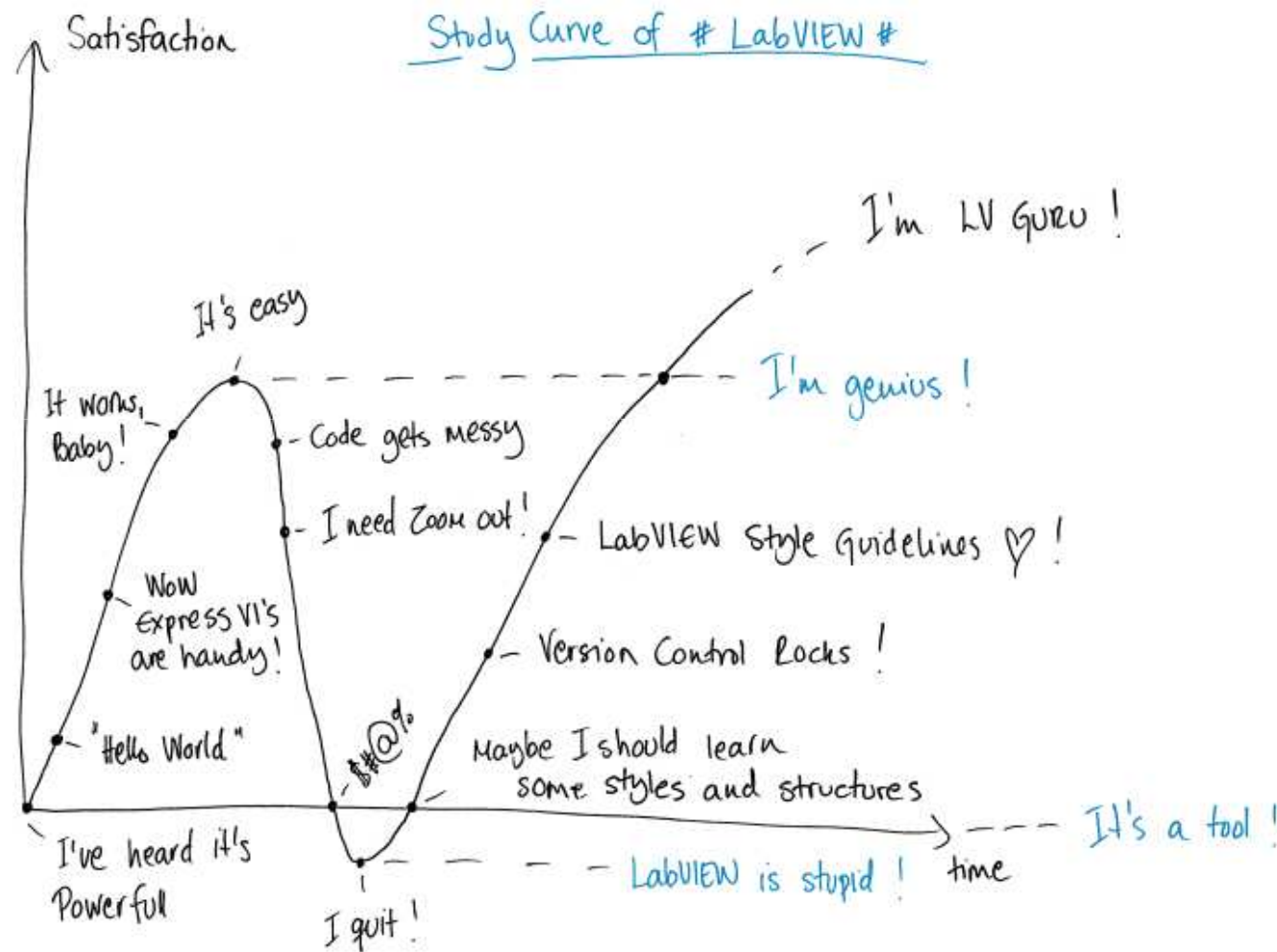


# LabVIEW Programmer Days

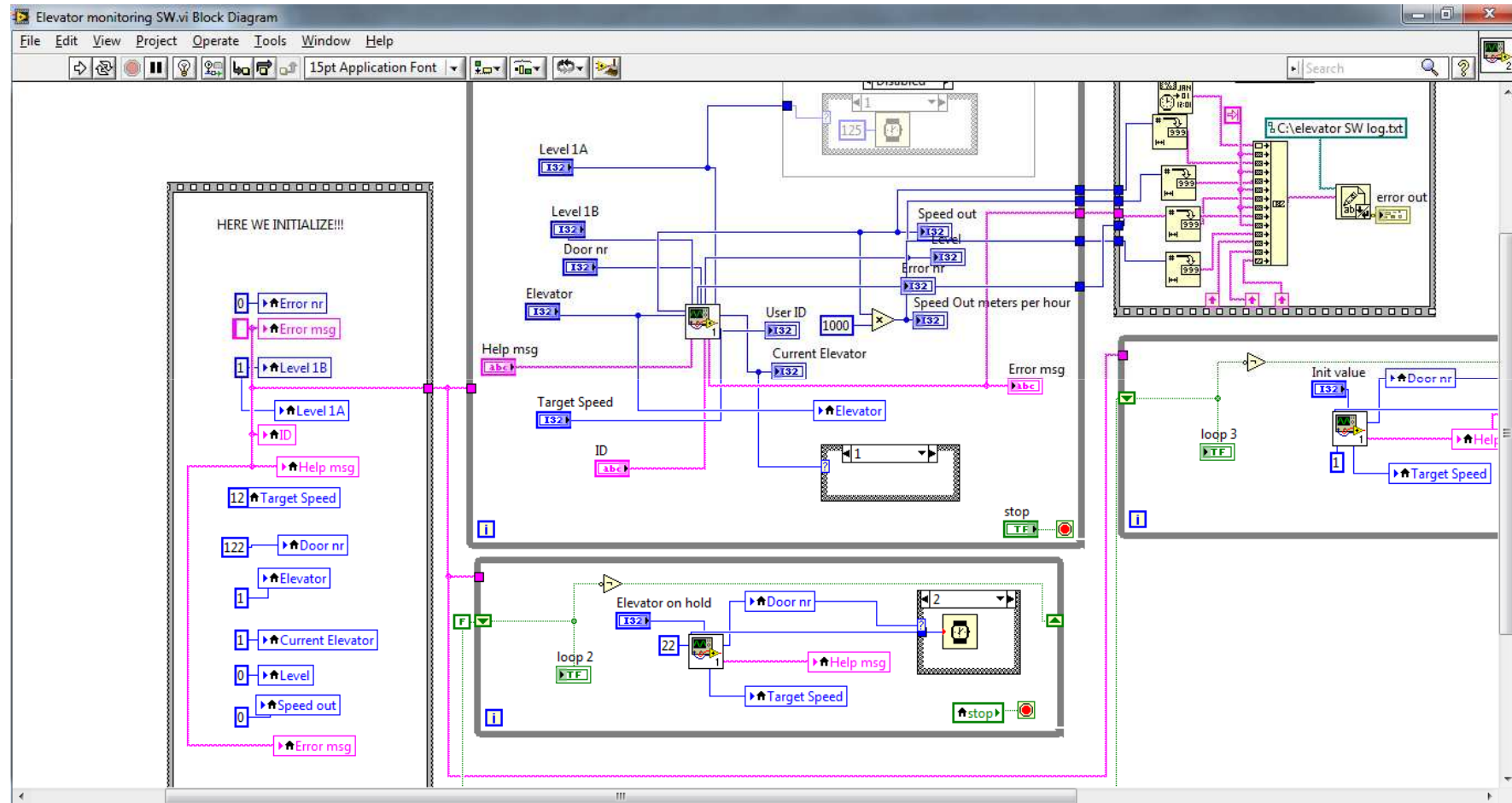
# Challenges when developing LabVIEW applications?



# Agenda

- How to start programming your LabVIEW Application?
- Common pitfalls
- Project Organization, File Naming and Control
- Structuring your application
- Error Handling
- Practical Tips and Tricks
- What is Next?

# Common Pitfalls



# How to avoid bad designs?

- How to start the project?
- Why should I use tools like LabVIEW Project Explorer, Version Control or LabVIEW Style Guidelines?
- Why the code structure is important?
- Why the Error Handling is essential part of every project? What are the benefits?

# LabVIEW Style Guidelines

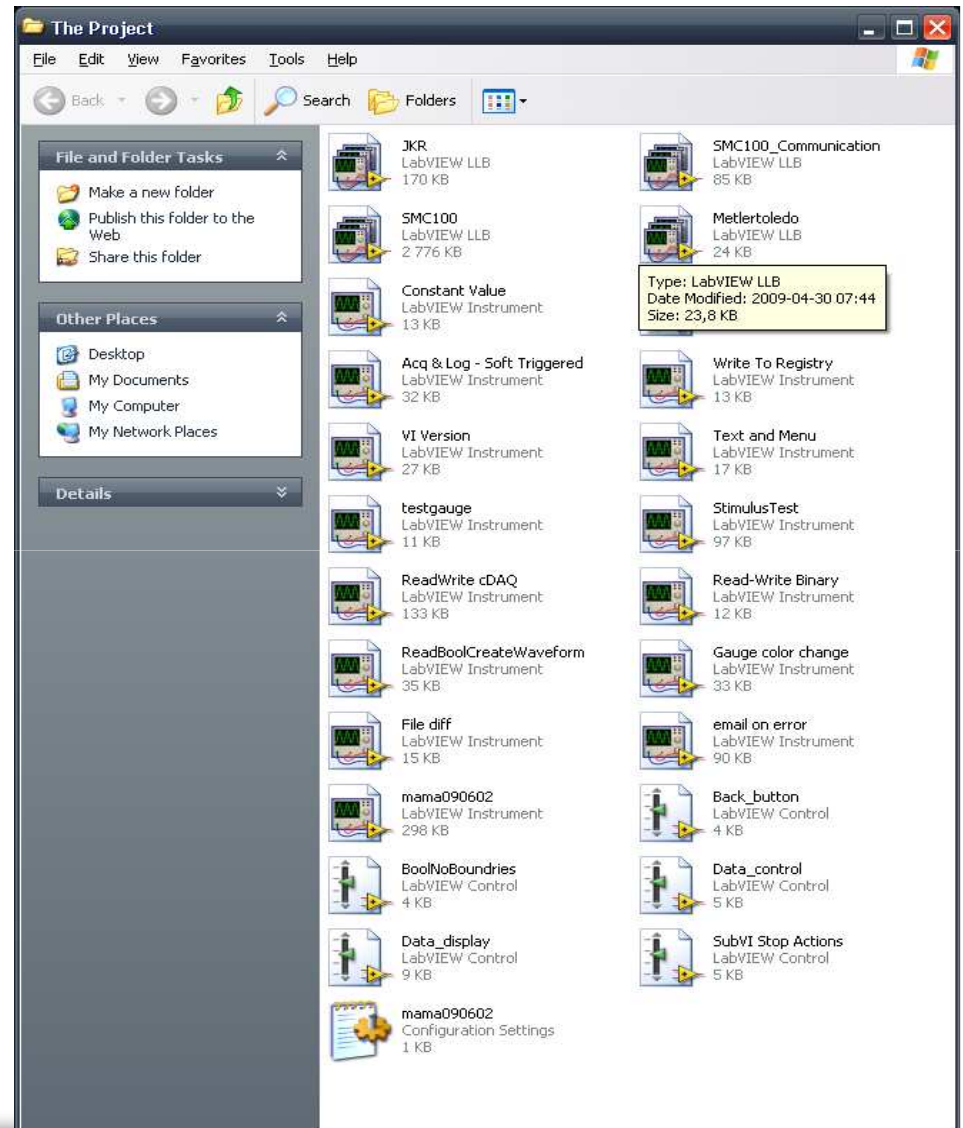
- Each developer has his own style and that can be a problem when multiple developers work on a project
- Inconsistent style makes software difficult to maintain and reuse.
- Select a set of guidelines that works for you and your development team and make sure everyone follows those guidelines.
- **LabVIEW Style Checklist:** <http://zone.ni.com/reference/en-XX/help/371361G-01/lvdevconcepts/checklist/>

# Project Organization, File Naming and Control

- Create the Folder Hierarchy (before you start coding)
- Use the LabVIEW Project
- Use proper source filenames:
  - Create unique filenames (never default names)
  - Do not abbreviate filenames
  - Identify the top-level Vis
- Use Source Control (for example: Perforce, Tortoise SVN, MS Visual SourceSafe)

# LabVIEW Projects

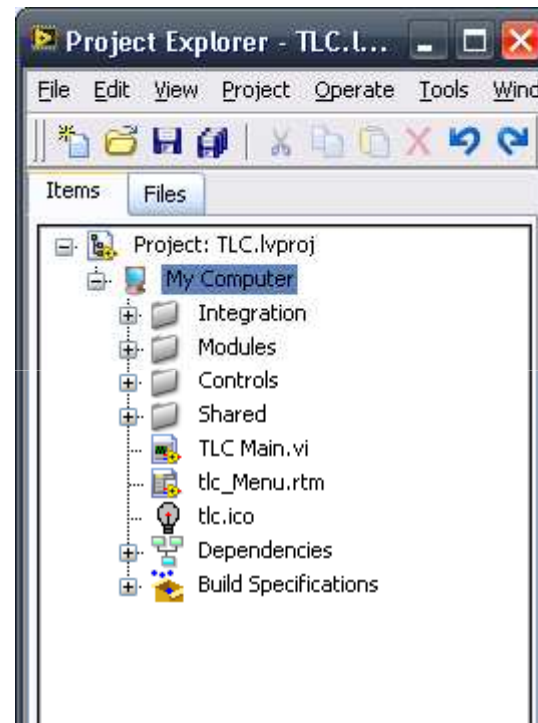
- Why should I use the Project Explorer?
- Which of these is the start VI?





