

УДК 681.3.001

АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ УСТРОЙСТВ  
ЛОГИЧЕСКОГО УПРАВЛЕНИЯ В ОДНОМЕРНОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

А.А.Койфман, Г.Н.Кулиш, Л.И.Макаров, В.А.Скоробогатов

При проектировании и построении систем логического управления средствами автоматики в микроэлектронном исполнении возникает ряд трудностей:

- высокие требования к технологии изготовления элементов системы;

- для каждого нового проекта системы управления процедуру изготовления элементов приходится повторять заново;

- высокие требования к надежности системы заставляют вводить большой резерв, и поскольку структура системы управления заранее определена, то этот резерв необходимо вводить либо на уровне элемента, либо на уровне подсхемы, либо всей схемы системы управления.

К трудностям в проектировании систем управления описываемого класса можно также отнести невозможность физического моделирования такой системы. В процессе моделирования можно было бы легко менять её параметры без изменения структуры с целью выбора наилучшего режима функционирования. Решение таких задач для сложных систем путем моделирования на ЭВМ требует большого количества времени. Поэтому целесообразно иметь для целей моделирования универсальное устройство, которое построено по тому же принципу, что и сама система управления.

Применение вычислительной среды [1] для реализации систем управления позволяет в той или иной мере избежать указанных затруднений. Однородность среды позволяет упростить технологию изготовления элементов, универсальность и возможность настройки позволяют на одном устройстве реализовать класс систем. Ме-

тоды повышения надежности, известные для обычных структур, могут быть использованы в среде. Вычислительная среда может служить средством для моделирования систем управления с целью выбора необходимых режимов работы системы, уточнения алгоритмов управления.

Работа посвящена изложению основных результатов по созданию автоматизированной системы проектирования устройств логи- ческого управления комплексами средств автоматики объектов на основе одномерных вычислительных сред.

§1. Проектирование устройств логического  
управления в одномерной вычислительной среде

Устройство логического управления (УЛУ) является частью управляющей системы. Роль УЛУ состоит в переработке информации, поступающей на его входы от двоичных датчиков, и выработке последовательности сигналов, которые управляют исполнительными механизмами управляющей системы. Возможные параметры УЛУ следующие.

Число исполнительных механизмов, число входов, число работающих состояний не превосходит  $3 \cdot 10^3$ , число состояний входного вектора -  $7 \cdot 10^3$ . Для задания условий работы УЛУ применяется ориентированный язык, и для него соответственно разработаны правила кодирования системы логических уравнений.

Предполагается, что логика УЛУ не должна иметь большой глубины, что наряду с другими соображениями предопределяло выбор одномерной среды как средства реализации УЛУ. Поскольку УЛУ является частью системы автоматического управления, которая распределена в некотором пространстве, то решается задача распределения в том же пространстве и УЛУ. В работе дается методика решения задачи для одномерного случая.

Основные требования, предъявляемые к УЛУ, можно сформулировать следующим образом.

УЛУ должно работать в течение заданного времени  $T$  с вероятностью безотказной работы  $P$ . Оно должно быть распределено в пространстве объекта и иметь минимальную сложность реализации в одномерной вычислительной среде (ОВСр).

Выходными данными для проектирования УЛУ является описание функционирования этого устройства, которое задается системой логических уравнений. Правые части уравнений являются монотонными, неповторными булевыми функциями независимых переменных в д.н.ф. Предполагается, что среда состоит из коммутирующего блока (К) и логического блока (Л), состоящего из многофункциональных элементов (МФЭ) (рис. I.1). Блок К предназначается для создания требуемой схемы путем настройки.

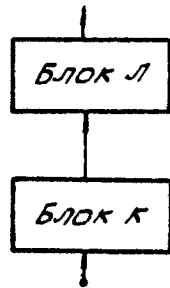


Рис. I.1

Предполагается, что любая переменная или её отрицание может быть подключена на любой вход любого элемента блока Л. Разветвление переменных может осуществляться вне коммутатора. Поэтому блок Л реализует монотонные и неповторные функции.

Для настройки может быть использован управляемый коммутирующий элемент, например, ПЭР [3,7].

Предполагается, что количество требуемых срабатываний ПЭР много меньше числа срабатываний, допускаемых техническими характеристиками. Необходимо, чтобы коммутатор мог осуществлять запоминание своего состояния после настройки и оставаться в этом состоянии при отключенном питании.

Число шин в коммутаторе, число входов и выходов в элементе среды могут быть различными. Поэтому можно говорить о классе сред, варианты которых могут различаться значениями указанных параметров. Будем также говорить, что среда однонаправленная, если сигнал распространяется только в одну сторону. Однонаправленность обеспечивается соответствующей схемой коммутатора. В работе рассматривается определенный вариант среды. Считается, что элементы блока Л — это интегральные схемы в стандартном корпусе, который имеет 14 выводов. Из них 2 идет на питание, два на выходы, два на управление и восемь на входы. Таким образом, МФЭ может выполнить набор функций алгебры логики от восьми переменных. Выходы элемента обеспечивают получение прямого и инвертированного значений функции. Управляющие входы позволяют подавать константы через блок К для настройки элемента на выполнение различных функций. Любой вход из оставшихся может быть использован для управления.

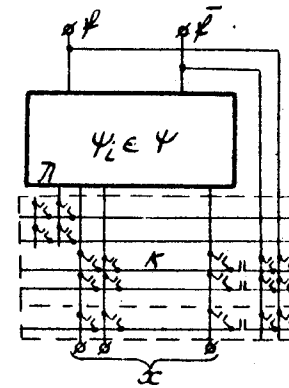


Рис. I.2

Функциональным базисом элемента среды будем называть набор  $\bar{\varphi} = \{\varphi_j\}$ ,  $j=1, k$ , функций  $\varphi_j$ , которые может выполнять МФЭ в различных состояниях настройки.

