USB Logic Analyzer

Shom Bandopadhaya
Advisor: Dr. James Irwin, Jr.

Bradley University
Department of Electrical and Computer Engineering

Presentation Outline

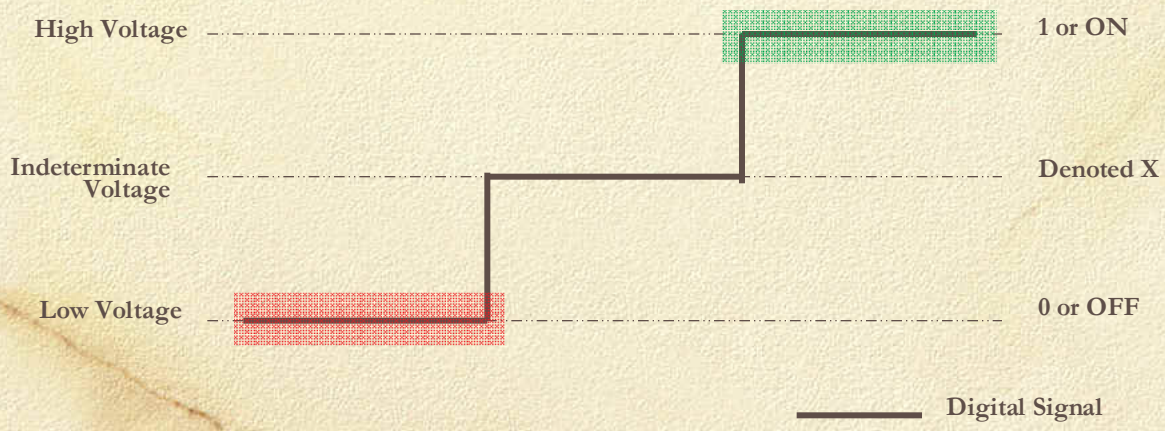
- Background Information
- Project Summary
- Project Goals
- Functional Description
- System Block Diagram
- Results
- Questions

Presentation Outline

- Background Information
- Project Summary
- Project Goals
- Functional Description
- System Block Diagram
- Results
- Questions

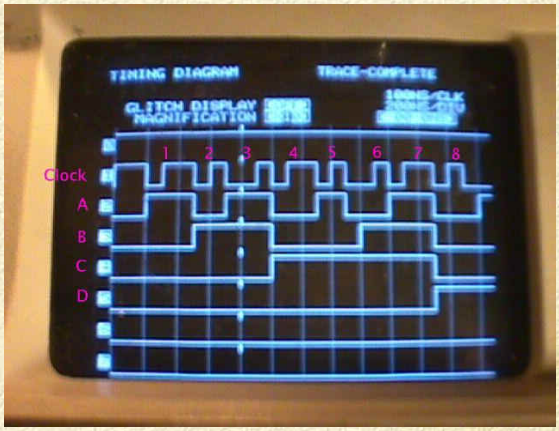
Logic Levels

- Logic Levels are used by electronic elements as a data and operation representation method. Logic Levels are based on a signal's voltage, a high voltage corresponds to 1 or ON; a low voltage corresponds to 0 or OFF.

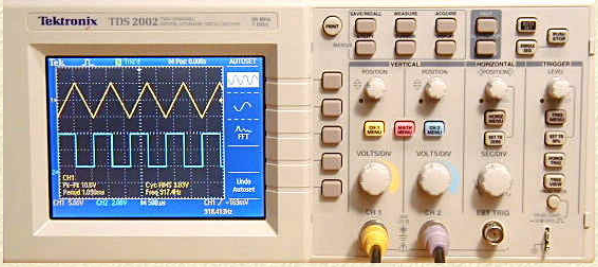


Logic Analyzers

- A logic analyzer displays signals in a digital circuit that are too fast to be observed by a human being. A logic analyzer captures digital data from a digital system that has too many channels to be examined with an oscilloscope and presents it to a user.



A HP 1615A Logic Analyzer showing a timing diagram.



A Tektronix oscilloscope showing square and triangle waveforms.

Presentation Outline

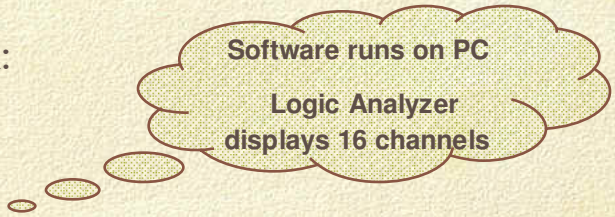
- Background Information
- Project Summary
- Project Goals
- Functional Description
- System Block Diagram
- Results
- Questions

Project Summary

- Provide Logic Analyzers for Junior Lab Workstations
- Workstations equipped with:
  - Personal Computer
  - Tektronix 2-channel oscilloscope (displays only 2 channels at a time)
- Juniors working with EMAC 8-bit µcontroller Boards (12MHz), CPLDs, GALs, Altera FPGA Boards (25 MHz)
  - Desired Sampling Rate = 50 MHz
  - Desired Data Width = 8 bits

