

ТОPAZ FLISR

(Определение и изоляция поврежденного участка сети, восстановление электроснабжения)

643.17480174.00001-01 31-09

ОПИСАНИЕ ПРОГРАММЫ

<i>Инв. №</i>	<i>Подпись и дата</i>	<i>Взам. инв.</i>	<i>Инв. №</i>	<i>Подпись и дата</i>

Москва 2023

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ИКЗ – изолятор короткого замыкания

КЗ – короткое замыкание

ОЗЗ – однофазное замыкание на землю

ОС – операционная система

РЗА – релейная защита и автоматика

FLISR – Fault Location, Isolation and Supply Restoration (определение и изоляция поврежденного участка сети, восстановление электроснабжения)

АННОТАЦИЯ

В данном программном документе приведено описание приложения TOPAZ FLISR (Определение и изоляция поврежденного участка сети, восстановление электроснабжения). Документ содержит сведения о логической структуре и функционировании данного приложения.

СОДЕРЖАНИЕ

1.	ОБЩИЕ СВЕДЕНИЯ	5
2.	ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ	5
3.	ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ	5
3.1.	Алгоритмы программы	5
3.1.1.	Алгоритм AFSR	5
3.1.2.	Алгоритм AFLO	7
3.1.3.	Алгоритм AFIS	8
3.2.	Используемые методы	8
3.2.1.	Алгоритм AFSR	8
3.2.2.	Алгоритм AFLO	8
3.2.3.	Алгоритм AFIS	9
3.3.	Структура программы с описанием функций составных частей и связи между ними	9
3.3.1.	Алгоритм AFSR	9
3.3.2.	Алгоритм AFLO	9
3.3.3.	Алгоритм AFIS	10
3.4.	Связи программы с другими программами	10
4.	ИСПОЛЬЗУЕМЫ ТЕХНИЧЕСКИЕ СРЕДСТВА	10
5.	ВЫЗОВ И ЗАГРУЗКА	11
5.1.	Способ вызова программы	11
6.	ВХОДНЫЕ ДАННЫЕ	11
6.1.	Характер, организация и предварительная подготовка входных данных	11
6.2.	Входные точки в программу	12
7.	ВЫХОДНЫЕ ДАННЫЕ	12
7.1.	Характер, организация и предварительная подготовка выходных данных	12

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Обозначение и наименование программы

Наименование программы – TOPAZ FLISR (далее в документе используется сокращенное название – приложение).

1.1. Программное обеспечение, необходимое для функционирования программы

Системные программные средства, используемые программой – операционные системы реального времени на основе ядра LINUX.

1.2. Языки программирования, на которых написана программа

Программа написана на языке C++.

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Программа предназначена для отделения поврежденных участков в распределительных сетях 6-20 кВ. Программа предназначена для работы в сетях с изолированной нейтралью, компенсированной нейтралью или нейтралью, заземленной через резистор при любых видах замыканий. Программа осуществляет определение местоположения повреждения, отключение поврежденного участка и восстановление электроснабжения неповрежденных элементов при наличии возможности.

Работа приложения может быть выполнена в полностью автоматическом режиме.

3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1. Алгоритмы программы

3.1.1. Алгоритм AFSR

Алгоритм AFSR – общая автоматика изоляции поврежденного участка распределительной сети и восстановления питания. Структурная схема алгоритма представлена на рисунке ниже.

