Gram Power Assignment

Directory & File Structure :

```
Project--|
        |-CSR_Requirements - Battery_Balancing_C++.pdf
        |-Code
            |-All_Data
                  |-data.c
                  |-data.h
            |- Battery_processing
                  |- Battery_Operation.c
                  |- BatteryOperation.h
            |- application.c
```

1. **Battery_Balancing_C++.pdf** – This document is Gram Power provided assessment document.
2. **application.c** – This .c file contains the main entry point of the code. There are three APIs in the file :
    - **Function main:** Here the data is collected from the user, stored into the buffers and further simulation is started.
    - **Function app_processing_routine** : This API check for 8v min. level and executes the API which simulates battery cells switch states.
    - **Function app_data_routine** : This API prints the updated data(Voltage ,Current & Switch states) on console for User.
3. **Battery_Operation.c** – This file contains the state machine for switch state sand calculation of currents & voltages.
    - **Function app_SwitchStateManager** : This API is the state manager for all the switch states and is supported by other functions of this file.
    - **Function app_Calculate_Currents_OneSwitch:** This API calculates the voltage left in one cell if one was ON.
    - **Function app_Calculate_Currents_TwoSwitch:** This API calculates the voltage left in two cells if two were ON**.**
    - **Function app_Calculate_Currents_ThreeSwitch:** This API calculates the voltage left in three cells if three were ON.
    - **Function app_CalculateVolatgesLeft_OneSwitch:** This API calculates the current of individual cell and system when one switch is ON.
    - **Function app_CalculateVolatgesLeft_TwoSwitch:** This API calculates the current of individual cell and system when two switches are ON.
    - **Function app_CalculateVolatgesLeft_ThreeSwitch:** This API calculates the current of individual cell and system when three switches are ON.

- **Function app_VoltageCompare:** This API calculates the effective voltage and does the comparison for next state decision.

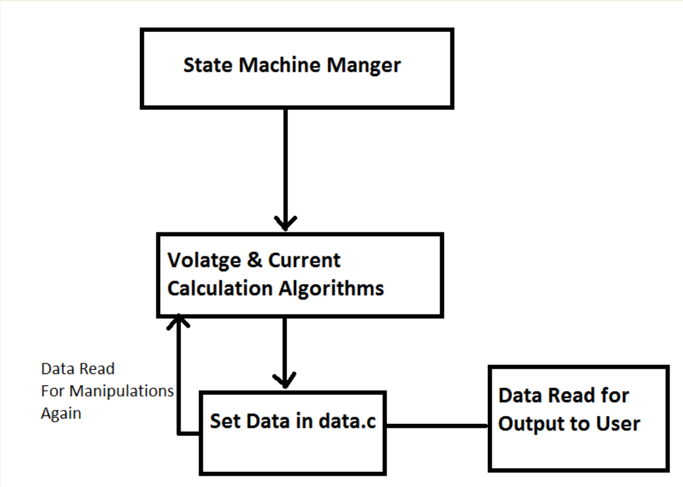
4. **data.c** – Tis files contains the getter setter methods for all the buffers and variable to keep the data abstracted from user read and write.

Implementation :

1. **High Level Design** :

   This big picture of the implementation contains a state machine which always keeps updating the state of the switches. After every 1sec in a state it recalculates the remaining voltages of the cells in order to understand the next probable state of the switches i.e. cell whose voltage is highest will have corresponding switch ON. Also further it calculates the voltage that Load will have in next state, currents for present states.

   Data is abstracted in data.c from user and manipulated only through battery_operation.c.



