# LabVIEW Lesson 5 – Clusters

**Lesson Overview**

- What are clusters?
- How to create a cluster.
- How to create a constant cluster.
- How to use the various cluster functions.

---

I. What is a cluster?

   a. A cluster is similar to the "structure" function in C-Programming.

   b. Clusters allow you to simplify your block diagram by grouping all types of data elements into one terminal. This reduces wire clutter by bundling all the loose wires from each component into one wire for the cluster.

   c. The only restriction with clusters, which is also the case with arrays, is that each cluster can only be either a control or indicator. No mixing of the two types can occur.

II. Creating a **Cluster**.

   a. For a control or indicator cluster, proceed as follows in the front panel:

      i. **Controls Palette → All-Controls → Array & Cluster → Cluster**

      ii. Once the cluster is placed on the front panel, you may resize it by dragging on one of the blue boxes on its border. Also notice that the cluster is empty, so you can determine the type by inserting either controls or indicators inside.

      iii. One very important item to note when creating a cluster is the order of its contents. As you insert components into a cluster, they are assigned a number that corresponds to the order in which they were inserted.

         1. Cluster order follows the same numbering pattern as arrays. The first component is assigned as the 0 element, the second as the 1 element, and so on.

         2. Why is cluster order important?

            a. The cluster order determines the order in which the elements appear as terminals on the **Bundle** and **Unbundle** functions, which will be covered later.

            b. When connecting a control cluster with an indicator or constant cluster, each cluster's contents must have the same data type order so that all elements are compatible between each connected cluster.

3. The cluster order can be viewed and modified by…
   - **Right-Click** on **Cluster Border** → **Reorder Controls in Cluster** → *(Should change the toolbar and clusters similar to* ***Figure 5.1****.)*

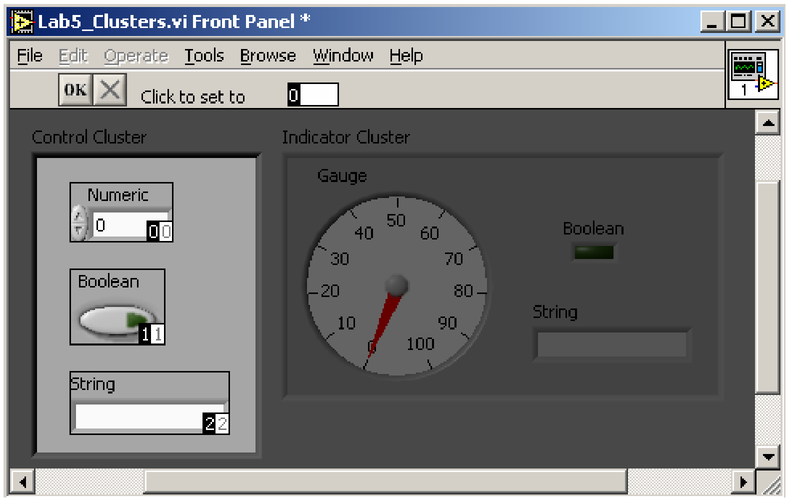


Figure 5.1: Cluster Reordering Window and Toolbar

a. The new toolbar contains a **Confirm Button**, a **Cancel Button**, and a "**Click to Set to**" **Text Box**, which indicates the position that will be assigned when the element is selected.

b. Next to each element is a white and black box. The white box indicates the element's current position, while the black box indicates the element's new position.

c. To change the order:

   1. Type the new order number in the **"Click to Set to" Text Box** in the toolbar.

   2. **Left-Click** on the element that you wish to change and the others will adjust accordingly.