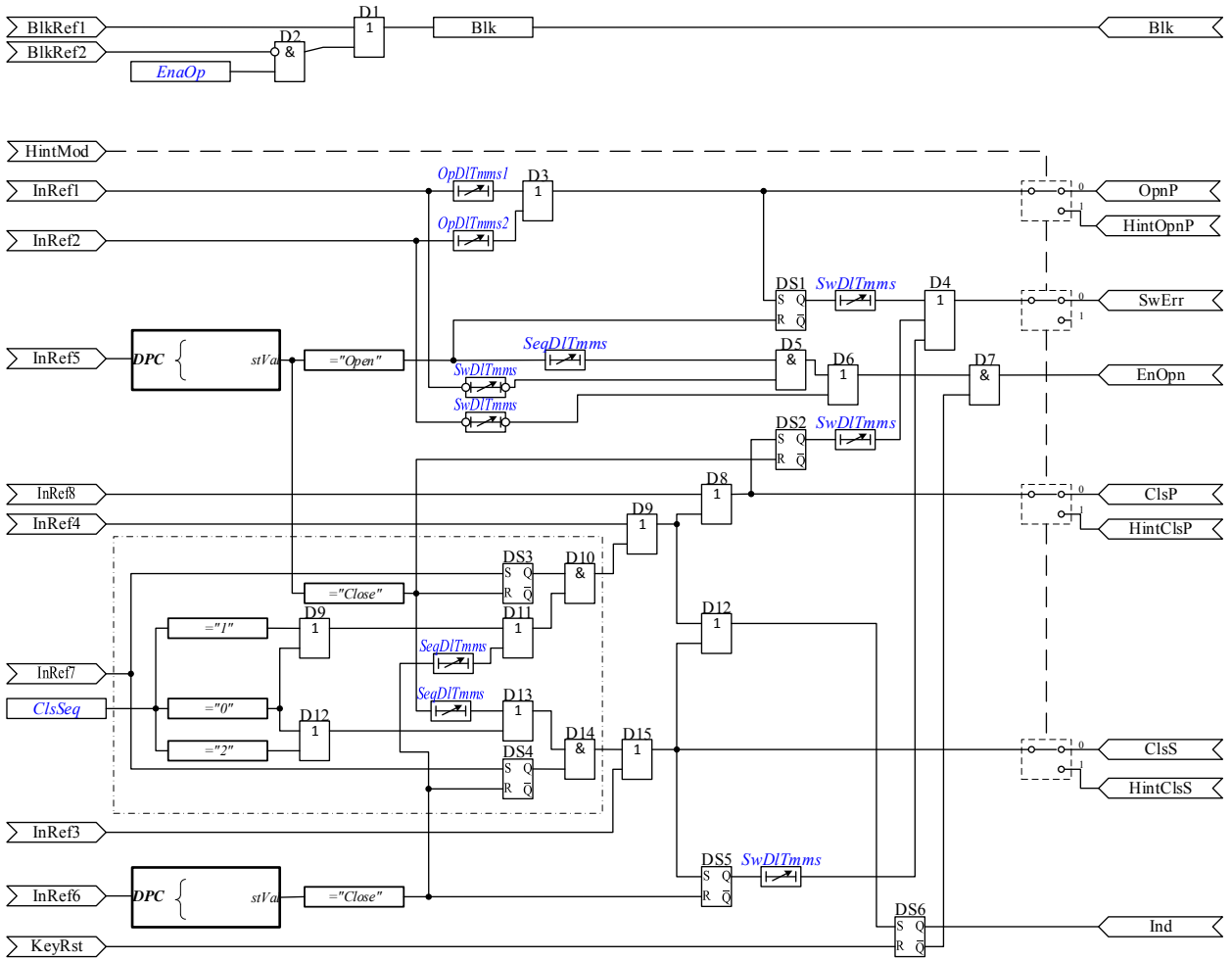


Рис. 1

3.1.3. Алгоритм AFIS

Алгоритм AFIS – автоматика изоляции поврежденного участка распределительной сети.

