How can an interlocking scheme be modelled?

If you wish to model an interlocking scheme in *PowerFactory* (with respect to this example) then it is highly recommended to read the DIgSILENT *PowerFactory* Technical Reference Documentation for *Instantaneous Overcurrent* model. This document explains the relay logic and signal structure of the *RelIoc* block, which will not be repeated here.

1 Basics

Basically, designing an interlocking scheme for a network uses chain interlocking model which will help in communication between the relays. The two models are:

- Sending Model which consists of three blocks: Relay, Link and Logic.
- Receiving Model which contains two blocks: Relay and Link.

Relay block is populated with the corresponding relay and *Link* block basically is used to link the communication between the relays. The application of this scheme can be understood using the network as shown in figure 1.1. You can setup the relay models as shown in the *PowerFactory* example called “Interlocking_Scheme.pfd”

As you can see in the figure 1.1, R1, R2 are the relays that are used to detect the downstream fault and R3, R4 will enable faster tripping upstream when there is no downstream fault. If R1 sees a downstream fault then it will communicate with R2 using sending model and later R3 will use receiving model to receive the communication sent by R2. Lastly, R4 will receive the communication from R3.

The cubicle Cub_1 of the Bus_1 where relay R1 is configured, also contains the sending relay model as shown in the figure 1.2. To get the details of the devices on a cubicle, *Right Click on the cubicle* → *Edit Devices*.

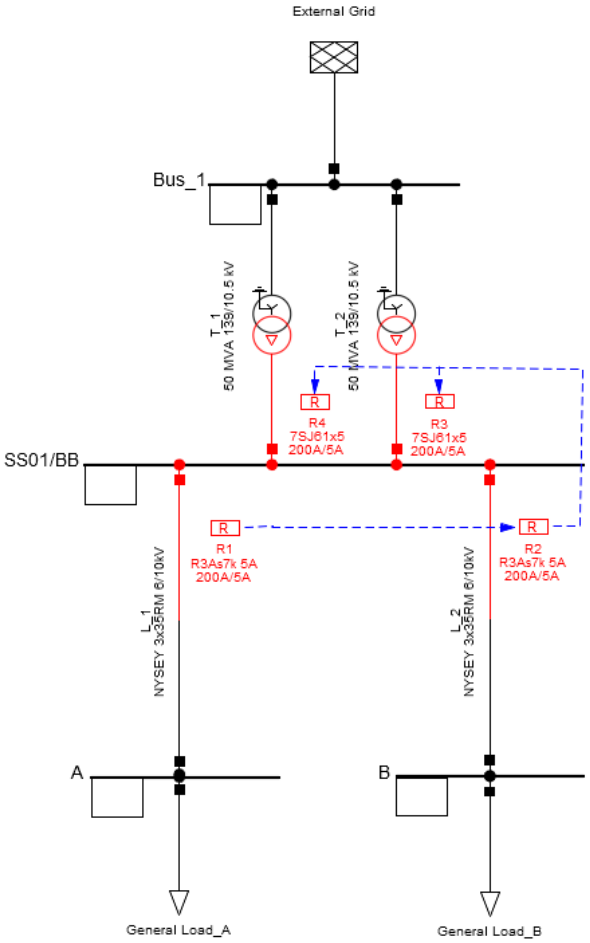

Figure 1.1: Network with chain Interlocking scheme model

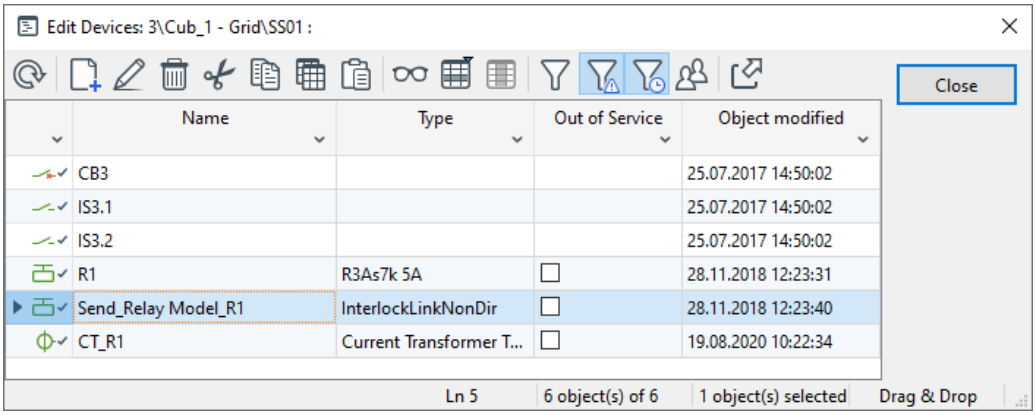

Figure 1.2: Cub_1 of Busbar Bus_1

The sending relay model for relay R1 is as shown in the figure 1.3. Since, R1 is the first relay to send the communication, Link block will be left empty.