Project Summary

- Provide Logic Analyzers for Junior Lab Workstations
- Workstations equipped with:
  - Personal Computer
  - Tektronix 2-channel oscilloscope (displays only 2 channels at a time)



- Juniors working with EMAC 8-bit µcontroller Boards (12MHz), CPLDs, GALs, Altera FPGA Boards (25 MHz)
  - Desired Sampling Rate = 50 MHz
  - Desired Data Width = 8 bits

Project Summary

- Provide Logic Analyzers for Junior Lab Workstations
- Workstations equipped with:
  - Personal Computer
  - Tektronix 2-channel oscilloscope (displays only 2 channels at a time)
- Juniors working with EMAC 8-bit µcontroller Boards (12MHz), CPLDs, GALs, Altera FPGA Boards (25 MHz)
  - Desired Sampling Rate = 50 MHz
  - Desired Data Width = 8 bits

Logic Analyzer samples over 100MHz

Max Data Width = 16 bits

Presentation Outline

- Background Information
- Project Summary
- Project Goals
- Functional Description
- System Block Diagram
- Results
- Questions

Hardware Goals

- Part of a different project, not in scope of this project
  - Design 16 channel PC board which:
    - Works with CMOS or TTL logic levels
    - Samples at upwards of 100 MHz
    - Acquires data based on Trigger Event
    - Interfaces with XEM 3001 FPGA board
  - VHDL code to interface FPGA board to PC

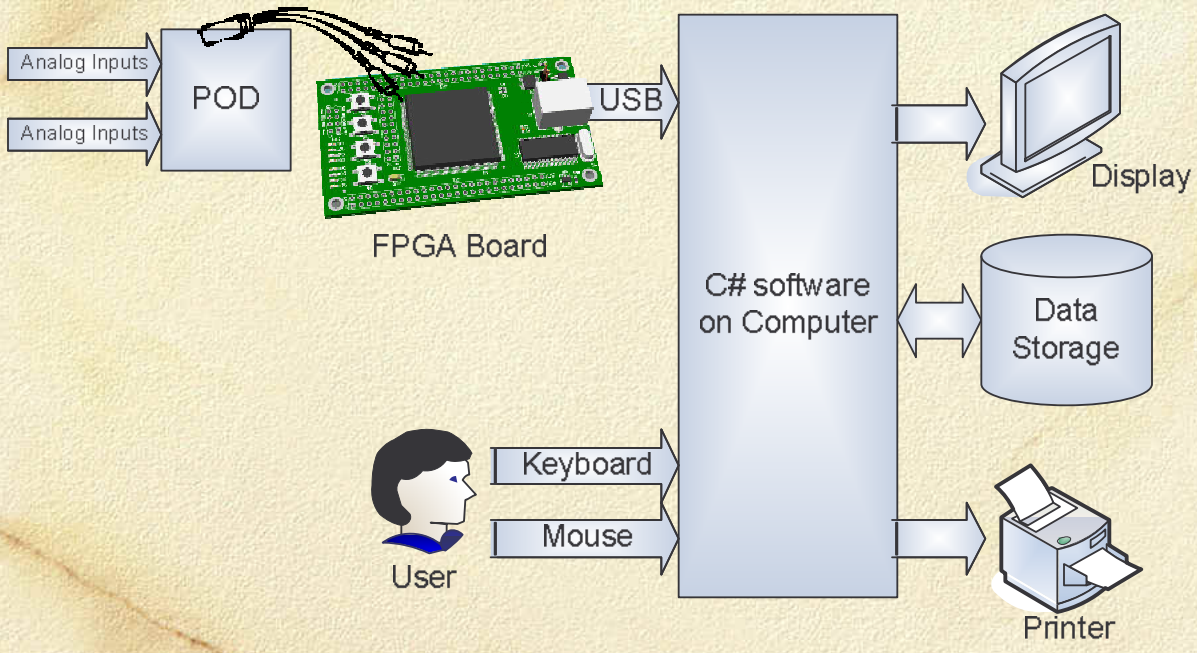
Original Project Goals

- **Software**
  - Display engine using NPlot graphing library
  - Plot acquired data as discreet logic signals
  - Combine data lines into data bus
  - Efficient data processing
  - Print/Save data capture