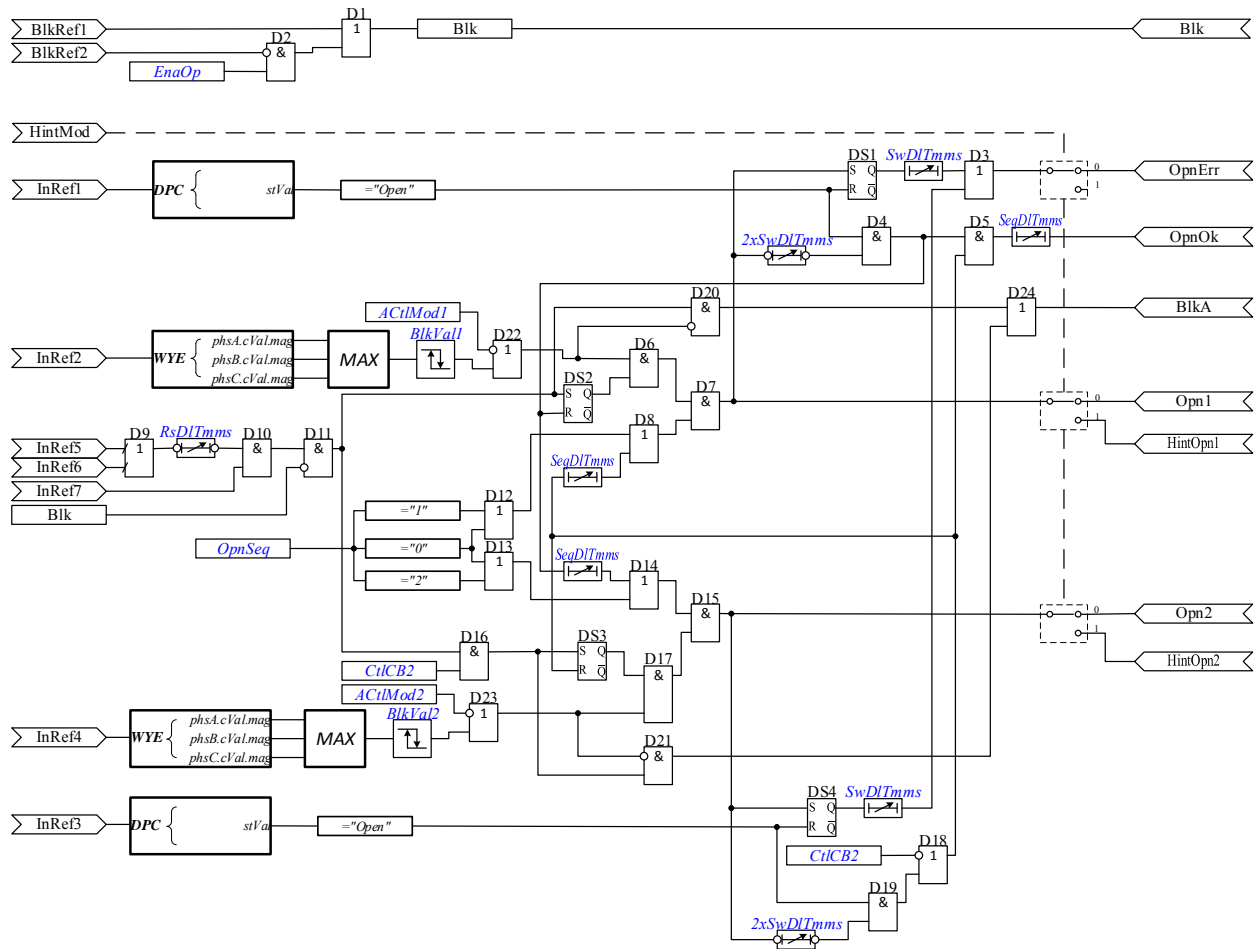


Рис. 3

3.2. Используемые методы

3.2.1. Алгоритм AFSR

- *AFSR::Init()* – начальная инициализация алгоритма;
- *AFSR::Reset()* – сброс текущего состояния алгоритма до начальных значений;
- *AFSR::UpdateModelInputs()* – предобработка и загрузка в алгоритм входных данных;
- *AFSR::Perform()* – расчет одной итерации алгоритма;
- *AFSR::PushModelOutputs()* – постобработка и выгрузка из алгоритма выходных данных.

3.2.2. Алгоритм AFLO

- *ALFO::Init()* – начальная инициализация алгоритма;
- *ALFO::Reset()* – сброс текущего состояния алгоритма до начальных значений;
- *ALFO::UpdateModelInputs()* – предобработка и загрузка в алгоритм входных данных;

- *ALFO::Perform()* – расчет одной итерации алгоритма;
- *ALFO::PushModelOutputs()* – постобработка и выгрузка из алгоритма выходных данных.

3.2.3. Алгоритм AFIS

- *AFIS::Init()* – начальная инициализация алгоритма;
- *AFIS::Reset()* – сброс текущего состояния алгоритма до начальных значений;
- *AFIS::UpdateModelInputs()* – предобработка и загрузка в алгоритм входных данных;
- *AFIS::Perform()* – расчет одной итерации алгоритма;
- *AFIS::PushModelOutputs()* – постобработка и выгрузка из алгоритма выходных данных.

3.3. Структура программы с описанием функций составных частей и связи между ними

3.3.1. Алгоритм AFSR

Алгоритм AFSR реализован с помощью наследования от абстрактного класса BaseProtection и переопределения его виртуальных методов, отвечающих за различные аспекты работы того или иного алгоритма РЗА. Каждый из описанных ниже методов вызывается вышестоящей подпрограммой в различные моменты времени:

- Метод *AFSR::Init()* вызывается в момент запуска прикладного ПО релейного терминала.
- Метод *AFSR::Reset()* вызывается в момент ввода алгоритма в действие после снятия блокировки по ключу задания режима работы блока (Mod) либо изменения состояния «горизонтальной связи».
- Метод *AFSR::UpdateModelInputs()* вызывается перед обработкой очередной итерации алгоритма для ввода входных данных алгоритма с SCL-дерева;
- Метод *AFSR::Perform()* инкапсулирует действия, реализующие расчетную итерацию алгоритма.
- Метод *AFSR::PushModelOutputs()* вызывается после очередной итерации алгоритма для вывода выходных данных на SCL-дерево.

