FUNCTIONAL GLOBAL VARIABLES IN LABVIEW

Creating a Bank Account Framework

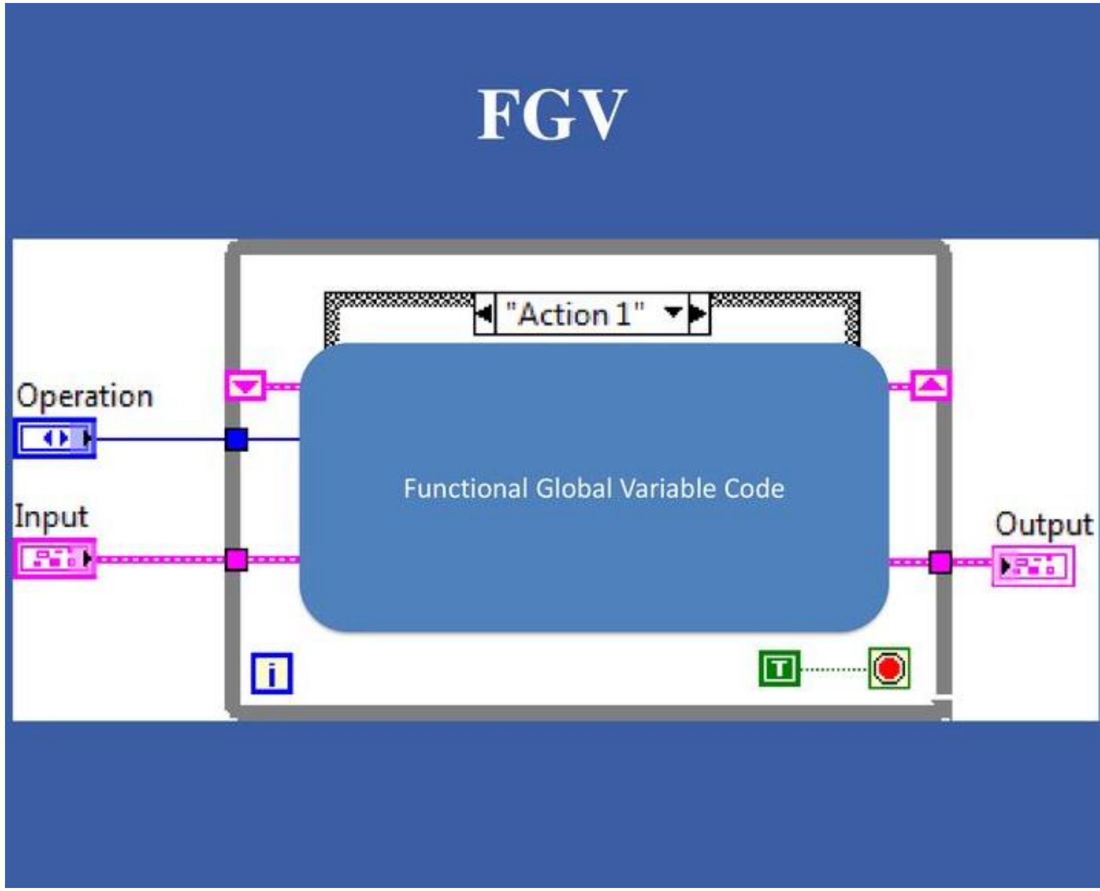


Table of Contents

- Creating the Functional Global variables
- Deposits at Bank Account
- Withdrawals from Bank Account
- Displaying the Current Balance
- Clear
- Exit
- Conclusion
- References