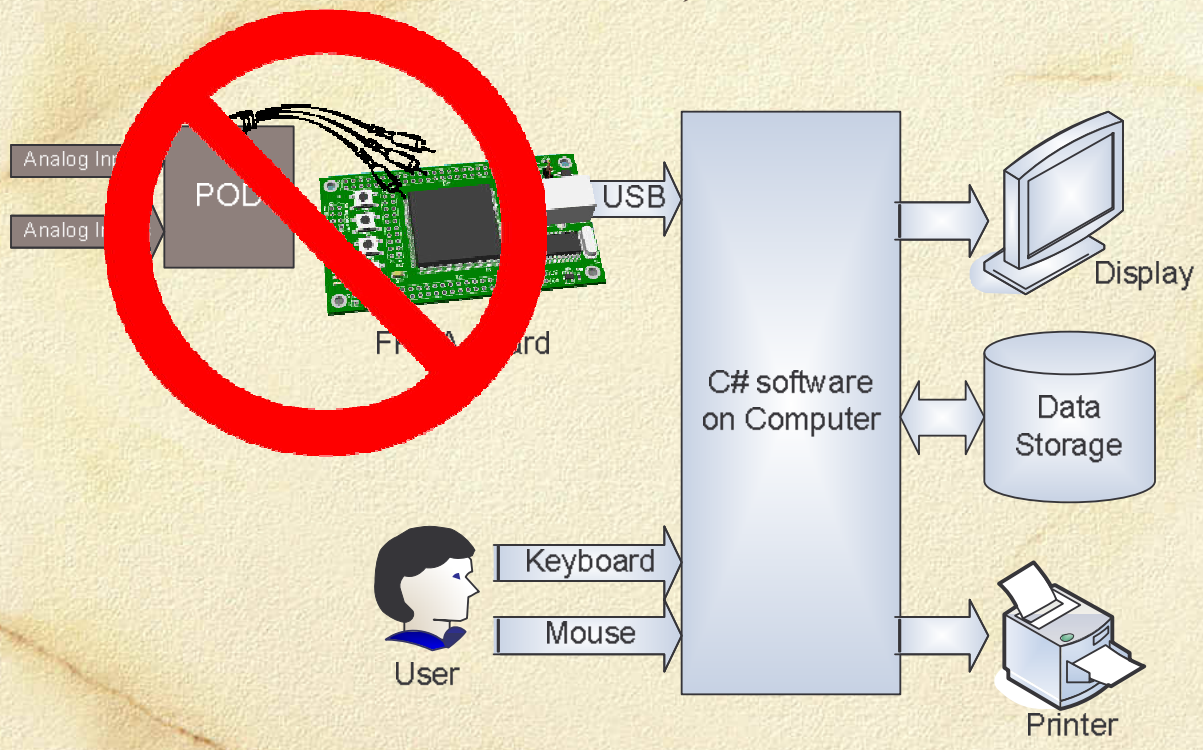
Presentation Outline

- Background Information
- Project Summary
- Project Goals
- Functional Description
- System Block Diagram
- Results
- Questions