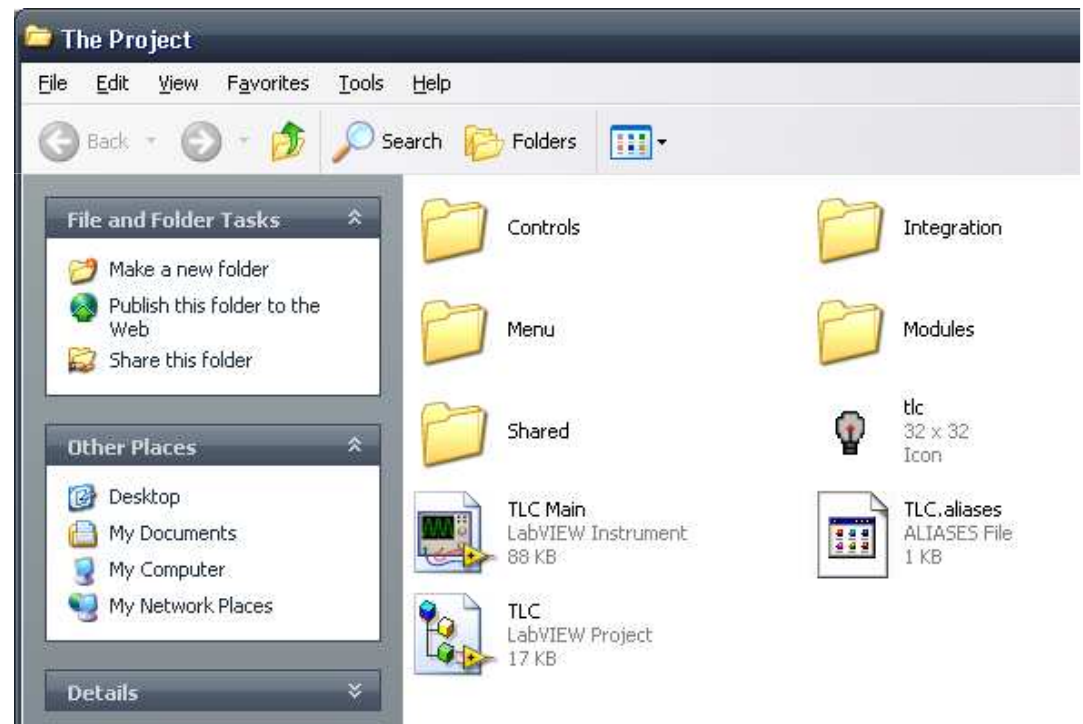
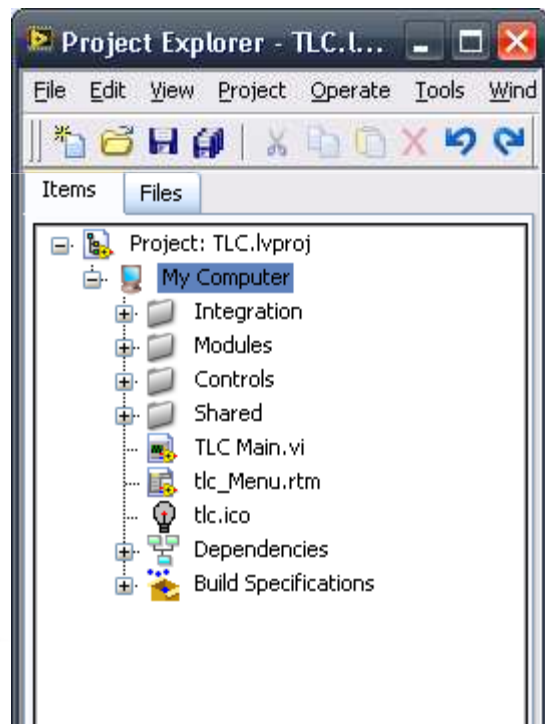
# Demo

## Project Explorer



# LabVIEW Projects

- Tip: Try to use the same file structure in your LabVIEW Project and on your hard disk



# Why is Structure Important?

- Think about structure before you start!
  - + Easier to understand, upgrade and maintain
  - + Easier project management
  - + Find problems earlier
  - + Self documenting
- LabVIEW has built in Templates
  - + No need to reinvent the wheel
  - + Get started faster

# Why is Structure Important?

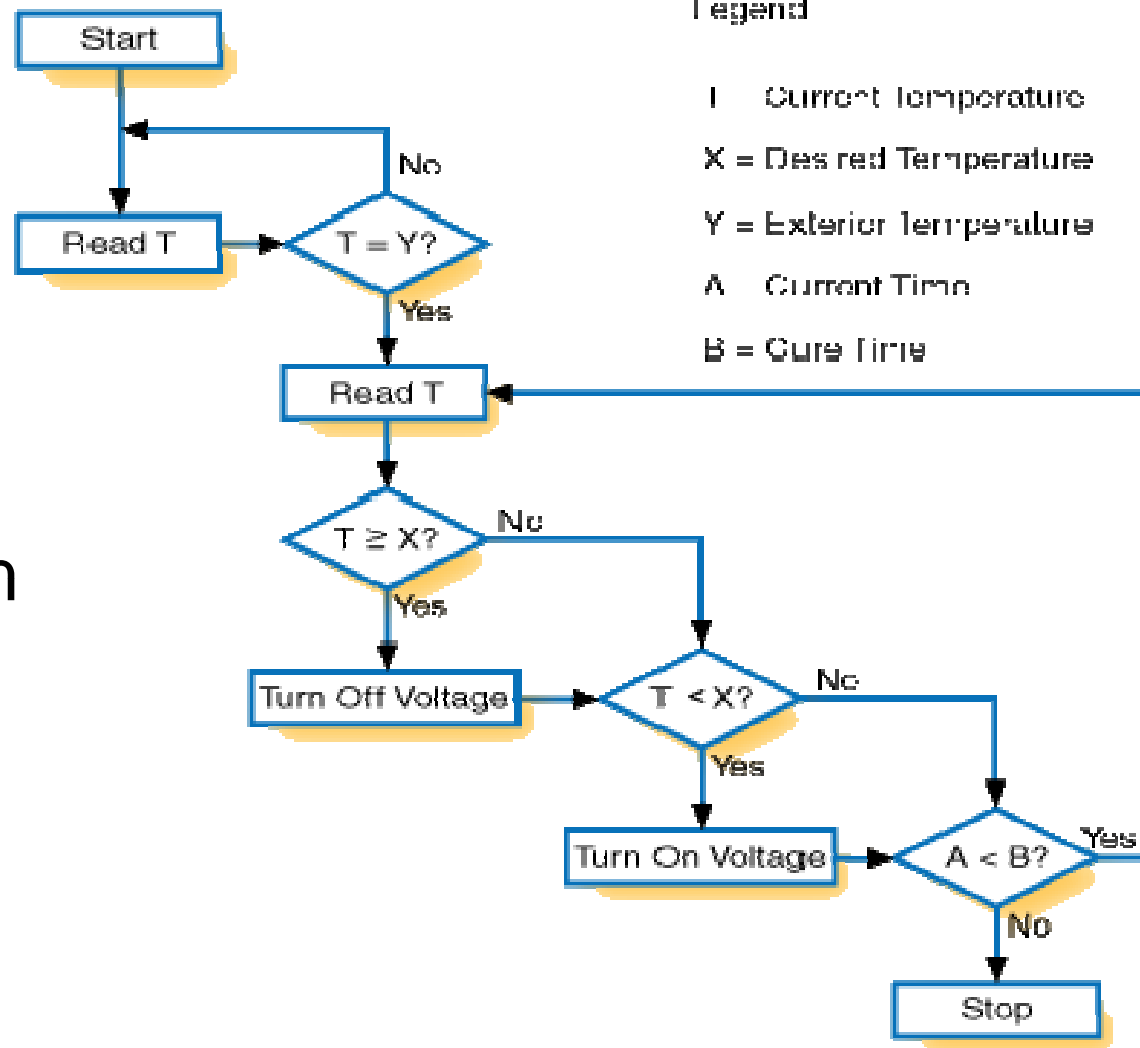
- Is Code & Fix a good practice?
- Easy to just start coding in LabVIEW

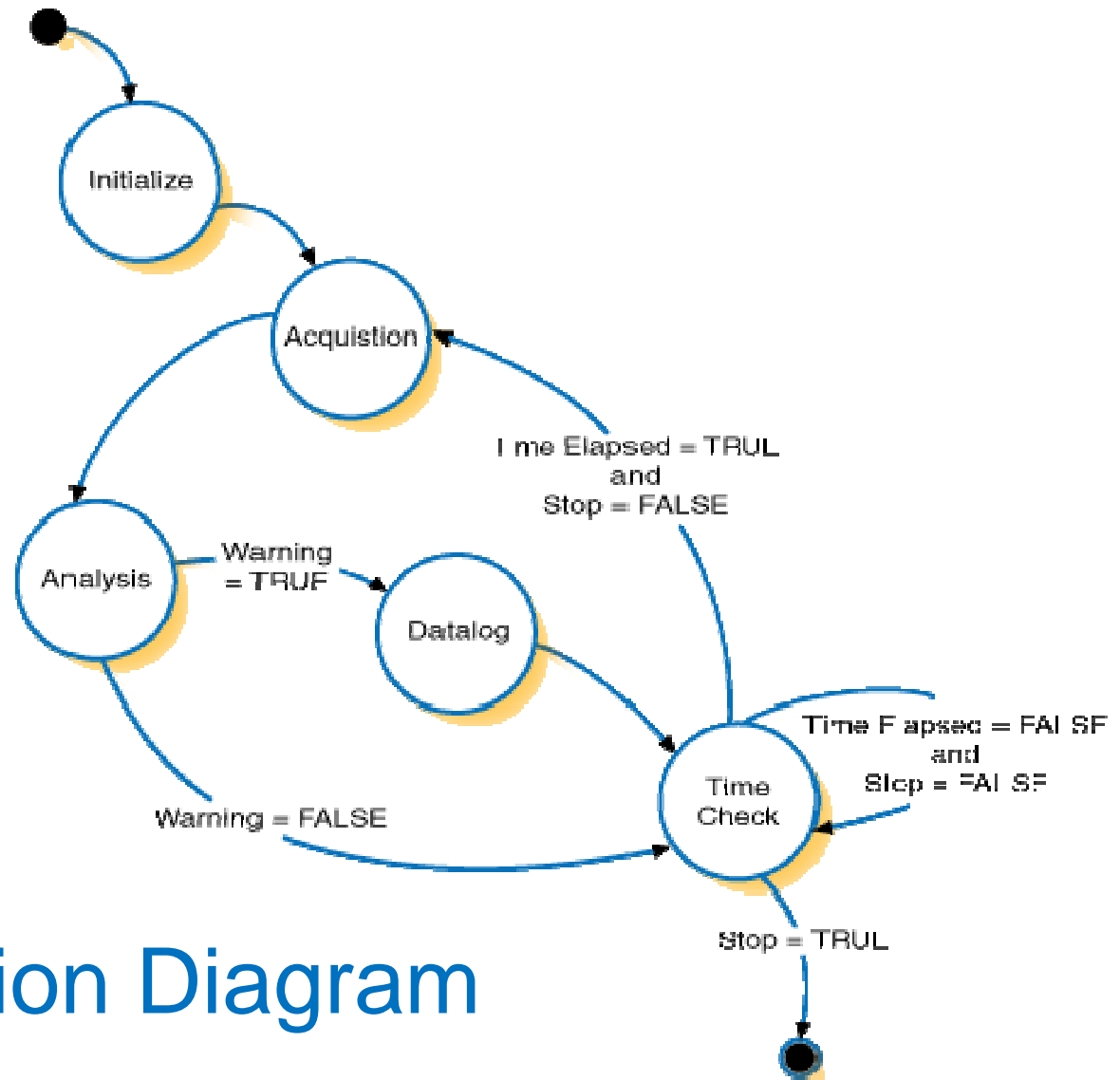
# How to start your main application

- Start on paper
- Requirements
- Program flow & data flow
  - State transition diagram
  - Flow chart