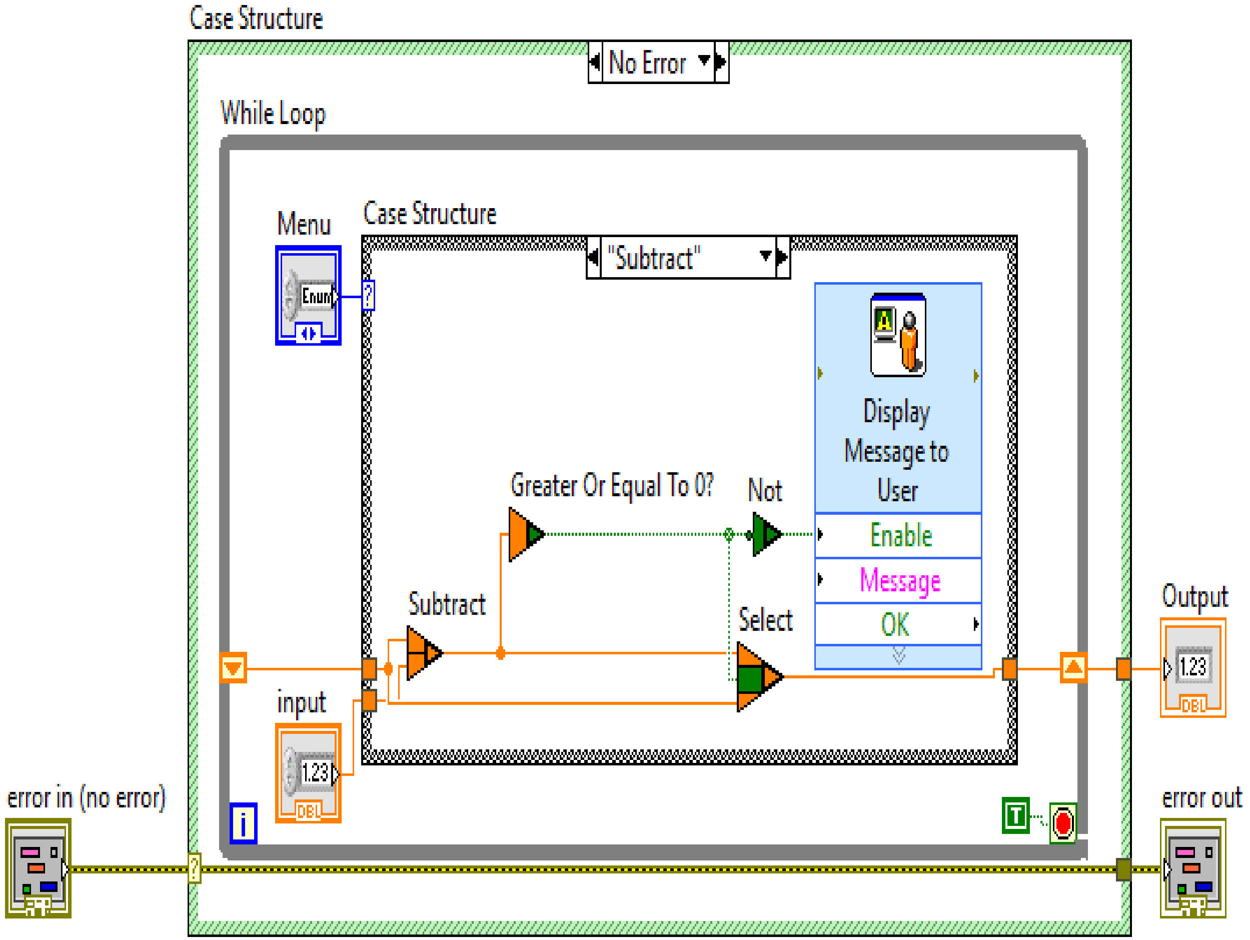
Creating the Functional Global Variable

- A Functional Global Variable (FGV) is a non-re-entrant VI featuring a while loop that iterates once and also has an uninitialized shift register.

- The purpose of this construction is to retain data between consecutive calls to FGV.

- FGV has a case structure with enumerated control to perform operations on the stored values.

- It also provides memory efficiency and avoids race conditions.

