Using LabVIEW for High Performance Computing

March 4, 2010

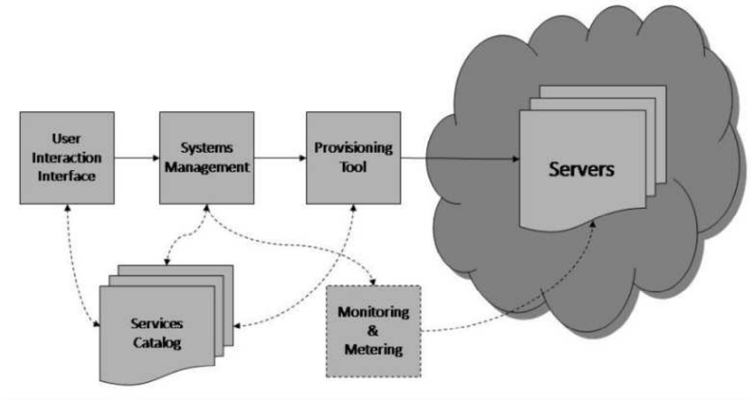
Terry Stratoudakis
terry@aleconsultants.com

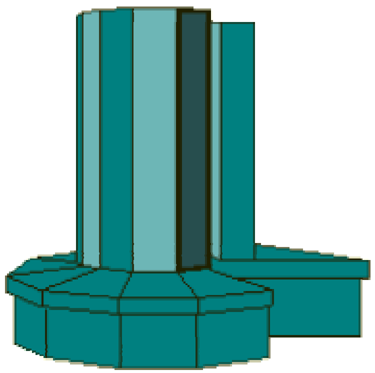
Agenda

1. What is High Performance Computing?
2. HPC Users
3. HPC Technologies
4. LabVIEW in HPC
5. Case Study: Option pricing on an FPGA

What is HPC?

- Solve advanced computation problems
- HPC is successor of Supercomputing
- Complex event processing
- Parallel computing



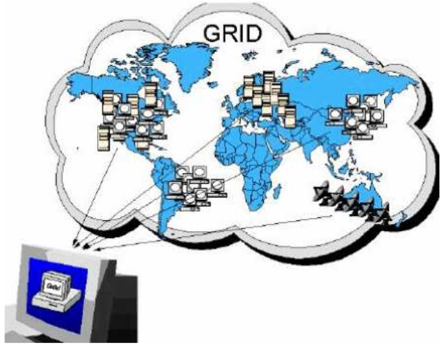


Users of HPC

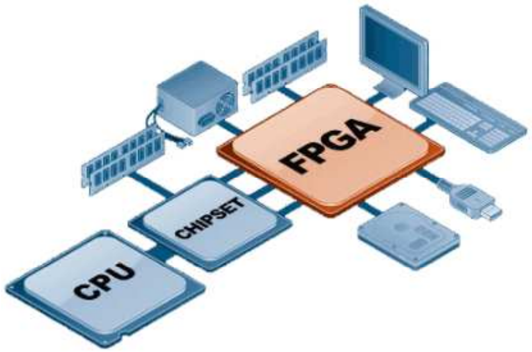
- Bioinformatics
- Cryptography
- Defense
- High Energy Physics
- Finance
- Telecommunications
- You

every time you use Google/search
for something, you are a user of HPC!