# Flowchart

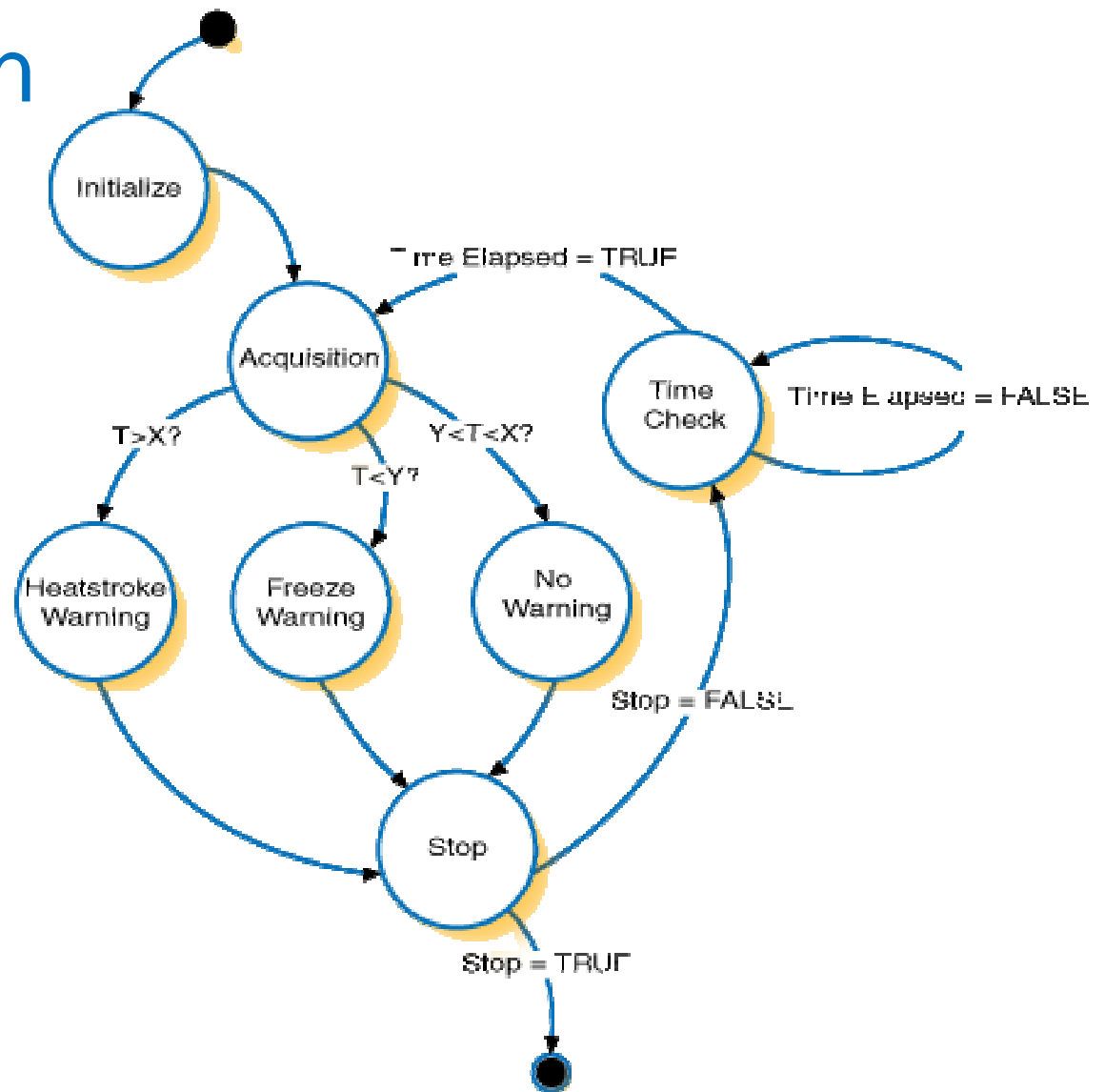
Visual design  
of an algorithm





## State Transition Diagram

# State Transition Diagram





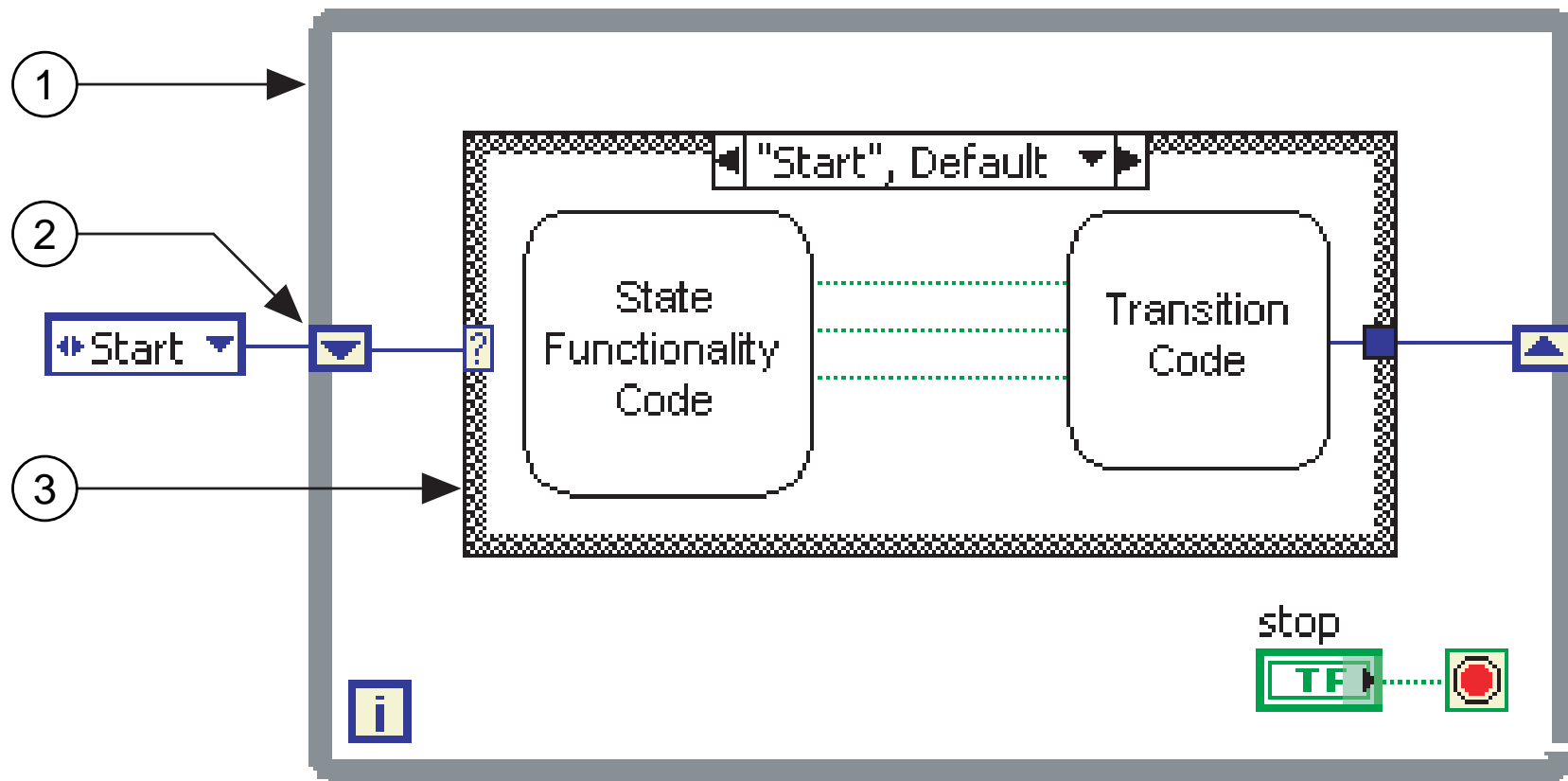
# Design Patterns

- State Machine
- Producer/Consumer (Data)
- Producer/Consumer (Events)
- + many more

# State Machine Design Pattern

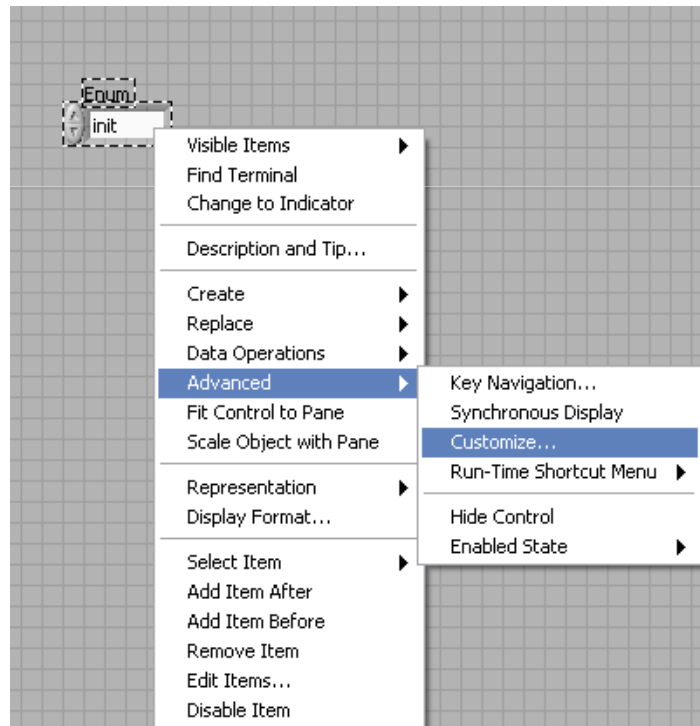
- You are facing these challenges:
  - Set of task to perform in sequence
  - Sequence is not fixed
  - Example: Acquiring data and performing actions based on signal level

# State Machine Design Pattern



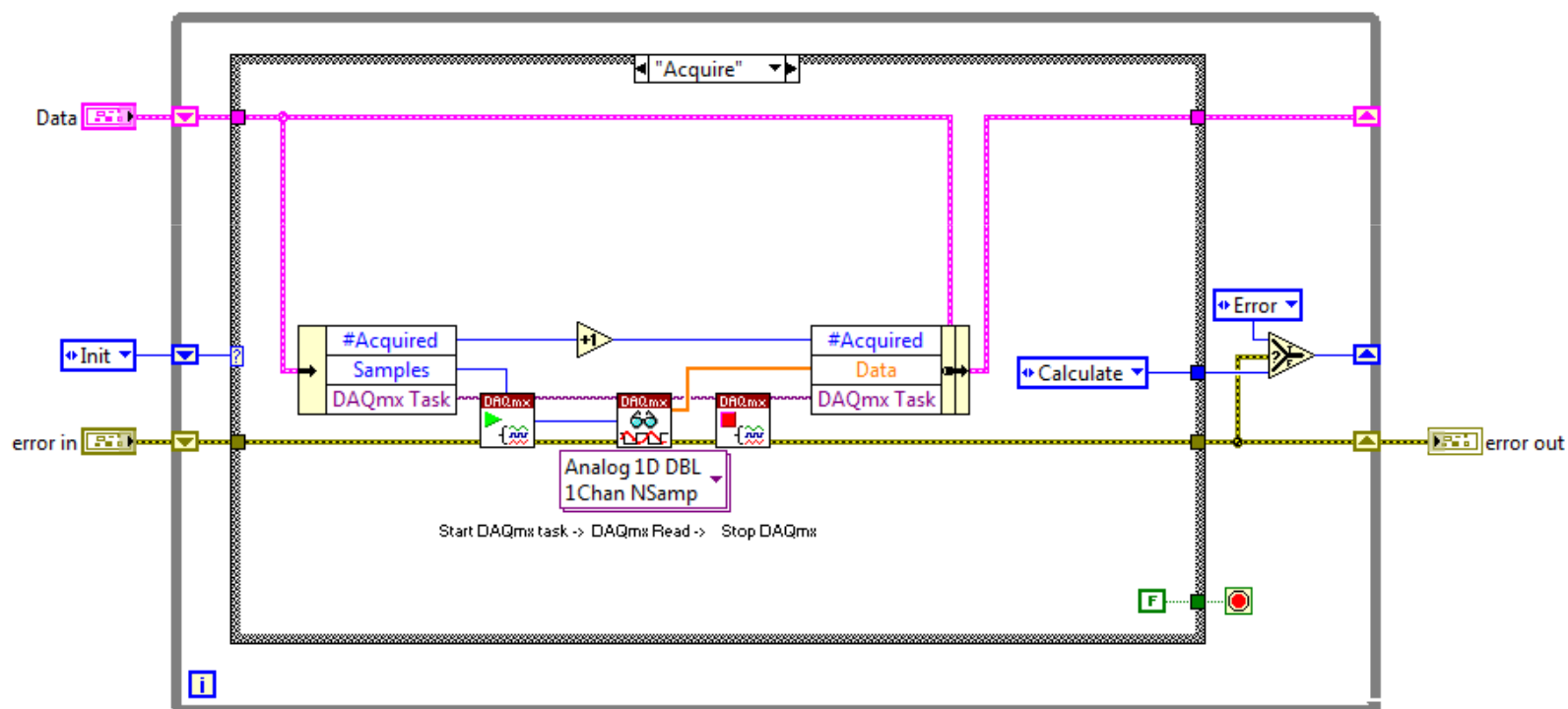
# State Machine Design Pattern

- Tip: Type Def the enum used to control the cases in the State Machine.



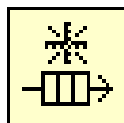
# Demo

## State machine with Cluster



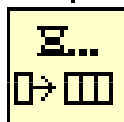
# Queues

## Obtain Queue



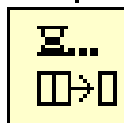
Create a Queue with a datatype and a name (optional)

## Enqueue Element



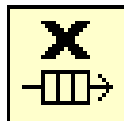
Put a piece of data last in the queue

## Dequeue Element



Take the first piece of data from the queue

## Release Queue



Release the queue to destroy it and free up the memory

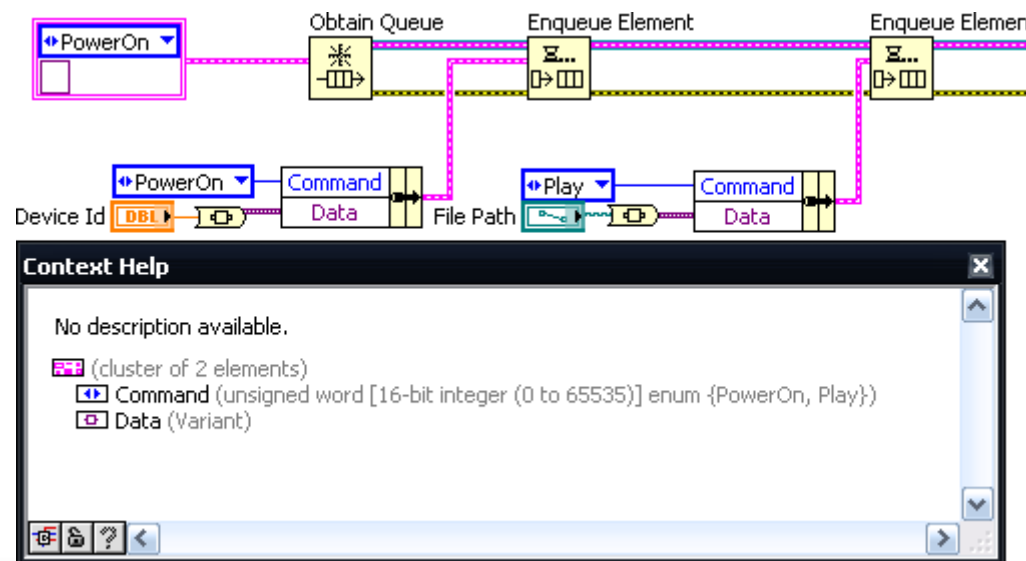
- Works just like a queue at the supermarket
  - First in, First out
  - Buffers data, allows different rates of Enqueue and Dequeue
  - Can be used to transfer data between loops in same VI or between VIs

- Normal Use and Behind the Scenes



# Queues

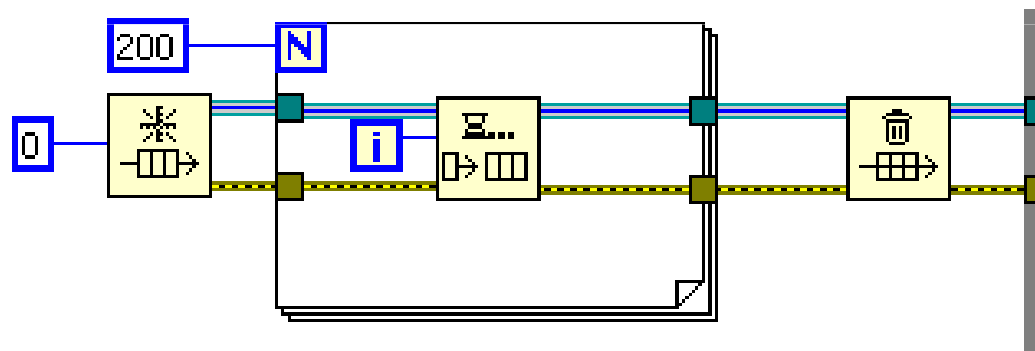
- Tip: Use cluster with Enum and Variant to get multifunction queue
  - Enum is command that explains datatype/action
  - Variant contains data of any type





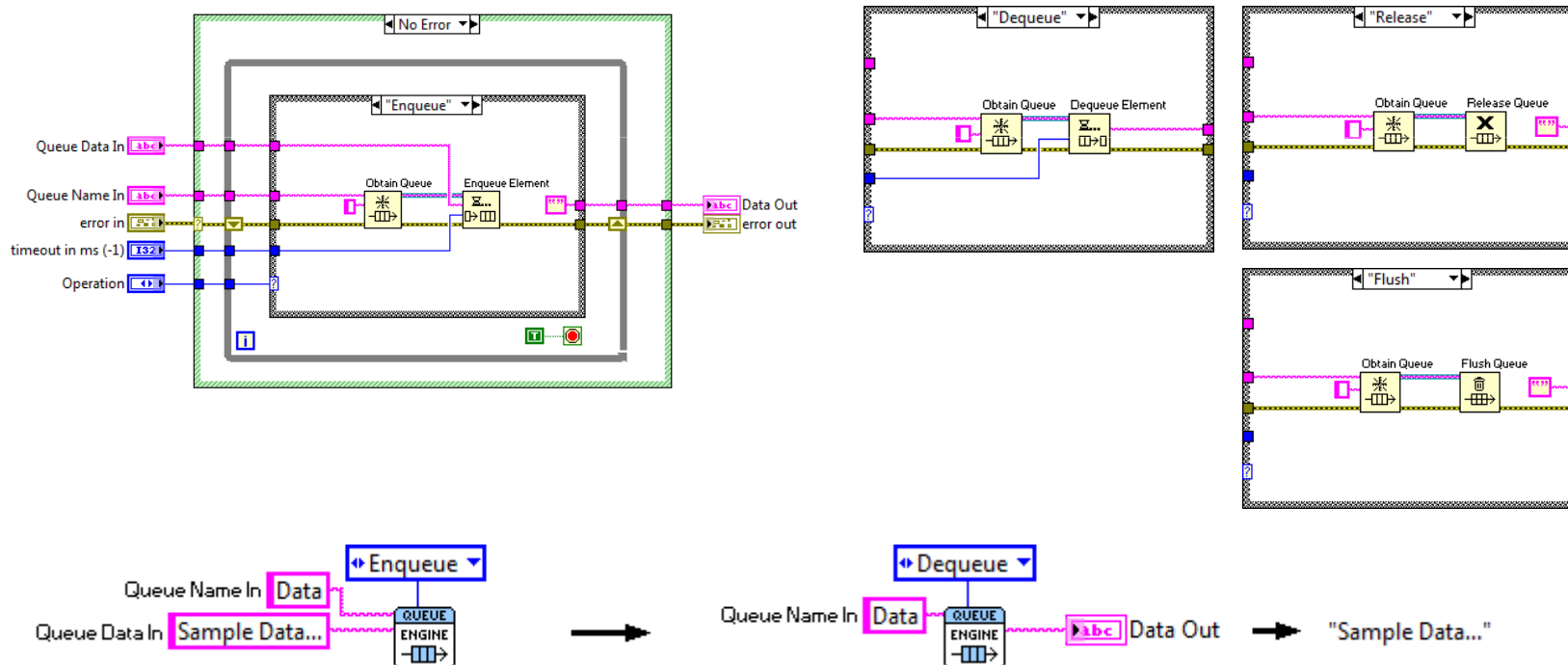
# Queues

- Tip: For time critical applications pre-allocate queue size for best performance



# Queues

- Tip: You can create a single VI to handle all Queue operations without using reference wires in the main program.