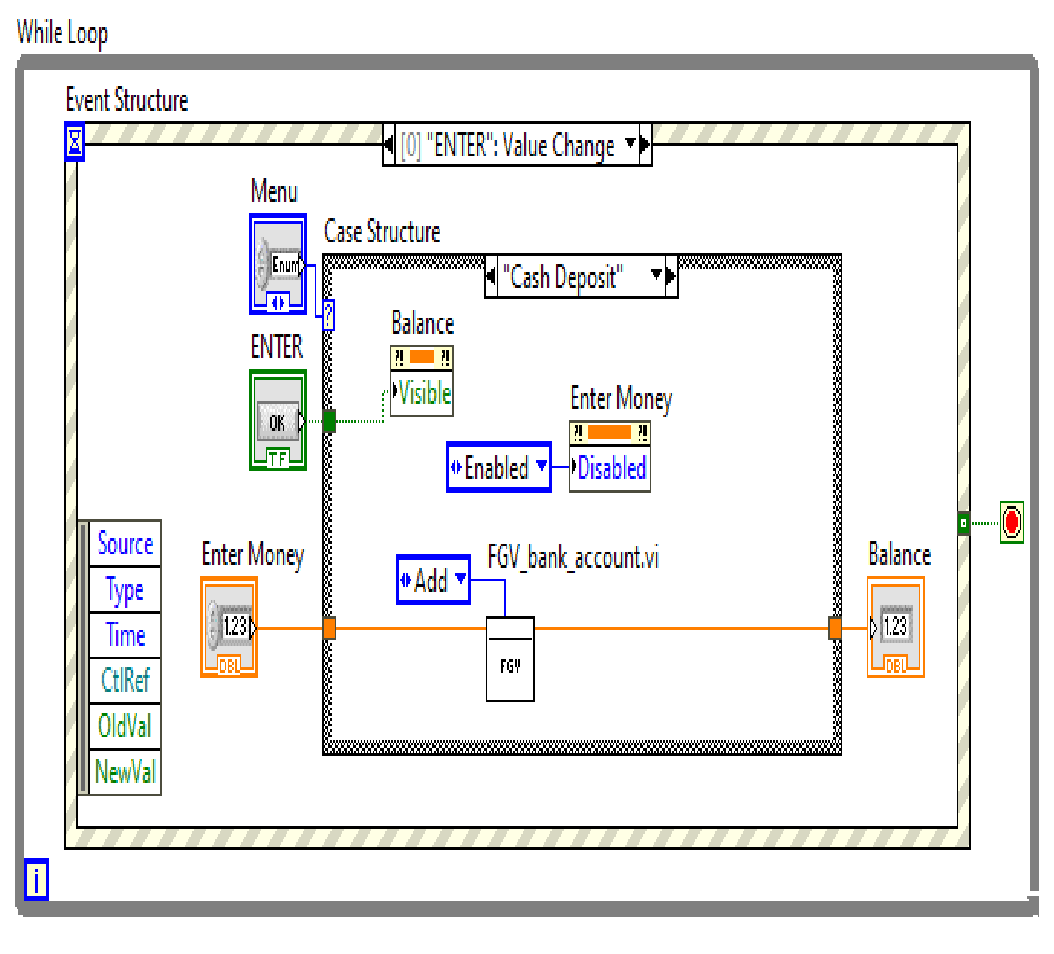


Deposits at Bank Account

- From menu, when the “Cash Deposit” is chosen and the amount is entered, the button triggers an event which executes the below operation.

- The deposit is performed inside the FGV and the balance is shown to the user.

- Inside the FGV, for the current operation the case structure “ADD” is being executed using Numeric Add function.

