

СООБЩЕНИЯ  
ОБЪЕДИНЕННОГО  
ИНСТИТУТА  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ  
ДУБНА



Ц8408  
К-736

З/III-75

10 - 8439

853/2-75

В.М.Котов

АЛГОРИТМЫ УПРАВЛЕНИЯ РАБОТОЙ  
СКАНИРУЮЩЕЙ СИСТЕМЫ  
"СПИРАЛЬНЫЙ ИЗМЕРИТЕЛЬ".

Часть II. Операционная система

**1974**

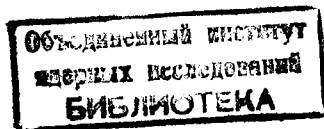
ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ  
ТЕХНИКИ И АВТОМАТИЗАЦИИ

10 - 8439

В.М.Котов

АЛГОРИТМЫ УПРАВЛЕНИЯ РАБОТОЙ  
СКАНИРУЮЩЕЙ СИСТЕМЫ  
"СПИРАЛЬНЫЙ ИЗМЕРИТЕЛЬ".

Часть II. Операционная система



## В В Е Д Е Н И Е

Последовательность операций по обработке с помощью сканирующей системы "Спиральный измеритель" снимков с трековых камер определяется конструкцией механической и электронной аппаратуры системы<sup>/1/</sup>, характерной особенностью которой является включение оператора в процесс измерения. Участие оператора в опознании нужного типа событий и измерение им нескольких точек в особо сложных случаях резко повышает эффективность системы в целом.

Рабочий режим управления сканирующей системой СИ может быть разделен на два подрежима:

- отсутствие заявок оператора и ожидание входной информации;
- обработка заданного алгоритма или группы их, осуществляемая как автоматически, так и при вмешательстве оператора.

Поэтому главное требование ко всей управляющей программе может быть сформулировано так: первый подрежим должен быть как можно короче, так как его длительность в основном определяет производительность системы. Лучшим способом выполнения этого требования является, конечно, автоматизация всех тех операций, вмешательство человека в которые не связано с опознанием и поэтому может быть устранено. Но весьма важно так организовать весь процесс измерения, чтобы те операции, которые остаются на долю оператора, были свойственны человеку, чтобы он даже испытывал желание поскорее их выполнить. К числу таких действий относится как раз вмешательство оператора при совмещении полюса спиральной развертки и вершины события при условии автоматического вывода изображения вершины в зону начала развертки.

Если ошибка невелика, то оператор очень быстро и с охотой производит центрирование, тем более, что он знает, что начинающийся после этого процесс спирального сканирования полностью автоматизирован и в течение 4-5 секунд можно ничего не делать.

Вмешательство оператора требуется также для подтверждения соответствия информации о данном событии (номер кадра, топология события и т.д.) той служебной информации, которая имеется в программе управления в блоке данных просмотра. Каждый раз при фиксации кадра в фильмовом канале оператору выдается на световое табло пульта связи вся служебная информация о данном событии, и если она соответствует тому, что оператор видит на обзорном экране, он должен нажать определенную кнопку пульта связи, после чего начинается следующая, полностью автоматизированная, операция по обмеру события. Для того, чтобы оператор быстрее ответил, программа управления одновременно с выдачей сообщения включает прерывистый звуковой сигнал, который убирается только после нажатия кнопки подтверждения.

Возможность чередования сравнительно простых и коротких действий оператора по вмешательству в процесс измерения с полностью автоматизированными операциями позволяет достигнуть очень высокой производительности обработки камерных снимков на СИ. (Рекордная производительность обработки, достигнутая на аналогичной установке *Spiral Reader* /2/ в Беркли США, составляет 200 событий в час). Описанию алгоритма управления работой сканирующей системы СИ посвящена данная работа.

#### АЛГОРИТМЫ УПРАВЛЕНИЯ СПИРАЛЬНЫМ ИЗМЕРИТЕЛЕМ

В процессе предварительного просмотра пленок данного эксперимента определяются номера кадров, содержащих исследуемое событие, его топология, а также количество событий на данном снимке и номера проекций, подлежащих измерению. Вместе с этим должны быть измерены для каждого события координаты расположения вершины события на снимке. Характерная особенность этого единственного для СИ процесса, требующего измерения при просмотре, состоит в том, что координаты вершины могут быть измерены достаточно грубо ( $100 \text{ мкм} < \Delta X < 1,5 \text{ мм}$  в плоскости пленки), что позволяет использовать просмотрные столы очень простой конструкции. Результаты просмотра записываются на магнитную ленту, которая затем вместе с просмотренными пленками поступает на СИ и устанавливается на один из 2-х магнитофонов управляющей ЭВМ /1/.