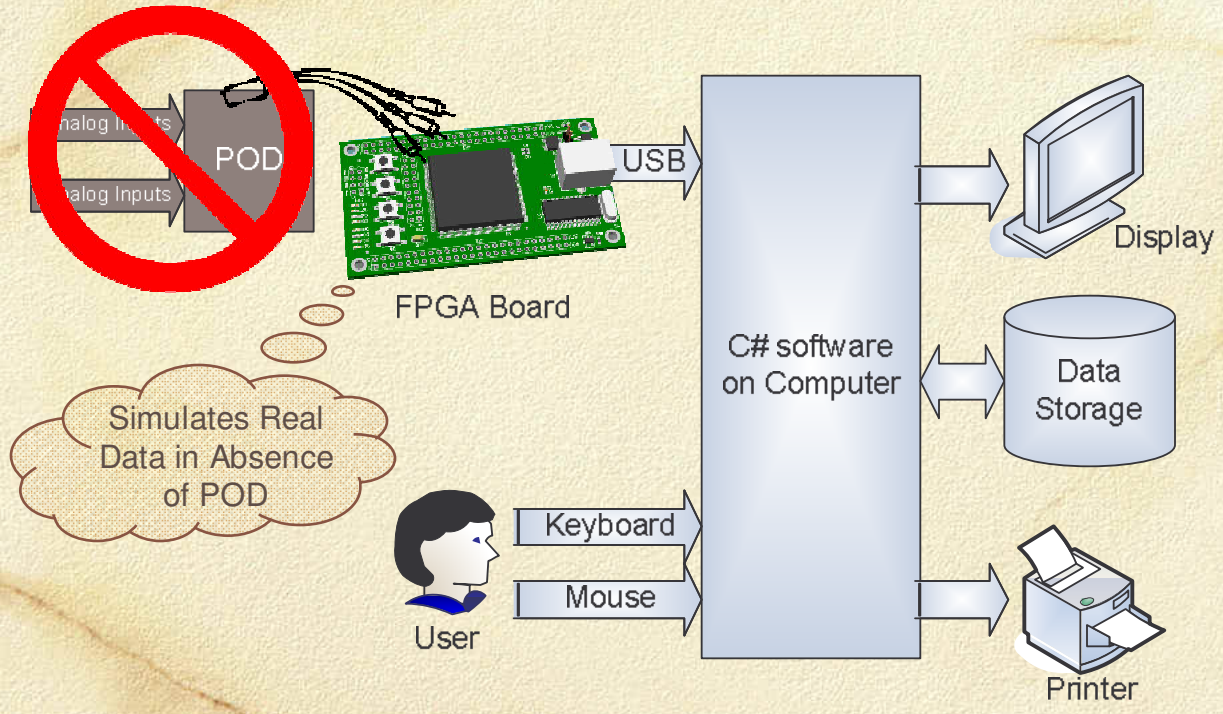
Functional Description



Functional Description



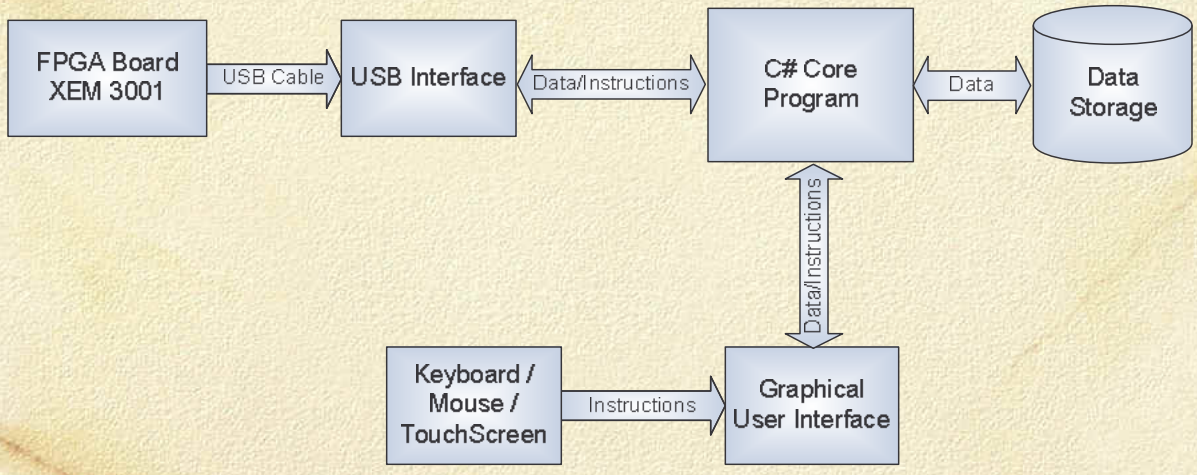
Functional Description



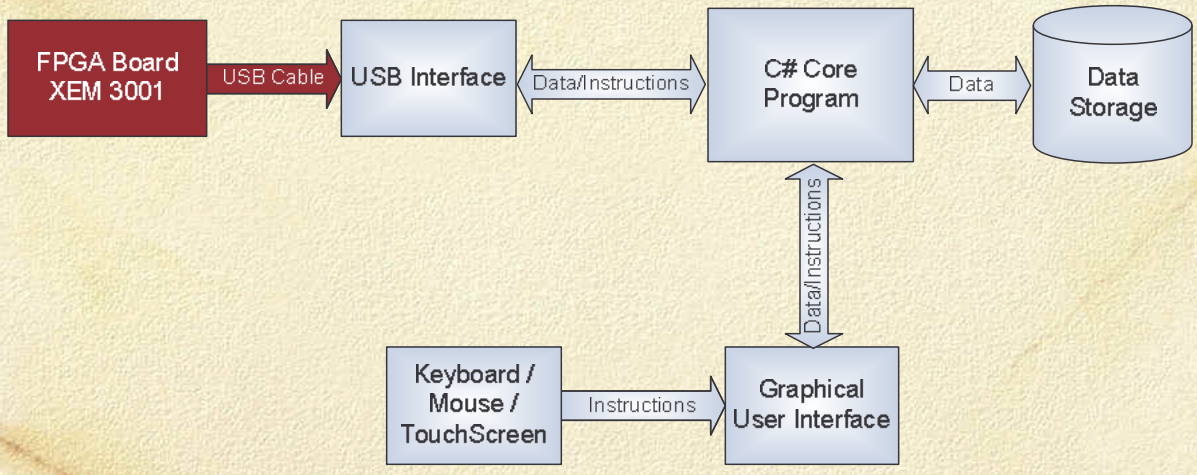
Presentation Outline

- Background Information
- Project Summary
- Project Goals
- Functional Description
- System Block Diagram
- Results
- Questions

System Block Diagram

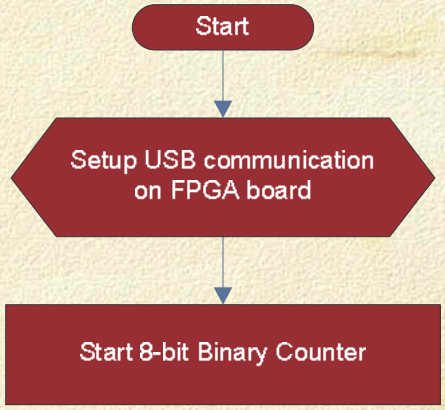


FPGA Board


FPGA Board
XEM 3001
USB Cable
USB Interface
Data/Instructions
C# Core
Program
Data
Data
Storage
Data/Instructions
Keyboard /
Mouse /
TouchScreen
Instructions
Graphical
User Interface