   3. **Left-Click** on **Confirm Button** to save changes or **Cancel Button** to revert back to original order.

iv. **Example 5.1**: Simple Control and Indicator Clusters

1. This example shows how to construct simple control and indicator clusters on the front panel as well as how to ensure that the cluster orders are compatible with one another.

   a. Creating the **Control Cluster**

   i. Insert a **Cluster** on the **Front Panel** and **Resize** it to fit three different controls.

   ii. Insert a **Numerical Control**
      - **Controls Palette → Num Ctrls → Num Ctrl**

   iii. Insert a **Boolean Push Switch**
      - **Controls Palette → Buttons → Push Switch**

   iv. Insert a **String Control**
      - **Controls Palette → Text Ctrls → String Ctrl**

b. Creating the **Indicator Cluster**
   i. Insert another **Cluster** on the **Front Panel** and **Resize** it so that it is about twice as wide as the other.

   ii. Insert a **Gauge**
      - **Controls Palette → Num Inds → Gauge**

   iii. Insert a **String Indicator**
      - **Controls Palette → Text Inds → String Ind**

   iv. Insert an **LED**
      - **Controls Palette → LEDs → Square LED**

c. Check the cluster order of the control cluster and then, verify that the cluster order of the indicator cluster is the same.  The Boolean and String elements should be out of place.  To reorder the elements proceed as follows:

   i. Type **1** in the "**Click to Set to**" **Text Box**.

   ii. **Left-Click** inside the black box that surrounds the **Boolean LED** *(this should change the **2** in the black box to a **1** for the LED and the **1** to a **2** in the black box for the String)*. The Front Panel should look similar to **Figure 5.2**, which shows the original order in the white boxes and the new order in the black boxes.

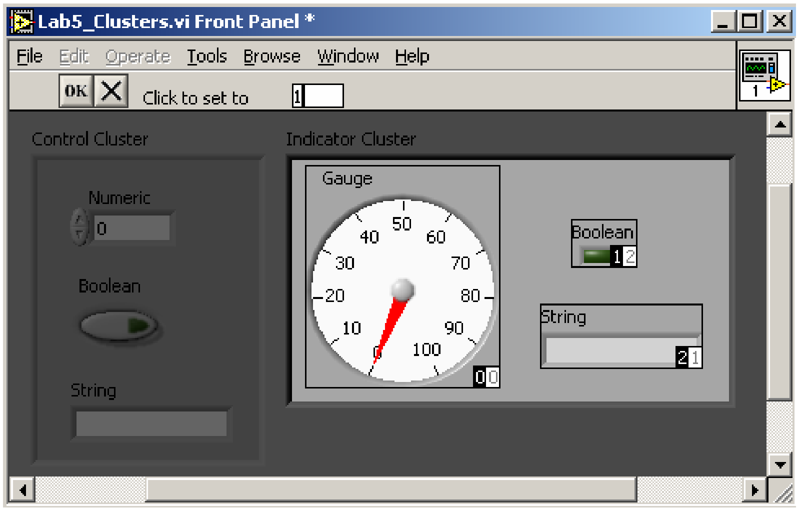


Figure 5.2: Indicator Cluster Order with Boolean Position Change

iii. **Left-Click** on the **Confirm Button** to accept the new order, which correctly pairs the data types in the two clusters.

d. The **Front Panel** should look similar to **Figure 5.3**.