После того, как оператор передает в управляющую ЭВМ исходную информацию о номере пленки и начальном кадре, установленном на каждом лентопротяжном механизме, начинается собственно процесс обработки снимков на СИ. Общая блок-схема программы управления и принцип действия СИ были рассмотрены в работе /1/, а в данной работе одновременно с более подробным обсуждением общего процесса управления работой СИ будут отмечаться отдельные алгоритмы или программы, входящие в модуль операционной системы СИ, названный монитором /3/, а также те функциональные алгоритмы, выполнение которых осуществляется при помощи внешних программ, составляющих библиотеку программ обслуживания, состав которой может изменяться в зависимости от данного физического эксперимента.

Обработка камерной фотографии на СИ начинается запросом оператора на считывание очередного блока данных просмотра с магнитной ленты и запись его в оперативную память ЭВМ. Формат данных просмотра на магнитной ленте стандартизирован и рассчитан на определенную форму записи результатов просмотра. Дальнейшее обслуживание этой заявки оператора, которая представляет собой внешнюю программу четвертого приоритета, осуществляется диспетчером. Структура этой программы в определенной степени зависит от способа фотографирования внутреннего объема камеры и получения стереопроекций. В результате работы этой программы определяется номер проекции и разница между текущим и искомым номерами кадров, значение которых вместе со знаком передается программе поиска кадра, входящей в состав монитора и работающей в реальном масштабе времени. В стартовую часть этой программы, выполняемую диспетчером, входит также группа программ для проверки условий, разрешающих работу лентопротяжных механизмов. После включения в работу программы поиска кадра, представляющей собой набор вклинивающихся программ /3/, диспетчером вызывается стартовая программа алгоритма управления перемещением измерительного стола в начальное положение, которая выполняется одновременно с вклинивающимися программами поиска кадра.

Необходимость перемещения измерительного стола в начальное положение вызвана тем, что лентопротяжные механизмы размещены на отдельных платформах, не связанных с измерительным столом /4/. Поэтому для того, чтобы кадр мог быть зафиксирован в фильмовом канале, измерительный стол перед прижимом кадра должен быть установлен в некоторое исходное состояние.

После завершения поиска и фиксации кадра номер его выдается оператору на табло. Если оператор отвечает нажатием кнопки пульта управления, подтверждающей соответствие, то диспетчер вызывает программу, которая образует петли пленки, что необходимо для перемещения измерительного стола с прижатой к нему пленкой в процессе измерения при неподвижных фильмопротяжных механизмах. Одновременно с этим вызывается алгоритм, выполняемый по вклинивающимся программам обслуживания прерываний, при помощи которого производится поиск такого же номера кадра на пленке, установленной на другом лентопротяжном механизме.

Когда петли пленки образованы, начинается процесс измерения координат реперных крестов, в алгоритм управления которым входит также внешняя программа из состава библиотеки обслуживающих программ. Все операции по определению количества измеряемых крестов и вычисления зоны их расположения на каждой проекции, а также контроль измеренных координат производится внешней программой. Если режим измерения координат реперных крестов полуавтоматический, то в результате работы этой программы определяются значения координат зоны первого измеряемого креста, которые являются исходными данными для выполнения уже в реальном масштабе времени программ движения измерительного стола в заданную точку. После отработки этой программы изображение реперного креста появляется на экране монитора перед оператором, и он центрированием креста с меткой на экране измеряет его координаты. Затем автоматически выводится на экран изображение следующего креста для центрирования и измерения и т.д. В автоматическом режиме вся процедура снятия значений координат производится без участия оператора.

По окончании измерения координат реперных крестов диспетчер вызывает следующую внешнюю программу - "VERT", которая производит вычисления значений координат вершины измеряемого события для данной проекции, используя данные просмотра, а затем опять вызывается входящая в монитор программа управления перемещением измерительного стола в заданную точку. После подтверждения топологии события оператор, центрируя вершину события с полюсом спирали, выдает заявку на начало сканирования. Если событие на кадре одновершинное и единственное, то обработка его на этом заканчивается и начинается измерение следующего кадра, содержащего другую проекцию. Блок-схема основных программ обслуживания заявок оператора приведена на рис.1, а на рис.2 поясняется алгоритм поиска очередной проекции.

Одновременно с выполнением функциональных алгоритмов управления в СИ производится также съем данных с различных отсчетных систем и датчиков исполнительных механизмов и передача их по каналу прямого доступа в оперативную память ЭВМ. Кроме того, в состав практически всех алгоритмов входят программы контроля за качеством результатов измерения и выполнением отдельных команд /5/.

Но тем не менее видно, что весь процесс измерения на СИ может быть разделен на три отдельных больших этапа:

- запрос данных просмотра и поиск кадра, содержащего нужный тип события;
- измерение координат реперных крестов и вершины события;
- спиральное сканирование и запись данных измерения на магнитную ленту.

Каждый из вышеназванных этапов начинается выполнением внешней подпрограммы из библиотеки обслуживающих программ, вызываемой диспетчером ОС по запросу от одной из кнопок пульта оператора. По этим

подпрограммам с использованием данных просмотра производятся все необходимые вычисления и задаются исходные данные для программ монитора операционной системы, при помощи которых и осуществляется управление процессом обработки намерных снимков на СИ.