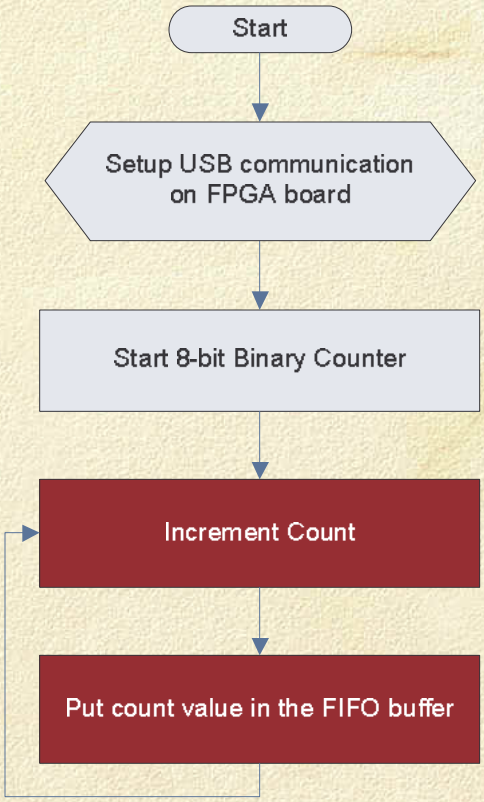
FPGA Software Flowchart

- Get FPGA ready to communicate with PC over USB
- Start Binary Counter

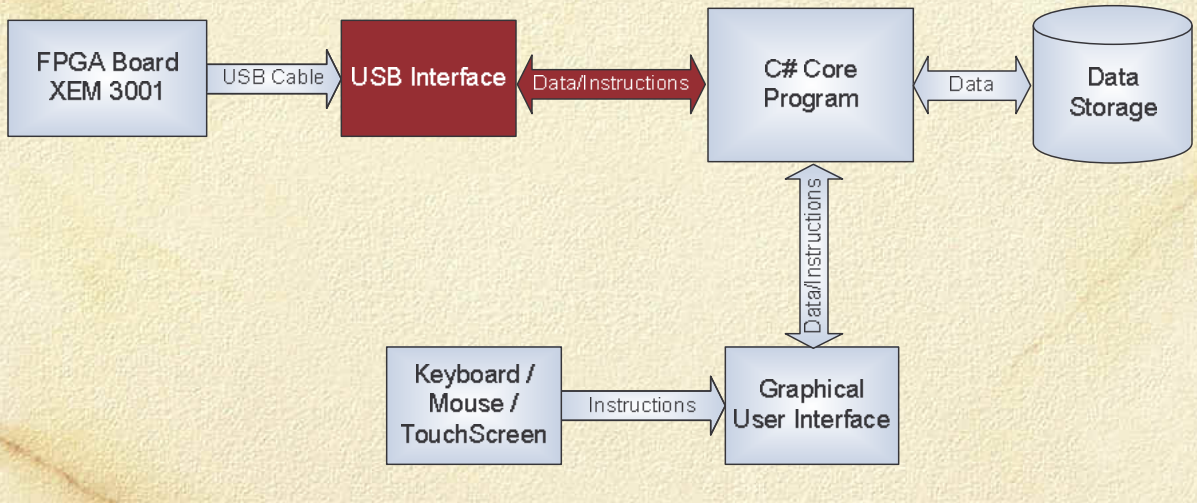


FPGA Software Flowchart

- Increment Binary Counter
- Make count value available on the FIFO buffer to be read by PC

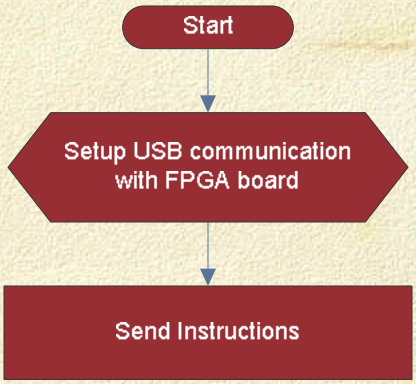


USB Interface



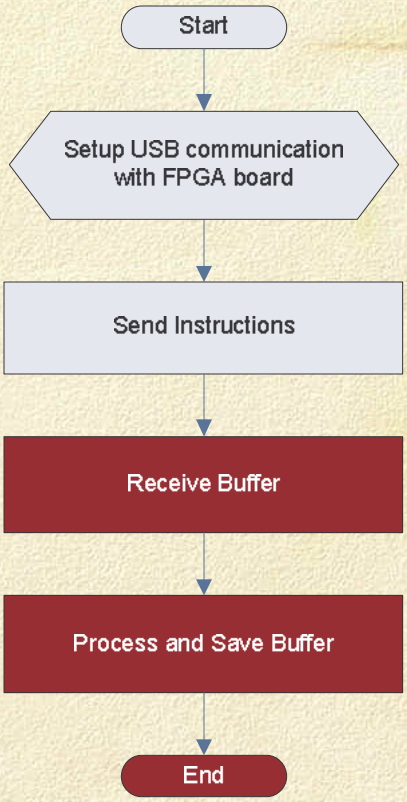
USB Interface Flowchart

- Get the PC ready to communicate with the FPGA board over USB
- Send instructions to the FPGA to read the FIFO buffer