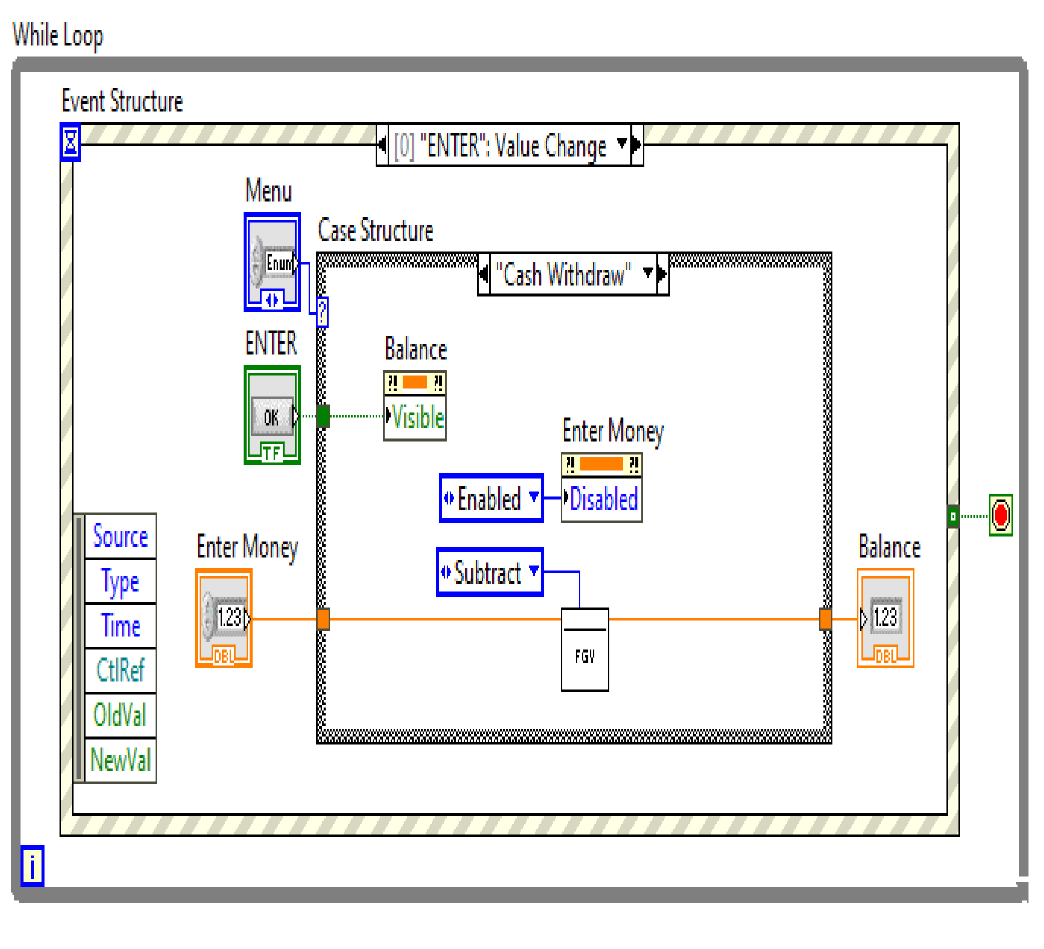


Withdrawals from Bank Account

- From menu, when the “Cash Withdraw” is chosen and the amount is entered, the button triggers an event which executes the below operation.