2. **Low Level Design & Algorithm**:

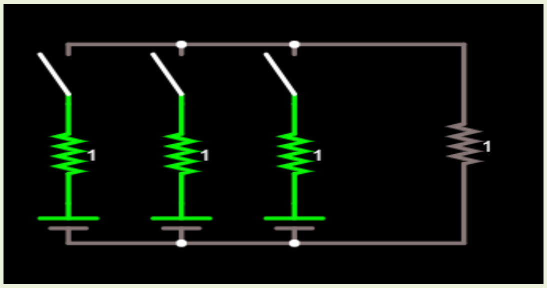
   Consider any switch state, when the assumed state is set the following things happens:
   - Switches which are open, their respective currents are set to zero.
   - Switched which are closed, their respective currents are calculated. This will have four scenarios:
     - No Switch is ON
     - One Switch is ON

  - Two Switches are ON
  - All three switches are ON
- Based on Above calculation system current is calculated as sum of all the available currents.
- System goes in discharge(sleep) for one second.
- Now after discharge of 1 sec, New voltages are calculated. This calculation based on switch status will have four scenarios:
  - No Switch is ON
  - One Switch is ON
  - Two Switches are ON
  - All three switches are ON
- The switches which were OFF have their corresponding voltage discharge 0 for that discharge cycle.
- Now with new voltage next switch state is calculated.

Calculation of Respective Currents

1. **No Switch is ON**

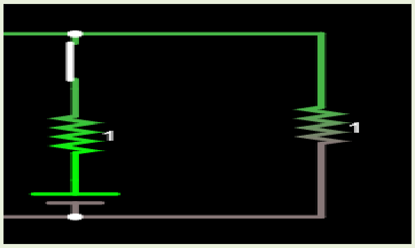
   When no switch is ON the currents will be zero as all the switches are open.



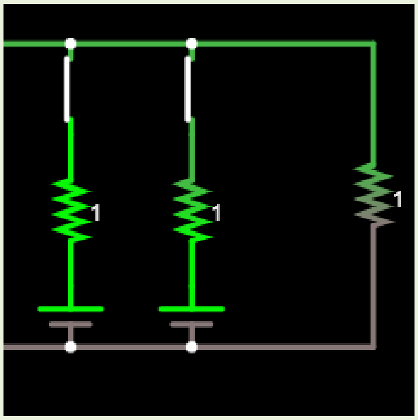
2. **One Switch is ON**

   In case of one switch is on R_1 and R_L goes in series. So total resistance is (R_1+R_L). Voltage would be the voltage of only connected cell i.e. V_n. SO current for particular cell will be:

$$I_n=V_n/(R_1+R_L)$$



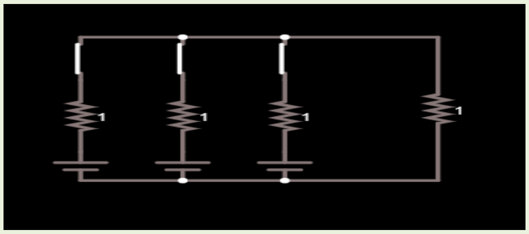
3. Two Switches are ON



In case of two cell C_n and C_m. Resistance for current I_n for cell C_n is (R_n series ($R_m || R_L$)) .

Resistance for current I_m for cell C_m is (R_m series ($R_n || R_L$)).
Voltage will be same if both the cells are connected together. By ohms law we can calculate I_n & I_m individually.

4. Three Switches are ON



This case can be taken as, here cell C_1,C_2,C_3 with current I_1,I_2,I_3, with internal resistance R_1,R_2,R_3 and Load Resistance R_L .

With respect to Cell C_1 R_1 is in series with parallel combination of ($R_2 || R_3 || R_L$)

With respect to Cell C_2 R_2 is in series with parallel combination of ($R_1 || R_3 || R_L$)

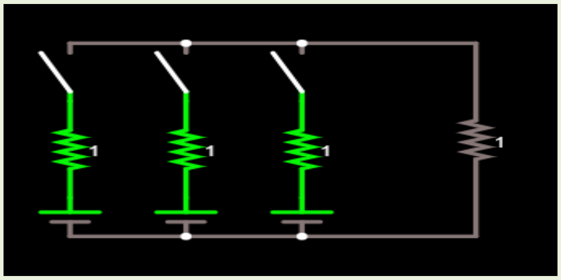
With respect to Cell C_3 R_3 is in series with parallel combination of ($R_1 || R_2 || R_L$)

Voltage will be common as all the cells are ON so voltages are same. With Resistances been known and Voltages been known, By Ohms law we can find I_1, I_1 and I_3.