HPC Technologies

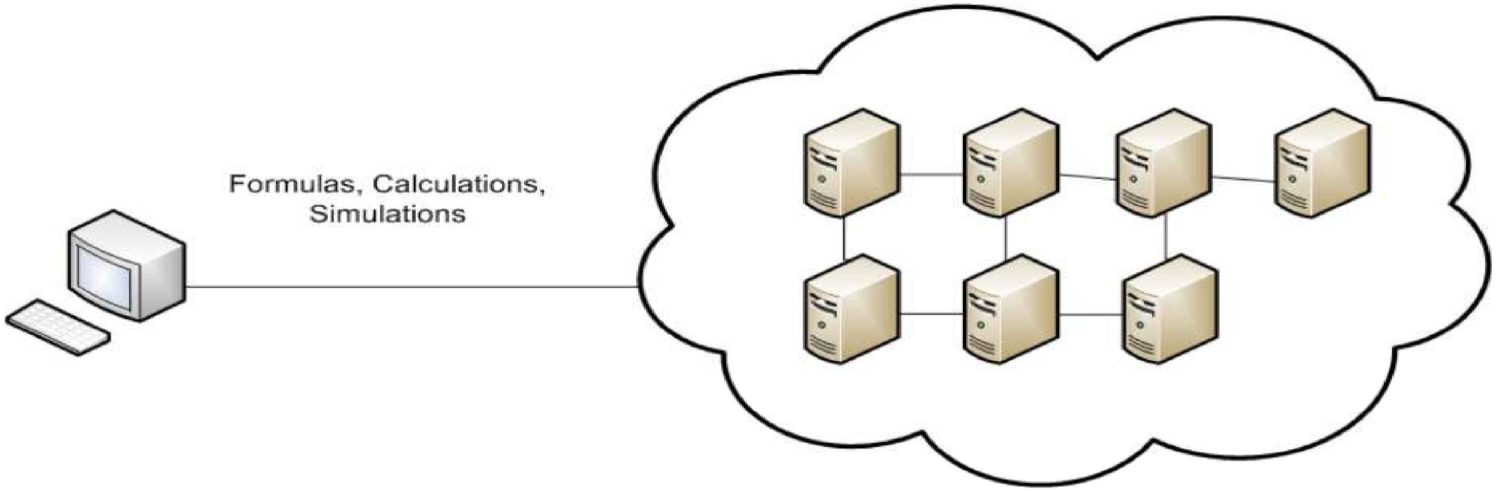


- Grid computing
- Multi-core & specialized processors
- Embedded
- Storage



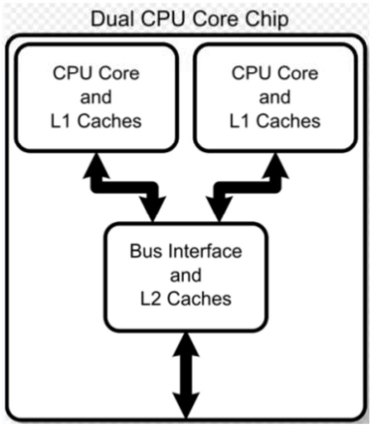
Grid Computing

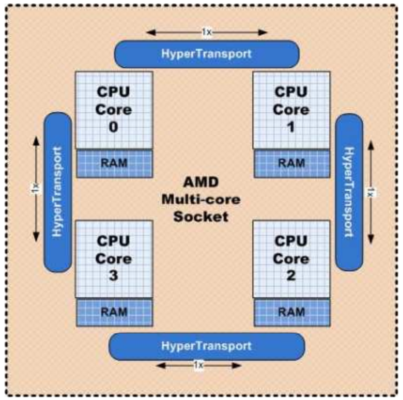
- Networked computers working together
- Most existing software <u>cannot</u> run "as is"
- Requires knowledge of parallel programming APIs and languages
- SaaS, Cloud, Cluster



Multi-Core Processors

- Solves temperature issues of 1-core
- Many processors on a chip
- Building block of a grid
- Similar challenges as for grid computing



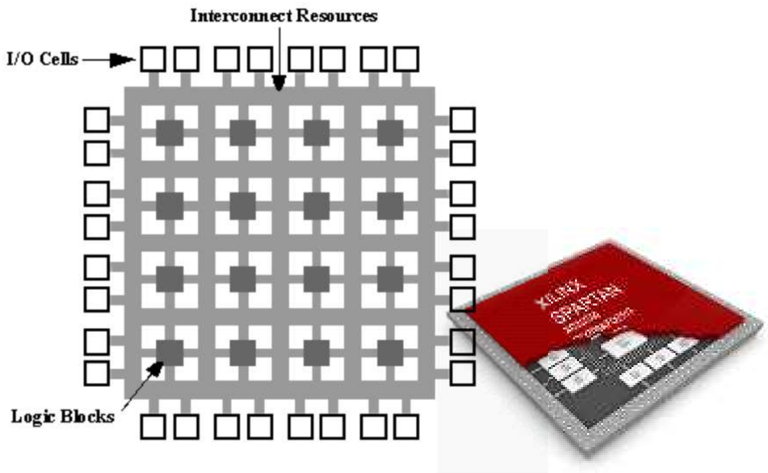


Specialized Processors

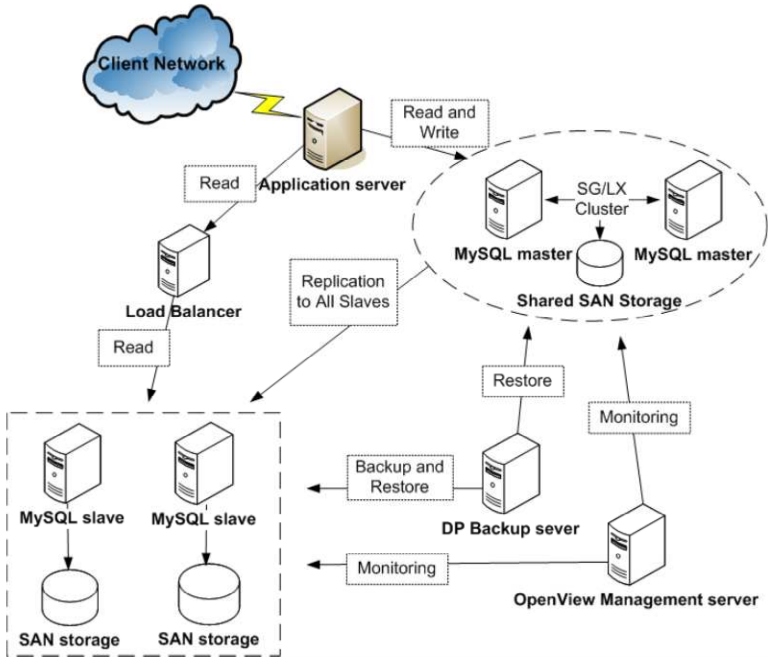
- CPUs too generic
- Optimized for certain calculations
- Examples
  - Graphical Processing Units (GPUs)
  - Digital Signal Processing (DSP)
  - Cell Processors
    - Example: grid of 10 networked Sony PlayStation 3

Embedded HPC

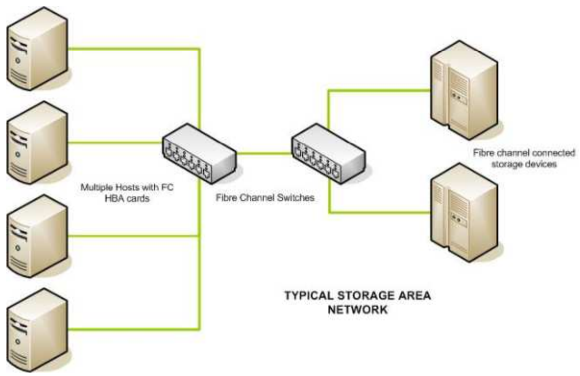
- Field Programmable Gate Arrays (FPGAs)
- Configured with Hardware Description Language
- *True* parallel execution



Storage



- Solid State HD
- RAID Arrays
- Storage Area Networks



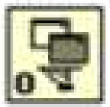
LabVIEW in HPC

- Grid Computing – VI Server
- Multicore
  - Parallel For Loops
  - Parallel Loops
  - Parallel Code
- GPU – CUDA interface to LabVIEW
- DSP – LabVIEW DSP Module
- FPGA – LabVIEW FPGA Module

VI Server

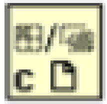
Create Reference to Object → **Operate on Properties Or Methods** → **Close Reference to Object** → **Check for Errors**