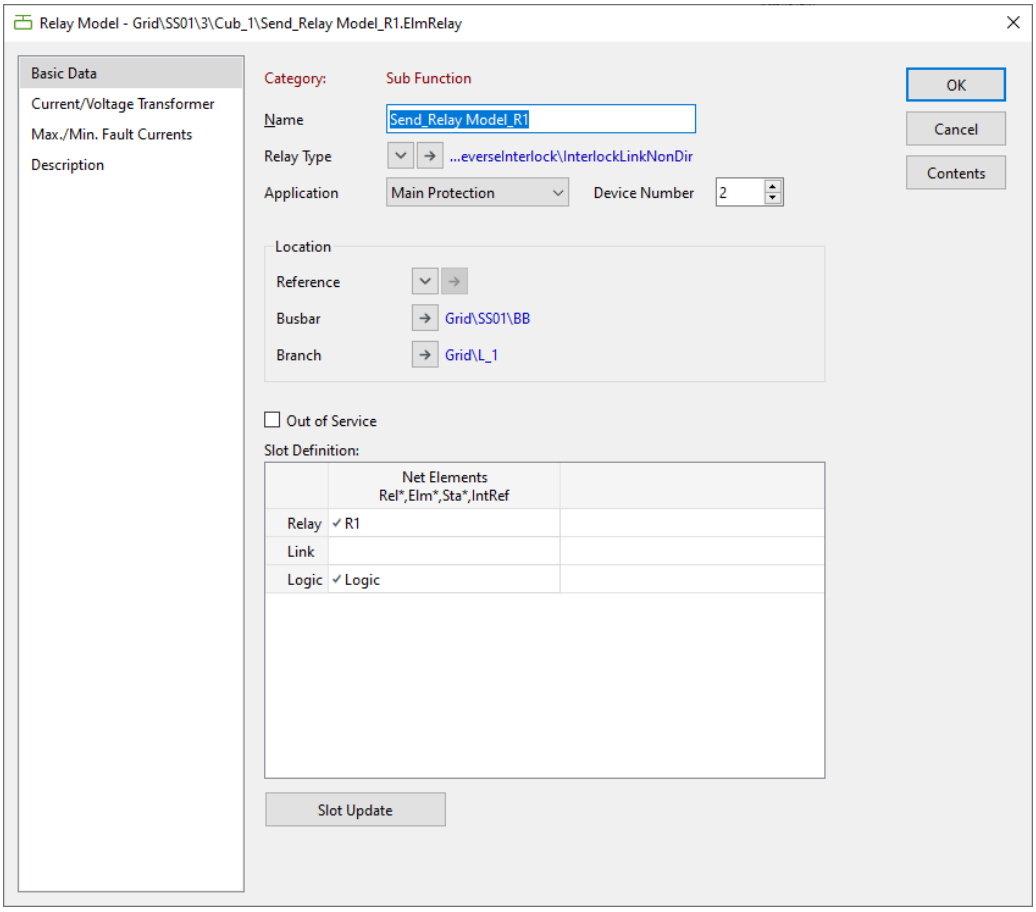

Figure 1.3: Send Relay Model for R1

Similarly, you can configure the sending relay model for relay R2. However, in this case the Link block should contain the signal from R1. Therefore, it should be populated with the Send_Relay_Model_R1. In order to understand the Send Relay Model in a better way, the frame for Send_Relay_Model_R2 is shown in the figure 1.4 as an example.

Next, Relay R3 will receive the communication from R2 using the receive model. The Link in the receiving model is populated with Send_Relay_Model_R2. The Receive_Relay_Model_R3 is as shown in the figure 1.6. Lastly, Relay R4 will receive the communication from R3 using the receive model. The Link in the receiving model is populated with Receive_Relay_Model_R3.

