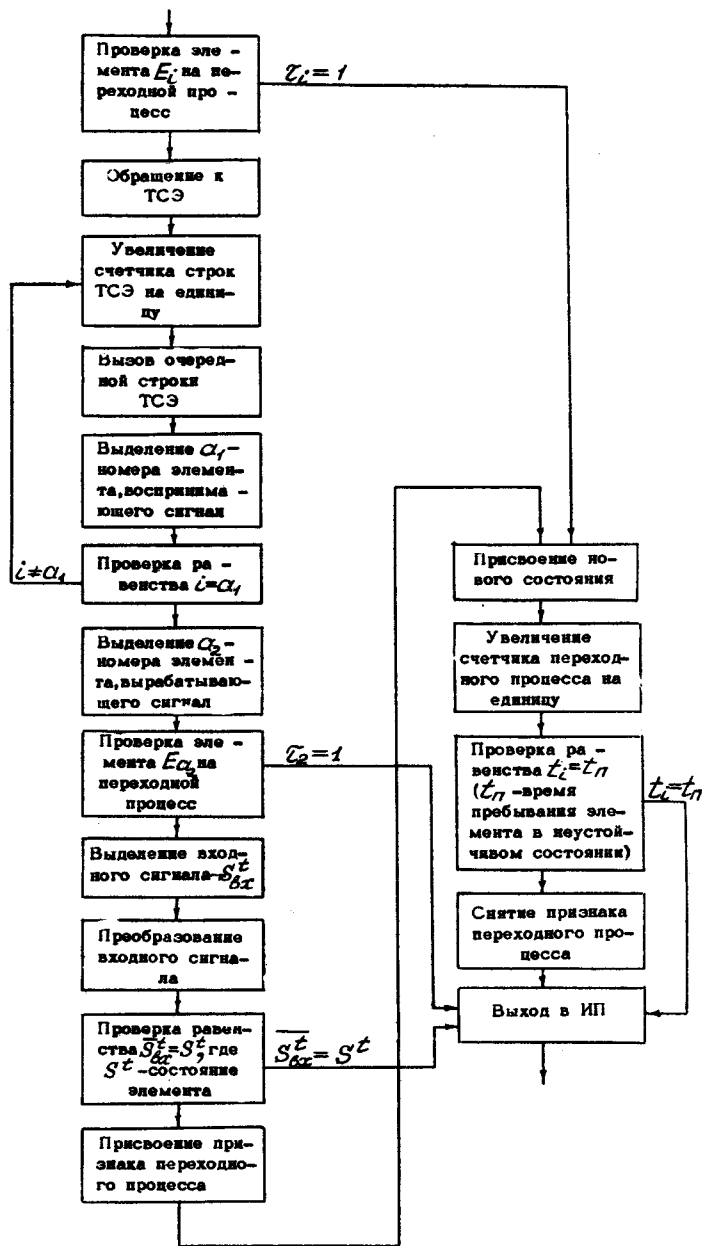


Рис. 2. Блок-схема подпрограммы инвертора.

тором происходит изменение входного сигнала t_{bx} , номер элемента, на входе которого происходит изменение сигнала α_{bx} , значение нового сигнала S_{bx} :

Разряды слова	0	I	2-6	7-18	19-27	28-39
Кодируемая информация	—	S_{bx}	—	t_{bx}	—	α_{bx}

В состав основной программы входит интерпретирующая программа (ИП), назначение которой — обработка блоков. Для сокращения времени моделирования вызов очередного блока из ВЗУ в данном такте производится в одном из двух случаев: либо при поступлении в данный блок нового сигнала (признак α'_s), либо при наличии признака "динамики" блока, т.е. при наличии в блоке хотя бы одного элемента, в котором $t_i \neq 0$ (признак α'_t).

Блок-схема ИП показана на рис. 1.

3. Программы моделирования функциональных элементов

По определению цифровой (дискретный) элемент может находиться в устойчивом состоянии (на выходе — фиксированный уровень, соответствующий состоянию "нуль" или "единица"), либо в состоянии переходного процесса. Поэтому входное воздействие можно рассматривать после того, как оно достигло фиксированного уровня. Подпрограммы элементов учитывают эту особенность цифровых элементов с помощью счетчиков тактов переходного процесса. При этом содержимое счетчика тактов переходного процесса, так же как и информация о состоянии элемента, хранится в строке ТЭ. Это позволяет представлять в виде подпрограмм функциональные элементы с одним выходом и одним или несколькими входами. При наличии более чем одного выхода (потенциальный триггер) элемент описывается несколькими строками ТЭ — по количеству выходов.