На рис. I.2 приведена схема элемента среды, состоящего из МФЭ и коммутатора. На рис. I.3 приведена схема соединения блоков Л и К.

Для обеспечения заданной надежности предлагается принцип группового резервирования в сочетании с непрерывным контролем внутри группы.

Для реализации непрерывного контроля использованы избыточность на уровне МФЭ и устройство, определяющее неисправность МФЭ элемента среды. Последнее в случае возникновения частичной неисправности, не приводящей к неправильному сигналу на выходе элемента, вырабатывает сигнал, отключающий этот элемент. Одновременно этот сигнал включает резервный элемент данной группы. В [4] приведена методика вычисления надежности группы и описание варианта группы с применением ПЭР. Схема устройства среды показана на рис. I.4.

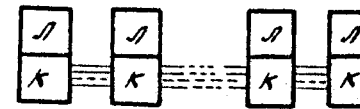


Рис. I.3

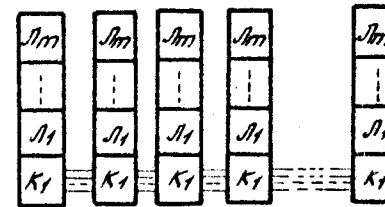


Рис. I.4

Таким образом, в классе сред можно выделить конкретный вариант среды, используя количественные значения следующих параметров:

- число коммутационных шин,
- число выходов и входов элемента,
- число и перечень функций базиса,
- кратность резервирования.

Эти параметры должны определяться так, чтобы реализация УЛУ на среде удовлетворяла исходным требованиям.

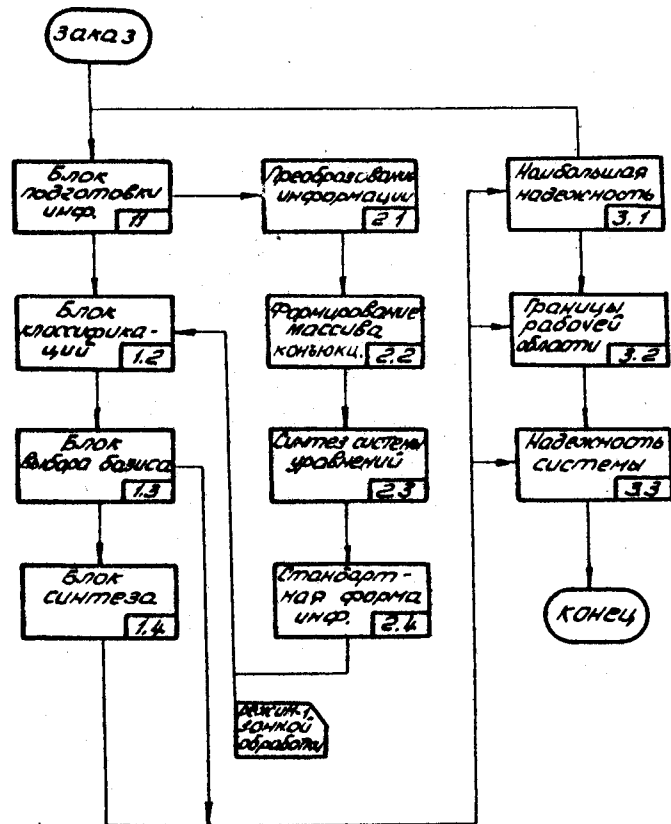


Рис. 1.5.

Система проектирования позволяет определять кратность резервирования, определять базис и программу настройки среды в условиях близкого к оптимальному распределения среды в прост-

ранстве объекта управления. В системе выделяются следующие комплексы:

- комплекс программ выбора базиса и синтеза функций в среде (§ 2),
  - комплекс программ рационального размещения логической сети УЛУ (§ 3),
  - комплекс программ обеспечения заданной надежности (§ 4).
- Общая блок-схема системы приводится на рис. 1.

## § 2. Выбор базиса и синтеза функций в среде

Комплекс программ выбора базиса и синтеза функций в среде состоит из следующих блоков:

- подготовки исходной информации,
- классификации,
- выбора базиса,
- синтеза функций в среде.

Параметры программ сведены в табл. 2.1 и 2.2.

Комплекс предназначен для выбора оптимального набора функций (базиса), которые должен выполнять элемент среды, и для автоматического получения программы настройки среды. Комплекс разработан на основе методов, изложенных в [5,6].

Блок подготовки исходной информации (рис.2.1) работает с системой логических уравнений (СЛУ), заданных в скобочной форме. СЛУ кодируется и записывается на магнитную ленту.

Программа 1.1.1 позволяет исправить ошибки, допущенные при кодировании и перфорации путем выдачи системы  $F$  на АЦПУ и сравнения её с исходной системой уравнений.

Программа 1.1.2 обеспечивает разбиение системы на массивы определенных размеров и формирование их на магнитной ленте в виде блоков, откуда эта информация может быть вызвана любым из комплексов для необходимости обработки.

Программа 1.1.3 раскрывает скобки в системе  $F$ , при этом подфункции в скобках заменяются одной и той же буквой с различными индексами. В результате работы на магнитной ленте получается последовательность блоков функций. С этой информацией работает блок синтеза функций.