Повышение устойчивости выполнения алгоритмов управления на СИ достигается тем, что для каждой стадии их измерения задается разрешенная комбинация кнопок пульта управления (назначение указано в таблице 1), которыми может пользоваться оператор, не вызывая нарушения работы системы. Порядок обработки сигналов прерывания от кнопок пульта в этом случае заключается в следующем: ЭВМ реагирует на нажатие каждой кнопки пульта, но если в выделенной ячейке оперативной памяти содержится разряд, соответствующего номеру данной кнопки, не равно логической единице, то вызова заявки диспетчером не производится, а оператору на световое табло выдается сообщение "Ошибка" с одновременным включением звукового сигнала (основные сообщения, выдаваемые на табло пульта связи стола оператора, общий вид которого представлен на рис. 3, приведены в таблице 2). Содержимое ячейки оперативной памяти, определяющей разрешенную комбинацию кнопок, устанавливается диспетчером ОС в соответствии с ходом выполнения алгоритма управления по результатам работ обслуживающих программ, вызываемых в начале каждого этапа.

Наряду с программными блокировками в электронную аппаратуру пульта (рис.4) включены логические схемы защиты от повторного нажатия кнопок. Для этого порядковый номер кнопки представляется в двоичном коде и по сигналу срабатывания нормально разомкнутых контактов заносится в регистр, сброс которого осуществляется по переднему сигналу срабатывания нормально замкнутых контактов любой из кнопок пульта. После этого выставляется сигнал запрета на занесение в этот

регистр до тех пор, пока в ЭВМ не будет считано содержимое этого регистра по сигналу прерывания, поступающему одновременно с занесением кода в регистр.

Для повышения помехоустойчивости при обслуживании прерывания от кнопок вводится еще и временная задержка на разрешение обслуживания следующего сигнала (рис.5).

Формирование адреса стартовой программы обслуживания выбранной кнопки производится внутри вклинивающейся подпрограммы, для чего двоичный код номера кнопки добавляется к начальному адресу, считанному из мультиплексора / I /, определяя тем самым адрес передачи управления для данной кнопки внутри резидента пульта управления (рис.5)

Каждая кнопка может быть подсвечена изнутри зеленым и красным цветом. Зеленый цвет получает кнопка, нажатая оператором последней, а красный подсвет включается у кнопки, которую надо нажать следующей, например, для подтверждения топологии. В этом случае подсвет мигающий, и одновременно включается звуковой сигнал. Постоянный красный подсвет сигнализирует оператору, что ЭВМ обрабатывает поданную ей при нажатии этой кнопки заявку на обслуживание. Обслуживание заявок пультов управления и связи производится по основным программам, повышение эффективности которых в значительной степени зависит от решения следующих двух основных вопросов:

- деление подпрограммы на блоки;
- организация входа в подпрограмму и выхода из неё.

Решение первой задачи связано с общей проблемой оптимизации программ и производилось по общим правилам сокращения времени счета или уменьшения объема памяти, необходимой для их размещения. Сокра-

щение объема программ при некотором увеличении времени выполнения было достигнуто широким использованием подпрограмм, т.е. выделением часто встречающихся процедур в отдельные циклы, обращение к которым может производиться неоднократно из различных точек программы. К числу таких повторяющихся циклов относятся прежде всего подпрограммы отсчета значений координат измерительного стола " XУРОЛН " и подпрограммы проверок выполнения условий, разрешающих дальнейшее выполнение алгоритма. Уменьшение расхода памяти, не приводящее к увеличению времени счета, было получено в результате оптимизации распределения памяти за счет использования одних и тех же её ячеек для размещения непересекающихся переменных.

Для внешних программ очень важной является вторая проблема - организация входа и выхода из внешних подпрограмм обслуживания. Следует отметить, что эти подпрограммы имеют два типа переменных: переменные, используемые только внутри программы в процессе счета, и переменные, соответствующие входным и выходным параметрам, при помощи которых данная подпрограмма общается с другими модулями операционной системы.

При проектировании системы СИ обращалось большое внимание на гибкость программы управления, т.е. на возможность добавления в её состав новых подпрограмм без изменения монитора. Выполнение этих условий требует, чтобы вход во внешние подпрограммы и выход из них был формализован, и тогда для монитора неважно, что и каким образом делает внешняя подпрограмма, пусть только результаты её работы будут представлены в заданном формате. Гибкость обмена в этом случае зависит от способности внешних подпрограмм обмениваться параметрами и работать с переменными, обладающими **COMMON** - свойствами.

Решение этой задачи в значительной степени определяется воз-

возможностями тех средств автоматизации программирования, при помощи которых разрабатывались программы управления.

Средства автоматизации программирования,  
программы редактирования и настройки

Структура общего алгоритма управления сканирующей системы СИ потребовала введения иерархии в его построение, что определило глубокую связь отдельных подпрограмм. Это предъявляет, в свою очередь, жесткие требования к средствам автоматизации программирования и прежде всего — к символическому языку в части обеспечения автоматической стыковки подпрограмм между собой и контроля взаимных управляющих и информационных связей. Информационно-логический характер программ управления усложняет их структуру и требует неоднократного применения в таких программах команд условного или безусловного перехода.