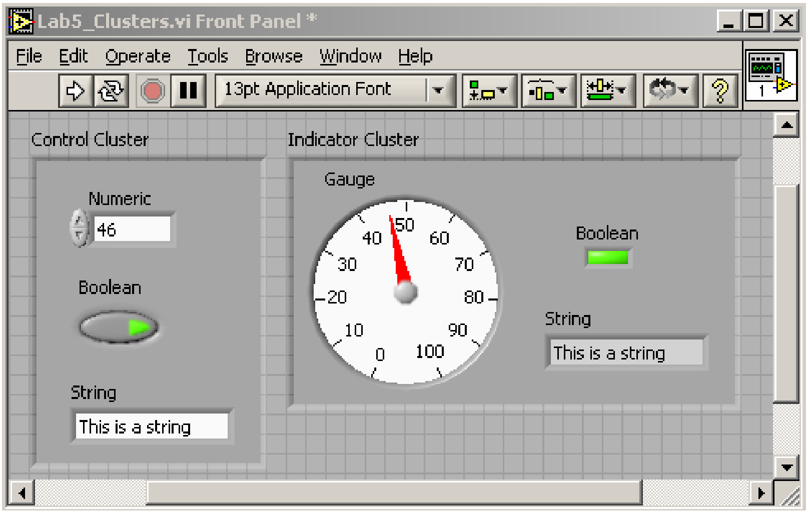
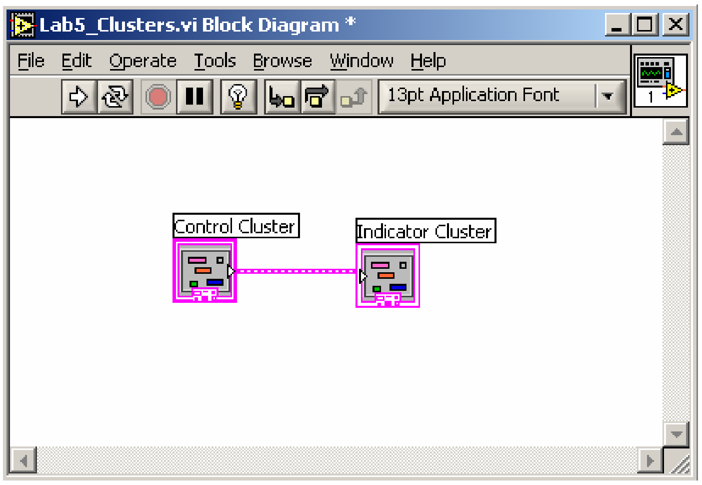


Figure 5.3: Front Panel of Simple Control and Indicator Clusters

e. Switch to the **Block Diagram** and **Wire** the **2 Clusters** together so that it looks similar to **Figure 5.4**.


Lab5_Clusters.vi Block Diagram *
File Edit Operate Tools Browse Window Help
13pt Application Font
Control Cluster
Indicator Cluster
Figure 5.4: Block Diagram of Simple Control and Indicator Clusters

f. **Left-Click** on the **Run Continuously Button** and then, try typing inputs into the **String Control**, switching the **Boolean Push Button**, and changing the **Numerical Control** value to see how it affects the **Indicator Cluster**.

b. For a **Cluster Constant**, proceed as follows in the **Block Diagram**:

i. **Functions Palette → All-Functions → Cluster → Cluster Constant**

ii. The cluster constant will be a blank box with a pink border. You can put any type of constant into this cluster as long as the data types are compatible with its mating cluster.

iii. An example of a constant cluster that corresponds to the clusters previously created is shown in **Figure 5.5**.

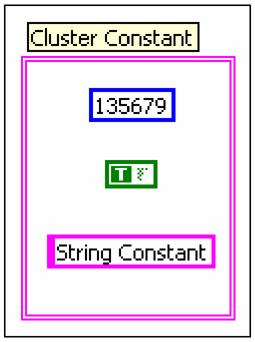


Figure 5.5: Cluster Constant in Block Diagram

III. Using the **Bundle** and **Unbundle** Cluster Functions.

a. 4 main cluster functions are available; however, 2 are assembly functions while the other 2 are disassembly functions:

i. **Bundle** and **Bundle by Name** :

1. These two functions perform the same operation; however, their terminals on the block diagram differ. Both functions can be used to either assemble or "bundle" elements into a new cluster or to change the values in an existing cluster without having to assign new values to all the elements.
2. The main difference between the two bundle functions is that the **Bundle** function lists the elements by data type in the terminal on the block diagram, while the **Bundle by Name** function lists the elements by their name.
3. Also, the **Bundle** function requires that the number of inputs into the function match the number of elements in the output cluster. The **Bundle by Name** function does not have this requirement as it references the data by name and not by element order.
4. When using the Bundle by Name function, choose which elements to display as terminals by **Left-Clicking** on the terminal of the element to change and then, **Selecting** the element that you want from the **Pull-Down Menu**. The **Unbundle by Name** function follows this same procedure.

ii. **Unbundle** and **Unbundle by Name** :