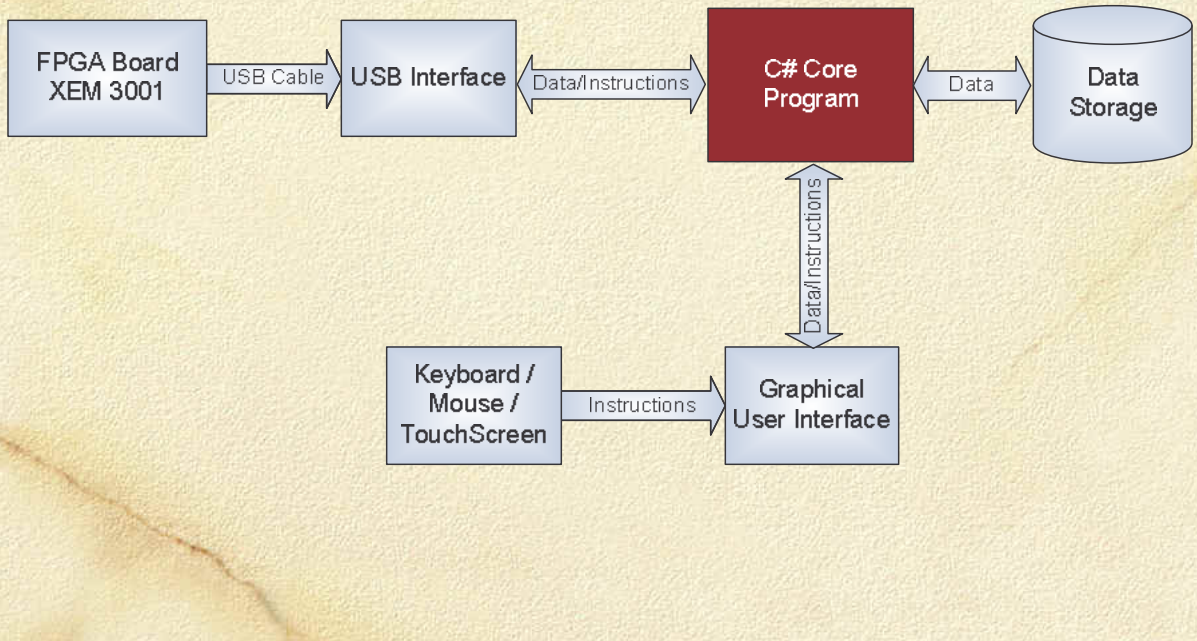


USB Interface Flowchart

- Receive data transmitted by the FPGA from the FIFO buffer
- Process the data that was received and save in software usable format

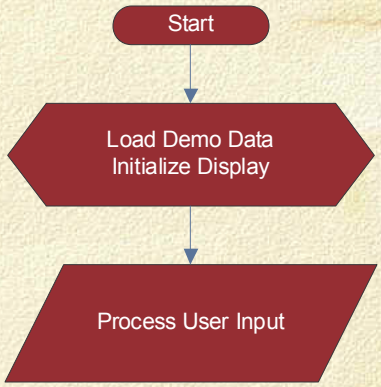


C# Core


FPGA Board
XEM 3001
USB Cable
USB Interface
Data/Instructions
C# Core
Program
Data
Data
Storage
Data/Instructions
Keyboard /
Mouse /
TouchScreen
Instructions
Graphical
User Interface

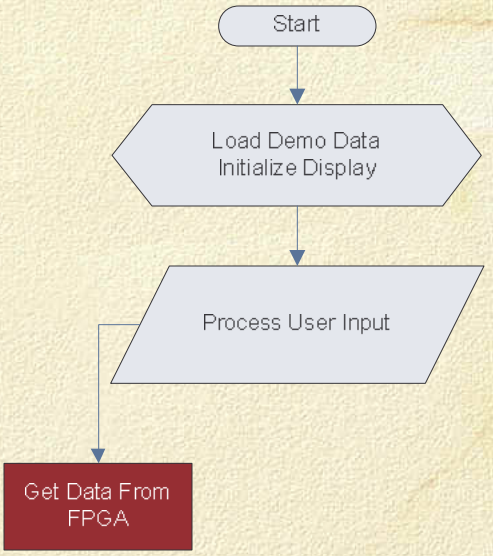
C# Core Flowchart

- This is the first module that loads when the application is started
- All other modules are then initialized
- Demo data is loaded and displayed
- The core waits for user input through the GUI