Open Application Reference



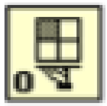
Property Node

Close Reference

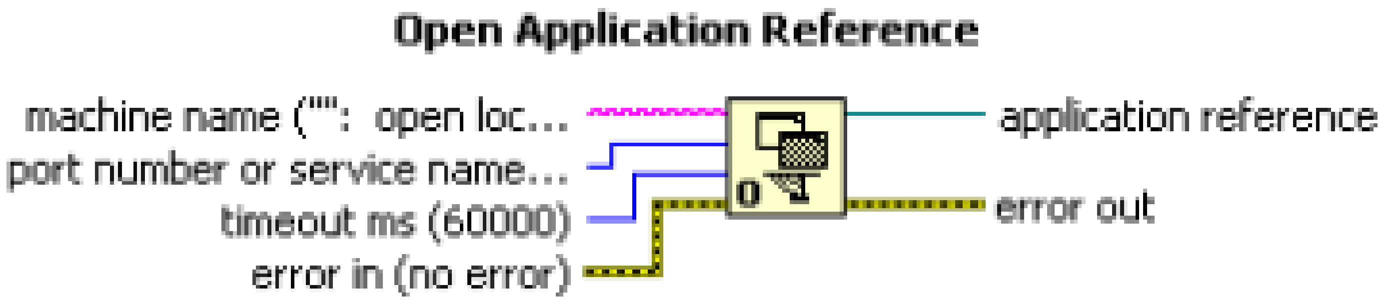


Simple Error Handler.vi

Open VI Reference

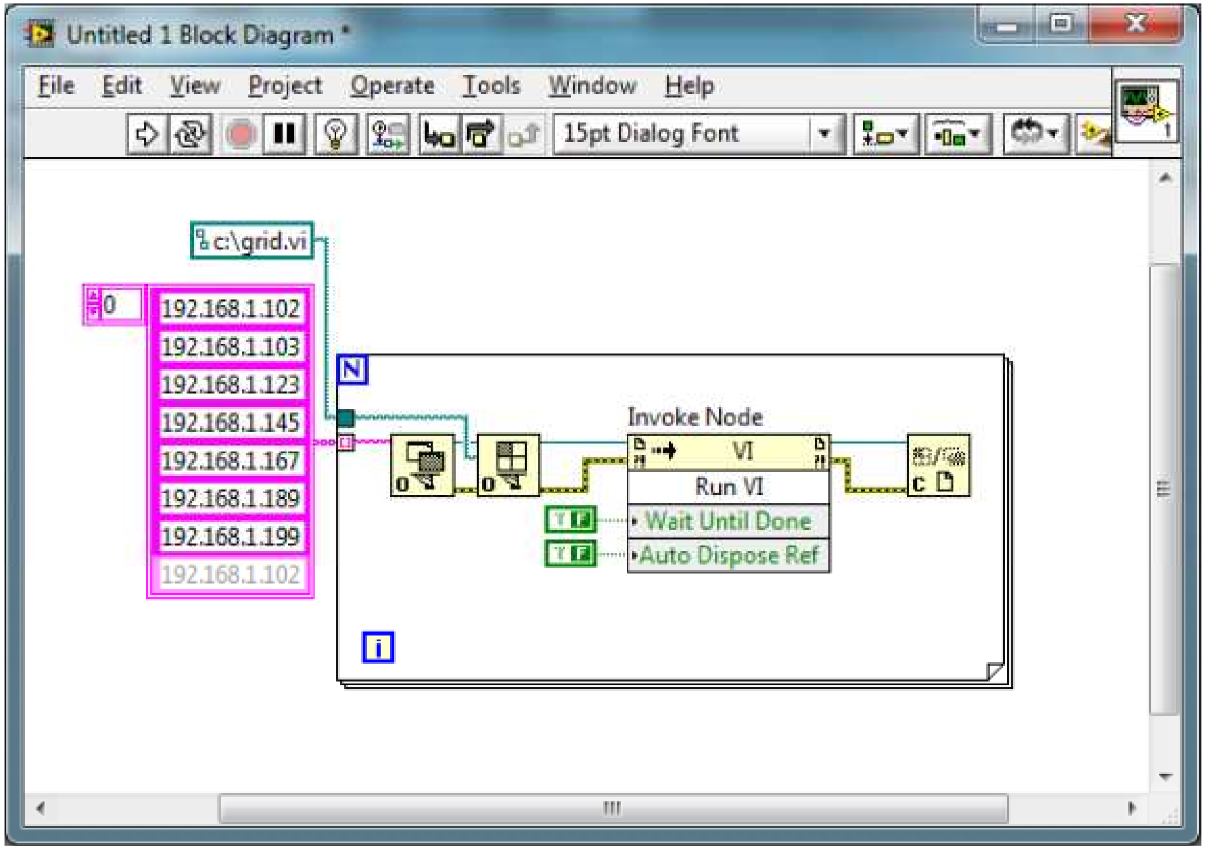


Invoke Node

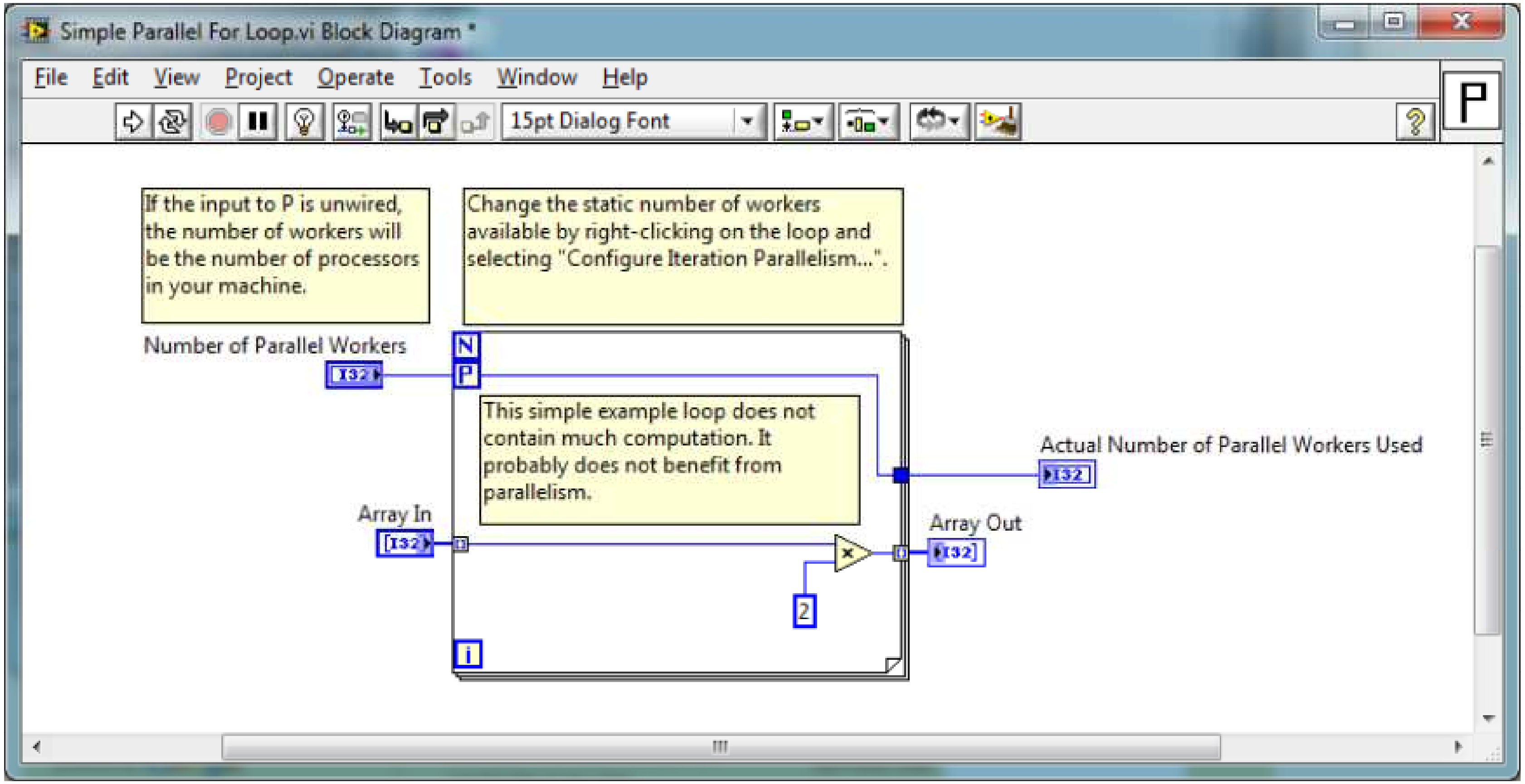


- Application reference—Input to property and invoke nodes
- To reference LabVIEW on a **remote computer**, set machine name to TCP/IP address or domain name