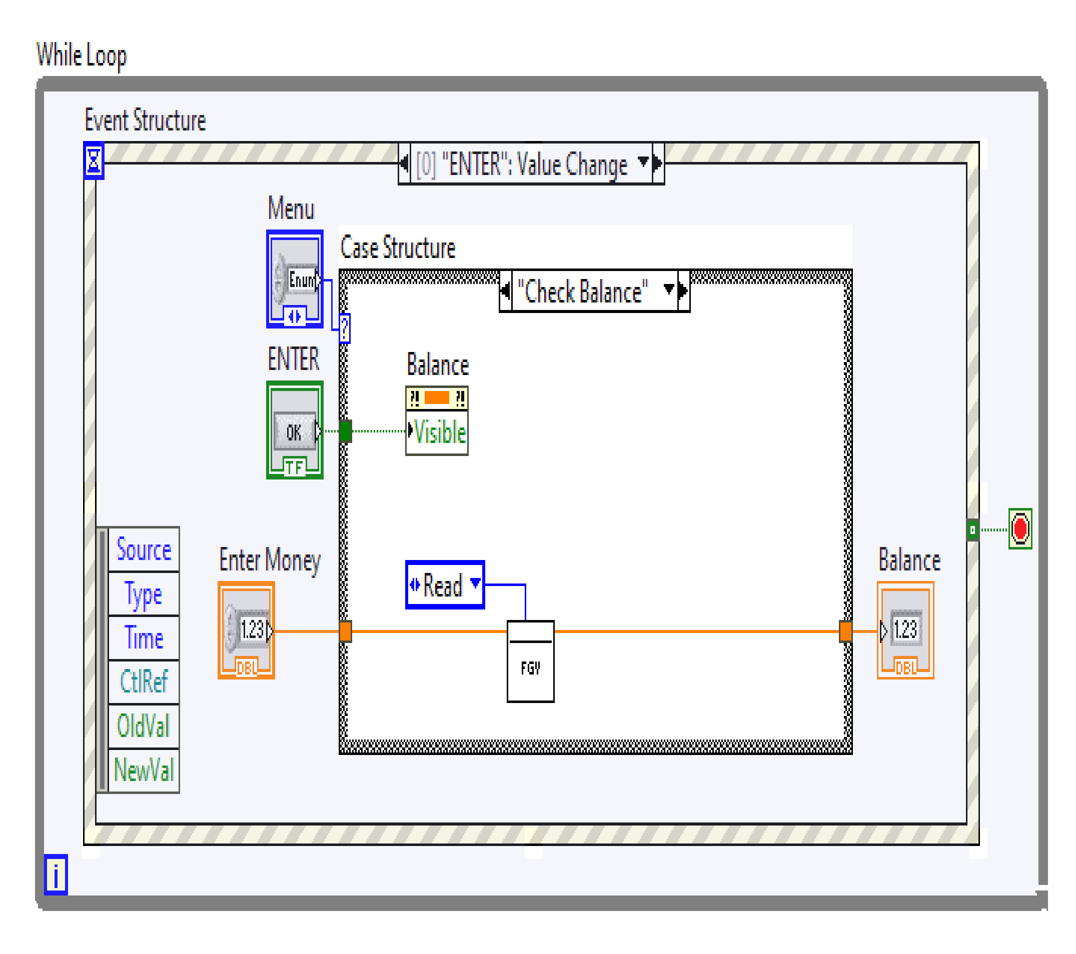
- The withdraw function is performed inside the FGV and the balance is shown to the user. For withdrawals more than available balance, the system displays an error message.

- Inside the FGV, for the current operation the case structure “Subtract” is being executed using Numeric Subtract function.