Таблица 2.1

Номер блока	Номер программы	Название программы	Время граф-схемы	Счет одного блока (мин)	Счет осний (мин)	Объем программы	Примечание
1	1.1	Расчетка		10		250 <sub>в</sub>	
	1.2	Формирование массивов		0,4		250 <sub>в</sub>	
	1.3	Раскрытие скобок	12	2,3	2,3 <sub>к</sub>	9 лист.	
	1.4	Система символов	7,5	0,15 <sub>п</sub>	0,15 <sub>пс</sub>	4 лист.	$C$ - константа, зависящая от числа обращений к ИМЛ, $п$ - число функций.
2	2.1	Разбиение на классы	II		$H(3 \times N + N^2 + C)$	7 лист.	$H$ - число частей в системе; $N$ - число классов; $N$ - число элементов в одном блоке; $\beta = 0,004$ ; $C$ - время считывания с ИМЛ одного блока.
	2.2	Выбор представителей	10		$2H(\alpha p N + C)$	6 лист.	$\alpha - 0,001$ ; $C - 0,03$ ; $p$ - число классов; $N$ - число элементов в блоке; $H$ - количество блоков.
	2.3	Список и система символов представителей	7		$\alpha_1 H + \alpha_2 V$	4 лист.	$H$ - количество блоков; $V$ - количество представителей; $\alpha_1 - 0,06$ ; $\alpha_2 - 0,04$ .
	2.4	Выход системы символов				100 <sub>в</sub>	
3	3.1	Выбор наборы функций	30		$\alpha \times AC \times RC \times SC$	34 лист.	$\alpha$ - среднее время вычисления сложности реализации одной функции; $\alpha < 0,001$ для функций, выходящих на более шести конъюнкций.
	3.2	Оценка сложности реализации					
	3.3	Генератор случайных чисел					

Таблица 2.2

4	4.1	Выбор длины сдвига	15		4 <sup>в</sup>	16 лист.	и) Время в минутах приведено для примера из (8) $L=4$ - макс. число векторов реализации; $L=240$ - утроенное макс. число переменных в функции; $Y=2$ - макс. число длины конъюнкций в функции; $C=4$ - число функций в базе; $M=7$ - макс. ранг конъюнкций в базе; $SH=8$ - число входов в элемент; $LZ=10$ - макс. число конъюнкций функции; $K'=5$ - число синтезируемых функций.
	4.2	Синтез	25		2 <sup>в</sup>	21 лист.	

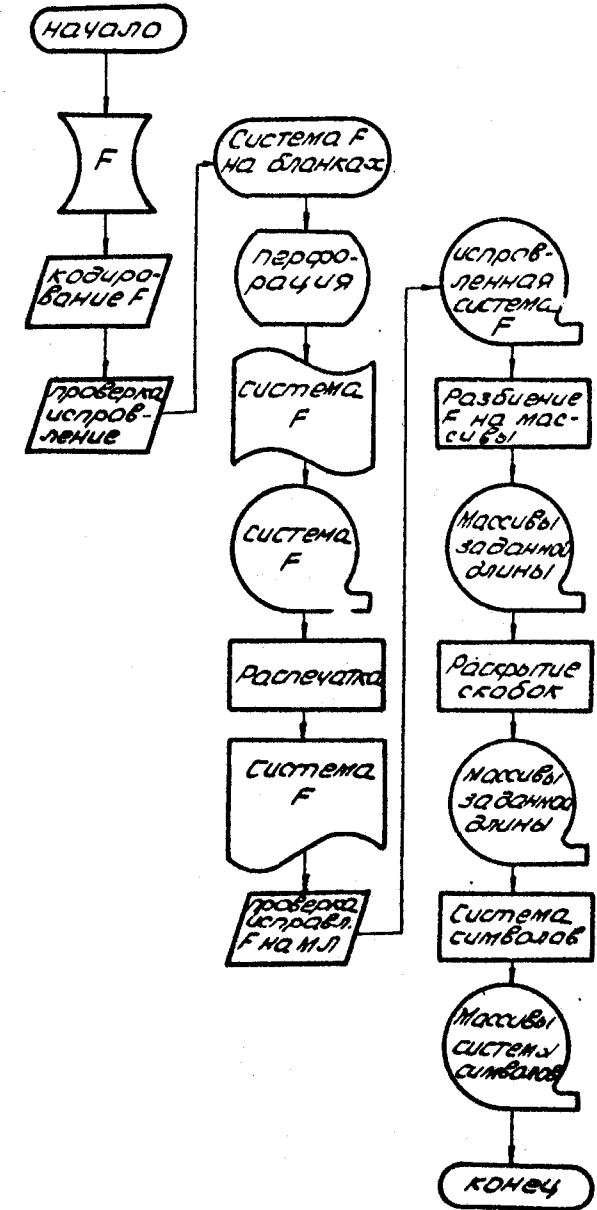


Рис. 2.1 Блок-схема блока подготовки исходной информации

Программа I.1.4 перерабатывает множество функций с исключенными скобками во взвешенную систему символов  $S$ , которая записывается на ленту массивами заданной длины. При этом не учитываются функции, символ которых тождественно равен единице. Каждая конъюнкция заменяется её рангом, символы разных функций отделяются разделительными символами.

Система  $S$  используется в качестве исходной информации для нахождения базиса элемента среды.

Блок классификации (рис.2.2) работает с системой символов  $S$  и состоит из следующих программ.

Программа I.2.1 разбивает множество функций  $F$  на классы [5]. Степень детальности разбиения  $F$  на классы задается рядом параметров, которые могут меняться по желанию разработчика.

Программа I.2.2 осуществляет выбор представителей классов и вычисление суммарной сложности классов.

Для вычисления сложности реализации одного класса используется формула

$$L^j = \sum_{i=1}^p a_i L_{\phi}$$

где  $L^j$  - сложность реализации  $j$ -го класса,

$L_{\phi}$  - сложность реализации фиксированной функции  $\phi_0$  из класса,

$p$  - количество функций в классе,

$a_i$  - коэффициент, характеризующий отличие  $i$ -й функции от функции  $\phi_0$ .

Сложность всего множества функций  $F$ , разбитого на  $k$  классов, равна

$$L(F) = \sum_{j=1}^k L^j$$

Таким образом, вычисление  $L(F)$  сводится к определению сложности реализации  $k$  функций. При этом число  $k$  можно менять в зависимости от желаемой точности и наличия машинного времени. В качестве представителя выбирается функция, сложность которой близка к среднему арифметическому сложностей функций класса. Представители классов нумеруются определенным образом и упаковываются в массив.