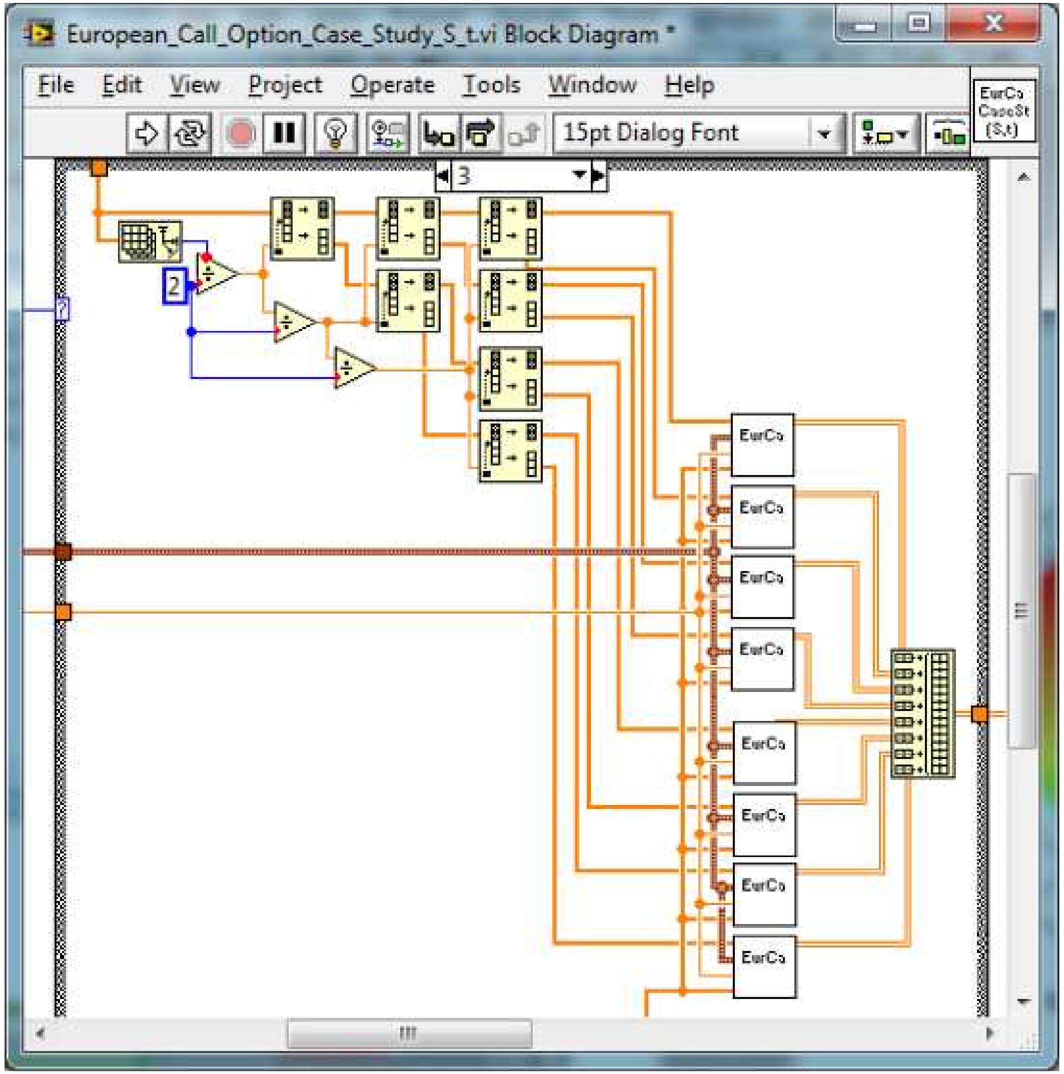
LabVIEW on a Grid Example



Multi-core Parallel For Loop



Multi-core Parallel Code



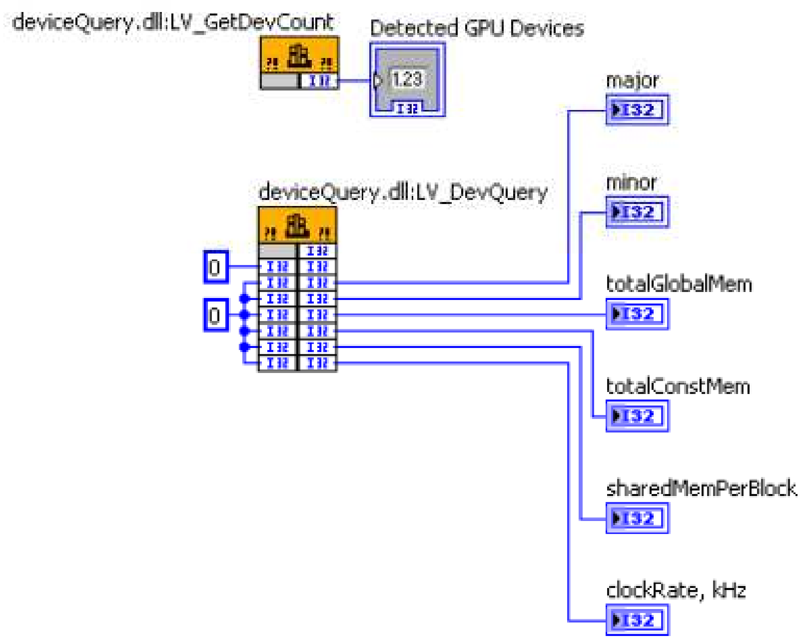
GPU – LabVIEW to CUDA Interface

deviceQuery.h | deviceQuery_cuda.c | deviceQuery_cuda.h | deviceQuery.c

```c
//================================================================================
//
// Title:       deviceQuery
//
//================================================================================

#include <utility.h>
#include "deviceQuery.h"

int LV_GetDevCount()
{
    int dev;
    dev = GetDevCount();
    return dev;
}

int LV_DevQuery(int dev,int *major, int *minor,
                         int *totalGlobalMem, int *totalConstMem, int *sharedMemPerBlock,
                         int *clockRate)
{
    int res;
    res = DevQuery(dev, major, minor,
                         totalGlobalMem, totalConstMem, sharedMemPerBlock,
                         clockRate);
    return res;
}

//================================================================================
// DLL main entry-point functions

int __stdcall DllMain (HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
    switch (fdwReason) {
        case DLL_PROCESS_ATTACH:
            break;
        case DLL_PROCESS_DETACH:
            break;
    }
    return 1;
}
```

http://decibel.ni.com/content/blogs/AndreyDmitriev/2009/04/09/using-nvidia-gpu-from-labview-with-cuda-and-cvi

GPU – LabVIEW to CUDA Interface



http://decibel.ni.com/content/blogs/AndreyDmitriev/2009/04/09/using-nvidia-gpu-from-labview-with-cuda-and-cvi

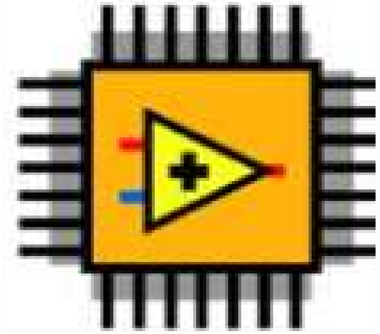
FINANCE CASE STUDY

OPTION PRICING ON AN FPGA

Field Programmable Arrays (FPGA)

- Introduced in 1987
- Customizable Integrated Circuit
- Millions of logic gates on a single chip
- Parallel Execution, Low Power Usage
- No Operating System

LabVIEW FPGA Module



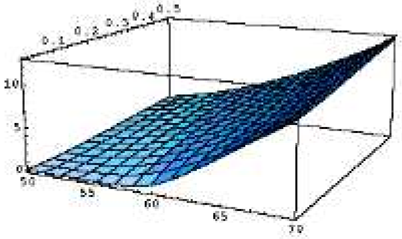
- Higher level of Abstraction
  - Reduce FPGA development by 75%
- Add-on to LabVIEW – since 2002
- Used in Defense, Biomedical, Telecom., Manufacturing

HPC to LabVIEW FPGA Process

1. Understand algorithm
    a. Look for ability to parallelize
    b. Identify math functions needed
        - e.g. logarithmic, division, multiply, exp, random numbers)
        - See NI IPNet (www.ni.com/ipnet)
2. Implement in LabVIEW FPGA
    a. Goal: run in single-cycled timed loop
    b. Pipelining
3. Test with simulated mode
4. Verification with known data