As mentioned in the figure 1.5, the signal *y_s* is the main signal coming out of both the overcurrent blocks. So, basically for relays R1 and R2 the signal *y_s* is the sending signal that is considered for blocking the incoming feeders.

For the incoming feeders, the receiving signal *iblock* as shown in the figure 1.7 is used in the relays R3 and R4.

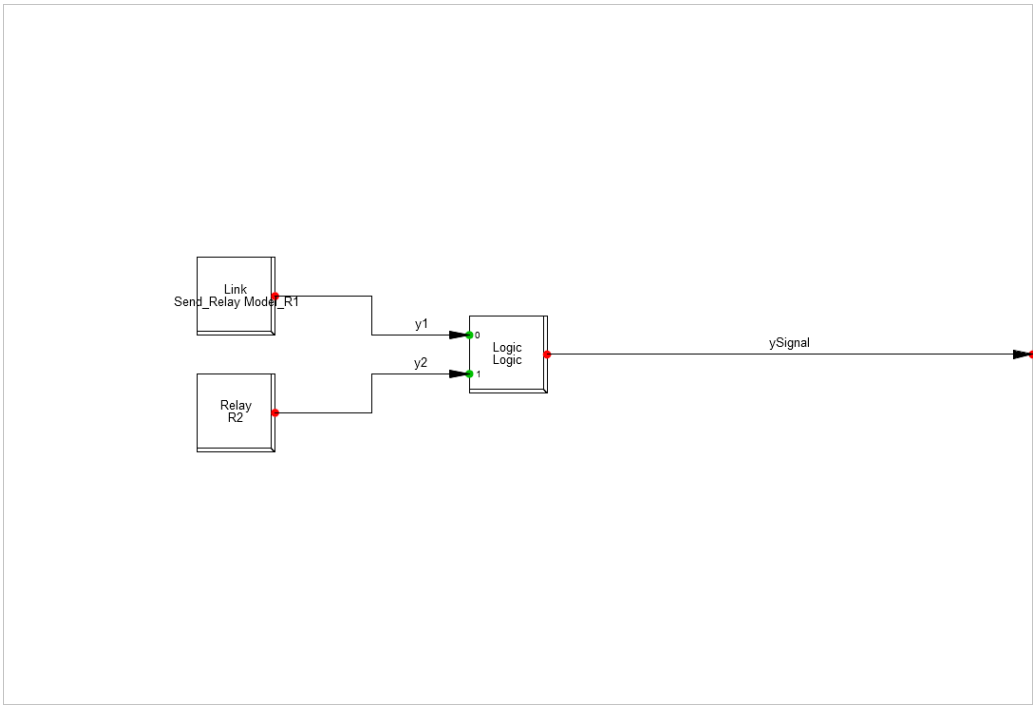

Figure 1.4: Block diagram for Send Relay Model R2

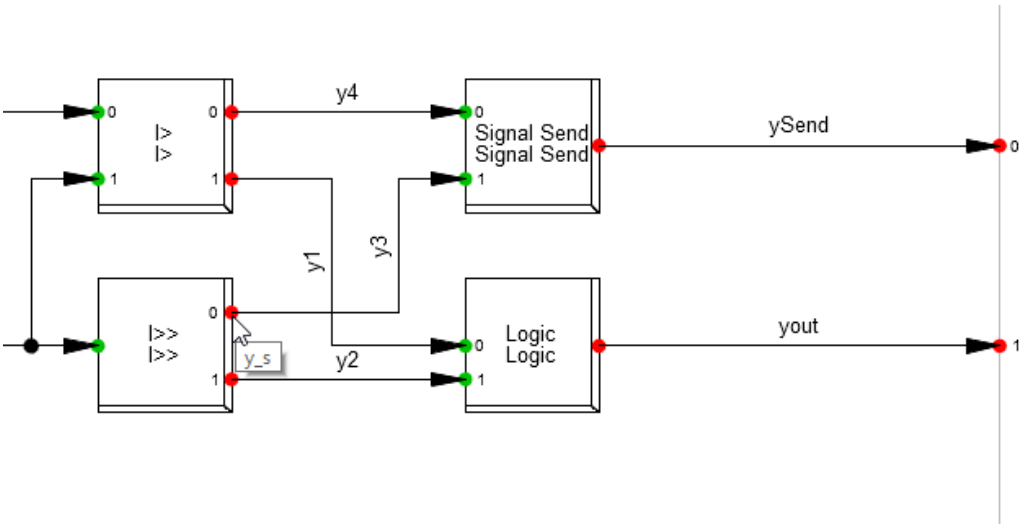