3.3.2. Алгоритм AFLO

Алгоритм AFLO реализован с помощью наследования от абстрактного класса BaseProtection и переопределения его виртуальных методов, отвечающих за различные аспекты работы того или иного алгоритма РЗА. Каждый из описанных ниже методов вызывается вышестоящей подпрограммой в различные моменты времени:

- Метод *AFLO::Init()* вызывается в момент запуска прикладного ПО релейного терминала.

- Метод *AFLO::Reset()* вызывается в момент ввода алгоритма в действие после снятия блокировки по ключу задания режима работы блока (Mod) либо изменения состояния «горизонтальной связи».
- Метод *AFLO::UpdateModelInputs()* вызывается перед обработкой очередной итерации алгоритма для ввода входных данных алгоритма с SCL-дерева;
- Метод *AFLO::Perform()* инкапсулирует действия, реализующие расчетную итерацию алгоритма.
- Метод *AFLO::PushModelOutputs()* вызывается после очередной итерации алгоритма для вывода выходных данных на SCL-дерево.

3.3.3. Алгоритм AFIS

Алгоритм AFIS реализован с помощью наследования от абстрактного класса BaseProtection и переопределения его виртуальных методов, отвечающих за различные аспекты работы того или иного алгоритма РЗА. Каждый из описанных ниже методов вызывается вышестоящей подпрограммой в различные моменты времени:

- Метод *AFIS::Init()* вызывается в момент запуска прикладного ПО релейного терминала.
- Метод *AFIS::Reset()* вызывается в момент ввода алгоритма в действие после снятия блокировки по ключу задания режима работы блока (Mod) либо изменения состояния «горизонтальной связи».
- Метод *AFIS::UpdateModelInputs()* вызывается перед обработкой очередной итерации алгоритма для ввода входных данных алгоритма с SCL-дерева;
- Метод *AFIS::Perform()* инкапсулирует действия, реализующие расчетную итерацию алгоритма.
- Метод *AFIS::PushModelOutputs()* вызывается после очередной итерации алгоритма для вывода выходных данных на SCL-дерево.

3.4. Связи программы с другими программами

В программе используется объектная модель, созданная в программе TOPAZ Model Creator.

4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Приложение поддерживает аппаратные средства, основанные на АРМ-архитектуре. Необходимые требования к аппаратной части пакет приложений представлены в таблицах ниже.

Таблица 1 – Требования к серверу доступа к данным

Наименование параметра	Значение
ЦП	Не менее 4-х ядер, не менее 1,2 ГГц
ОЗУ	Не менее 4 Гб
ПЗУ (системный накопитель)	SSD, не менее 8 Гб
ПЗУ (накопитель БД)	4 x 2.5-inch HDD, емкость носителя определяется количеством узлов в сети и глубиной архивирования, RAID10
ОС	Операционные системы реального времени на основе ядра LINUX
Ethernet	Не менее 2 шт, 1 Гбит/с

Таблица 2 – Требования к АРМ пользователя

Наименование параметра	Значение
ЦП	Не менее 4-х ядер, не менее 1,2 ГГц
ОЗУ	Не менее 4 Гб
Видеокарта	Дискретная
ПЗУ (системный накопитель)	SSD, не менее 8 Гб
Диагональ монитора, не менее	27"
Ethernet	Не менее 2 шт, 1 Гбит/с

Для осуществления работы приложения на линии должно быть установлено следующее оборудование:

- автоматика определения наличия устойчивого повреждения на участке питающей сети;
- автоматика определения (локализации) поврежденного участка сети между двумя коммутационными аппаратами;
- автоматика отключения (изоляции) поврежденного сегмента;
- автоматика восстановления электроснабжения потребителей.

5. ВЫЗОВ И ЗАГРУЗКА

5.1. Способ вызова программы

Программа работает в автоматическом режиме. Запуск осуществляется после автозагрузки программы в ОС.

6. ВХОДНЫЕ ДАННЫЕ

6.1. Характер, организация и предварительная подготовка входных данных

Входными данными являются пользовательские данные (текстовая или цифровая информация) и данные от первичных источников информации (оборудование подстанции), организация хранения в соответствии со структурой БД системы SCADA.

6.2. Входные точки в программу

Классы *AFSR*, *AFLO*, *AFIS*.

7. ВЫХОДНЫЕ ДАННЫЕ

7.1. Характер, организация и предварительная подготовка выходных данных

Выходные данные организованы в виде узлов SCL-дерева в соответствии с МЭК 61850.