При выборе символического языка для автоматизации программирования приходилось учитывать также особенности работы с большим количеством булевых переменных и малоразрядных величин, необходимость использования которых определяется логическим характером управляющих алгоритмов. Плотная упаковка таких величин в оперативной памяти и доступность их для различных программ требует определенной абстрактности символического языка программирования.

В результате анализа и оценки перечисленных выше требований при разработке программ управления были выбраны два типа символического ассемблера управляющей ЭВМ спирального измерителя. Первый из них — PALIII /6/, входящий в состав математического обеспечения, поставляемого заводом-изготовителем этой вычислительной машины, — представляет собой машинно-ориентированный символический язык, являющийся фактически покомандным автокодом машины. Выходная двоичная

перфолента программы выдается в абсолютных двоичных адресах и загружается в память "двоичным загрузчиком".

Основная трудность работы с таким ассемблером заключается в необходимости учета при программировании особенностей страничной организации оперативной памяти ЭВМ. Все ссылки из текущей в другие страницы памяти (размер каждой страницы 128 ячеек) требуют использования косвенной адресации. Кроме того, и переход из одного куба памяти в другой требует выполнения целого ряда операций.

В связи с этими особенностями PALIII нашел ограниченное применение (в основном для написания тестовых программ проверок и функционального контроля). Весь комплекс управляющих программ был разработан с использованием символического ассемблера SABR, входящего в состав алгоритмического языка FORTRAN BK для PD P8 фирмы DEC /6/, который с небольшими изменениями был адаптирован для управляющей ЭВМ СИ. Основные особенности этого языка состоят в том, что в результате первого прохода транслятора FORTRAN BK генерируется программа на языке SABR, который в то же время представляет собой самостоятельный символический машинно-ориентированный язык, близкий по своим свойствам к автокодному языку. Это разрешает естественным образом стыковывать программы, написанные как на ФОРТРАНе, так и в SABR, потому что после выполнения первого прохода транслятора ФОРТРАНа (компилятора) выдается промежуточная для ФОРТРАНа лента, но в формате ассемблера, и поэтому уже на уровне второго прохода образуется единая программа.

В результате трансляции ассемблер выдает двоичную перфоленту программы, но в относительных адресах. Для загрузки этой ленты в оперативную память используется специальный (перемещающий) загрузчик "Linking Loader".



Каждый раз после окончания очередного прохода транслятора ФОРТРАНа и выдачи перфоленты с результатами память ЭВМ может быть полностью освобождена, а операционная система ассемблера " **RUN-TIME LINKAGE ROUTINES** ", загружаемая вместе с "перемещающим загрузчиком", занимает около 400 ячеек каждого куба оперативной памяти. Если в процессе загрузки программы "перемещающий загрузчик" определяет, что загружаемая программа не помещается в память, то он выдает на перфолатор в **RIM** -формате часть программы, которая затем может быть введена в оперативную память уже **RIM**-загрузчиком на то место, где был размещен сам "перемещающий загрузчик", не трогая его операционной системы (т.е. тех 400 ячеек каждого куба, о которых было сказано выше). Таким образом, **FORTRAN 8K** в отличие от версии **FORTRAN 4** оставляет свободной для пользователей фактически все 8K оперативной памяти.

Но основное достоинство этой системы математического обеспечения, которое определило широкое применение её при разработке программ управления СИ, состоит в том, что собственно ассемблер не только разрешает использовать всю библиотеку стандартных программ ФОРТРАНа, но позволяет работать также с внешними подпрограммами пользователя, которые имеют точку входа **ENTRY** и могут транслироваться отдельно, как обычные подпрограммы ФОРТРАНа. Вызов таких внешних подпрограмм производится при помощи псевдоинструкции **CALL**, и каждая из подпрограмм может иметь до 64 аргументов для передачи фактических параметров. Кроме того, при программировании с использованием ассемблера автоматически учитываются ссылки вне текущей страницы, а вызов внешних подпрограмм, имеющих точку входа **ENTRY**, может производиться независимо от номера куба оперативной памяти, откуда производится вызов и где размещена вызываемая подпрограмма.

В состав ассемблера включено большое число псевдоинструкций, используемых для распределения оперативной памяти. Ассемблер позволяет также организовывать при помощи псевдоинструкции **TEXT** текстовые строки. Совместное использование этих псевдоинструкций с имеющимися в составе библиотеки стандартных программ передачи и приема данных с телетайпа упрощает выдачу сообщений и диагностики оператору.

Перемещающий загрузчик после загрузки программы пользователя выдает карту памяти и количество незанятых страниц каждого куба. Это позволяет производить оперативный контроль за процессом загрузки, добиваясь высокой плотности упаковки программ.