1. These two functions also perform the same operation, which is to split up or "unbundle" a cluster into its individual components.
2. The **Unbundle** function splits the cluster into its individual components based on cluster order and data type.
3. The **Unbundle by Name** function allows the user to choose which elements to extract from the cluster based on each elements name.
4. The same regulations exists for the unbundle functions as did for the bundle functions. The **Unbundle** function will have the same number of output terminals as the number of elements in the cluster, while the **Unbundle by Name** function only displays terminals for the elements the user chooses.

iii. To access any of these functions:
- **Functions Palette → All-Functions → Cluster → Desired Function**

b. **Example 5.2**: Temperature VI with Clusters.

i. PROBLEM: The user would like to acquire the temperature at three different points along a metal specimen that is exposed to a heat source on one end and a cold bath on the other. The thermocouples will be placed one near the heat source, one on the cold bath end, and one halfway between the two.

ii. This VI will show how clusters may be incorporated into a VI to solve this problem. Additionally, the use of the different bundle and unbundle functions will be demonstrated as well as how to incorporate a constant cluster. Note: Actual temperature measurements will not be used; instead, the temperatures will be simulated using a random number generator.

iii. Begin by opening up "**TempForLoop.VI**" that we created in Lesson 4.

iv. Starting with the **Block Diagram**.

1. Delete the **Boolean Toggle Switch Terminal** and Replace it with a **Boolean True Constant**.
    - **Functions Palette → Arith/Compare → Boolean → True Constant** 🟩 **→ Wire to "Thermometer" SubVI**
2. Delete the terminals for the "**Temperature (Deg F)**" **Chart** and the "**Current Temp**" **Numerical Indicator**.
3. Compact the remaining **Temperature Group** and Copy **twice** *(Select group → Hold Ctrl and Drag)*. These will simulate the three temperature readings since we don't have access to thermocouples and DAQ (Data Acquisition) equipment.
4. Move the **Iteration Terminal** and everything attached to it to the **top** of the **For Loop**.
5. You may need to stretch out the **For Loop** so that you have plenty of room to work.
6. The **Block Diagram** should look similar to **Figure 5.6**.

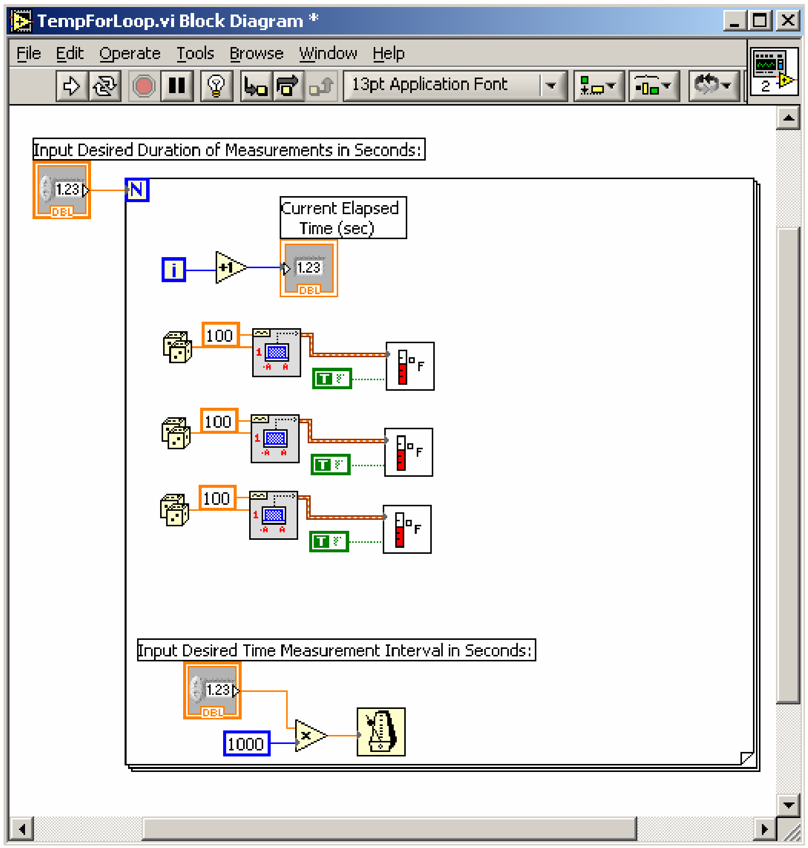

TempForLoop.vi Block Diagram *
File Edit Operate Tools Browse Window Help
13pt Application Font
Input Desired Duration of Measurements in Seconds:
Current Elapsed Time (sec)
100
100
100
Input Desired Time Measurement Interval in Seconds:
1000