Figure 1.5: Block diagram for Relay Model R2

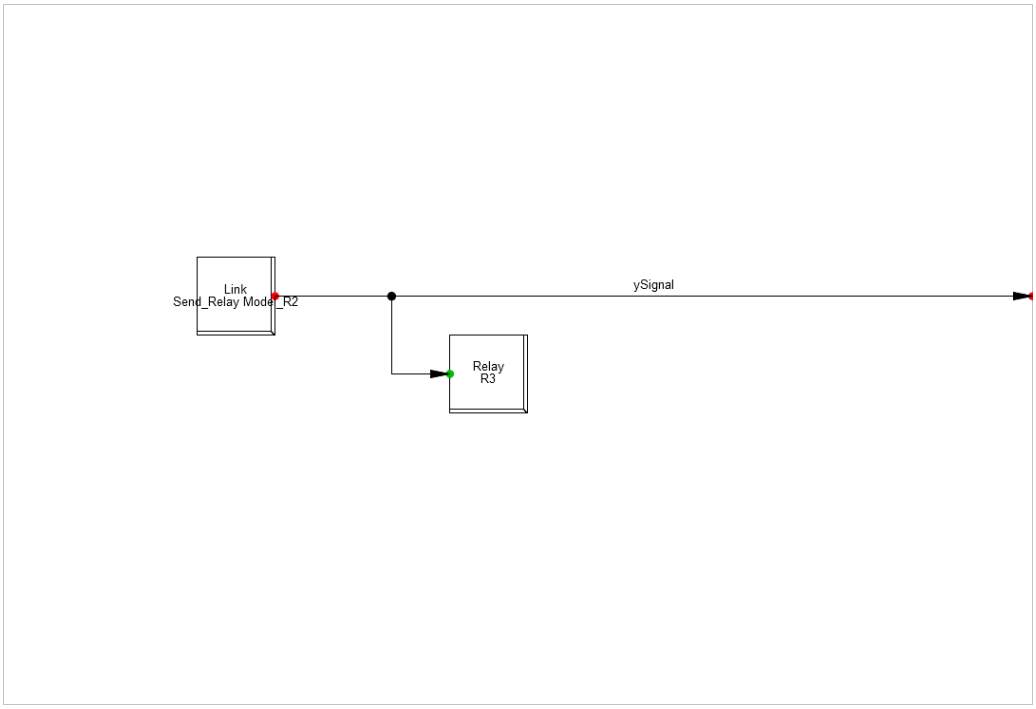

Figure 1.6: Block diagram for Receive Relay Model R3

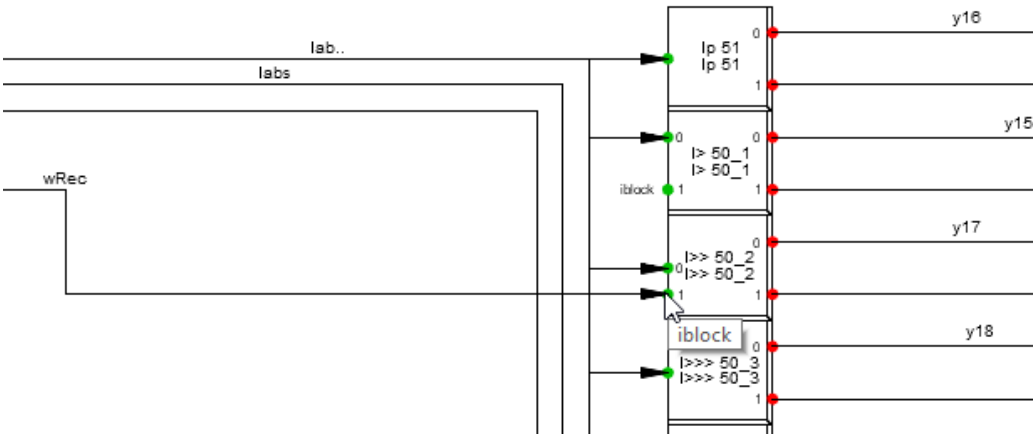

Figure 1.7: Block diagram for Relay Model R3

2 Testing of Interlocking Scheme

The example below shows that if a fault occurs in protection area of relay R1 or R2, the faster stage of R3 and R4 are being blocked. As shown in the figure 2.1, when a fault occurs on the line L_1, the relays R3 and R4 are being tripped in the delayed stage.