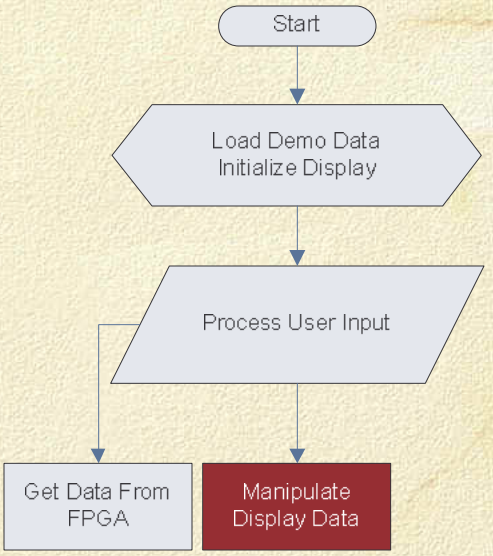
C# Core Flowchart

- Call the USB Interface module and acquire data from the FPGA



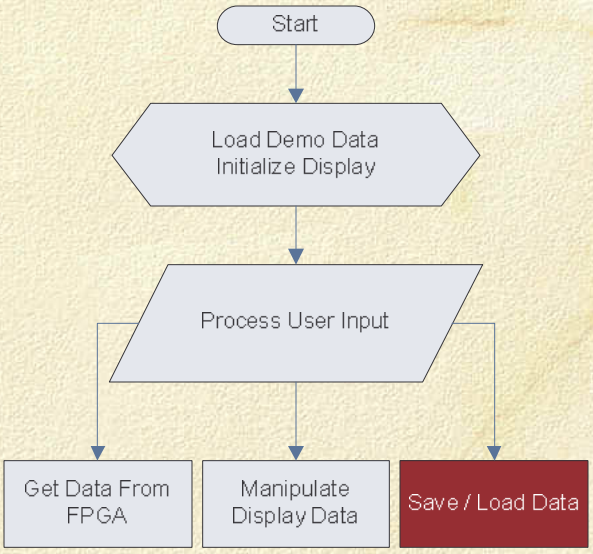
C# Core Flowchart

- Process acquired data and send it to the GUI to be displayed on the monitor

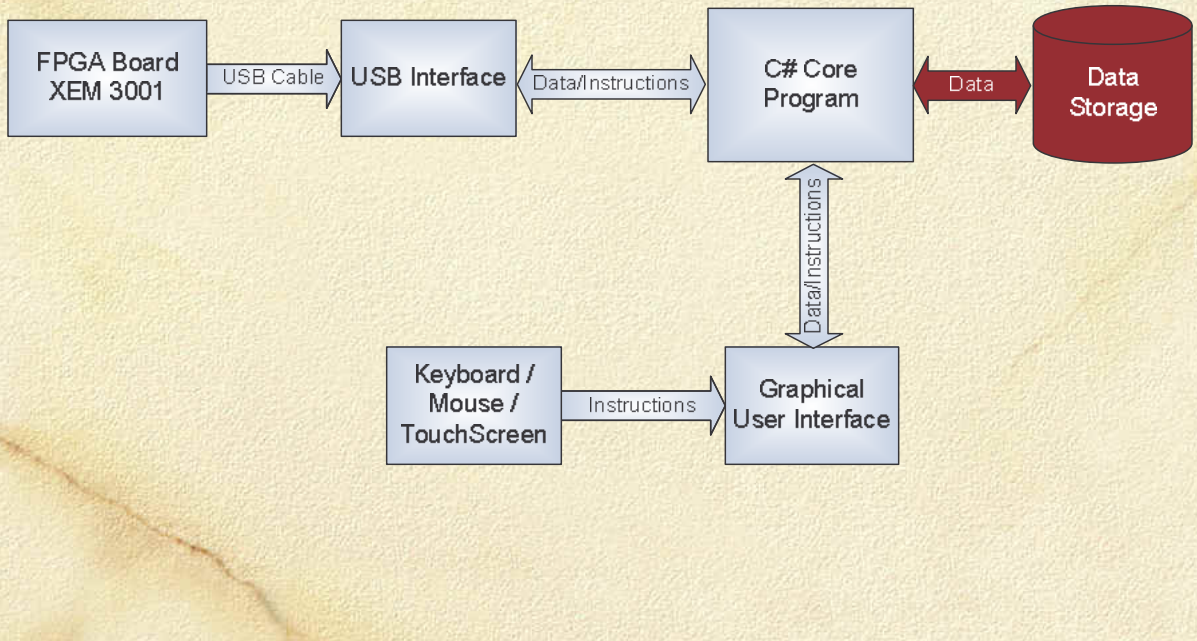


C# Core Flowchart

- Save the data on the disk, either as an image or text data

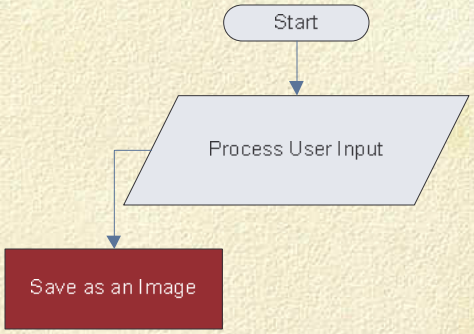


Data Storage



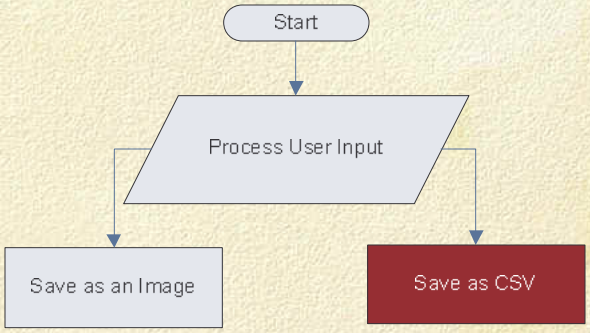
Data Storage Flowchart

