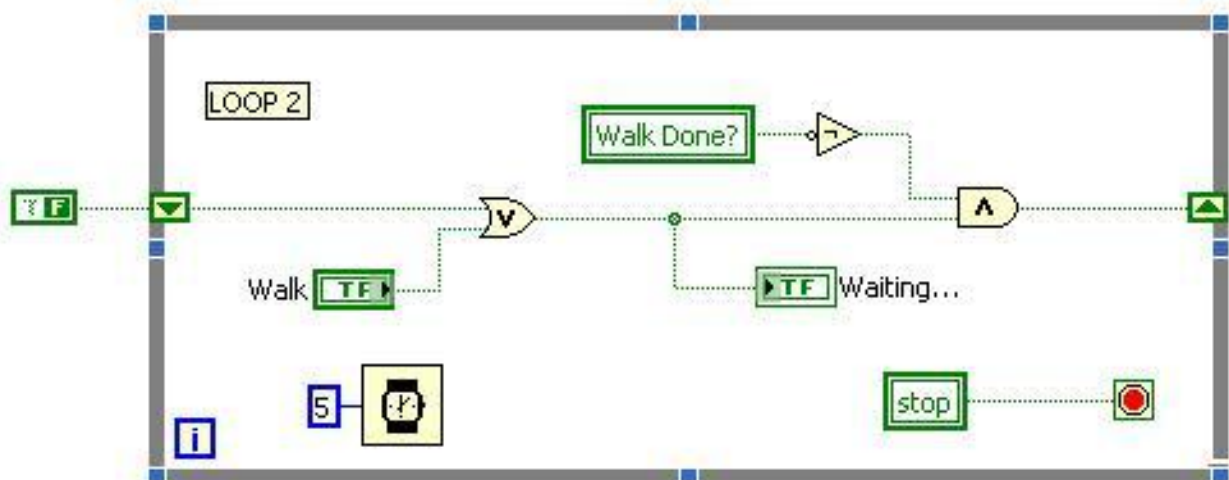


**Post-Grad  
LabVIEW Course  
Exercise 13  
G Boorman**

## Exercise 13 – Improving the Pedestrian Lights

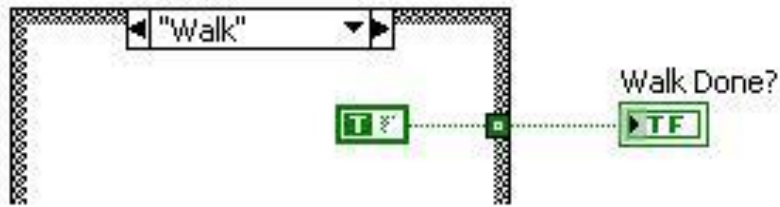
This exercise overcomes the limitation of the previous Pedestrian Lights. A second, parallel, While Loop is created to assess the status of the *Walk* button to ensure only one *walk* is allowed per cycle of the lights and to prevent the *Walk* button being ‘un-pressed’. Two *local variables* are used to communicate between the parallel loops.

- 1) Make a copy of *Simple Pedestrian Lights.vi* and name it *Improved Pedestrian Lights.vi* and add it to the Project.
- 2) Ensure the *Walk* button has the Mechanical Action *Latch When Pressed*.
- 3) Add a second While Loop to the diagram and include a shift register. Create a front panel Boolean Indicator, named *Walk Done?*, and then do Advanced>>Hide Indicator for this indicator.
- 4) Add a second Boolean Indicator named *Waiting...*
- 5) Make the contents of the second While loop look like the following:



- a) Add a *local variable* (from structure palette), right-click and do Select Item>>Walk Done?. Right-click and do Change to Read.
  - b) Add a second *local variable*, link to Stop and do Change to Read.
  - c) Make sure you understand what the code in this diagram.
- 5) In the original While Loop add another *local variable*, link it to *Waiting...* and wire to the selector in case *Red*.

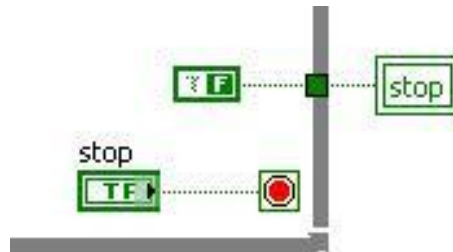
- 6) In the *Walk* case, add a *True* constant and wire to *Walk Done?*. Right-click the tunnel and do *Use Default If Unwired*. It should look like the following:



A Boolean value will default to *False*, and since only one case needs to be *True*, using *Use default If Unwired* can be done in this case.

**Note:** Other data-types all have defaults (integers default to Zero, strings to Empty Strings etc), but be very careful if using *default* on custom controls – unexpected VI behavior is often possible and difficult to debug!

- 7) On the front panel, change the *Mechanical Action* to *Switched when Pressed*.
- 8) Add another local variable linked to Stop and leave it as *writeable*. Wire as below:



- 9) Run the VI and check it works as expected. Is the use of *local variables* understandable? Is the implementation efficient?

## End of Exercise