Ниже приводятся два примера моделирования функциональных элементов.

Подпрограмма инвертора. Инвертор реализует следующую функцию:

$$S^{t+t_n} = \overline{S^t_{bx}}$$

где S^t_{bx} - сигнал на входе инвертора в момент t ;
 S^{t+t_n} - сигнал на выходе инвертора после окончания переходного процесса.

На рис.2 показана блок-схема подпрограммы инвертора.

Подпрограмма потенциального триггера. Как известно, работа триггера описывается следующим образом:

$$S^{t+t_n} = \overline{S^t_{bx1}} \cdot \overline{S^t_{bx0}} \cdot S^t \vee S^t_{bx1} \cdot \overline{S^t_{bx0}};$$

$$S^t_{bx1} \cdot S^t_{bx0} = 0,$$

где S^t - состояние триггера в момент t ;
 S^{t+t_n} - состояние триггера после окончания переходного процесса;
 S^t_{bx1}, S^t_{bx0} - сигналы на входах триггера в момент t .

Блок-схема подпрограммы потенциального триггера приведена на рис.3.

4. Основная рабочая программа

В п. I. показано, каким образом осуществляется моделирование системы. Нетрудно видеть, что структура модели цифровой системы повторяет структуру модели подсистемы, но вместо таблицы элементов и таблицы соединений элементов здесь используется таблица блоков и таблица соединений блоков. Такая иерархическая структура позволяет расширять цифровую модель для систем более высокого порядка.

Как указывалось выше, блоки хранятся в ВЗУ, а в ОЗУ хранятся адреса и информация о блоках. Рассмотрим кодирование этой информации.

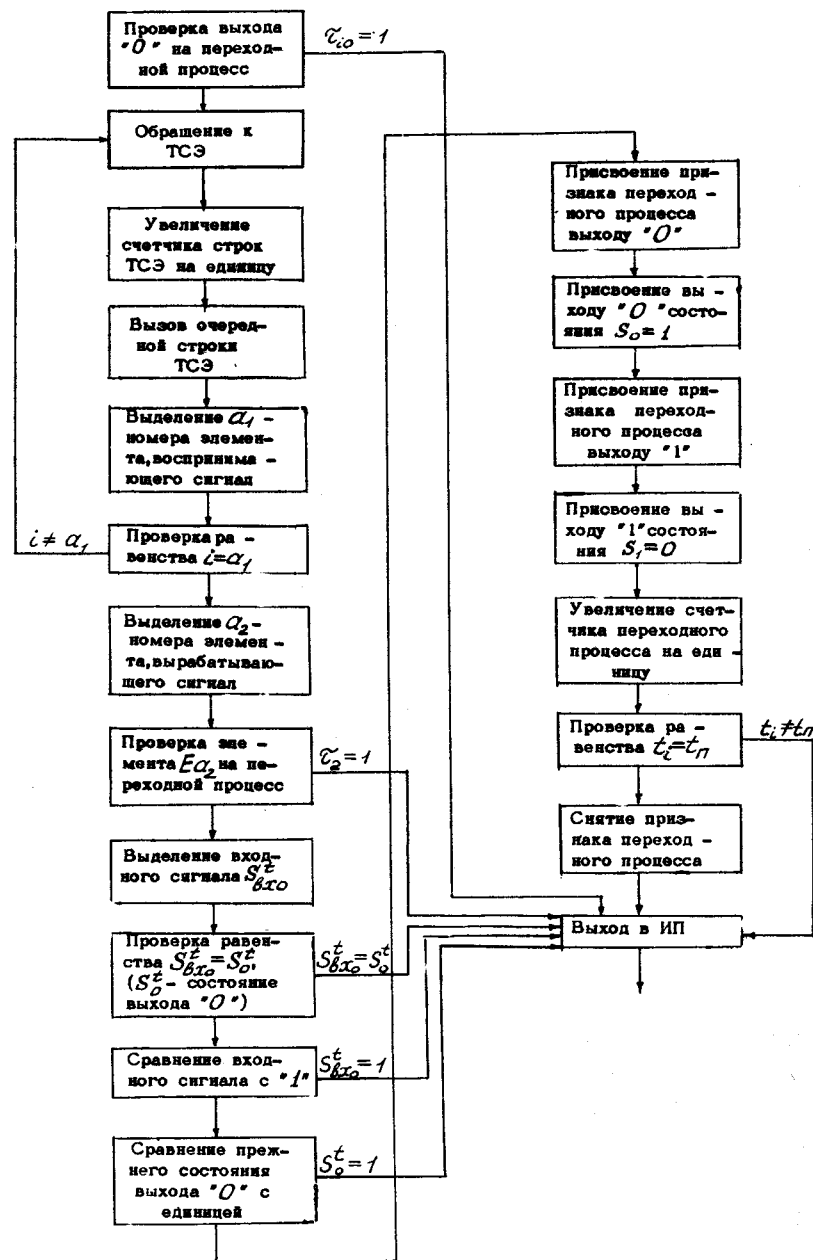


Рис. 3. Блок-схема подпрограммы потенциального триггера.