Figure 5.6: Modified Block Diagram for TempForLoop.VI

v. Switch to the **Front Panel**.

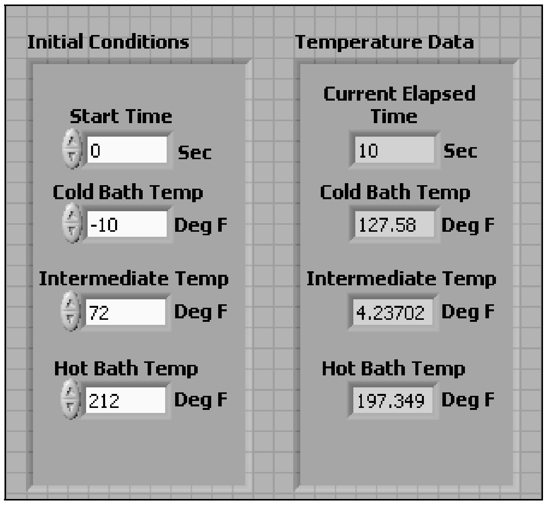
1. Delete the following text: "**Indicate Temperature Scale**…", "**Deg C**", and "**Deg F**"

2. Insert **2 Clusters** (one will be for initial values and one will be for actual current values).
   - **Controls Palette → All-Controls → Array & Cluster → Cluster**

   a. Name one cluster "**Initial Conditions**" and the other "**Temperature Data**"

   b. Configure the **2 Clusters** as shown in **Figure 5.7** (Ignore the values).

   c. **Note**: The "**Current Elapsed Time**" **Numerical Indicator** was drug into the "**Temperature Data**" **Cluster**.



Initial Conditions

Start Time 0 Sec

Cold Bath Temp -10 Deg F

Intermediate Temp 72 Deg F

Hot Bath Temp 212 Deg F

Temperature Data

Current Elapsed Time 10 Sec

Cold Bath Temp 127.58 Deg F

Intermediate Temp 4.23702 Deg F

Hot Bath Temp 197.349 Deg F

Figure 5.7: Temperature VI Cluster Configurations

3. Insert **2 Waveform Charts**, which will be used to plot the temperature data over time and allow for any overall trends to be observed.
   - **Controls Palette → Graph Inds → Waveform Chart**

   a. Name one graph "**Temperature Plot**" (*This one will be used to display the temperature at each location.*) and the other "**Temperature Changes**" (*This one will be used to display the change in temperature at each location.*)

   b. In order to display the 3 temperatures on each graph, expand the top-right box upwards by dragging its border. This box will serve as your Legend, so you will want to rename the plot lines.

   c. Each **Waveform Chart** should look similar to **Figure 5.8** without the anything plotted. Plotting techniques will be discussed in the next lesson.