# Producer/Consumer (Data)

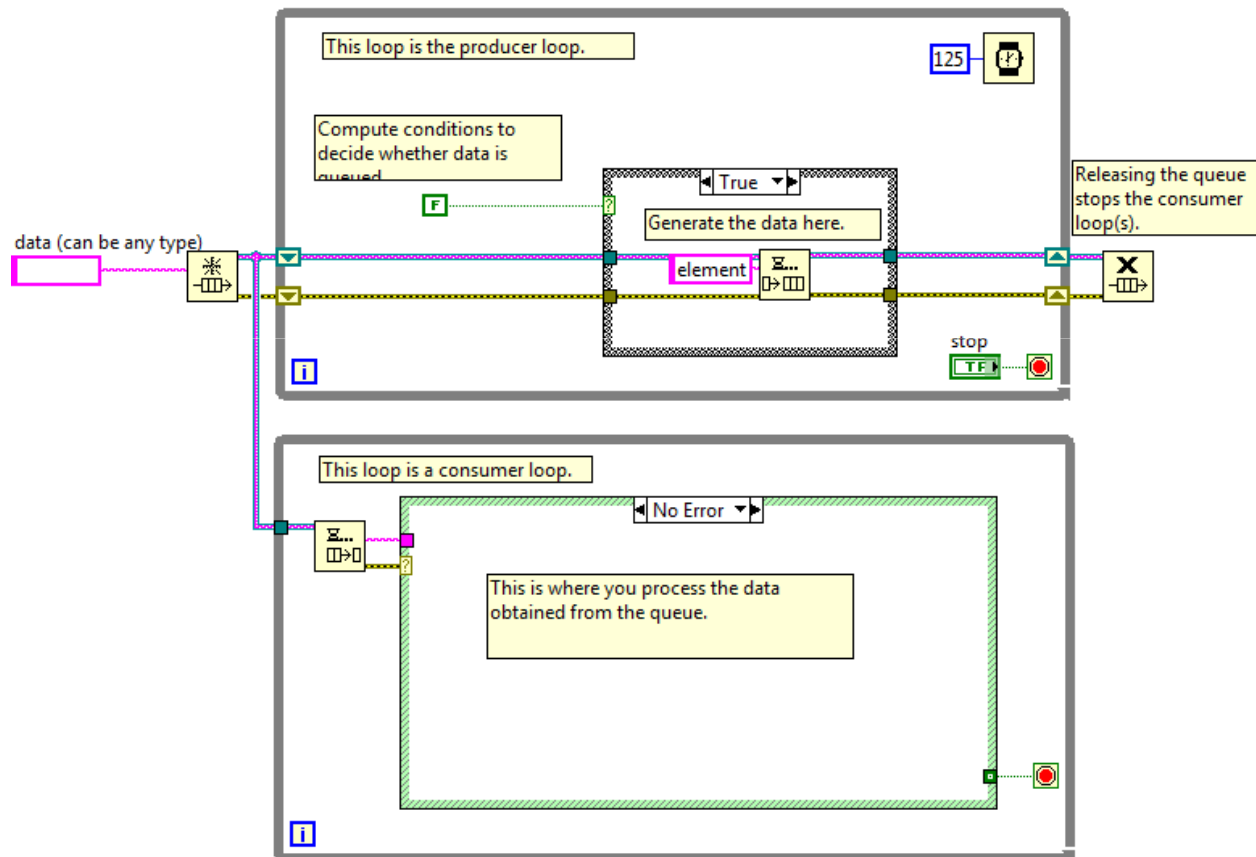
- You are facing these challenges:
  - Need to do intensive processing
  - Amount of code too much for one loop, takes too long to execute
  - Different parts in the program need to execute at different speeds
  - Can't afford to lose data
  - **Example:** Time critical data acquisition with intensive analysis

# Producer/Consumer (Data)

- Enhanced data sharing between multiple loops running at different rates
- Two categories of processes; those that produce data and those that consume data
- Use when you need to acquire multiple sets of data that must be processed in order

# Producer/Consumer (Data) Design Pattern

This template is for the Producer/Consumer design pattern.

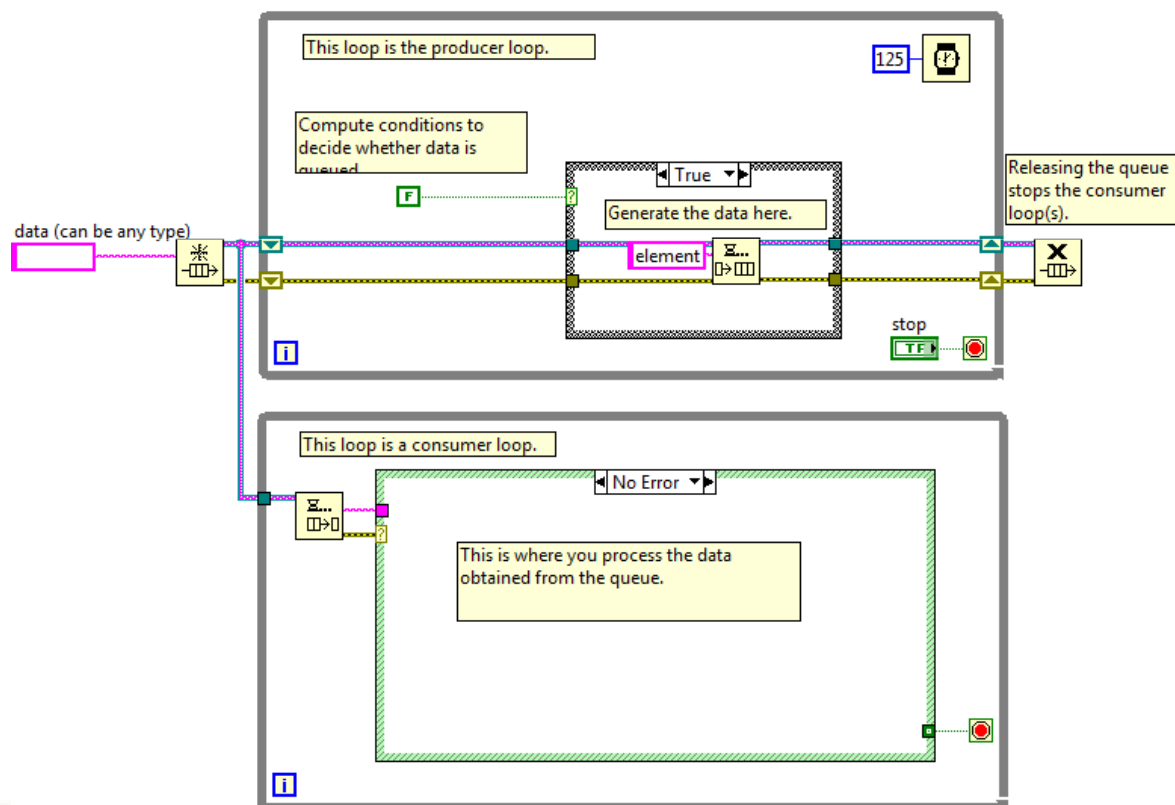


- Benefits
  - Acquire multiple sets of data that must be processed in order
  - Queues data, no data loss
- Considerations
  - Timing the producer
  - Sending data from the consumer to the producer

# Demo

## Producer/Consumer (Data)

This template is for the Producer/Consumer design pattern.



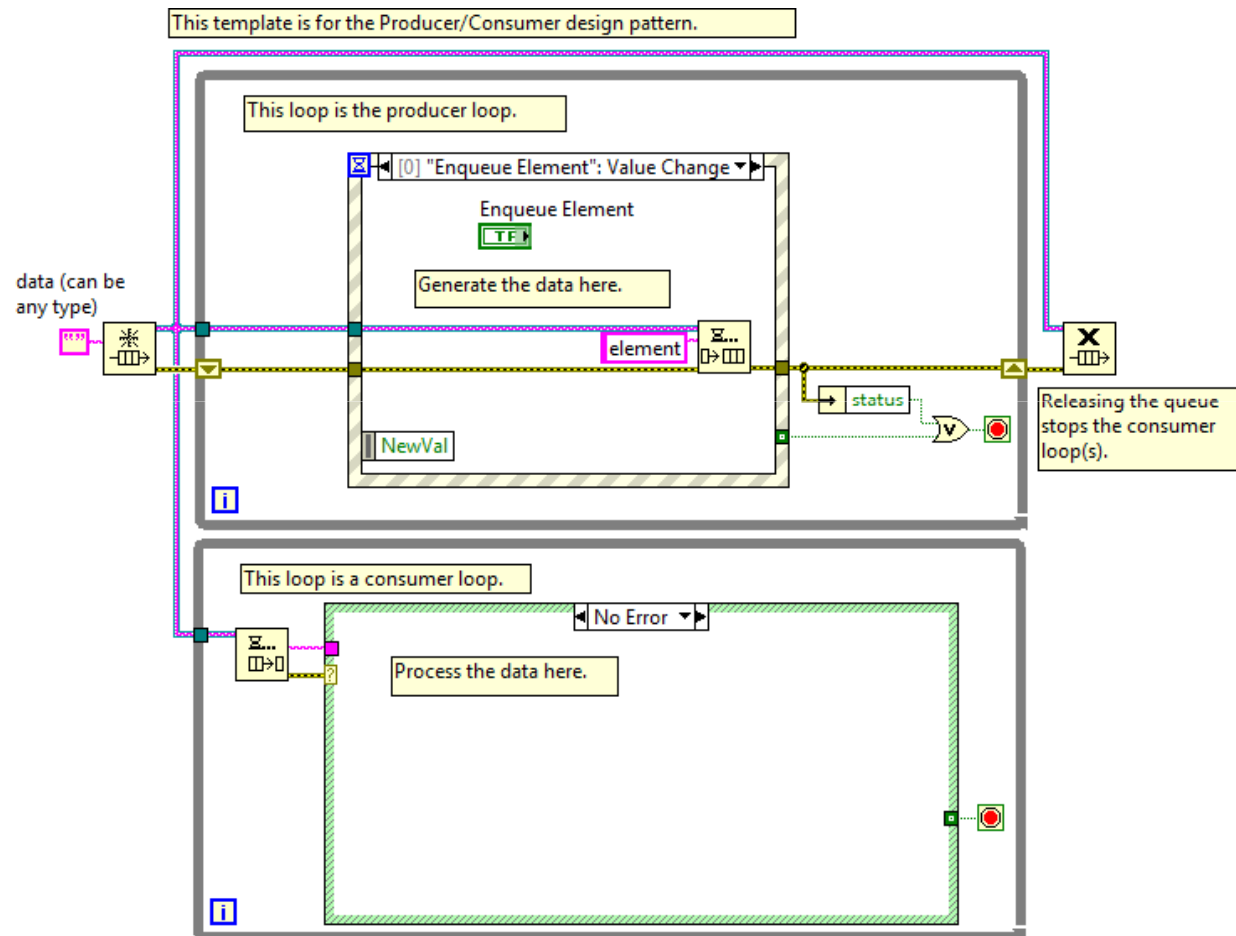
# Producer/Consumer (Events)

- You are facing these challenges:
  - User controls application via GUI, no set sequence
  - Heavy data processing required
  - GUI needs to respond directly at all times
  - Example: Any heavy processing, user interactive application

# Producer/Consumer (Events)

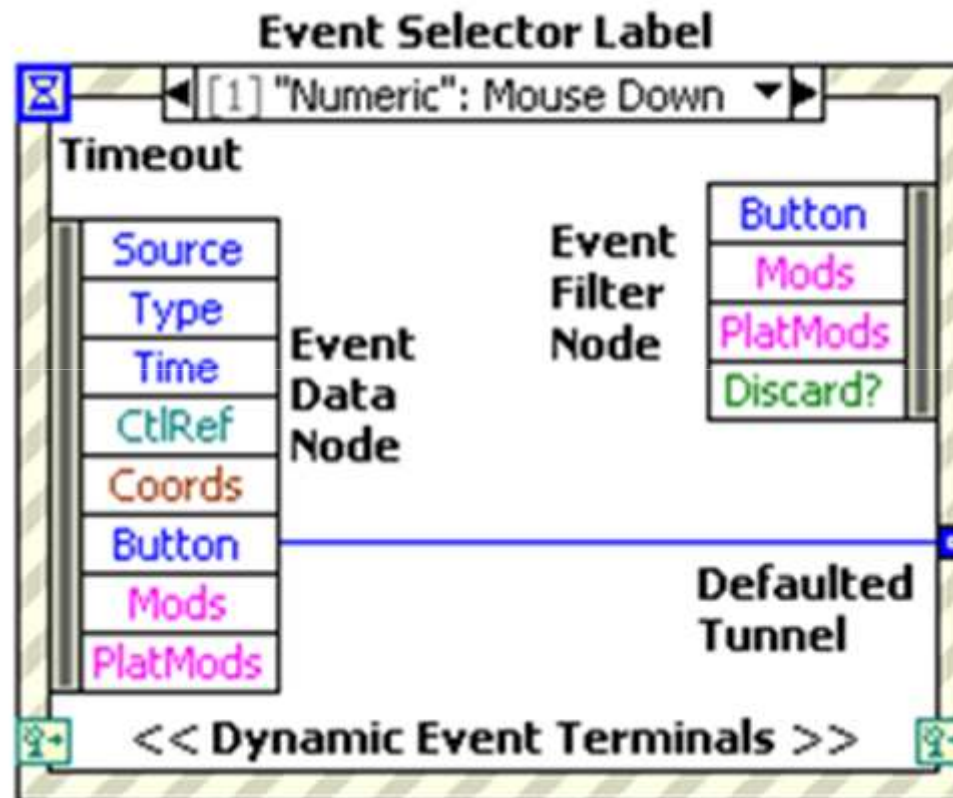
## Benefits

- Efficiently responds asynchronously to the user interface
- Queues can transfer any data type