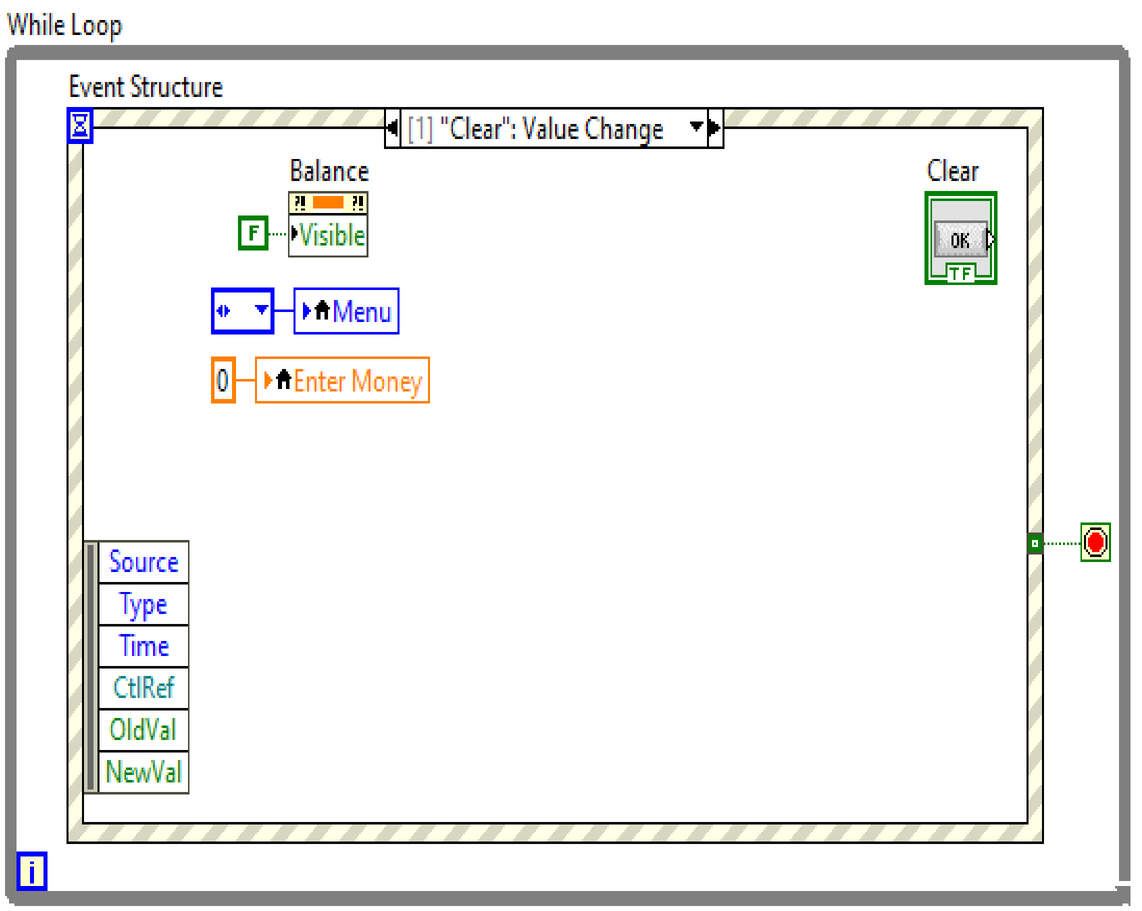
Displaying the Current Balance

- From menu, when the “Check Balance” is chosen, the button triggers an event which executes the below operation.
- The balance function is performed inside the FGV and the balance is shown to the user.
- Inside the FGV, for the current operation the case structure “Read” is being executed and it just reads out the value stored in the shift register.