Структура команд управления СИ и возможности ассемблера определять команды пользователя разрешают задавать мнемоническое значение этим командам и включать их в постоянную таблицу символов ассемблера. В этом случае производимый ассемблером в процессе трансляции синтаксический контроль охватывает и введенные команды управления.

Перечисленные особенности делают этот символический язык очень эффективным при написании программ управления. Однако полное использование свойств ассемблера, особенно при написании вклиниваемых программ обслуживания прерывания, требует очень осторожного и внимательного анализа. Основное требование в этом случае заключается в следующем: вклинивающаяся программа должна быть написана так, чтобы при выполнении её не использовалась собственная операционная система ассемблера. При определенном навыке программирования на этом языке можно, в случае необходимости, максимально приблизить программу, написанную в **SABR**, к автокодному варианту и исклю-

чить в конкретном случае вклинивающихся программ какое-нибудь использование подпрограмм собственной операционной системы.

### ЗАКЛЮЧЕНИЕ

Разработка программ управления СИ потребовала длительного времени (трудоемкость создания программ управления СИ составляет около 15 человеколет) и производилась в несколько этапов, одновременно с настройкой электронной аппаратуры системы. Многообразие логических связей и неоднократное преобразование информации в системе предопределяет наличие множества ошибок, особенно на первых этапах настройки, на поиск и устранение которых уходит основная часть времени разработки системы. Поэтому быстрота внесения изменений в программы управления и длительность полного цикла получения исправленной программы играют очень большую роль в ускорении процесса настройки и в известной мере влияют на осуществимость системы в целом.

Существенную роль в ускорении процесса настройки операционной системы СИ сыграло использование служебных программ редактирования, которые позволяют оперативно вносить изменения в программу и, контролируя синтаксис вносимых изменений, помогают находить грубые формальные ошибки.

При разработке программ управления СИ широко использовался символический редактор *Symbolic Editor* фирмы DEC для ЭВМ PDP8 /6/. Необходимость использования программ редактирования связана также с тем, что для управляющей ЭВМ СИ, как и вообще для малых ЭВМ, весь процесс подготовки программы для ввода в машину производится при помощи ленточного перфоратора с выдачей результатов на перфоленду. Поэтому возможность изменения сдвига и переноса отдельных команд или целых блоков программы, а также сегментирование и выдача отдельных блоков, осуществляемые программой редактирования,

значительно ускоряет весь процесс подготовки программного обеспечения. Команды программы редактирования ориентированы на использование телетайпа *ASR -33*.

Для настройки программ управления совместно с работой электронных устройств применялись программы *DDT -8* и *ODT -8* фирмы DEC для ЭВМ PDP-8. Эти служебные программы позволяют включать в работу отдельные подпрограммы, останавливать выполнение их в заданной программистом точке программы, выдавать на телетайп содержимое всех рабочих регистров ЭВМ, изменять их содержимое, а также выполнять, в случае необходимости, итерационные процессы и организовывать циклы в программе. К сожалению, применение этих служебных программ невозможно при комплексной настройке программ управления в реальном масштабе времени при работе с использованием прерываний или внепрограммного обмена.

В заключение автор выражает признательность А.Е.Селиванову, И.И.Скрылю и Г.А.Ососкову за полезные обсуждения и благодарит Г.Н.Буланову, Т.А.Филимонову, Р.Х.Кутуева, В.Г.Васильева за помощь, оказанную ими при создании программного обеспечения спирального измерителя .

### ЛИТЕРАТУРА

1. В.М.Котов и др. ОИЯИ, IO-7939, Дубна, 1974.
2. G.G.R.Lynch. UCRL-17328, 1967, Berkeley.
3. В.М.Котов. ОИЯИ, II-794I, Дубна, 1974.
4. В.И.Зайцев и др. ОИЯИ, IO-7949, Дубна, 1974.
5. Г.Н.Буланова и др. ОИЯИ, IO-8I89, Дубна, 1974.
6. *Introduction to Programming*. DEC 1970.

Рукопись поступила в издательский отдел  
II декабря 1974 года.

Таблица I

Назначение кнопок пульта управления СИ

- |   |  |
|---|--|
| 1. Проекция № 1   | Продвинуть пленку на соответствующем лентопротяжном механизме на один кадр вперед (используется при начальной установке или при ошибке в поиске кадра).              |
| 2. Проекция № 2   |  |
| 3. Проекция № 5   |  |
| 4. Проекция № 4   |  |
| 5. Вершина  | Измерить ХУ координаты вершины события и начать спиральное сканирование.   |
| 6. Точка останова   | Измерить ХУ координаты точки останова трека измеряемого события.   |
| 7. Точка излома трека                                     | Измерить ХУ координаты точки излома трека измеряемого события.   |
| 8. Особая точка № I<br>(точка трека постороннего события) | Измерить ХУ координаты точки трека постороннего события, расположенного вблизи измеряемого события.  |
| 9. Особая точка № 2<br>(точка трека измеряемого события)  | Измерить ХУ координаты точки измеряемого события в сложном случае, например: наличие "узкой пары" треков или блика на треке и т.д.                                   |
| 10. Измерение реперных крестов                            | Подтвердить топологию события и начать перемещение ХУ стола в зону первого реперного креста.   |
| 11. Пропустить следующую операцию                         | Произвести изменение в последовательности операций при обработке снимков.  |
| 12. Следующая операция ("Дальше")                         | Начать очередную операцию. Например: измерение второй вершины или следующего события на данном кадре. Если событие на кадре последнее, то произвести смену проекций. |

Таблица 2

СВЕТОВОЕ ТАБЛО №1 (КОМАНДА ЗАНЕСЕНИЯ 6534).