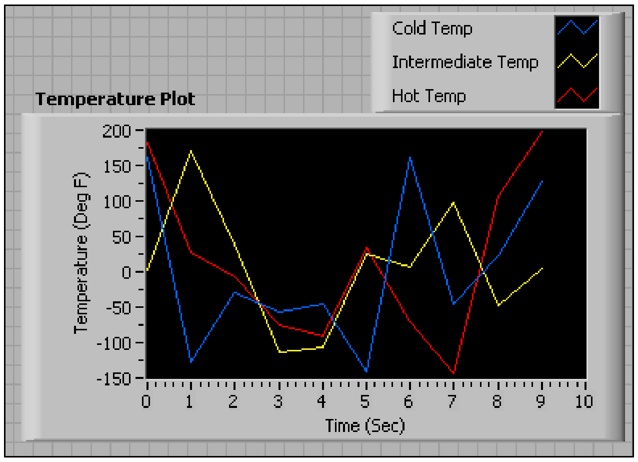
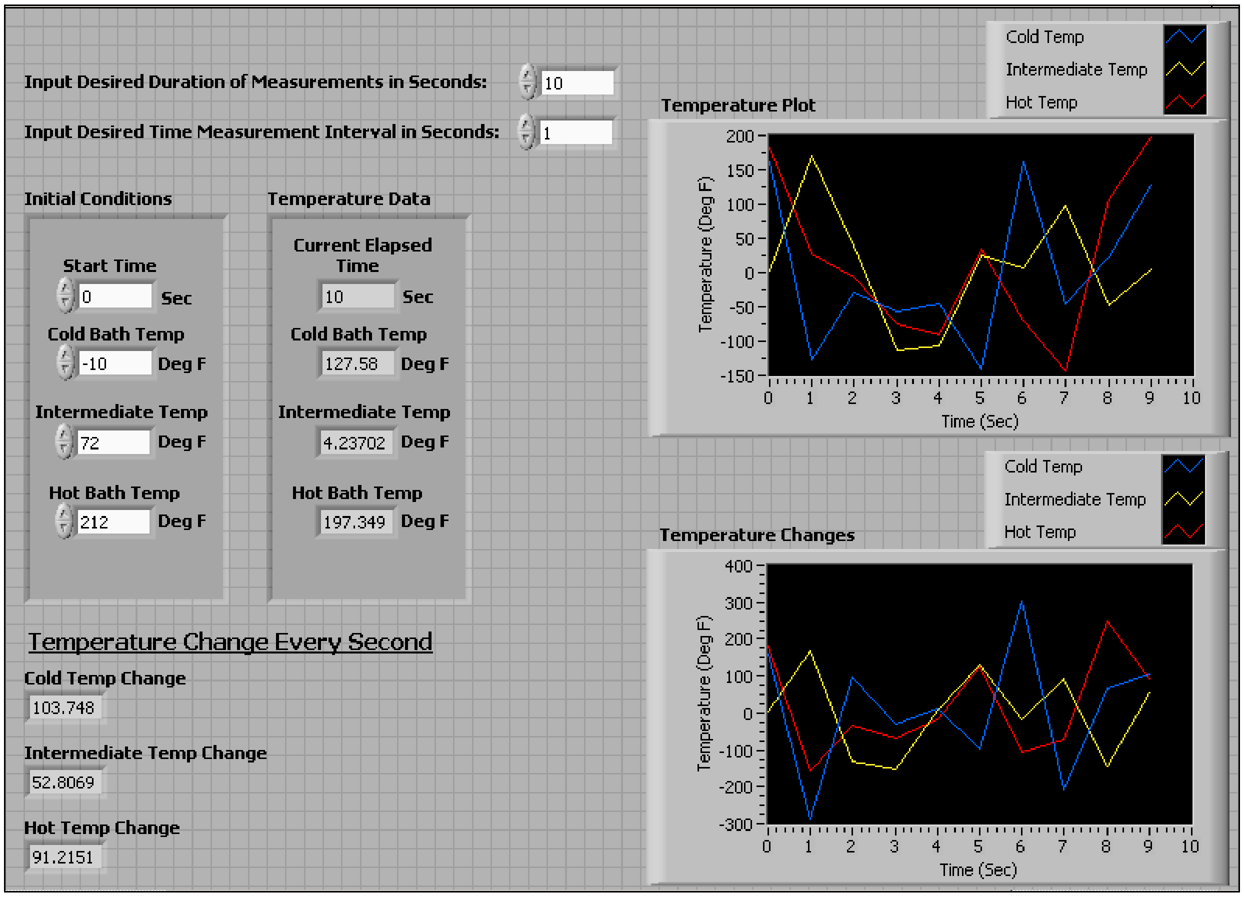


Figure 5.8: Temperature Plot Waveform Chart Configuration

4. Lastly, insert **3 Numerical Indicators**, which will be used to display the instantaneous change in temperature for each location. Name the **Numerical Indicators** appropriately.