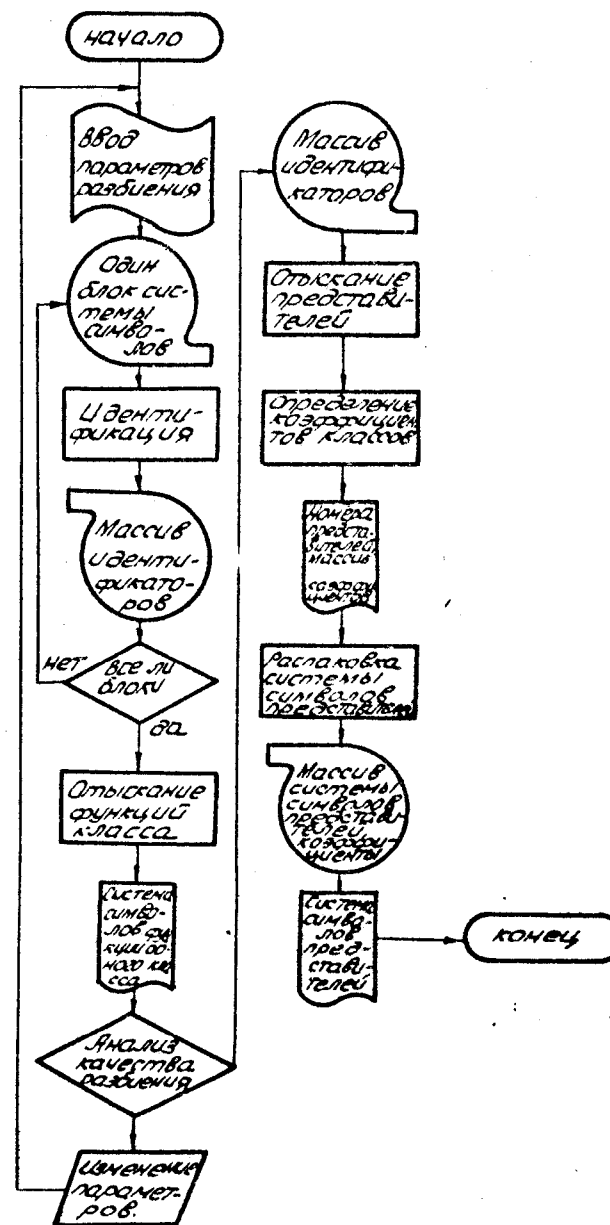


Рис. 2.2 Блок-схема блока классификации

Программа 1.2.3 по номерам представителей оценивает системы символов функций-представителей, распаковывает и записывает ранги конъюнкций в один массив, а число конъюнкций данного ранга-в другой.

Программа 1.2.4 используется после классификации функций исходной системы, т.е., когда для каждой функции, представленной в виде системы символов, построен код, определяющий принадлежность её к некоторому классу. Находятся группы совпадающих кодов, и выводятся на печать системы символов, соответствующих этим кодам. Вывод на печать необходим для оценки качества классификации.

Блок выбора базиса (рис.2.3) выбирает базис, близкий к оптимальному. Он состоит из трех программ 1.3.1, 1.3.2, 1.3.3, реализующих метод направленного поиска с адаптацией.

Первая программа осуществляет выбор базиса, вторая вычисляет сложность реализации множества функций при заданном базисе и третья - генерирует случайные числа, равномерно распределенные в интервале  $(0,1)$ .

Осуществляется выбор заданного числа функций базиса из некоторой генеральной совокупности, имеющей  $N$  функций. Выбор функций базиса управляется вектором  $P$  вероятностей выбора. Вначале вероятности выбора всех  $N$  функций считаются равными. В соответствии с вероятностями при помощи программы 1.3.3. выбирается фиксированное число функций базиса. Вычисляется оценка сложности реализации  $F$  при таком базисе. Выбор проводится определенное число раз. Вероятности выбора функций, вошедших в набор и давших наибольшую оценку, уменьшаются, а давших наименьшую оценку, увеличиваются на одну и ту же величину.

Начиная с некоторого минимального значения, вероятности остаются неизменными. Изменение вектора  $P$  с последующей оценкой повторяется заданное число раз. Выбирается базис, давший минимальную оценку сложности реализации множества  $F$ . Имеется возможность принудительного включения в базис любых функций.

Блок синтеза функций в среде (рис.2.4) состоит из двух программ 1.4.1 и 1.4.2. Блок позволяет для заданной системы уравнений  $F$  и выбранного базиса получить программу настройки вычислительной среды.