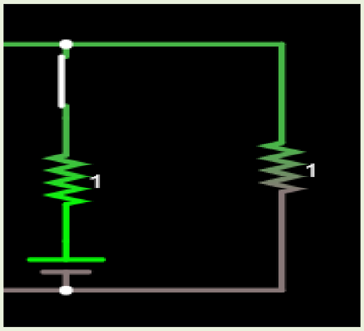
Calculation of Respective Voltages Left

Assumption : The discharge curves of all cells are same with slope = -0.1 (Can be configured through Macro)

1. **No Switch is ON**
   When no switch is ON, no voltage discharge has taken place so voltages remain same.

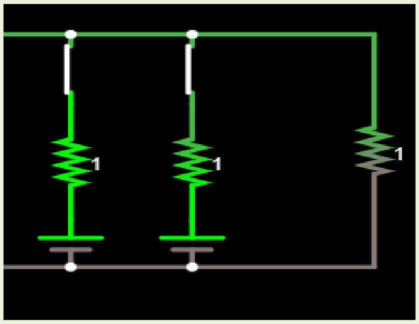


2. **One Switch is ON**



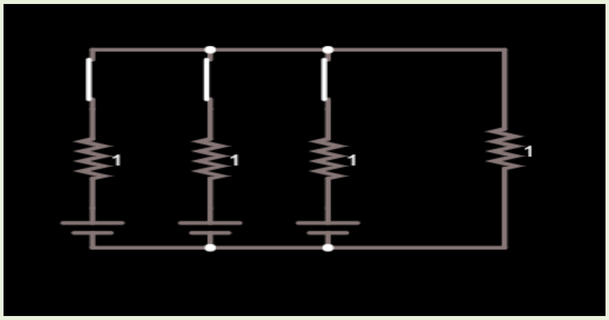
In case of one switch is ON 0.1part of total voltage will be discharged(approx10%). So Left voltage can be calculated by linear subtraction.

3. **Two Switches are ON**



In case of two cell C_n and C_m, Since the load requirement is same and the voltages of both the cells are same discharge will be approx 5%(0.1/2).

4. Three Switches are ON



This case can be taken as, here cell C_1,C_2,C_3, since voltage is same and load is again common as for above cases the discharge time would be 3times less compare to case 2. So discharge will be approx.3.33% (0.1/3).

Calculation of Voltages for Comparison:

NEED : This voltage calculations are needed to considered as voltage drop on Load will not be same as that of a cell voltage. There will be a minimal voltage drop on the internal resistances too.

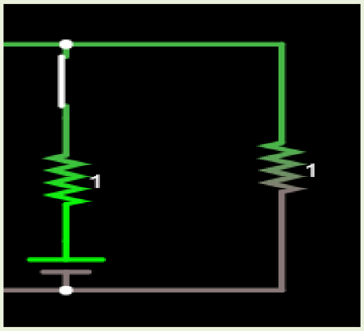
This can be a considerable loss too if in case of old(or deteriorated) cell. So this cannot be neglected in system voltage calculation. Once the individual drops are calculated across the internal resistances then we can have the three nodal resistances for all the three cells.

These potentials can then be compared with each other to find the maximum potential across the load.

Calculation goes as:

Assumption : Cells do not charge each other.

The Aim is to find potential drop across internal resistance R_n from cell C_n. There will not be any effect of other cell as per the mentioned assumption. So with this we can consider circuit as below :



Internal Resistance Rn and the Load RL are in series. Net resistance will be R_n+R_L. Voltage drop would be same as potential of cell C_n viz.. V_n. By ohms law Current In :

$$= V_n/(R_n+R_L)$$

With this we can find drop across Rn as:

$$Vn_{temp} = I_n*R_n$$

By this way we can find effective cell potential for Load RL from L_{th} and m_{th} cell as $VL_{temp\&}$ Vm_{temp} These three Voltages can be compares, and cell with highest effective voltage can be connected into the circuit. Rest all the cells can be kept with opened switch. In this way switch positions can be determined.