СООБЩЕНИЕ	СОБЫТИЕ	КРЕСТ	КАДР	ПРОЕКЦИИ	Н М Л	ОБОЮРНУЮ ТОЧКУ	ПЕРЕСКОП	ФИЛЬМ	СТОЛ ХУ	ВЕРШИНЕ СОБЫТИЯ	400	1000	1400	2000	2400	3000	3400	4000	4400	0000	ПРЕЖДЕ ЧЕМ КОД	
ИЗМЕРЕНО	X	X	X	X	X	X	X	X	X	X												
БУФЕР ПОЛОН	X	X	X	X	X	X	X	X	X	X												
СМЕНИТЬ	X	X	X	X	X	X	X	X	X	X												
ПЕРЕМЕРИТЬ	X	X	X	X	X	X	X	X	X	X												
НЕ ГОТОВ	X	X	X	X	X	X	X	X	X	X												
ИЗМЕРЯЙ	X	X	X	X	X	X	X	X	X	X												
ТОПОЛОГИЯ	X	X	X	X	X	X	X	X	X	X												
ДАЛЬШЕ	X	X	X	X	X	X	X	X	X	X												
ИДЕТ ТЕСТ	X	X	X	X	X	X	X	X	X	X												
ОШИБКА	X	X	X	X	X	X	X	X	X	X												

СВЕТОВОЕ ТАБЛО №2. (КОМАНДА ЗАНЕСЕНИЯ 6131).