Black-Scholes Option Valuation

- Published in 1973
- Basis for Quantitative Finance
  - Equity price modeled as stochastic time series
- Pricing of Options and Corporate Liabilities
- Basis for multi-trillion dollar Options Trading
- Computed with a Monte Carlo Simulation



$$dS_t = \mu S_t\,dt + \sigma S_t\,dW_t$$

$$\nu = \frac{\partial V}{\partial \sigma}$$

$$u(x,\tau) = \frac{1}{\sigma\sqrt{2\pi\tau}}\int_{-\infty}^{\infty} u_0(y)e^{-(x-y)^2/(2\sigma^2\tau)}\,dy.$$

$$\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{z^2}{2}}\,dz$$

$$dV = \left(\mu S\frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2}\right)dt + \sigma S\frac{\partial V}{\partial S}\,dW.$$

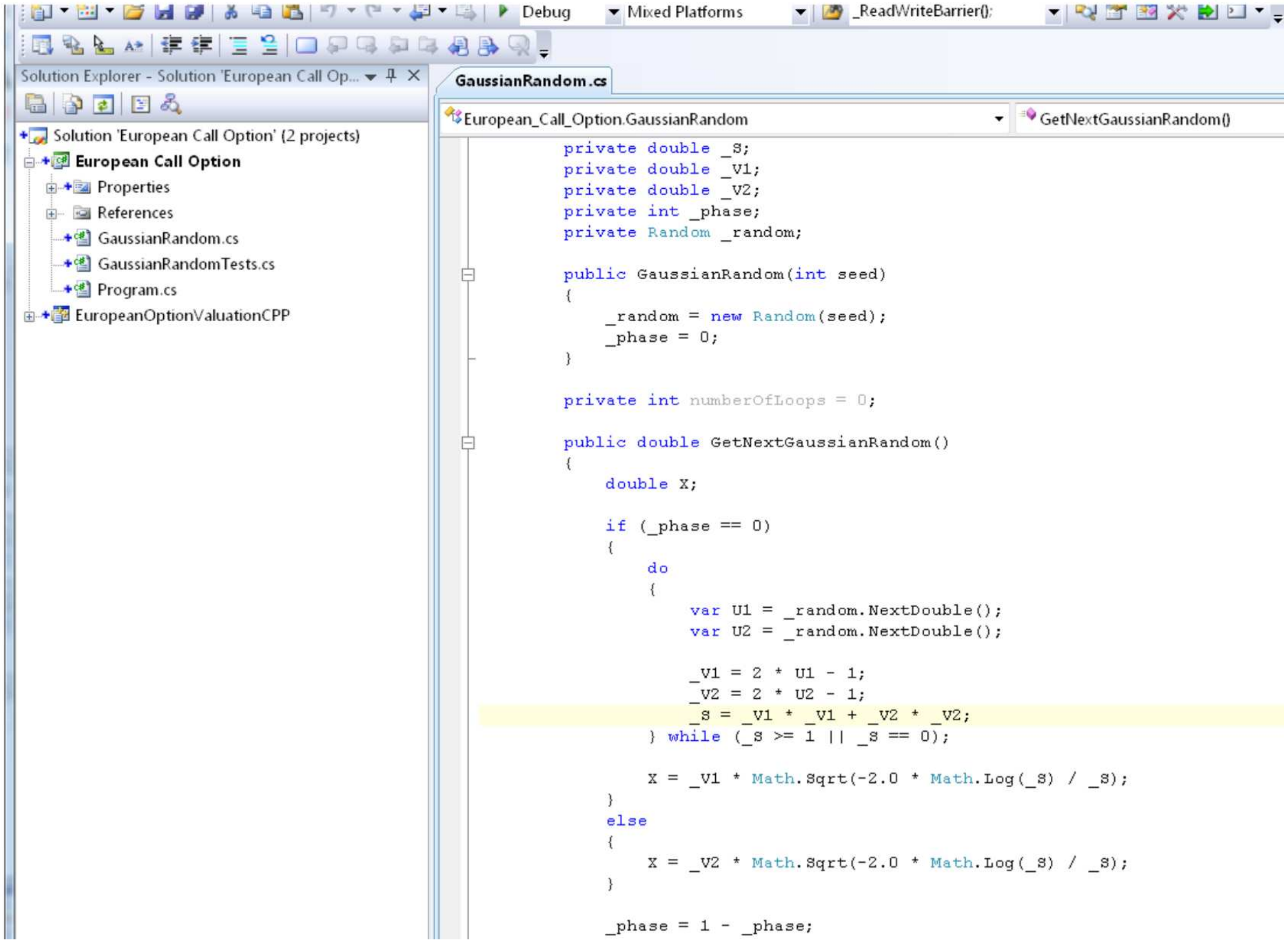
Challenge

- Program Black–Scholes Option Valuation:
  - NI Compact-RIO platform (Xilinx FPGA)
    - Running National Instruments LabVIEW 8.6.1
  - Alienware Area-51 7500 Dual Core
    - Running Microsoft Visual C# .NET 2.0
- Benchmark
  - Development time
  - Execution time
  - Energy Consumption