Таблица блоков (ТБ). Каждая строка ТБ содержит номер блока B_j и отметку о наличии "динамики" - α_s .

Таблица соединений блоков (ТСБ). В этой таблице перечисляются соединения, в каждое из которых входят номера связанных между собой элементов из разных блоков и информация, передаваемая между элементами. Передача информации S ($S=1$ или $S=0$) $E_i[B_j] \rightarrow E_i'[B_j']$ кодируется следующим образом:

Разряды слова	0-6	7-18	19	20	21-27	28-39
Кодируемая информация	B_j'	E_i'	S	α_s	B_j	E_i

Признак α_s служит для указания состояния элемента E_i в предыдущем такте. На рис. 4 приведена упрощенная блок-схема основной программы.

Примечание. I. Блок 28 включает в себя программы, с которыми взаимодействует моделирующая система. Назначение этих программ определяется той целью, с которой производится моделирование, они позволяют автоматизировать анализ проектируемых устройств.

Работа блока B_0 существенно отличается от работы других блоков. Назначение этого блока - перерабатывать таблицу входных сигналов в сигналы на входах блоков. Для этого используется ТСБ. После каждого вызова в ОЗУ блока B_0 производится присвоение новых входных значений элементам ТСБ в соответствии с закодированной временной диаграммой. В отличие от других блоков, вызов блока B_0 производится в каждом такте.

5. Результаты

Описанная система моделирования была разработана для ЭВМ "Урал-4" и опробована для различных схем ЭВМ. Интерпретирующая программа для отдельно взятой подсистемы занимает 110 ячеек оперативной памяти. Допускается до 20 типов элементов, до 200 элементов и до 300 изменений входных сигналов. Хотя в ЭВМ количество типов элементов достигает 50 и более, каждое от-

дельное устройство ЭВМ содержит элементы не более 20 типов.

Разработаны две программы, взаимодействующие с моделирующей системой:

1. Программа печати временных диаграмм. Эта программа взаимодействует с основной программой на каждом такте и позволяет получать с помощью алфавитно-цифрового печатающего устройства (АЦПУ) одновременно до 60 графиков выходных сигналов. На графиках указываются номера элементов в блоке, для которых печатаются временные диаграммы, номера начальных и конечных тактов для каждого интервала времени (через каждые 10 тактов программа печатает звездочку на оси времени). При необходимости печатаются входные временные диаграммы.

Печать временных диаграмм может осуществляться в различном временном масштабе.

2. Программа контроля. Эта программа позволяет установить правильность выполнения временных и логических соотношений моделируемых устройств. Исходной информацией для этой программы является таблица, в которой указывается номер такта, номер элемента и значение сигнала на выходе данного элемента в данном такте. Программа контроля сравнивает информацию о состоянии элементов с той информацией, которая получена в результате работы моделирующей программы. При несопадении программа контроля выдает на АЦПУ номер элемента, его состояние в данный момент и номер такта.

Отметим одну существенную особенность моделирующей системы. Как уже указывалось, программа последовательно обрабатывает блоки $B_0, B_1, B_2, \dots, B_M$, причем после работы каждого блока информация для других блоков заносится в ТСБ и ТБ. Эту программу легко реализовать на вычислительной системе, состоящей из N машин, где $N \geq M+1$. Каждая ЭВМ вычислительной системы должна обработать свой блок и в конце такта обменяться информацией с другими ЭВМ.

В выполнении данной работы принимали участие инженеры С.В. Заграничная, Г.П. Прищепа, А.Р. Усынина, за что автор выражает им свою благодарность.