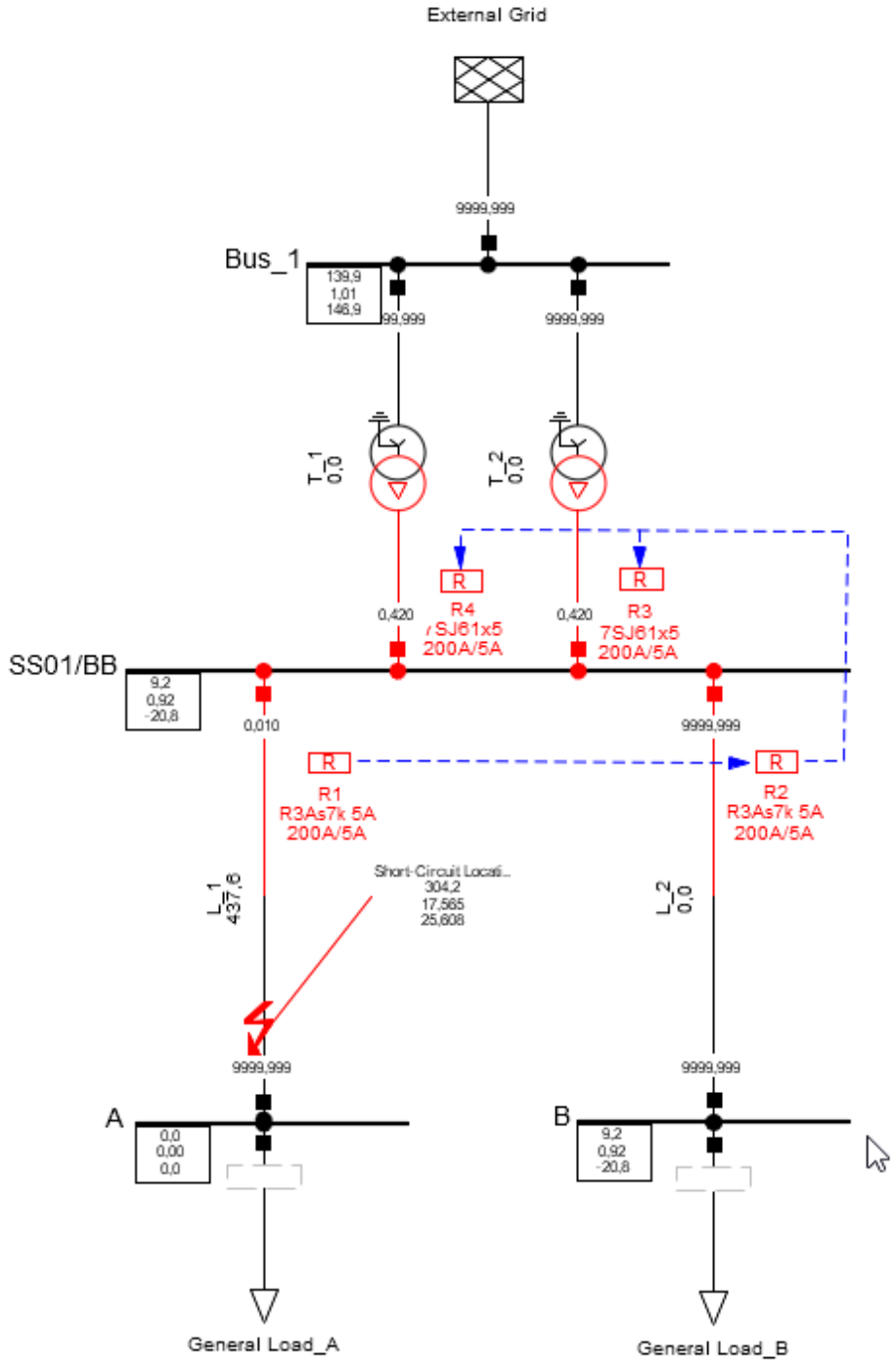

Figure 2.1: When fault occurs on Line L_1

and if the fault occurs at the substation SS01, the faster stages of R3 and R4 will trip, as shown in the figure 2.2.