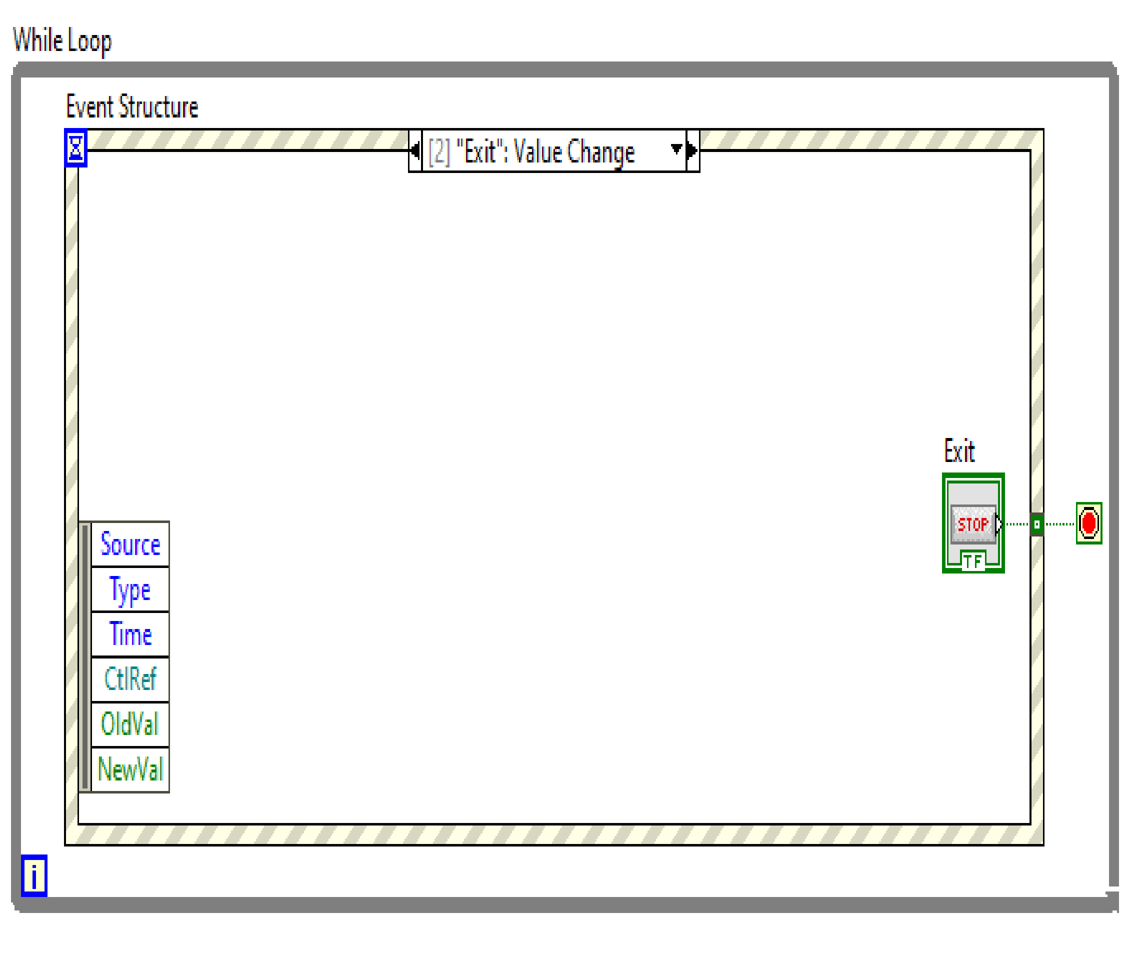
Clear

- From the main screen when Clear button is selected, it triggers an event which executes the below operation.
- The balance field is not displayed on the screen.
- The “Enter Money” text field is set to 0.
- This option is provided to the user to clear out the previously selected operation and choose the new one.
- There is no use case of FGV here.



Exit

- From the main screen when Exit button is selected, it triggers an event which executes the below operation.
- The entire execution of the while loop is stopped along with this the LabVIEW execution is terminated.
- There is no use case of FGV here.



Conclusion

- Functional Global Variable Vis has outer case structure that implements standard error behaviour.
- In this project, the While-loop, Event and inner case structure is used for various operations of the system.
- We also have used the numeric VIs and functions for various arithmetic and comparison operations.
- The shift register is used to store the values in the looping operations. The port on the right side pointing upwards transfers the values to the port on the left side. This shifted value is used for the next operation.

References

- https://learn-cf.ni.com/teach/riodevguide/code/rt_functional-global-variable.html
- https://www.ni.com/docs/de-DE/bundle/labview/page/lvconcepts/shift_registers_concepts.html
- https://www.ni.com/docs/de-DE/bundle/labview/page/glang/numeric_functions.html
- https://www.ni.com/docs/de-DE/bundle/labview/page/glang/property_node.html
- https://labviewwiki.org/wiki/Reentrant_VI