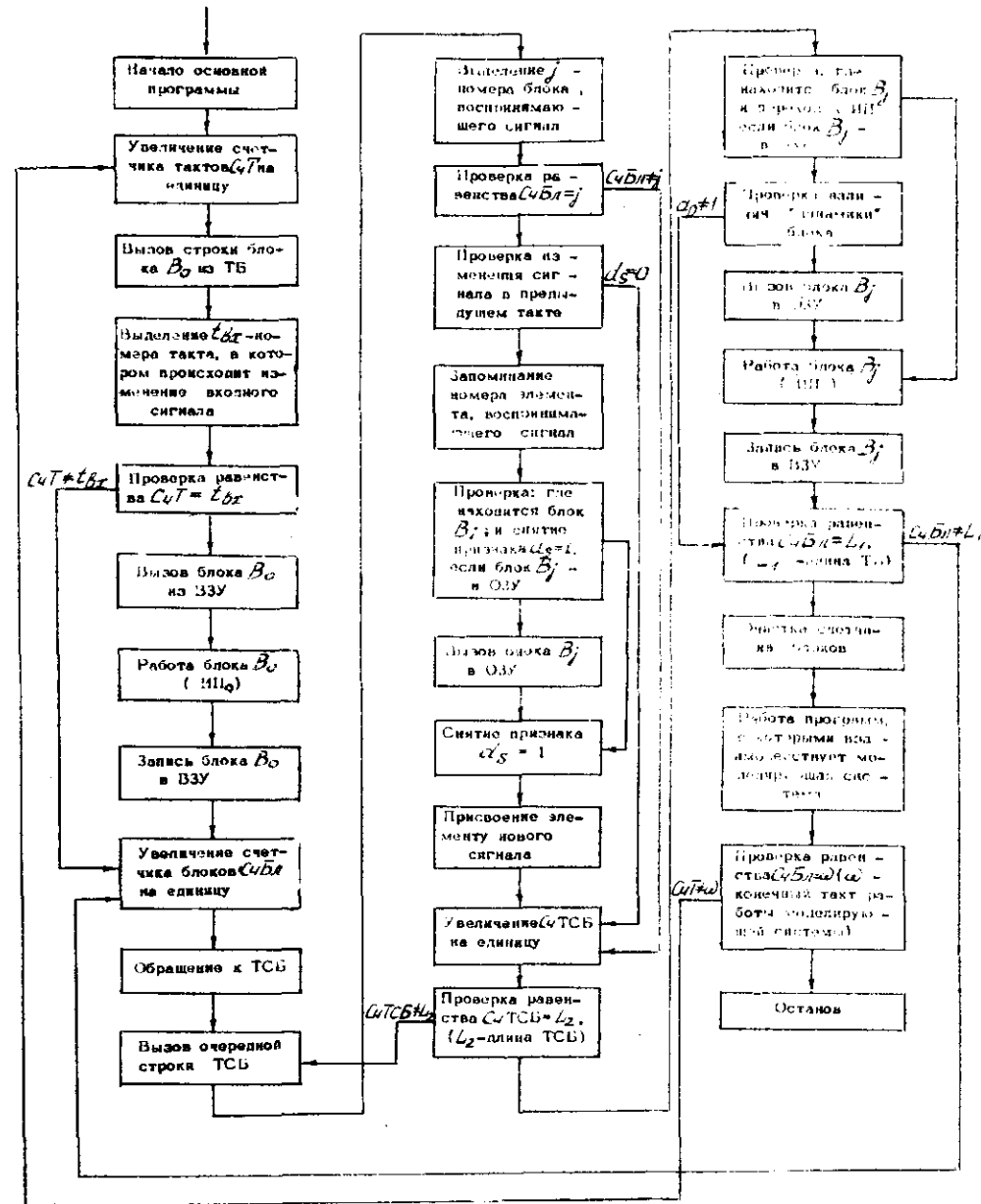


рис. 4. Блок-схема основной программы.

Л И Т Е Р А Т У Р А

1. Э.В. Евреинов, Ю.Г. Косарев. О методике разработки вычислительных систем. Новосибирск, -Вычислительные системы, 1963, выпуск 6, стр. 3 - 20.
2. G.N. Stockwell. Computer logic testing by simulation. - IRE Trans. Mil. Elect. 1962, Mil-6, N3, p.275 - 282.
3. M. Lehman, Rayna Wshed, Z. Netter. The checking of computer logic by simulation on a computer. - Computer Journ., 1963, 6, N2, p.154-162.
4. Y. Chu. An ALGOL-like computer design language. Commun. ACM, 1965, 8, N 10, p.607-615.
5. Н.М. Ермолаева. Машинный контроль схем устройств. - Вычислительная и информационная техника. Москва, ВИНТИ, 1962, стр. 123-124.
6. И.Я. Лендеу. Об автоматизации проектирования ЭЦВМ. - Автоматика и телемеханика, 1964, 25, № II, стр.1581-1587.

Вычислительный центр,
г. Красноярск

Поступила в редакцию
I.УП.1966 г.