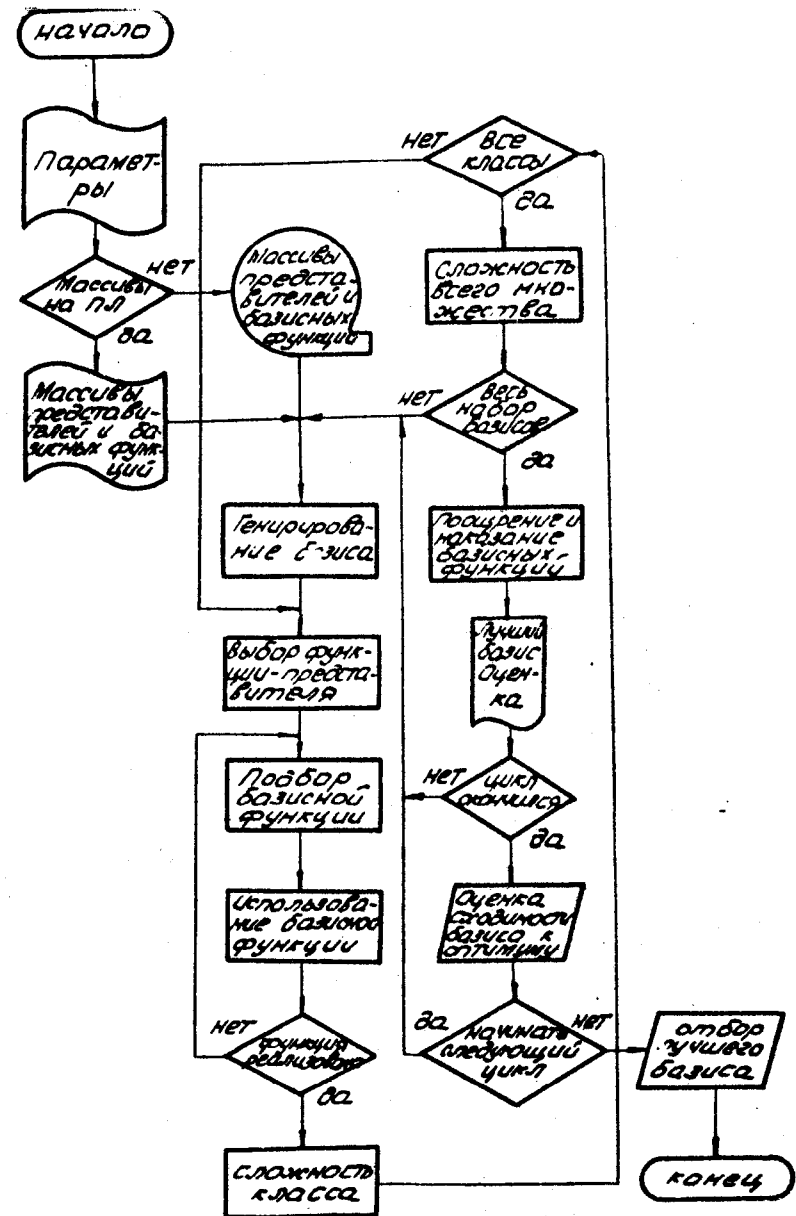


Рис. 2.3 Блок-схема выбора базиса

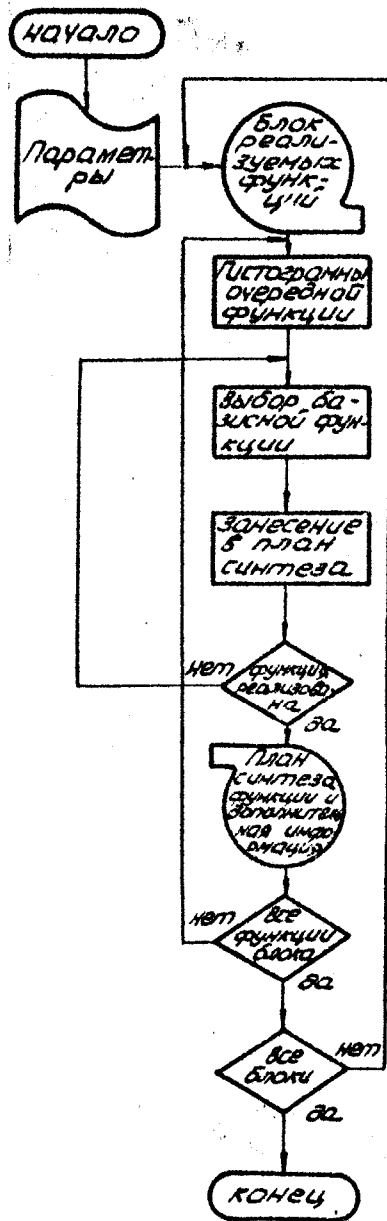


Рис. 2.4

Программа I.4.I находит план синтеза для каждой из функций  $f \in F$  и записывает их на магнитную ленту. Построение плана производится на основе процедуры наискорейшего спуска.

Строится гистограмма конъюнкций. По ней строятся две приведенные гистограммы: одна — для "коротких" конъюнкций, вторая — для остатков "длинных". Ранги длинных конъюнкций и их остатков запоминаются как два массива  $A_1$  и  $A_2$ . Одновременно в план синтеза вносятся векторы, соответствующие реализации длинных конъюнкций (без остатков). Основным шагом алгоритма состоит в следующем. В  $A_1$  и  $A_2$  выбирается конъюнкция максимального ранга, и в базисе подбираются все функции, способные реализовать данную конъюнкцию. Для каждой функции базиса находится набор конъюнкций, которые можно реализовать этой функцией, так, что максимизируется значение  $D$

$$D = \sum_{i=1}^g x_i \cdot i - \frac{m}{k} x_0,$$

где  $x_i$  — число конъюнкций ранга  $i$ , реализованных на элементе среды;  $m$  — максимальный ранг конъюнкции в