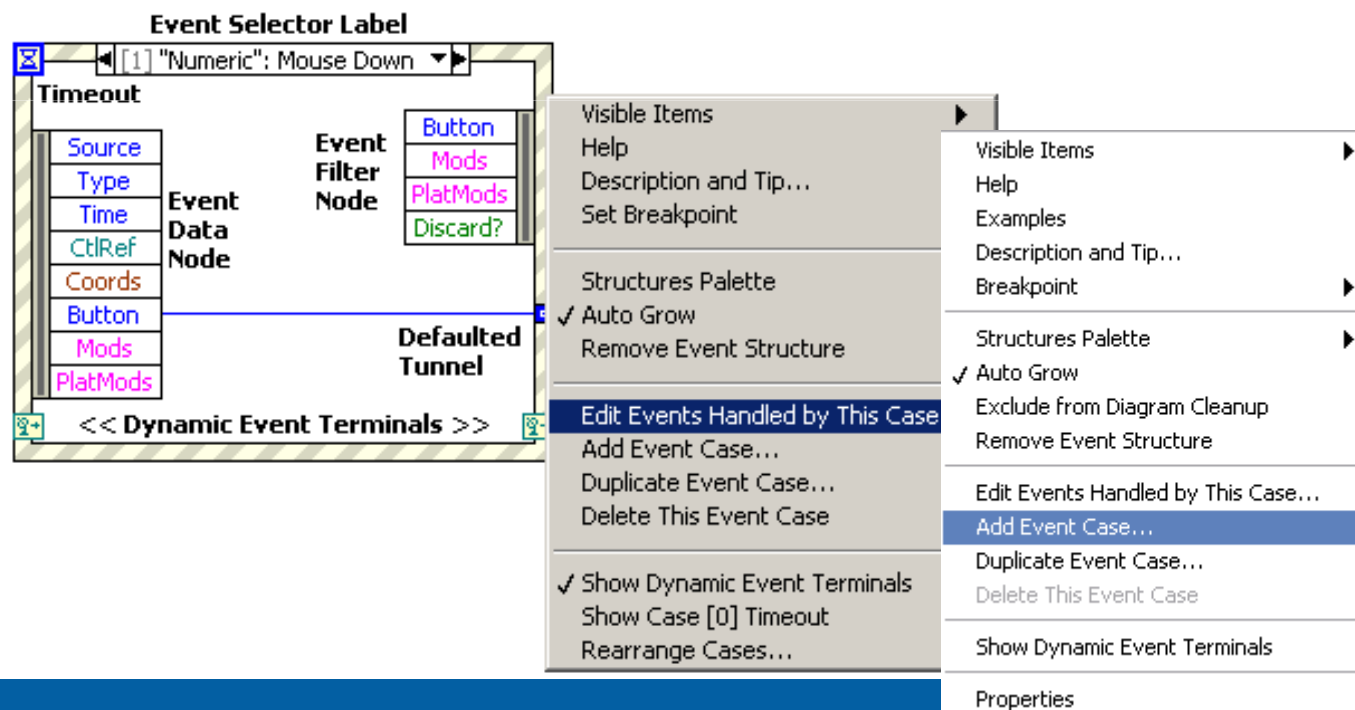


# Events



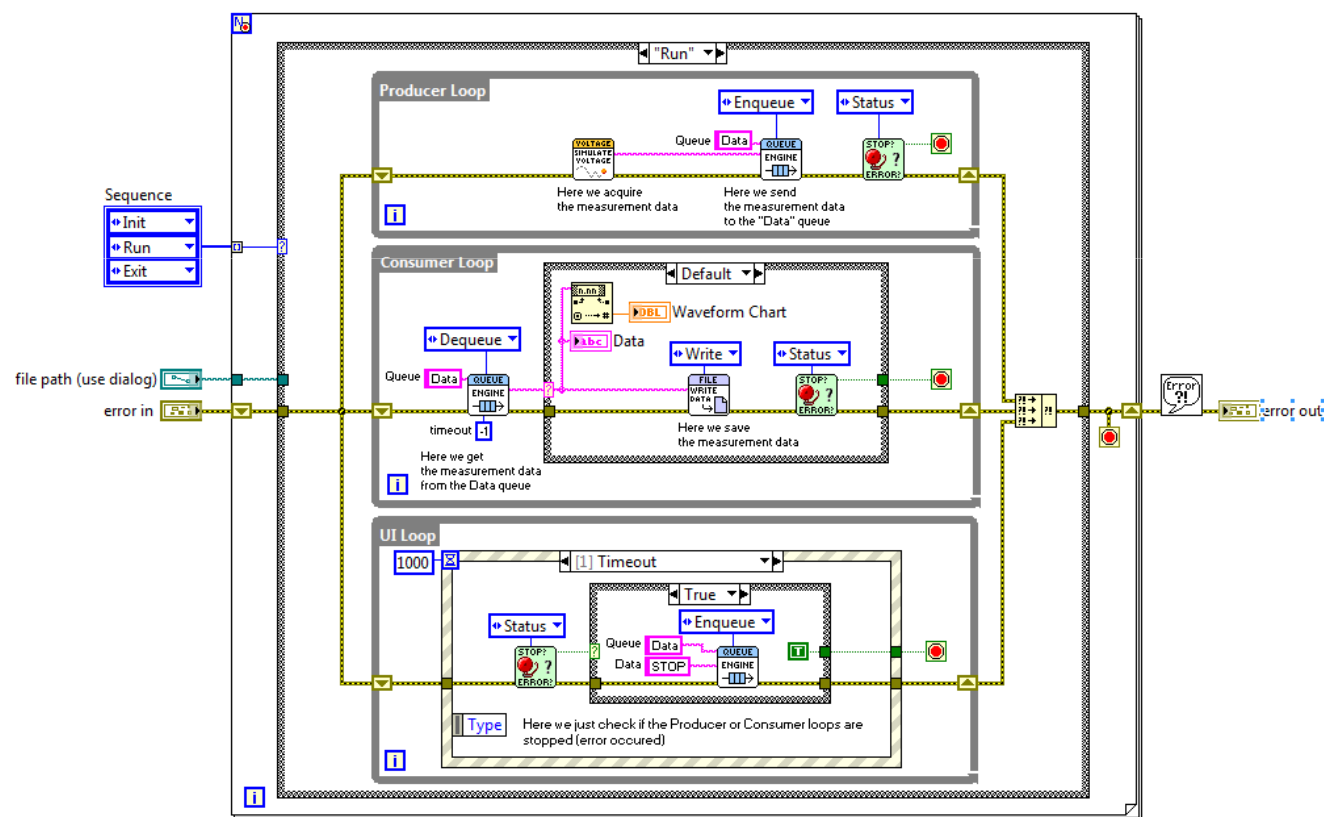
# Events - Configuration

- Use a dialog box to configure events by right-clicking the Event structure border and selecting **Add Event** or **Edit Events Handled by This Case** from the shortcut menu

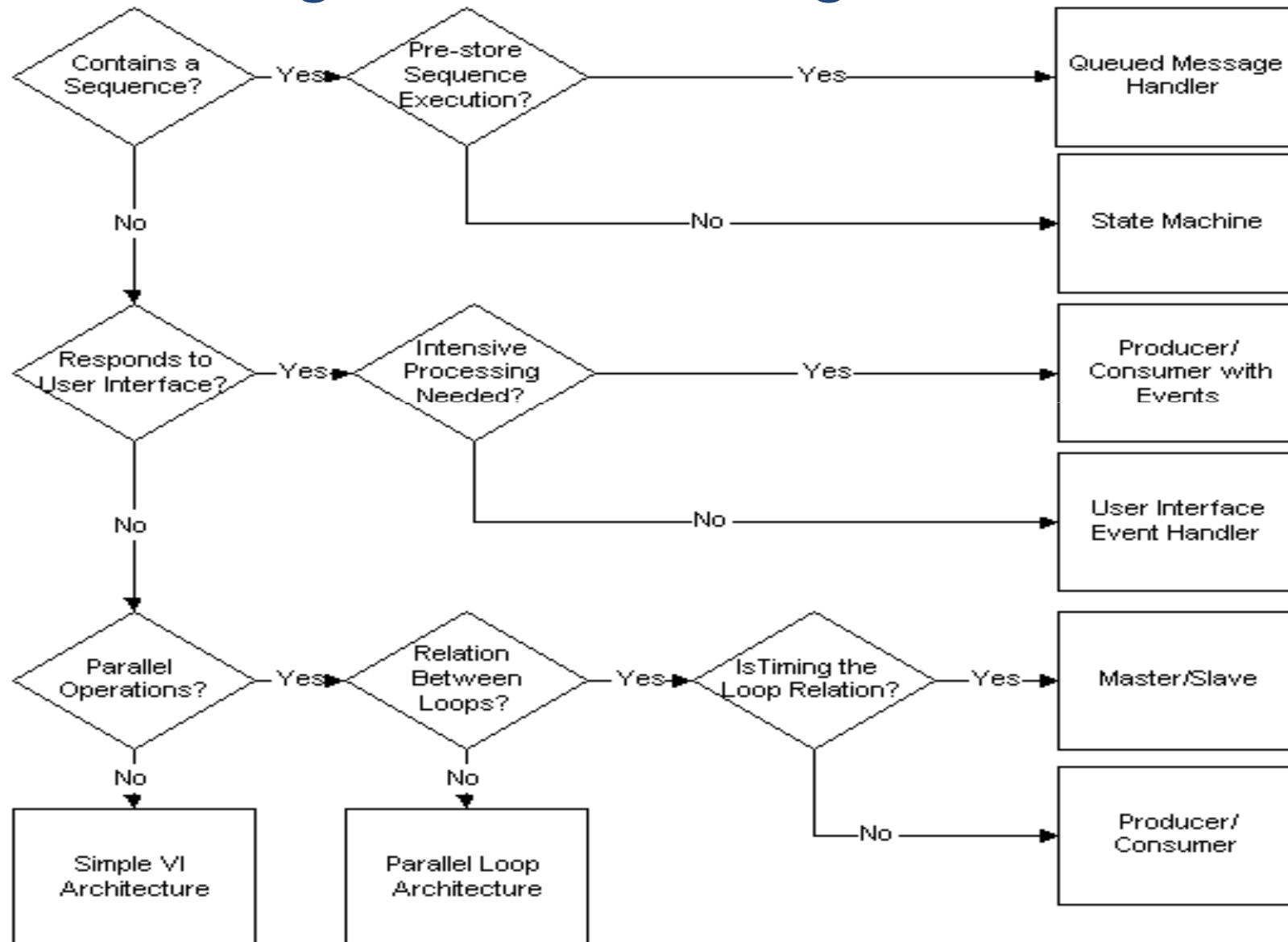


# Demo

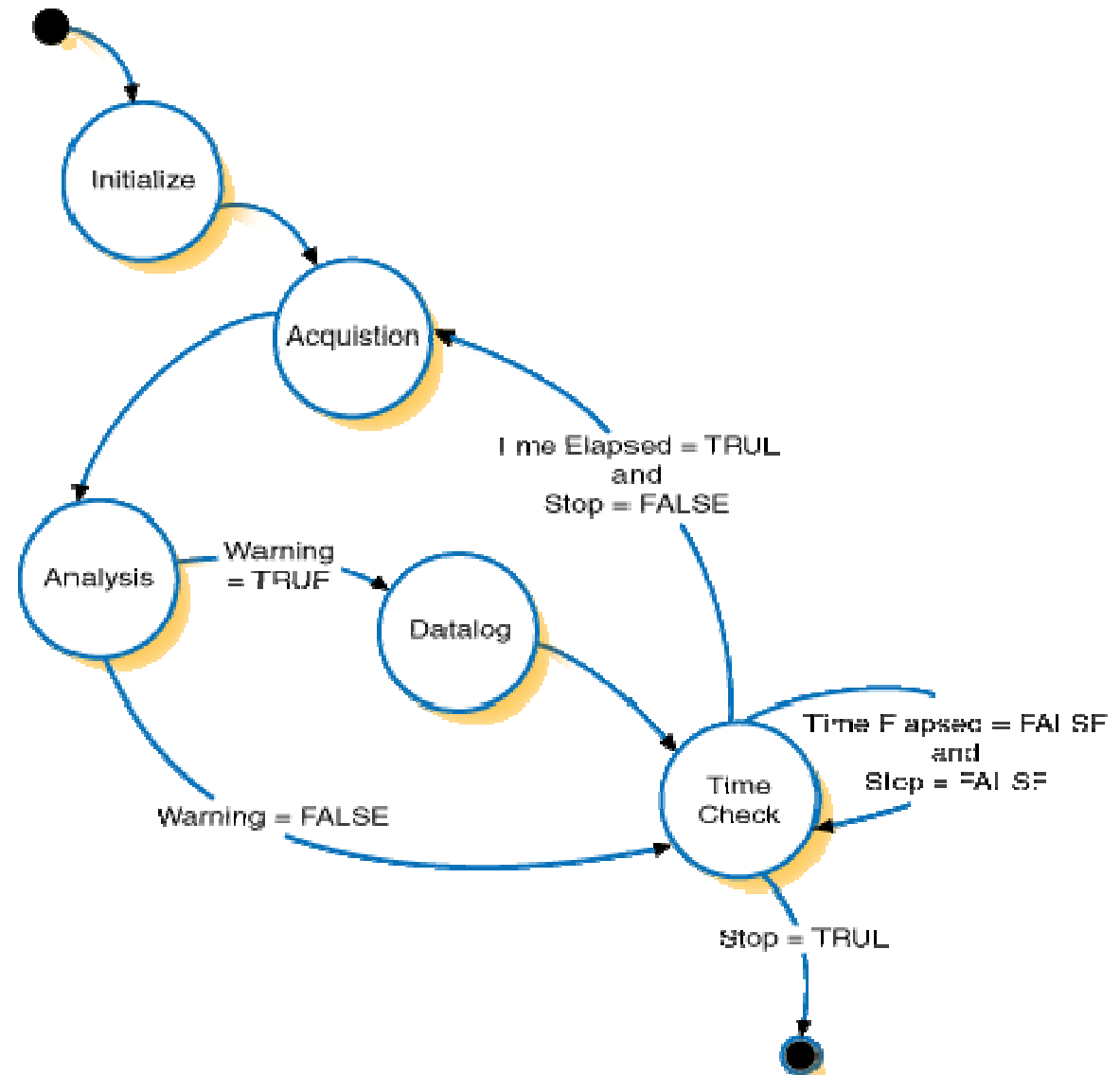
## Producer/Consumer/User Interface



# Choosing the best Design Pattern



# Case



BREAK

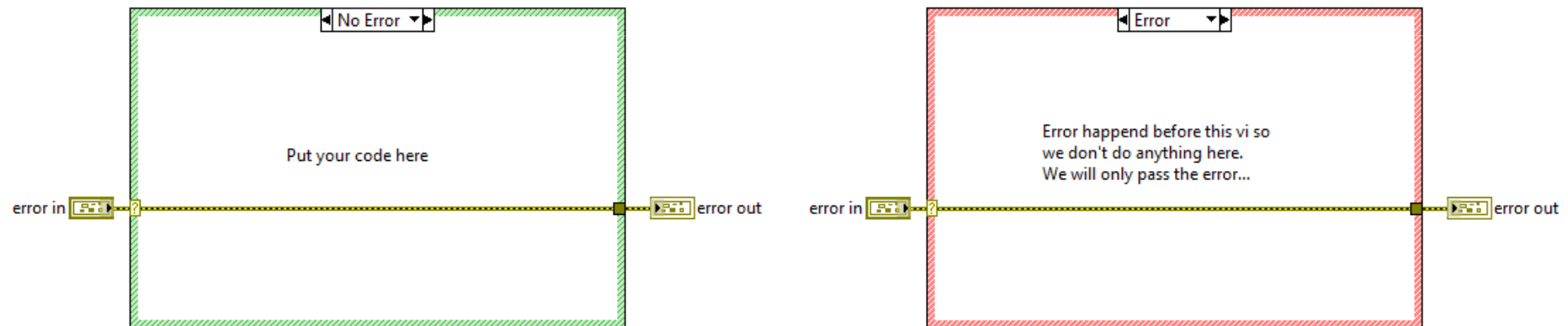
# Error Handling



- LabVIEW applications execute as programmed and therefor are not immune to bugs.
- Error handling is essential part of any application.
- Debugging LabVIEW application without proper error handling is same as programming blindfolded.
- So how to handle errors?