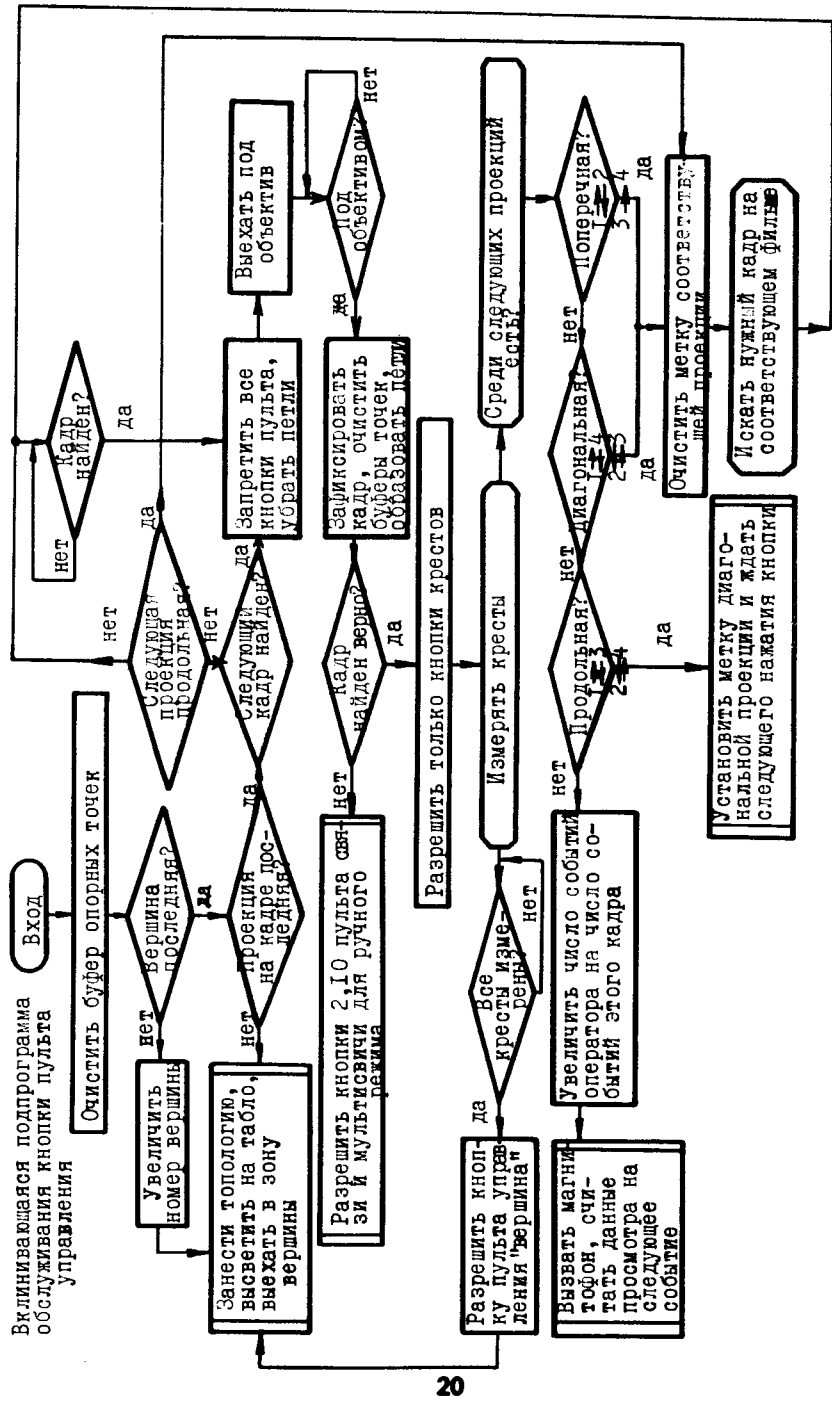


Рис. 1. Блок-схема программы обслуживания кнопки "дальше" пульта управления.

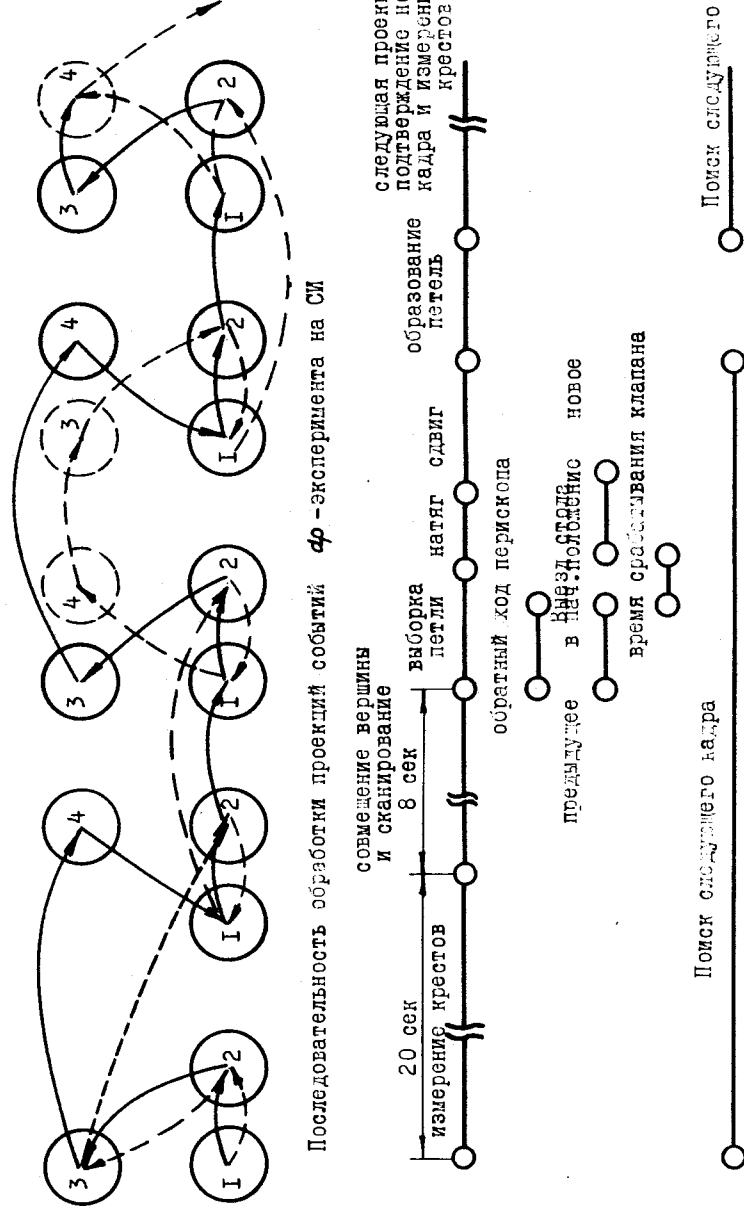


Рис. 2. Последовательность обработки события на СИ и график выполнения операции.



Рис.3. Общий вид стола оператора.