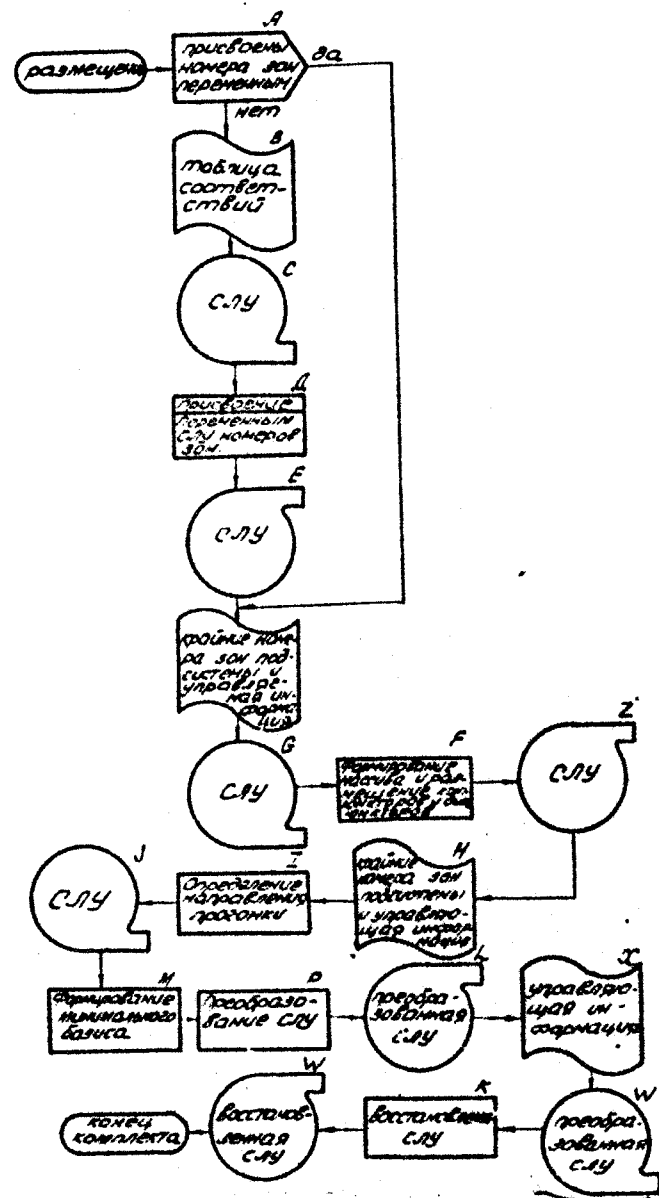


Рис. 3.1 Блок-схема комплекса рационального размещения

базисе;  $k$  — число коммутационных шин;  $x_0$  — число дизъюнктивных входов, реализованных на элементе при данной функции.

Среди всех рассмотренных базисных функций выбирается та, для которой значение  $D$  максимально. Сформированный вектор заносится в план синтеза. После этого корректируются гистограммы с учетом того, что часть функций уже реализована: некоторые компоненты уменьшаются. Следующий шаг начинается с выбора в гистограммах  $A_1$  и  $A_2$  конъюнкций максимального ранга. Процесс построения плана заканчивается, когда в гистограммах все компоненты равны нулю.

Программа 1.4.2 производит построение программы настройки среды в соответствии с полученным планом и выдает результат на печать. Программа имеет ограничения. Она предназначена для среды с двумя шинами. При изменении параметров среды ряд процедур программы должен быть заменен аналогичными, учитывающими специфику элемента.

В [6] приведен пример реализации функции на ОВСр, полученной при помощи блока синтеза.

### § 3. Рациональное размещение сети

Комплекс программ рационального размещения сети в пространстве объекта состоит из четырех программ (рис.3.1).

Программа 2.1 присвоения переменных уравнений из СЛУ координат пространства объекта (блоки  $A, B, C, D, E$ ).

Программа 2.2 формирования массива конъюнкций и дизъюнкций (блоки  $F, G$ ).

Программа 2.3 синтеза системы уравнений в пространстве объекта (блоки  $H, I, J, M, P, L$ ).

Программа 2.4 приведение новой СЛУ к стандартной форме (блок  $K$ ).

Характеристики программ сведены в таблицу 3.1.

Модель пространства есть прямоугольник, разбитый на 3 зон (рис.3.2). Номера зон названы координатами пространства объекта.

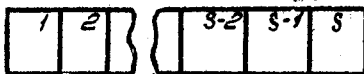


Рис.3.2

В [8], например, описаны задачи разрезания (разбиения) графа на подграфы с наименьшим суммарным числом ребер, связы-

Таблица 3.1

Номер программы	Время тран-сляции (мин)	Счет од-ного блока (мин)	Счет общий (мин)	Объем программы
2.1	7	(0 - 50)		2 лист.
2.2	14	15		10 лист.
2.3	35	2 - 30*		20 лист.
2.4	5-7	15		5 лист.

\*)  $\mu$  — число различных координат, для которых производится расчет.

вающих подграфы. Однако при автоматизации проектирования схем бывает выгоднее процедуру разбиения осуществить на уровне описания схемы, например, системных булевых уравнений. В этом случае система уравнений должна быть разбита на подсистемы так, чтобы соответствующие сети имели наименьшее суммарное число связывающих линий. Этот подход используется в данной системе проектирования.

Входной формой служит массив системы уравнений, сформированный в блоке подготовки информации.

Комплекс имеет два режима работы. В первом режиме во входном массиве переменные не имеют соответствующих им координат зон. Процедуру присваивания переменным координат зон выполняет программа 2.1. Во втором режиме массив входных переменных задается с соответствующими координатами и работа начинается с программы 2.2.

Программа 2.2 предназначена для формирования массива функций и фиксации некоторых элементов сети в определенных зонах. Для функции  $f_i \in F$  осуществляется замена каждой конъюнкции новой переменной, формируется массив с функциями от новых переменных и таблица соответствия переменных конъюнкций. Определяются интервалы координат, в которых сосредоточены наборы переменных для каждой функции и для каждой конъюнкции.

Производится присвоение выходной переменной  $f_i$  координаты зон (где имеются  $x$  переменных функции  $f_i$ ), ближайшей к координате соответствующего выходного полюса сети. В той же зоне фиксируются конъюнкции, зависящие только от переменных данной зоны.



Для данной программы 2.3 необходим ввод о ПФЛ граничных номеров зон расположения переменных. В основу программы положена процедура формирования и перемещения полюсов сети из зоны в зону. Начальное направление перемещения выбирается произвольно, а перемещение начинается с крайней зоны. Выбирается множество новых переменных, имеющих выходы в соседние зоны. Находится минимальный базис, т.е. минимальное множество конъюнкций, из которых могут быть образованы все конъюнкции данной зоны. Элементы базиса фиксируются в пределах данной зоны. При этом очевидно, что число линий, идущих из зоны в зону, равно числу элементов минимального базиса. После этого формируется набор входных полюсов из полюсов независимых переменных новой зоны и из полюсов элементов минимального базиса предыдущей зоны. Для новой зоны также находится минимальный базис. Процедура выполняется для всех зон в прямом и обратном направлениях.

После окончания процедуры подсчитывается и выдается на печать суммарное число линий связи между зонами.

В результате получается логическая сеть, эквивалентная исходной.

Программа 2.4 приводит массив полученных новых функций к стандартному виду и записывается на МЛ. Эта информация может быть использована как исходная для остальных комплексов системы.