# Error Handling

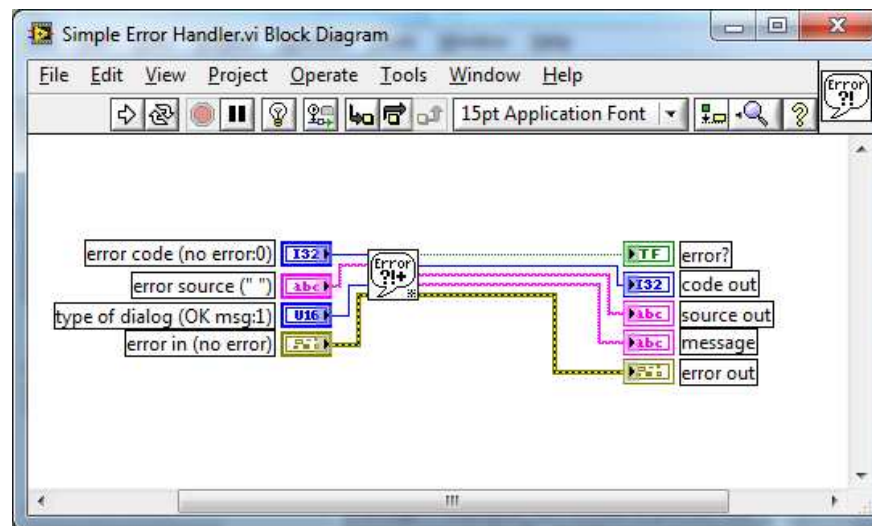
- Use the "No Error / Error" case with most of your subVIs!





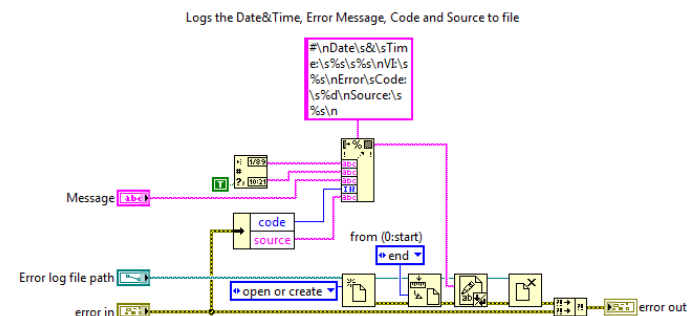
# Error Handling

- Use General Error Handler VI instead of Simple Error Handler VI
  - Simple Error Handler VI is just calling General Error Handler VI and hiding some of its inputs.



# Error Handling

- Log all Errors
  - Users may not report every error to developers
  - Even the minor errors may indicate bugs in the code. For example ini-file may be in the wrong place and user has to always seek the right path.
  - Errors may happen within specific time so there might be a memory leakage or some other problem that log file could help to discover.

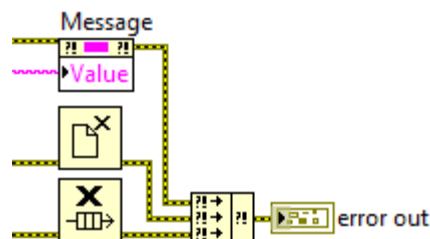


# Error Handling

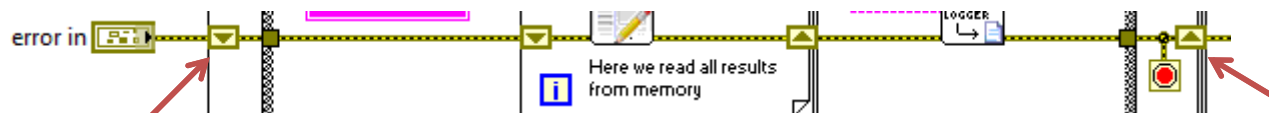
- Can I create custom Error Codes? Yes you can! And you should!
- Error Codes in the range of -8999 to -8000 and 5000 to 9999 are reserved to user-defined errors
- Maintain user-defined error codes within an XML file
- Use Error Code File Editor:
  - **Tools -> Advanced -> Edit Error Codes**

# Error Handling

- Use Merge Errors function for parallel execution

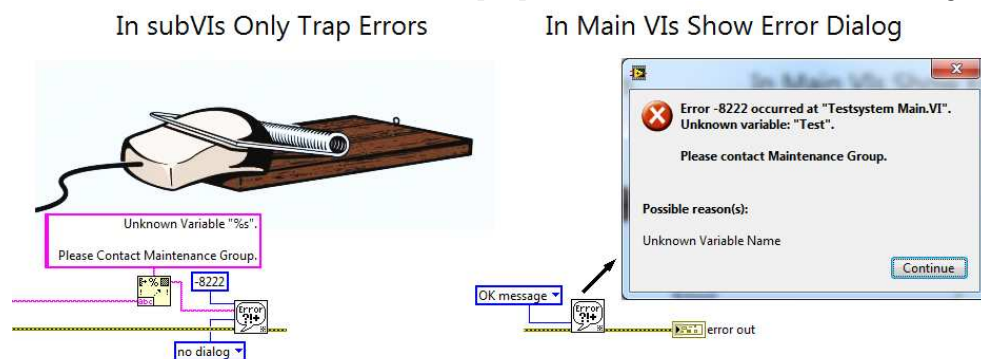


- Use shift registers in loops with Error cluster
  - For loop with 0 iterations will clear the Error (if shift register is not used)!



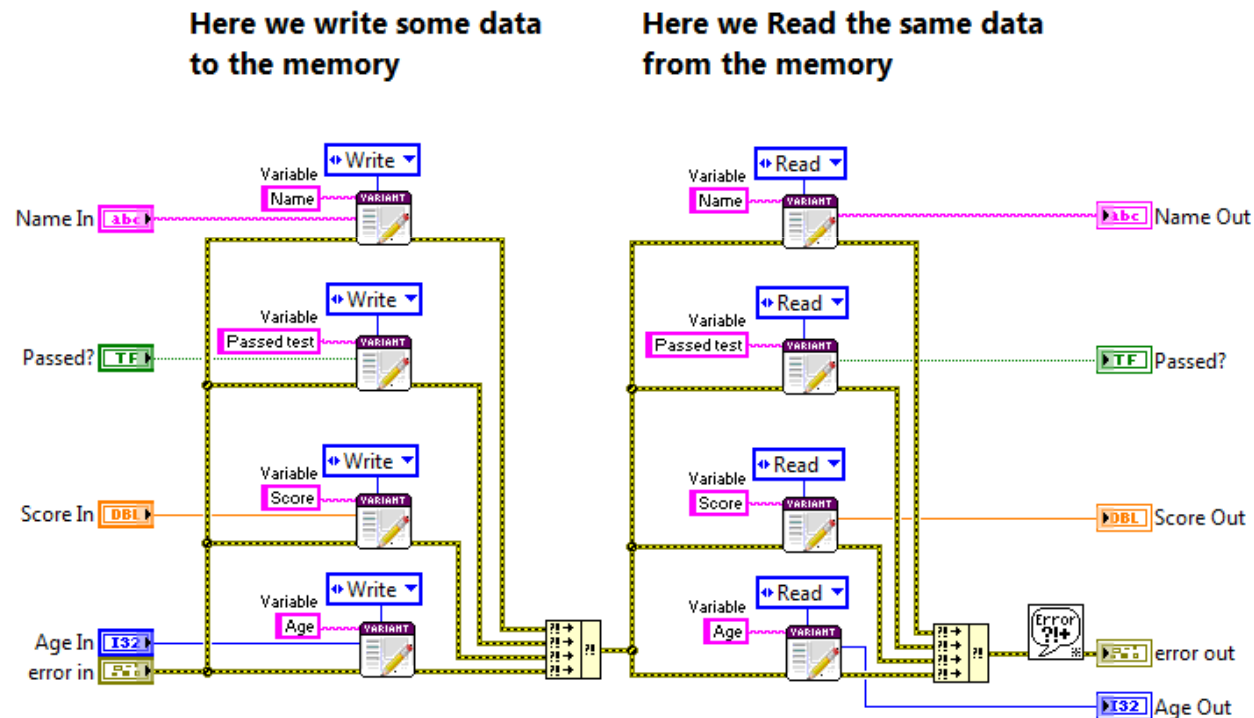
# Error Handling

- Do not popup Error Window in lower level VIs. Only acknowledge errors there and pass that data forward with error cluster to the Main VI level.
- Use Error Popups in the Main VIs to show user where the error happend and why.




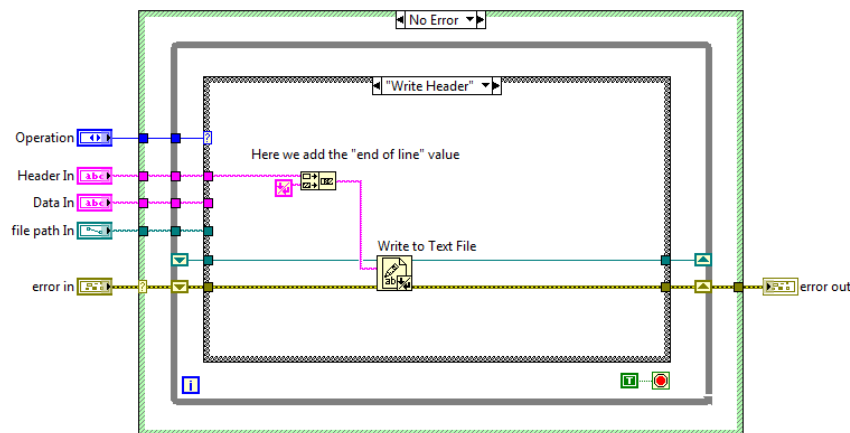
# Tips and Tricks

- High Speed Data Storage with Variant Attributes



# Tips And Tricks

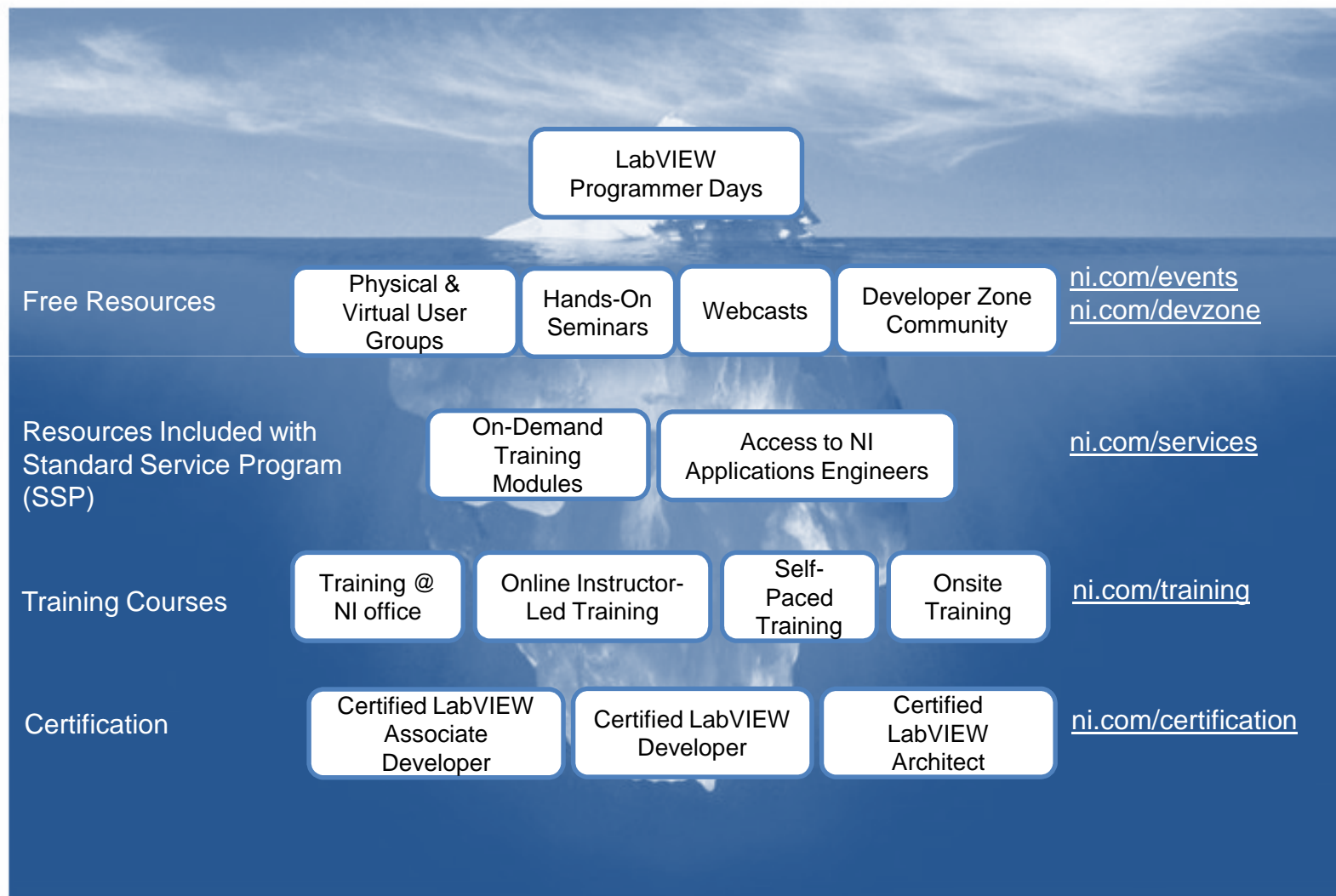
- Save measurement data easily with Smart Logger VI.
  - One VI will handle all the file IO tasks
  - Fast & Easy to Use
  - No reference wires needed in the main application



# What is Next?

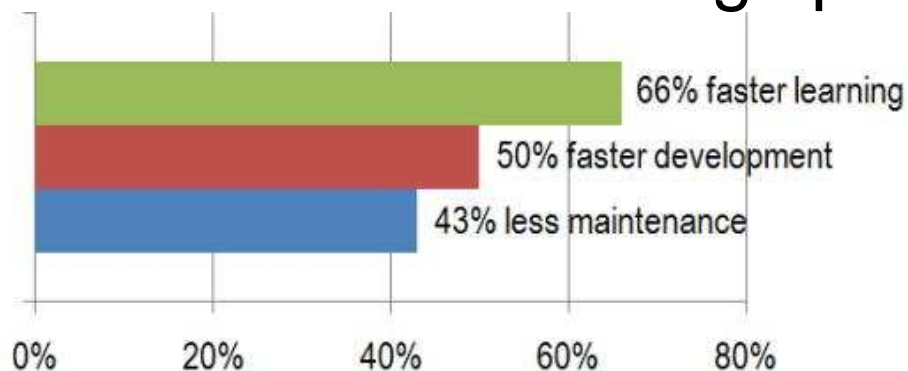


# LabVIEW Programmer Days: Just the Tip of the Learning Iceberg



# Understanding the value of NI Training

- Overall, approximately 89 percent of respondents of our survey categorized the return on investment (ROI) for NI training courses as average to extremely high!
- In addition, the average NI training customer achieved the following specific benefits:



# Online LabVIEW Community

Discussion  
Forums

File  
Sharing

Groups

R&D Bloggers

Product  
Feedback

[ni.com/community](http://ni.com/community)

# The Power of the Platform

110,000+ online members  
350,000+ support posts nine languages  
300+ user groups worldwide  
2,000+ robotics experts through FIRST  
400,000+ children through LEGO  
1,000+ job postings online