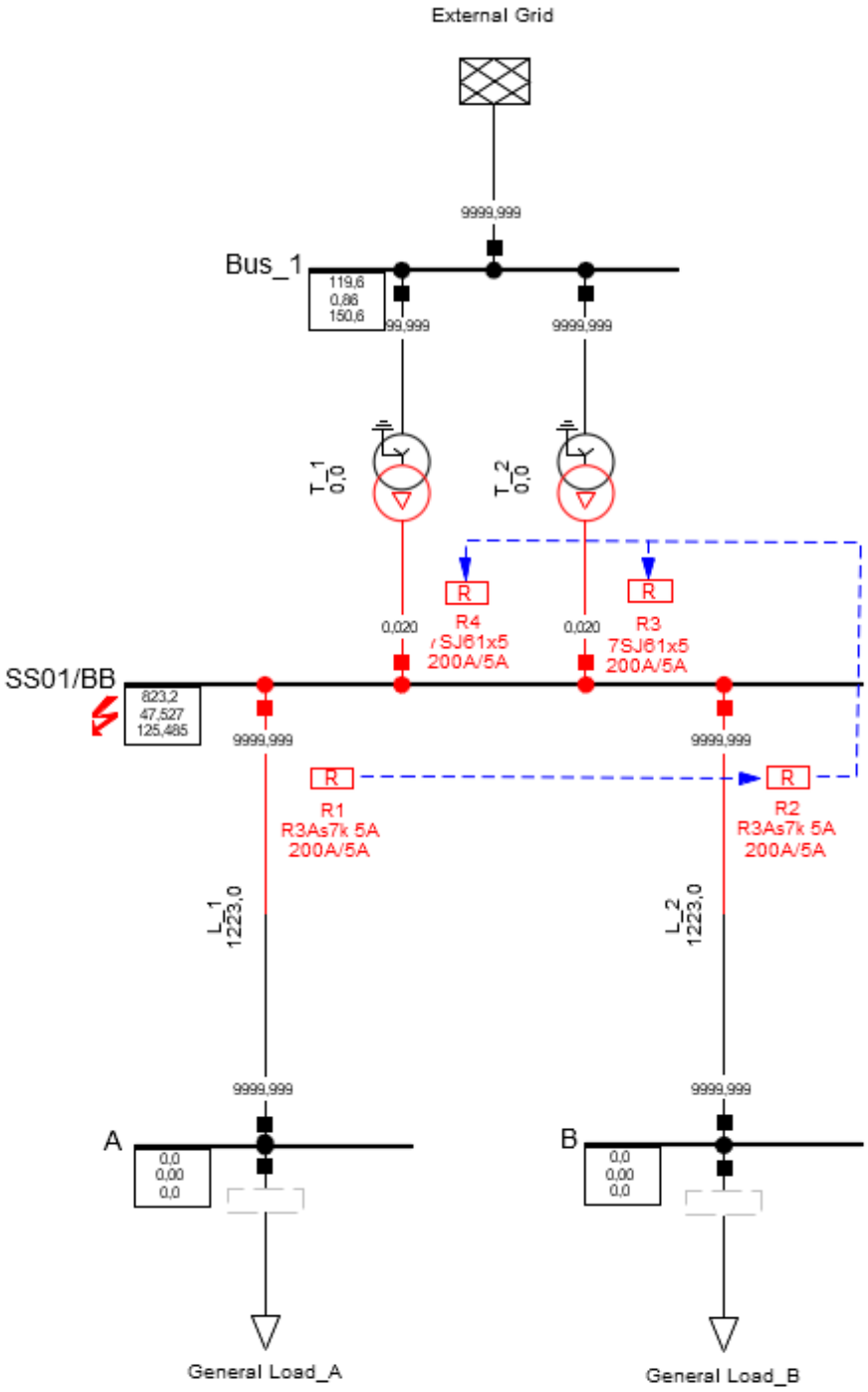

Figure 2.2: When fault occurs on Substation SS01