#### § 4. Надежность

Комплекс программ предназначен для вычисления значений надежности системы при заданном диапазоне значений надежностных характеристик компонент выбранной схемы элемента и группы. Он позволяет получать общую картину распределения надежности системы, если выбрана схема элемента, и существенно облегчает процесс оценки степени пригодности тех или иных компонент для использования их в проектируемом устройстве.

Комплекс состоит из трех программ (рис.4.1), разработанных на основании методики, изложенной в [4].

Программа 3.1 находит наибольшую требуемую надежность элемента ОВСр ([4] п.3).

Программа 3.2 находит границу рабочей области ([4] п.4).

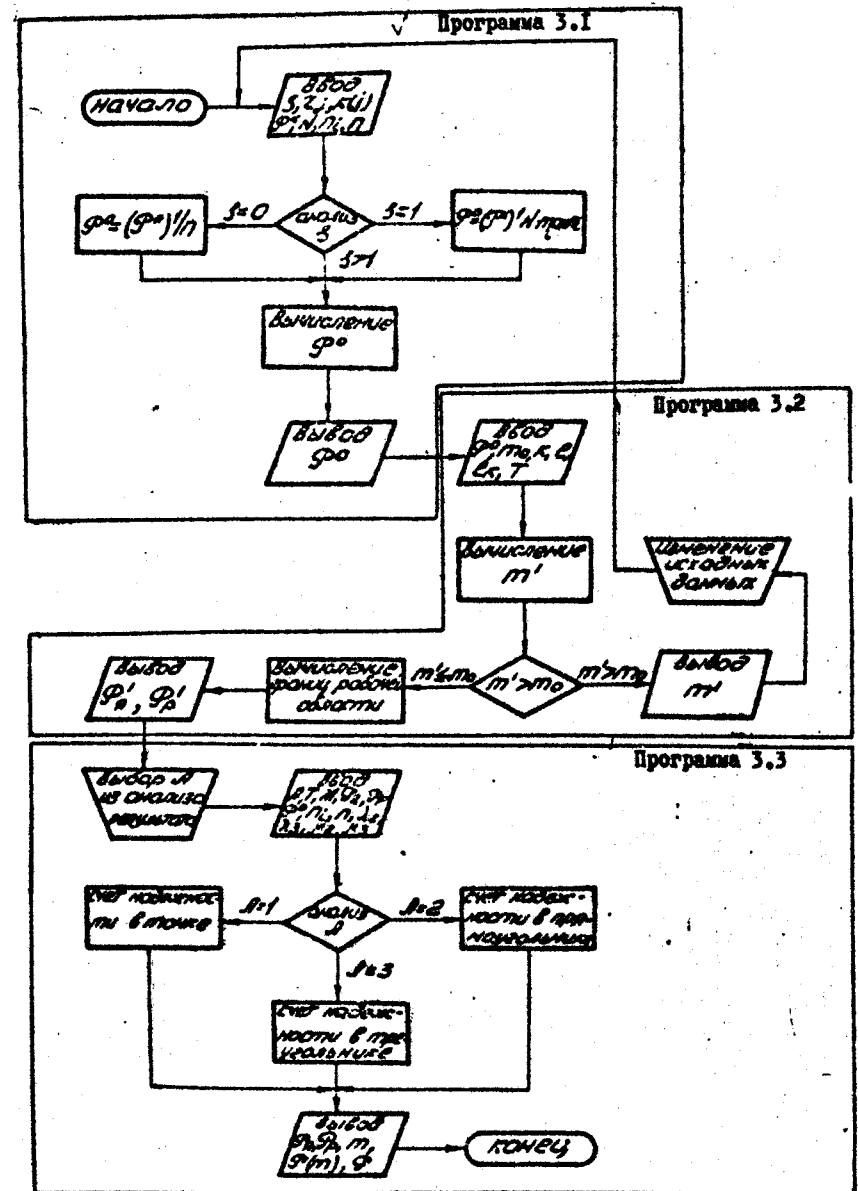


Рис. 4.1 Комплекс расчета надежности

Программа 3.3 вычисляет надежность системы в заданном диапазоне надежностных параметров элементов ОВСр ([4] п.п. 2.3).

Характеристики программы сведены в табл. 4.1

Т а б л и ц а 4.1

Но - мер про-грам-мы	Счет од - ного бло-ка (мин.)	Объем программы	Примечание
3.1	10	3 лист.	$N = 787$ функций
3.2	10	2 лист.	$n = 1697$ элементов
3.3	25	5 лист.	$m_0 = 5$

Рассмотрим подробно перечисленные программы.

Программа 3.1 имеет следующий массив исходных данных:

$P^*$  - требуемая надежность системы функций,

$k(j)$  - важность  $j$ -го класса функций,

$z(j)$  - число функций в  $j$ -м классе,

$s$  - число классов функций,

$N = \sum_{j=1}^s z_j$  - число функций системы,

$n_j$  - сложность реализации в ОВСр  $i$ -й функции,  $i = \overline{1, N}$ ,

$n$  - сложность реализации в ОВСр системы из  $N$  функций.

Величины  $P^*, k(j), z_j, s, N$  задаются заказчиком, а величины  $n_j, n$  вычисляются на этапе синтеза (§ 3).

Результатом работы программы 3.1. является величина наибольшей требуемой надежности элемента ОВСр.

В программе 3.1 предусмотрены три способа вычисления величины  $P^0$  в зависимости от параметра  $s$ .

Если  $s > 1$ , то  $P^0$  вычисляется согласно методике, изложенной в [4] п.3.

Если  $s = 1$ , то  $P^0$  вычисляются по формуле  $P = (P^*)^{1/N \cdot \max n_i}$ .

Если  $s = 0$ , т.е. не делается никаких предположений о важности функций, то  $P^0$  вычисляется по формуле  $P = (P^*)^{1/n}$ .