## Community



## Collaboration

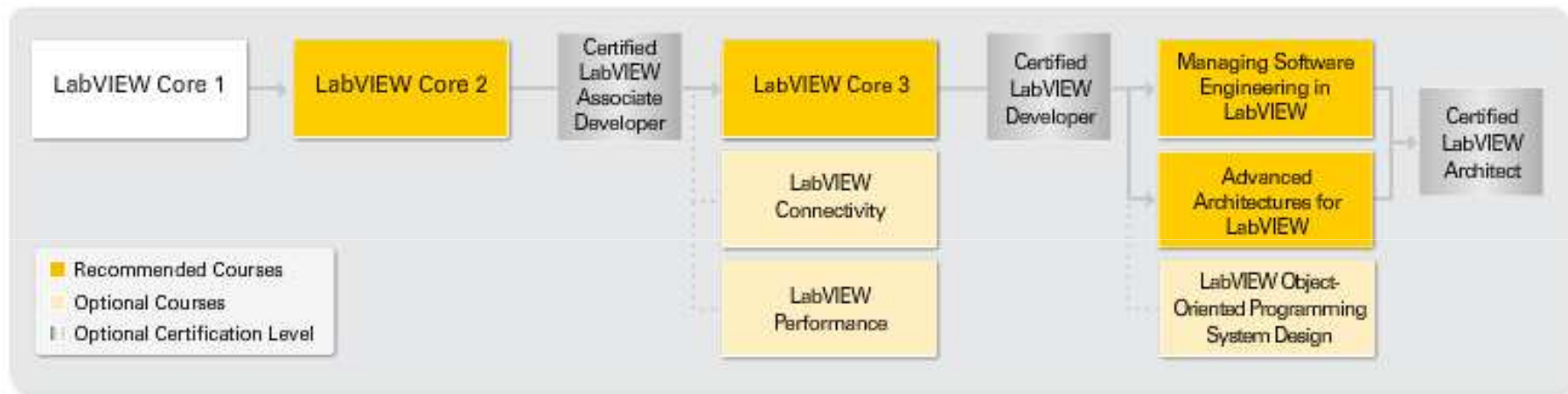
400+ third-party add-ons  
400+ solution partners  
1,000+ value added resellers  
35+ classroom training courses  
15+ online training courses

## Connectivity

NATIONAL INSTRUMENTS  
**LabVIEW™**

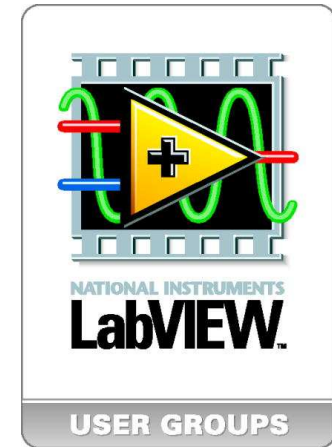
6,000+ example files  
7,000+ instrument drivers  
1,000+ motion drives  
1,000+ smart sensors  
500+ industrial cameras  
1,000+ Third-party PAC devices

# LabVIEW Courses



# User Groups

- Browse existing user groups
- Explore user group content
- Host your user group on ni.com
  - Share presentations
  - Display a group calendar
  - Communicate with other members
  - Read meeting notes



[ni.com/usergroups](http://ni.com/usergroups)



# LVUG.NL

## LabVIEW User Group





<http://lavag.org/>

LAVA - Windows Internet Explorer

http://lavag.org/

Home - Northern Region Univers Std by Adobe Typ...

Facebook LAVA

### Software & Hardware Discussions

Forum	Stats	Last Post Info
<b>LabVIEW General</b> Post questions here that don't fall into any other LabVIEW programming category listed below.	2,217 Topics 10,697 Replies	Today, 05:52 AM In: Quality of Labview Application By: ShaunR
<b>LabVIEW (By Category)</b> <ul style="list-style-type: none"> <li>Application Design &amp; Architecture</li> <li>Object-Oriented Programming</li> <li>User Interface</li> <li>Remote Control, Monitoring and the Internet</li> <li>VI Scripting</li> <li>Certification and Training</li> <li>Application Builder, Installers and code distribution</li> <li>Development Environment (IDE)</li> <li>Source Code Control</li> <li>Database and File IO</li> <li>Calling External Code</li> <li>Machine Vision and Imaging</li> <li>TestStand</li> <li>Real-Time</li> <li>PDA</li> <li>Embedded</li> <li>Linux</li> <li>Apple Macintosh</li> </ul>	5,498 Topics 30,485 Replies	Today, 10:14 AM In: SMS using ethernet port modem By: rociologa
<b>Hardware</b> Hardware problems, questions, driver development, DAQ, Sensors, GPIB, Serial, Instrument control and any hardware specific issues.	1,426 Topics 4,954 Replies	08 January 2011 - 03:26 AM In: Trouble with a couple of Co... By: SuperS_5

### Resources

Forum	Stats	Last Post Info
<b>Code Repository (Certified)</b> Discussions and support topics linked to the Code Repository certified code. <ul style="list-style-type: none"> <li>Code Repository (Uncertified)</li> </ul>	186 Topics 1,033 Replies	Yesterday, 10:30 AM In: [Discuss] State Editor for ... By: Ton Plomp
<b>Code In-Development</b> Use this forum to discuss code that may or may not qualify for the code repository but you just need somewhere to upload it and share with the LabVIEW community.	45 Topics 359 Replies	Today, 09:52 AM In: PHP Calendar Question By: Ton Plomp

Worldwide Incorporation  
 HK Offshore Onshore & Mainland China Co. Formation  
[www.topworldreg.com](http://www.topworldreg.com)

Open company in Belize  
 All you need to register offshore Fast & safe services in Belize  
[Company-Express.com/b...](http://Company-Express.com/b...)

Top Reputation

Aristos Queue	156
Daklu	130
ShaunR	115
vugie	105
jpgcode	102
François Normandin	100
crelf	94
Yair	86
PaulG.	79
jcardmody	74

Internet | Protected Mode: On



## Links of interested:

- Developers Zone:  
<http://zone.ni.com/>
- NI Support  
<http://www.ni.com/Support>
- Alliance Program  
<http://www.ni.com/alliance/>
- Training  
<http://www.ni.com/training>
- LabVIEW Fundamentals Exam  
[http://www.ni.com/training/labview\\_exam.htm](http://www.ni.com/training/labview_exam.htm)

# Sweden 2011



**NIDays 2011 – The LabVIEW Conference** - offers you a full day with a range of innovative technical sessions. New in 2011 is the special focus on LabVIEW use.

More information and registration, go to [ni.com/sweden/nidays](http://ni.com/sweden/nidays)

**The Graphical System Design Awards** – the technical paper contest that showcases the most innovative uses of virtual instrumentation in teaching or research as well as from engineers in industry.

One winner will be recognized for his achievement with a prize and his paper will be published on our website and be presented on NIDays 2011.

Prizes: iPad, Trophy, Keynote time on NIDays

More details on [ni.com/sweden/gsdawards](http://ni.com/sweden/gsdawards)

# Denmark 2011



**NIDays 2011 – The LabVIEW Conference** - offers you a full day with a range of innovative technical sessions. New in 2011 is the special focus on LabVIEW use.

More information and registration, go to [ni.com/denmark/nidays](http://ni.com/denmark/nidays)

**The Graphical System Design Awards** – the technical paper contest that showcases the most innovative uses of virtual instrumentation in teaching or research as well as from engineers in industry.

One winner will be recognized for his achievement with a prize and his paper will be published on our website and be presented on NIDays 2011.

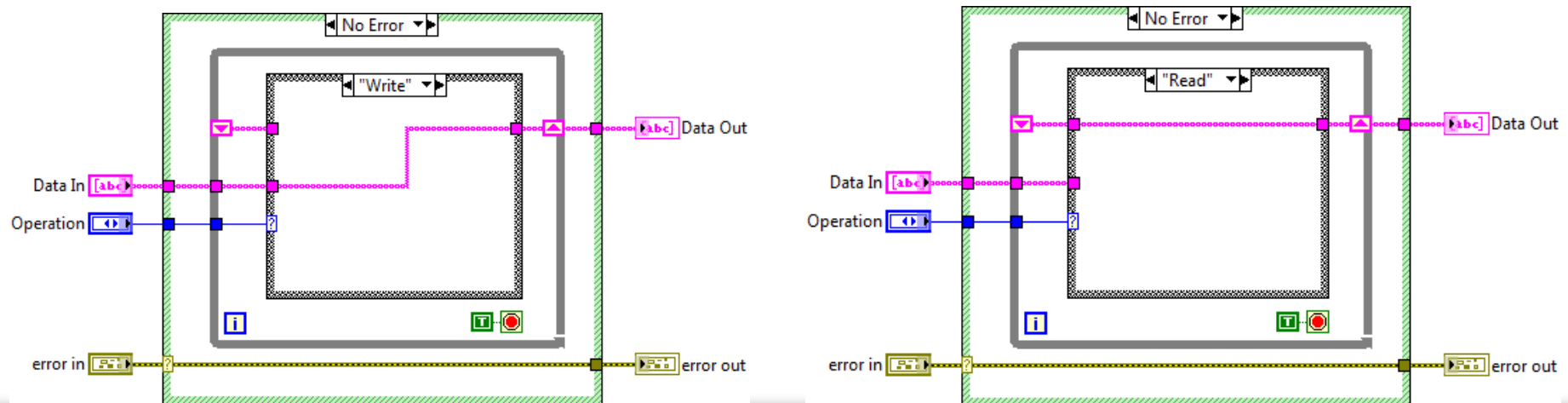
Prizes: iPad, Trophy, Keynote time on NIDays

More details on [ni.com/denmark/gsdawards](http://ni.com/denmark/gsdawards)

# Questions?

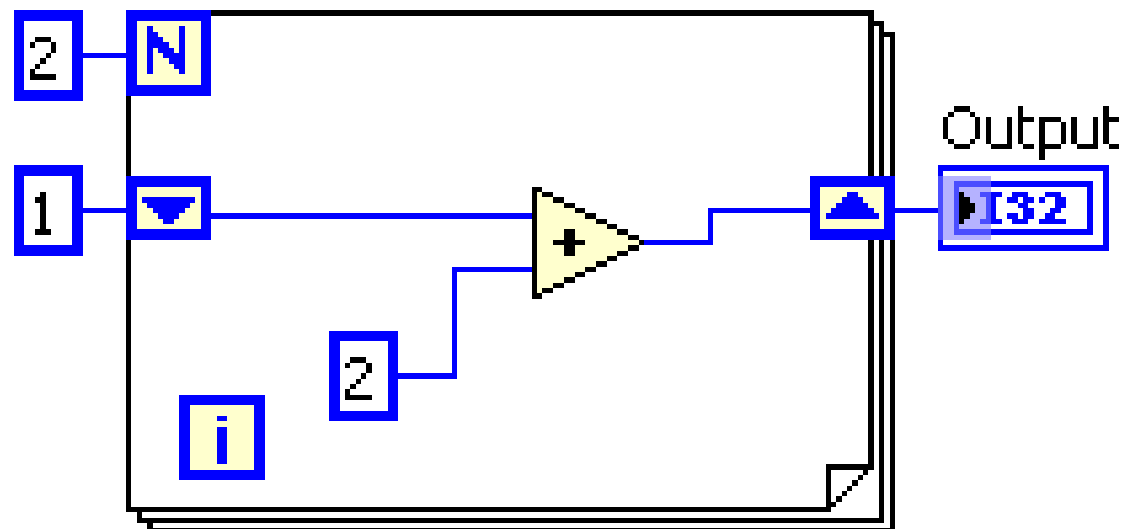
# Functional Global Variable (FGV)

- Fast
- More memory efficient and reliable than Local or Global Variables
- Expanded Functionality possible



# Shift Registers

- When using loops, you often need to remember data from previous iterations
- Shift registers transfer values from one loop iteration to the next



# Shift Registers - Initializing

Run once

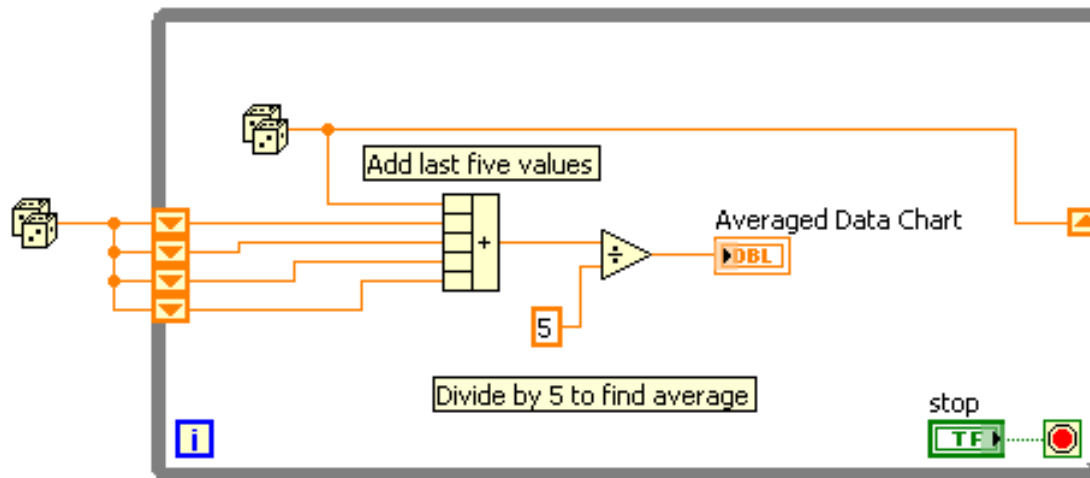
VI finishes

Run again

Block Diagram	1st run	2nd run
<p>Initialized Shift Register</p>	Output = 5	Output = 5
<p>Not Initialized Shift Register</p>	Output = 4	Output = 8

# Stacked Shift Registers

- Stacked shift registers remember values from multiple previous iterations
- Right-click the left shift register and select **Add Element**





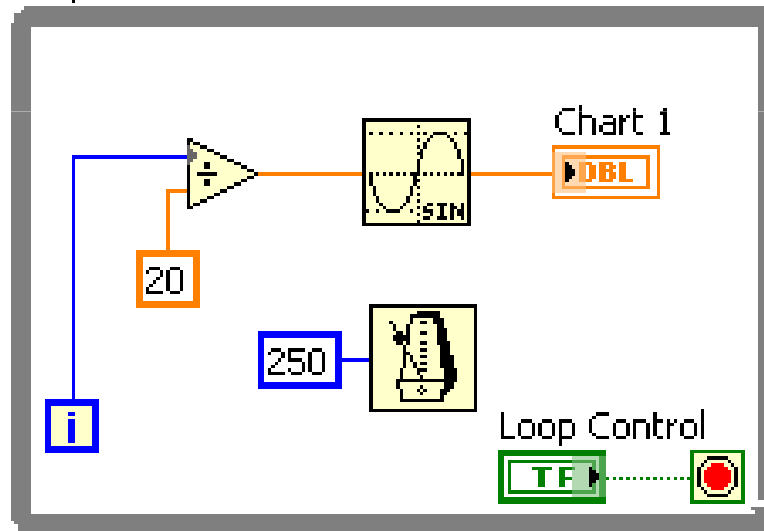
# Local Variables

- Local Variables make no use of the standard Data Flow principle
- Ideal for situation when wires cannot be used, for example between separate loops
- Do not overuse! Shift Registers are more effective
- More advanced cases use Queues or Notifiers instead