5. The **Front Panel** should now look similar to **Figure 5.9** without all the data.


Input Desired Duration of Measurements in Seconds: 10

Input Desired Time Measurement Interval in Seconds: 1

Initial Conditions

Start Time 0 Sec

Cold Bath Temp -10 Deg F

Intermediate Temp 72 Deg F

Hot Bath Temp 212 Deg F

Temperature Data

Current Elapsed Time 10 Sec

Cold Bath Temp 127.58 Deg F

Intermediate Temp 4.23702 Deg F

Hot Bath Temp 197.349 Deg F

Temperature Change Every Second

Cold Temp Change 103.748

Intermediate Temp Change 52.8069

Hot Temp Change 91.2151

Figure 5.9: Front Panel for Temperature Cluster VI

vi. Switch to the **Block Diagram.**

1. Drag all elements created on the Front Panel into the For Loop.

2. Currently, 3 temperatures are being generated as well as the time at which the temperature was taken. The 3 temperatures and the time generator need to be bundled so that they can be wired to the "**Temperature Data**" **Cluster**.

   a. Insert a **Bundle by Name** function inside the **For Loop**
      - **Functions Palette → All Functions → Cluster → Bundle by Name**

   b. Wire the "**Initial Conditions**" **Cluster Terminal** to the **Top Terminal** of the **Bundle by Name** function. **4 Terminals** should appear with the Names of the inputs to the **Bundle by Name** function.

   c. Wire the **Iteration Terminal** group to the top terminal "**Start Time**" of the **Bundle by Name** function. Also, connect the 3 temperature generator groups to their appropriate terminals.

   d. Finally, wire the **Output Terminal** of the **Bundle by Name** function to the "**Temperature Data**" **Cluster**.

3. Now the instantaneous temperatures will be displayed in the "**Temperature Data**" **Cluster**, but the user would also like a plot of all 3 temperatures for the duration of the experiment.

   a. Insert the **Unbundle by Name** function inside the **For Loop**, which will be used to extract just the 3 temperatures, and resize it by dragging its bottom border down so that 3 terminals are showing.
      - **Functions Palette → All Functions → Cluster → Unbundle by Name**

   b. Branch a wire from the wire connecting the **Bundle by Name** function to the "**Temperature Data**" **Cluster** to the **Input Terminal** of the **Unbundle by Name** function. (*Again, the names of the elements inside the cluster will appear in the terminals.*)

   c. Change the names in the terminals so that the 3 temperatures are displayed and that the **Cold Temp** is on **Top** and the **Hot Temp** is on **Bottom**.
      - **Left-Click on Terminal** (*Drop-Down Menu should appear*) **→ Select Name of Element to assign to Terminal**

   d. Insert the **Bundle** function so that we can wire them into the "**Temperature Plot**" **Waveform Chart**, which only accepts one input.

   e. Wire the **Outputs** from the **Unbundle by Name** function to the **Inputs** of the **Bundle** function and then, wire the **Bundle** function's **Output** to the "**Temperature Plot**" **Waverform Chart**. (*You will notice that the Bundle function terminals only show data types instead of names.*)

4. Lastly, the user wants to display the instantaneous temperature changes as both a digital readout and a graph.

   a. Add a **Set of Shift Registers** so that the previous loop's temperatures are stored for comparison to the current loop's temperatures.