Программа 3.2 вычисляет координаты  $P'_q, P'_p$  вершин треугольной рабочей области, т.е. нижние границы надежности ячеек и реле.

Массив исходных данных для программы 3.2:

$m_0$  - наибольшая допустимая кратность резервирования группы,

$T$  - время работы УЛУ,

$k$  - число ячеек в элементе группы,

$l$  - число реле в элементе группы,

$l_k$  - число реле в коммутаторе группы,

$P^0$  - требуемая надежность группы.

Величины  $m_0, T$  задаются заказчиком, величины  $k, l, l_k$  определяются выбранной схемой элемента ОВСр, а величина  $P^0$  вычисляется с помощью программы 3.1.

Программа 3.2 оценивает величину  $m'$  - кратность резервирования, минимально необходимую для получения  $P^0$ . Если  $m'$  превышает  $m_0$ , то величина  $m$  выводится (это говорит о необходимости изменения исходных данных), если нет, то вычисляются и выводятся на печать нижние границы надежности ячейки  $P'_q$  и реле  $P'_p$ .

Программа 3.3 вычисляет таблицу максимальных значений надежности  $P$  системы функций для заданной области счета, лежащей внутри рабочей области, и для значений кратностей резервирования, не превышающих  $m_0$ .

Массив исходных данных:  $m_0, T, k, l, l_k, P^0, n, P'_q, P'_p, M = T/\sigma$  - число интервалов  $\sigma$ , на которое разбивается время  $T$ . Кроме того, задаются нижние и верхние границы интервалов изменения параметров  $\lambda_2, \lambda_3, \mu_2, \mu_3$  [4] и числа точек счета внутри этих интервалов.

В программе 3.3 с помощью переменной  $A$  предусмотрены три способа задания области счета в координатах  $P_q, P_p$ . Если  $A = 1$ , то область счета задается одной точкой с координатами  $P_q, P_p$ .

Если  $A = 2$ , то область счета является прямоугольником, границы и числа точек счета по каждому параметру  $P_q, P_p$  которого должны быть заданы.

Если  $A = 3$ , то область счета является треугольником, границы и число точек счета по каждому параметру  $P_q, P_p$  которого должны быть заданы. Счет в каждой точке области счета проводится путем последовательного увеличения кратности резервирования, начиная с кратности, равной 2. Счет прекращается при достижении кратности  $m_0$ , или при достижении значения надежности группы величины  $P^0$ , или при достижении значения надежности группы максимальной величины. При  $A = 2$  и  $A = 3$  счет начинается с на-

больших значений  $P_q, P_p$ . Для фиксированного значения  $P_q$  перебираются значения  $P_p$ , начиная с наибольшего; если в какой-либо точке требуемая надежность группы не достигается, то берется новое значение  $P_q$ .

На печать выводятся координаты точки счета  $P_q, P_p$  кратность резервирования  $m$ , при которой достигается максимальная надежность группы  $P(m)$ , надежность системы  $P = [P(m)]^n$  и величина  $P(m) \geq P^0$ .

Результат расчетов для системы уравнений, состоящей из 787 функций, приводится в [4].

В классе одномерных вычислительных сред для случая однонаправленной среды с  $n$ -входными МФЭ, которые могут быть настроены на выполнение любой функции из выбранного базиса, разработана система автоматизированного проектирования (САПР) УЛУ, описываемая системами логических функций бесповторных в д.н.ф.

САПР позволяет:

1. Находить рациональное размещение УЛУ в пространстве объекта.

2. Находить базис элемента среды, который при выбранном методе синтеза обеспечивает близкие к минимуму число элементов УЛУ.

3. Находить программу настройки среды на реализацию заданного УЛУ, обеспечивающую близкое к минимуму количество элементов.

4. Находить надежность параметры элементов, обеспечивающих требуемую надежность устройства.

Система проектирования содержит 167 листов на АЛГОЛе ( $\approx 25\ 000$  команд). Она позволяет существенно сократить время разработки УЛУ, а использование среды позволяет расширить область её применения.

#### Л и т е р а т у р а

1. ЕВРЕЙНОВ Э.В., КОСАРЕВ Ю.Г. Однородные универсальные вычислительные системы высокой производительности. Новосибирск, 1966.

2. МОСКОВ Б.А., СКОРОБОГАТОВ В.А. Реализация судовых логических функций в блочной вычислительной среде. Материалы ко II Всесоюзной конференции по вычислительным системам. Новосибирск, "Наука", 1969.

3. ДЯТЛОВ В.Л., РОГАЛЕВ А.И. Электромеханические пленочные элементы. "Вычислительные системы", Новосибирск, вып. 49, стр. 132-146.

4. КОЙФМАН А.А., МАКАРОВ Л.И., МОСКОВ Б.А., НАЗАРОВ Н.И., СКОРОБОГАТОВ В.А. Надежность устройств логического управления в одномерной вычислительной среде. Настоящий сборник, стр. 129-138.

5. КОЙФМАН А.А., МОСКОВ Б.А., ПАНКОВ А.В., ПОПКОВ В.К., СКОРОБОГАТОВ В.А. Выбор базиса элемента одномерной вычислительной среды. Настоящий сборник, стр. 112-118.

6. КОЙФМАН А.А., МАКАРОВ Л.И., ПАНКОВ А.В., ПОПКОВ В.К., СКОРОБОГАТОВ В.А. Синтез устройств логического управления в одномерной вычислительной среде. Настоящий сборник, стр. 119-128.

7. КОЙФМАН А.А., МАКАРОВ Л.И., МОСКОВ Г.А., СКОРОБОГАТОВ В.А. Решение о выдаче авторского свидетельства от 15.XI-72 на заявку № 1696388/18-24.

8. МОРОЗОВ К.К., МЕЛИХОВ А.Н. и др. О компоновке элементов сложных схем. Препринт - 72-10, Киев, 1972.

Поступила в ред.-изд.отд.  
22 декабря 1972 г.