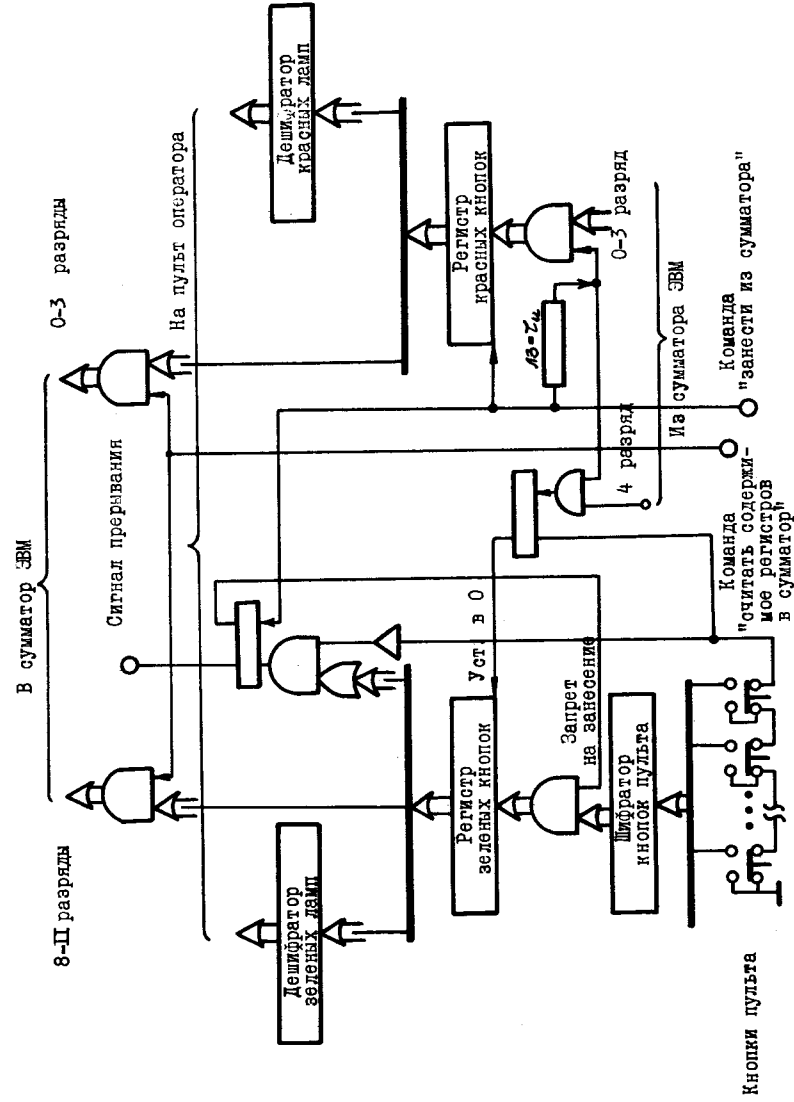


рис.4. Логическая схема пульта управления.

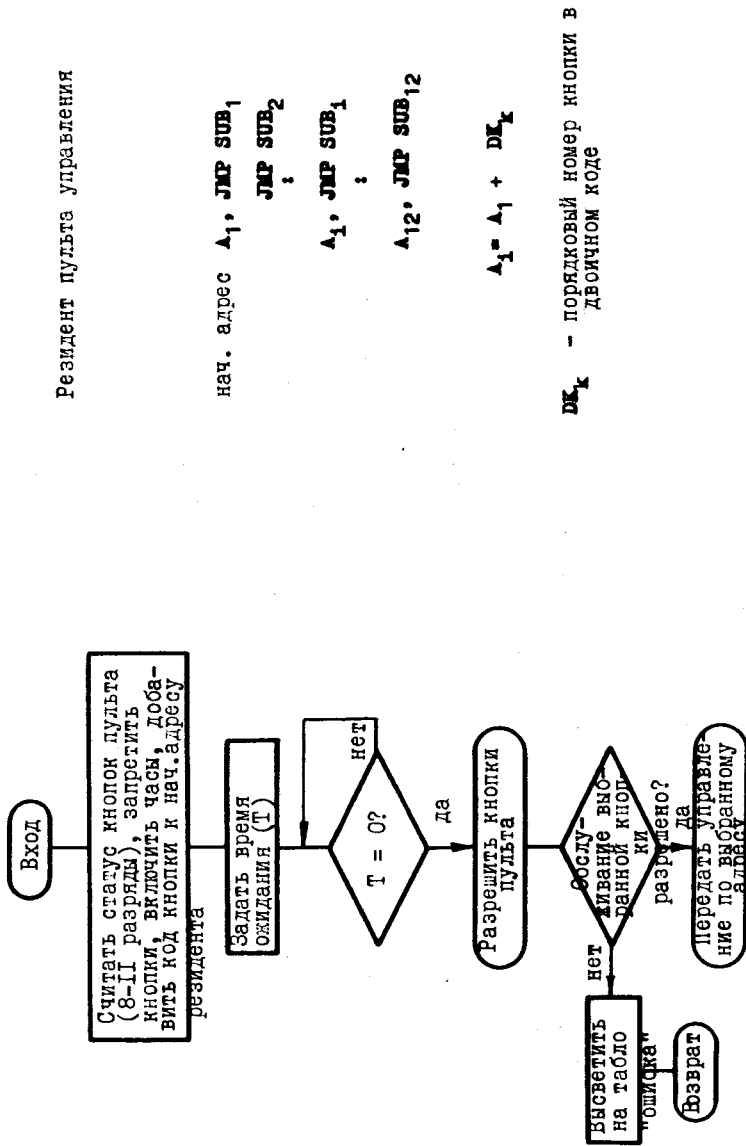


Рис. 5. Программа обслуживания пульта управления.