- **Right-Click** on the **For Loop Border → Add Shift Register**

b. Branch a wire from the wire connecting the **Bundle** function to the “**Temperature Plot**” **Waveform Chart** to the **Right Shift Register**.

c. Create a **Cluster Constant** outside the **For Loop** next to the **Left Shift Register** and wire to the **Left Shift Register**. This cluster will initialize the **For Loop** and **Shift Register** every time the VI is ran.

d. Insert **3 Numerical Constants** inside the **Cluster Constant** and leave at 0, since the initial temperature change is 0.

e. Insert the **Subtract** function, branch a wire from the wire connected to the **Right Shift Register** to the **Top Terminal** of the **Subtract** function, and then wire the **Left Shift Register** to the **Bottom Terminal**.

f. Wire the **Subtraction Output** to the “**Temperature Changes**” **Waveform Chart**.

g. Now to display the individual temperature changes, insert the **Unbundle** function.

h. Wire the **Subtraction Output** to the **Unbundle** Input and the **Unbundle Outputs** to their corresponding **Numerical Indicators**. *Remember, the order of the elements in the cluster as these should correspond to the individual indicators.*

5. Your **Block Diagram** should look similar to **Figure 5.10**.

6. Save VI as “**3TempCluster.VI**” and then, initialize some values and run.

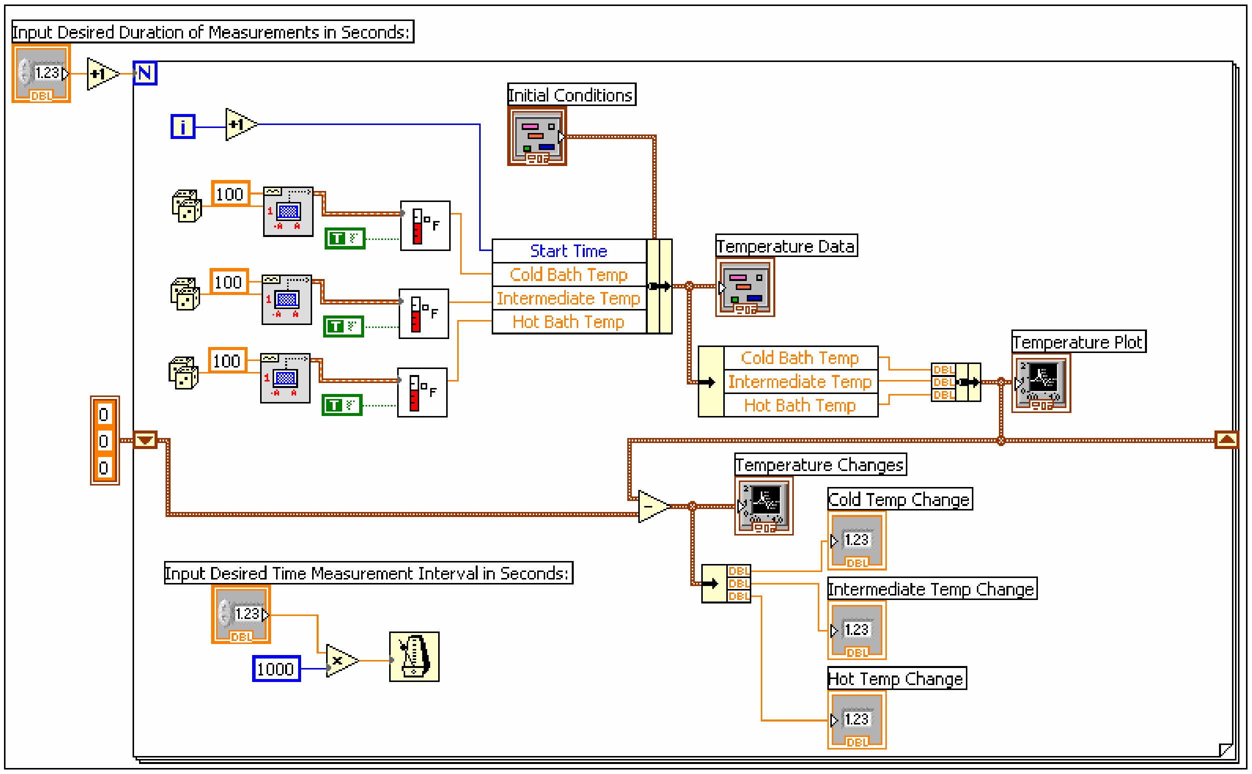


Figure 5.10: Block Diagram for Temperature Cluster VI