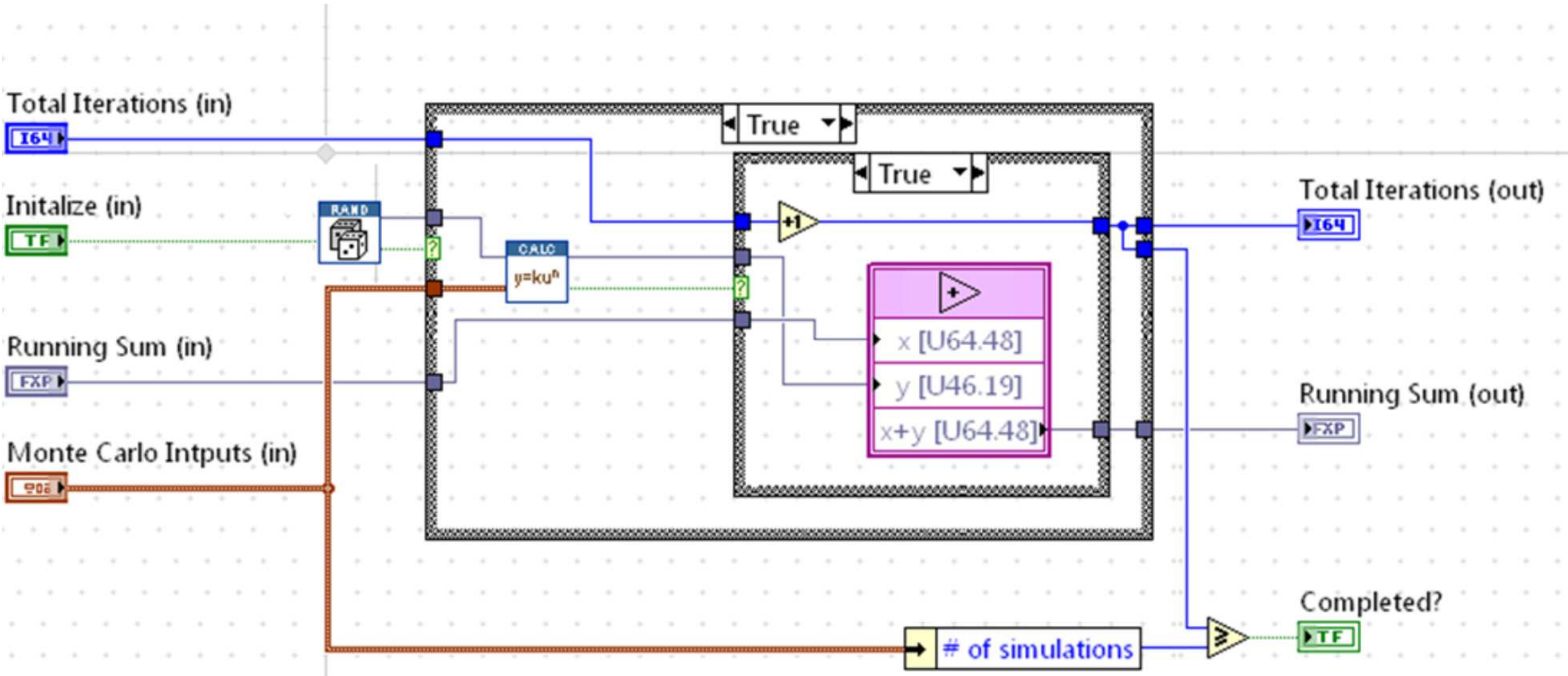
Visual C# on Dual-Core PC

- Microsoft Windows Vista Ultimate Edition
- High-Performance Gaming Machine
- 3.0 GHz Intel Core 2 Duo E6850
- SATA RAID-0 10,000 RPM Hard Drives
- 4 GB RAM
- .NET 2.0 Runtime