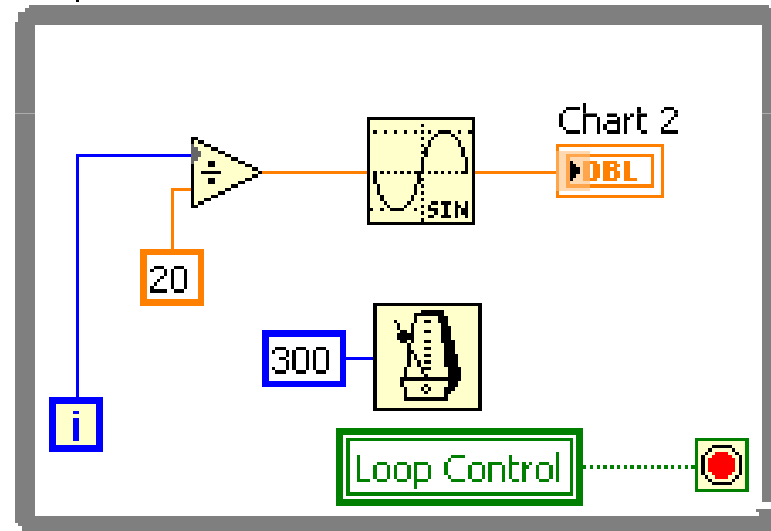
# Local Variables

- Good use case: Use local variables to pass stop command between parallel loops

Loop 1

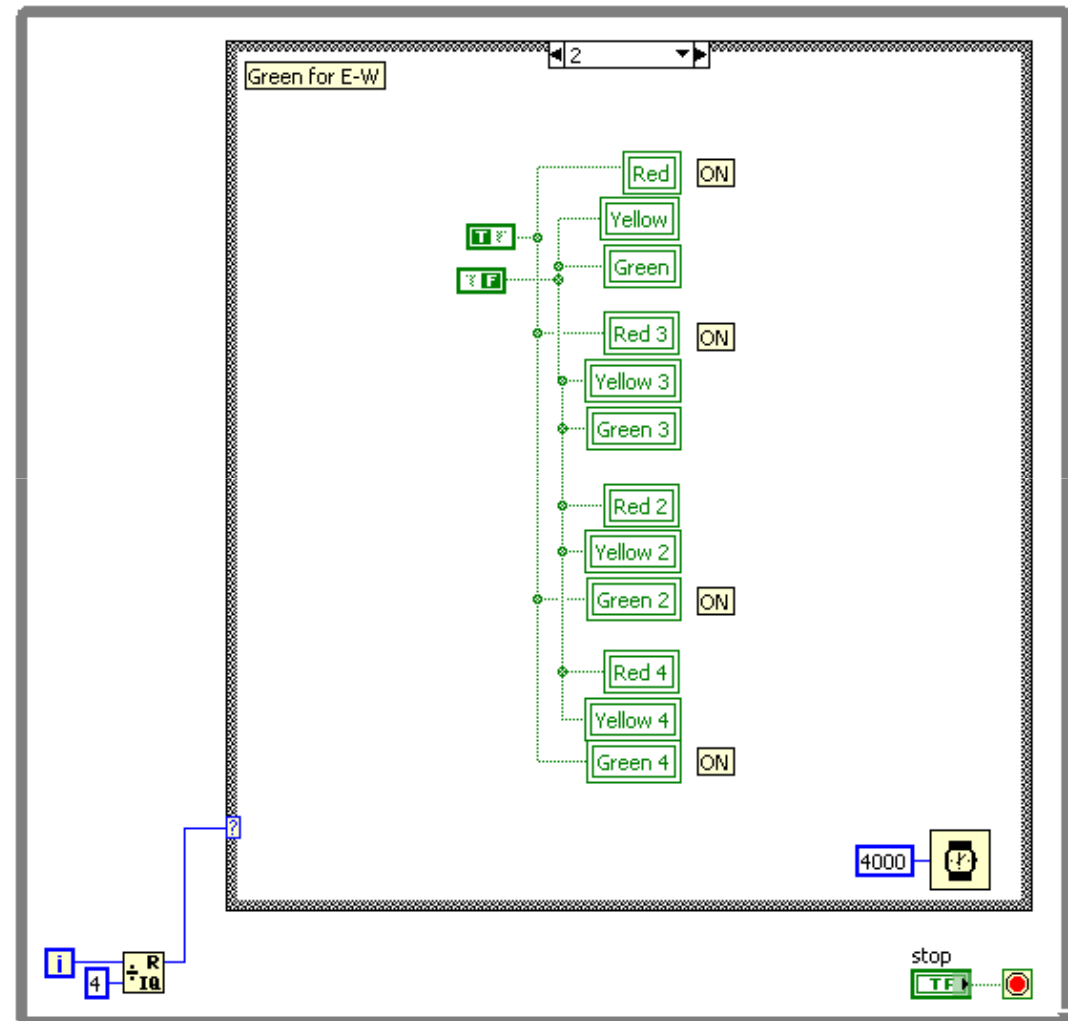


Loop 2



# Local Variables

- Bad use case:  
Main Architecture
- Use Cluster and  
Shift Registers  
instead!



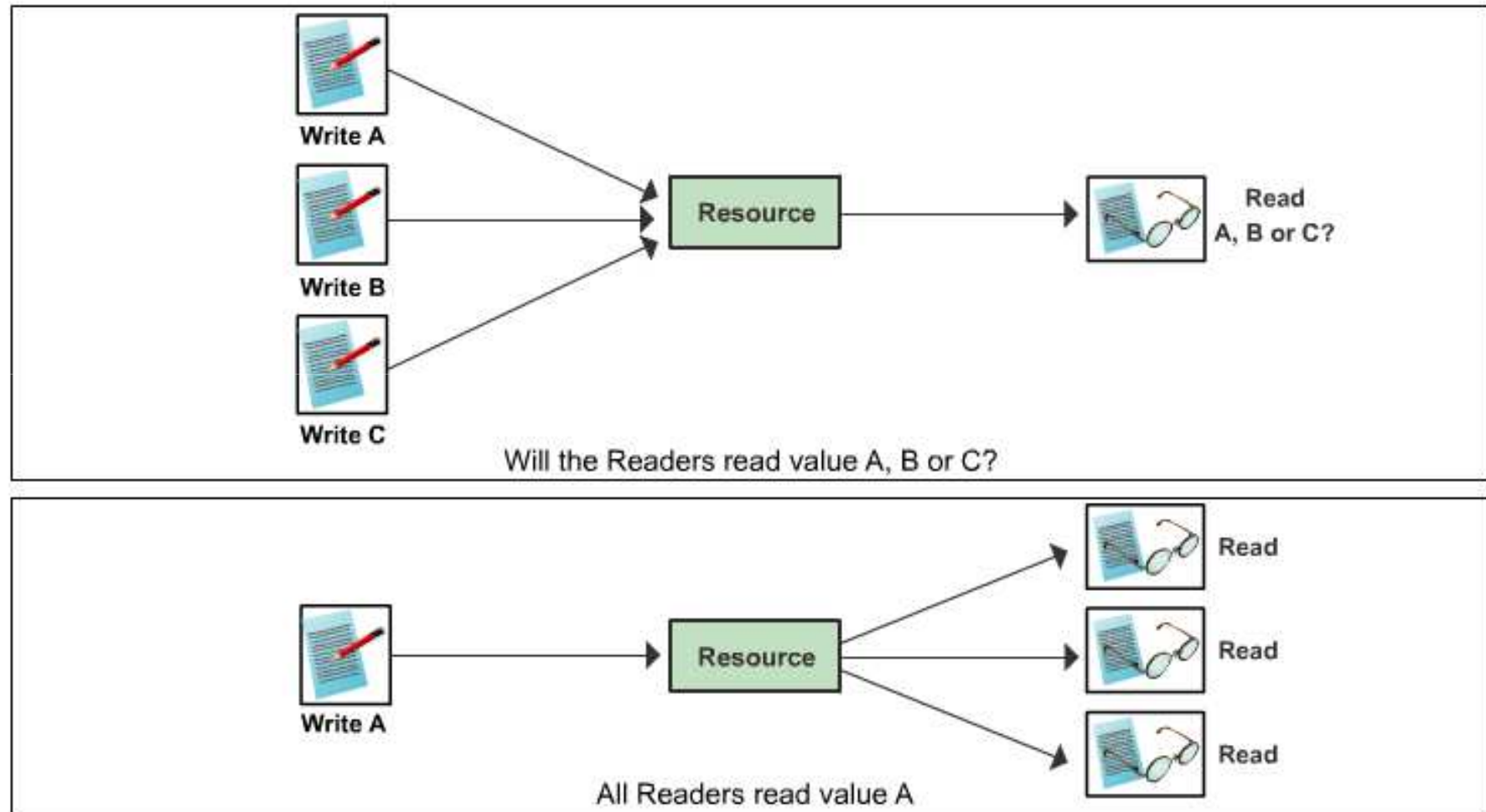
# Race Conditions

- A race condition is a situation where the timing of events or the scheduling of tasks may unintentionally affect an output or data value
- Race conditions are a common problem for programs that execute multiple tasks in parallel and share data between the tasks

# Race Conditions

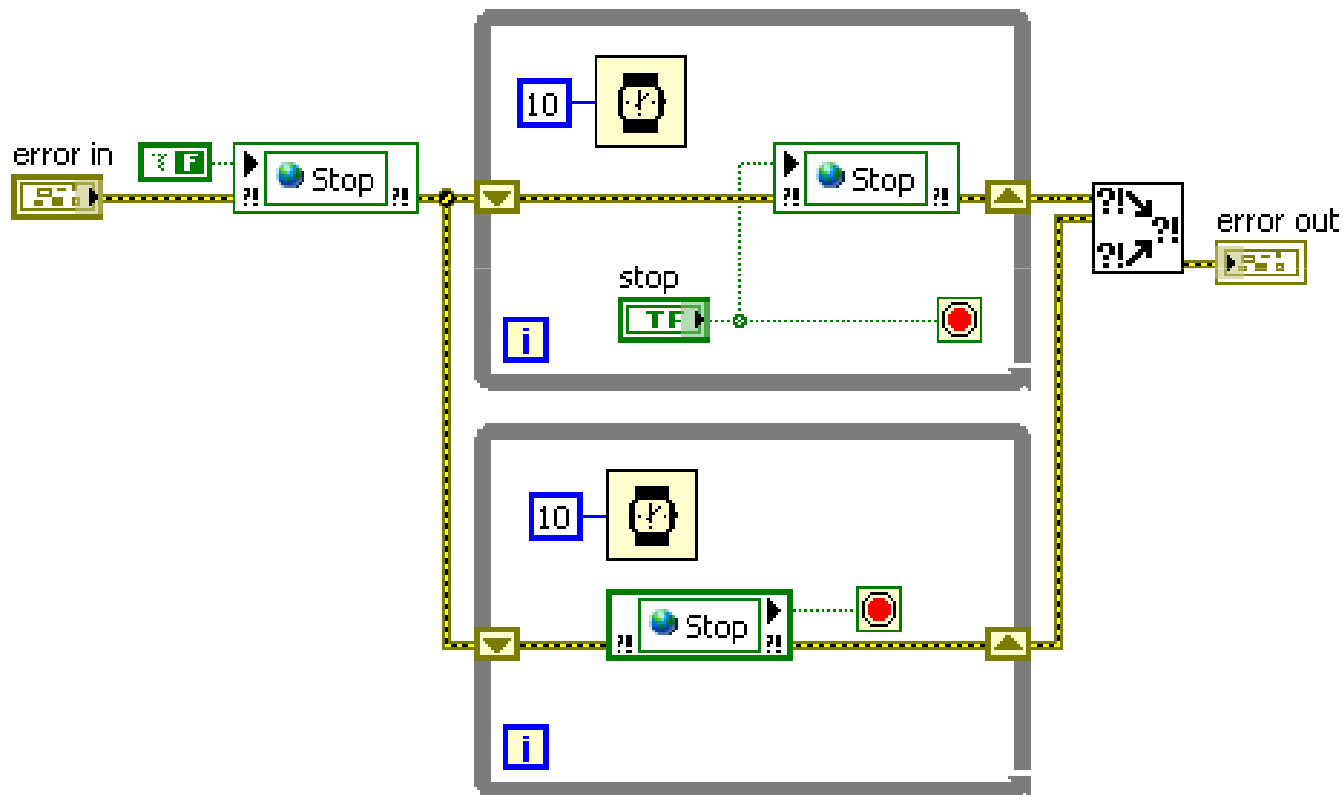
- Race conditions are very difficult to identify and debug
- Often, code with a race condition can return the same result thousands of times in testing, but still be capable of returning a different result
- Avoid race conditions by:
  - Controlling shared resources
  - Properly sequencing instructions
  - Identifying and protecting critical sections within your code
  - Reducing use of variables

# Race Conditions – Shared Resources



# Local Variables

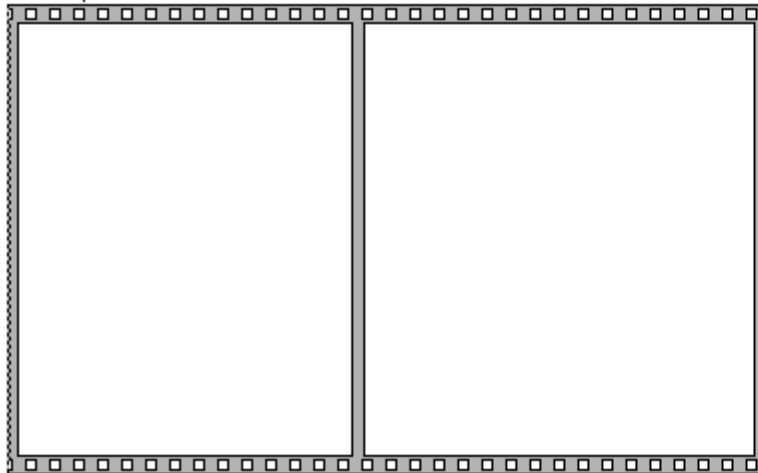
- Tip: Don't forget to initialize!



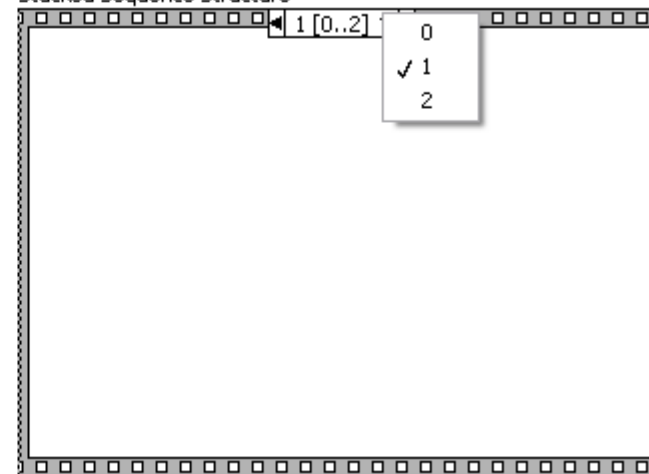
# Sequence Structures

- Performs actions in a specific order
- Needs to run its course, no abort
- Both Flat and Stacked

Flat Sequence Structure



Stacked Sequence Structure





# Sequence Structures

- Use only in special cases
  - Run code without error inputs at special time
  - FPGA Code
- Not as Main Structure of your application