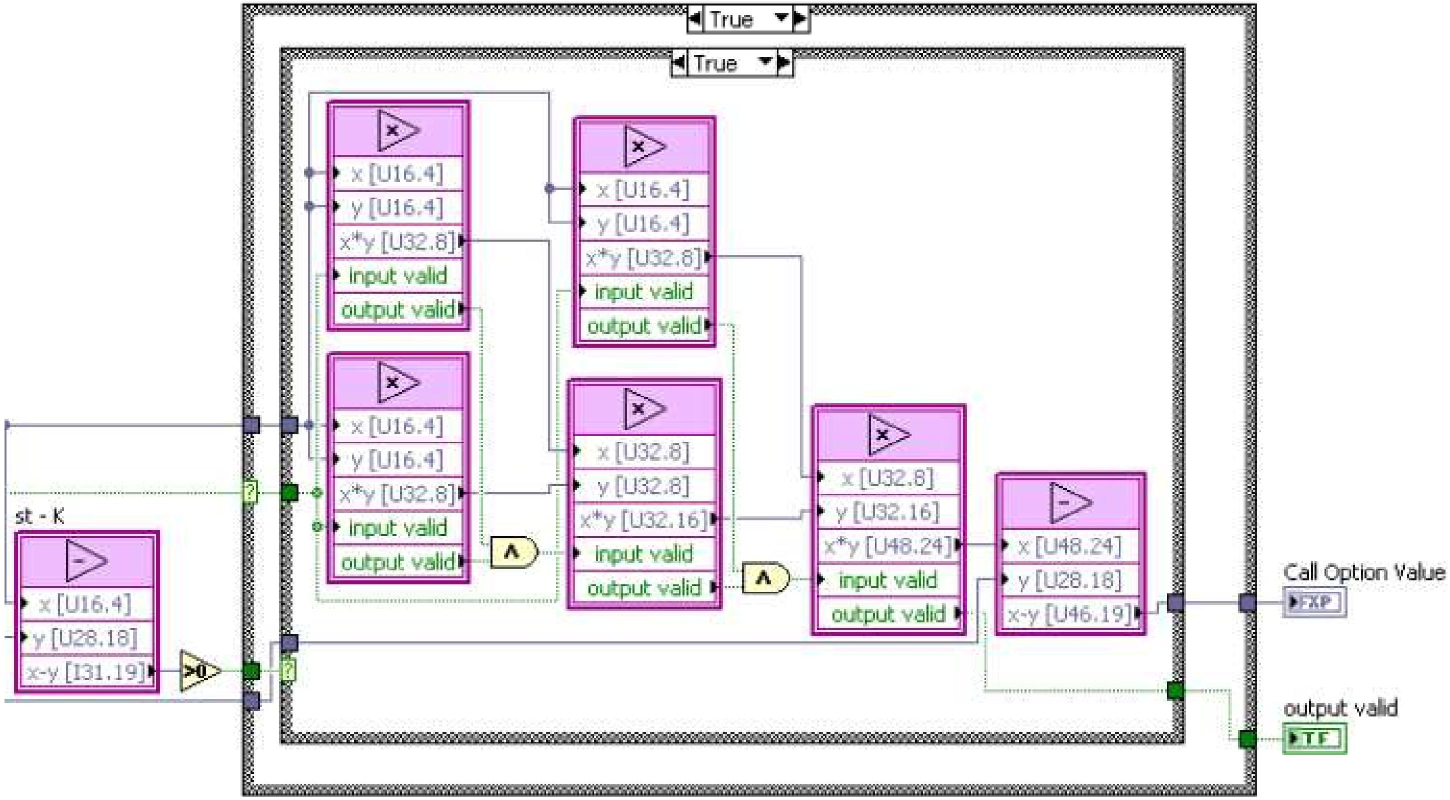
Black Scholes - Visual C#



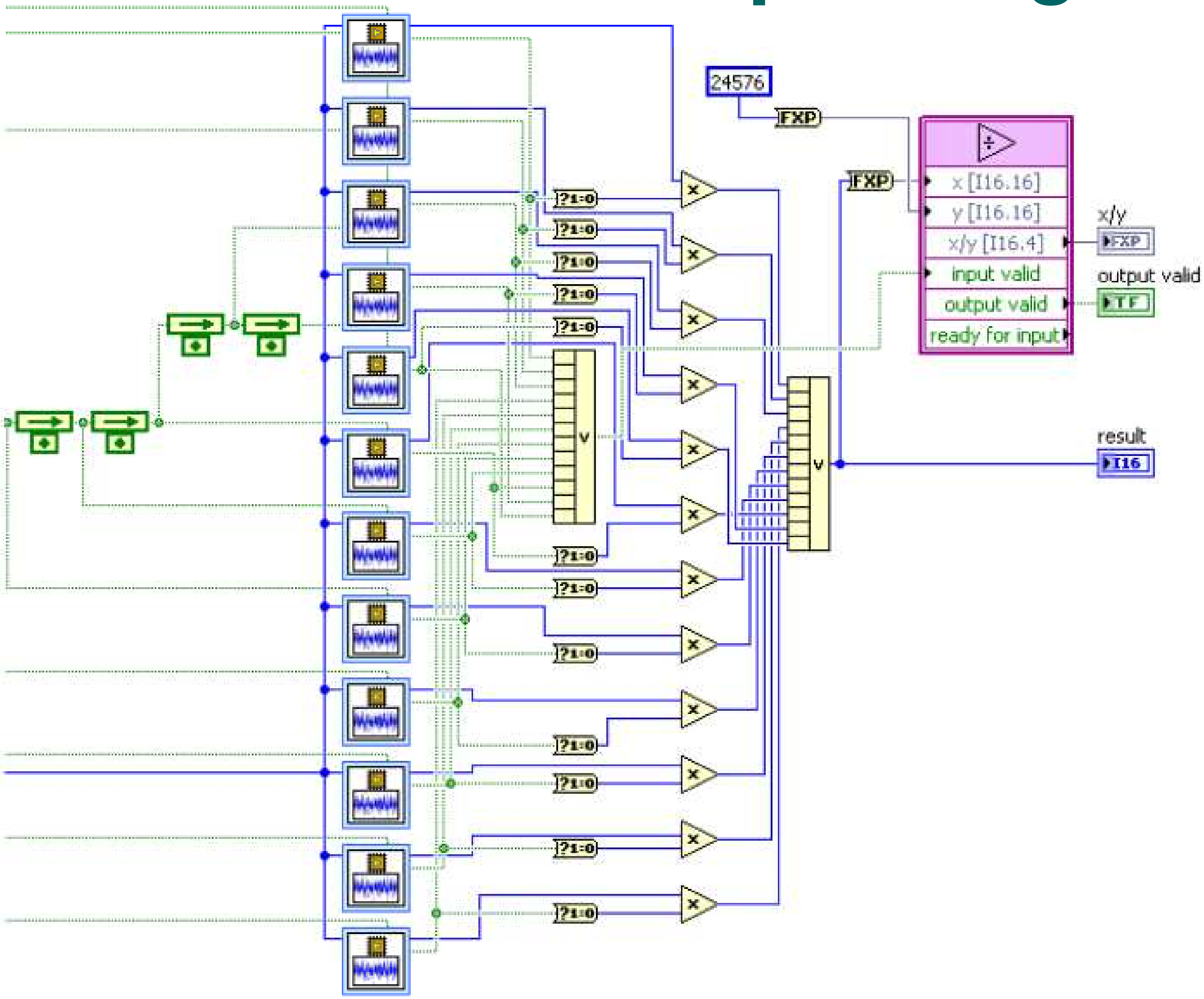
Black-Scholes on LabVIEW FPGA



LabVIEW FPGA – Fixed Point Math



LabVIEW FPGA – Pipelining



Results

- Development times were comparable
- LabVIEW on FPGA ran 59X faster
- LabVIEW on FPGA had 33X energy reduction
- Compact-RIO takes up 1/8 the space

More info at WallStreetFPGA.com

Benefits of LabVIEW FPGA for HPC

- Case Study Results
  - Quick development
  - Energy efficient
  - Fast execution
- LabVIEW for FPGA can be faster than text based programming running on a grid

ALE SYSTEM INTEGRATION

www.aleconsultants.com – info@aleconsultants.com

- Based in Long Island, New York – projects nationwide
- National Instruments Certified Alliance Partner
  - All developers have National Instruments Certification
- Experience:
  - Test Labs, Manufacturers, Mil/Aero, Finance
  - Over 14 Years Test & Automation experience
  - Expertise in variety of instrument manufacturers' products
- Programming:
  - LabVIEW, LabWindows/CVI, TestStand, Visual Studio

Terry Stratoudakis, P.E.

Education/Certifications

- B.S., M.S. in Electrical Engineering, Polytechnic University
- NI Certified LabVIEW Developer and Certified Prof. Instructor
- New York State licensed Professional Engineer

Experience

- Test Engineer at Underwriters Laboratories for six years
- Former Assistant Adj. Prof. at NYC College of Technology
- Co-founder and President of ALE System Integration