- Save acquired and manipulated data for future reference
- As one of four image formats
  - PNG
  - JPEG
  - BMP
  - GIF

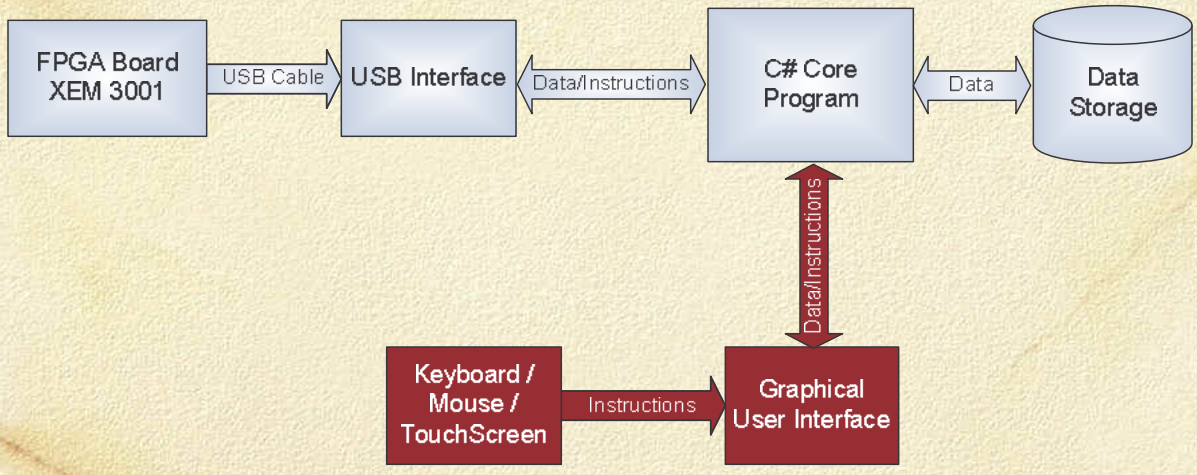


Data Storage Flowchart

- Save as CSV [Comma Separated Values] file
- Can be loaded into the application later on to view and manipulate further

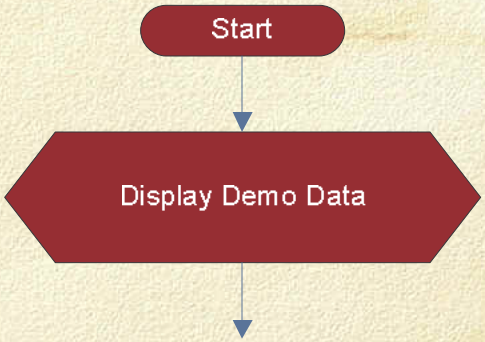


Graphical User Interface (GUI)



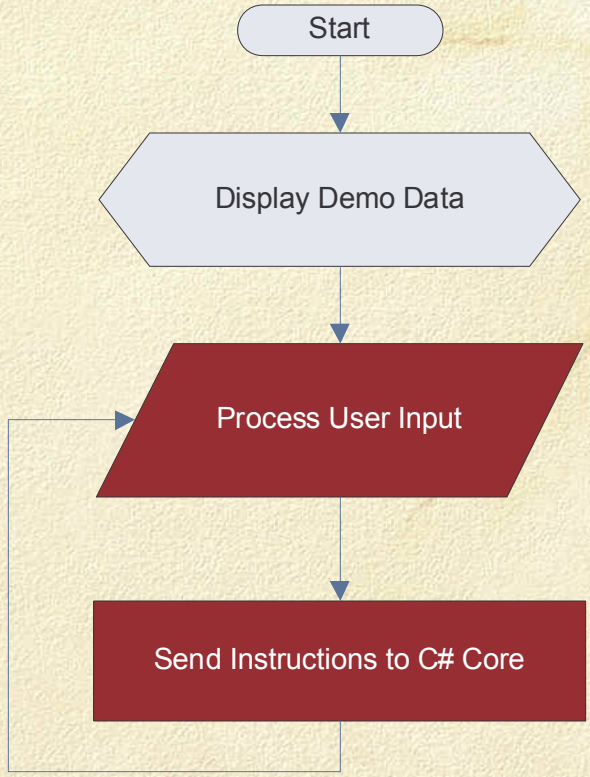
GUI Flowchart

- Start the front end to the application
- Display demo data and all the user controls



GUI Flowchart

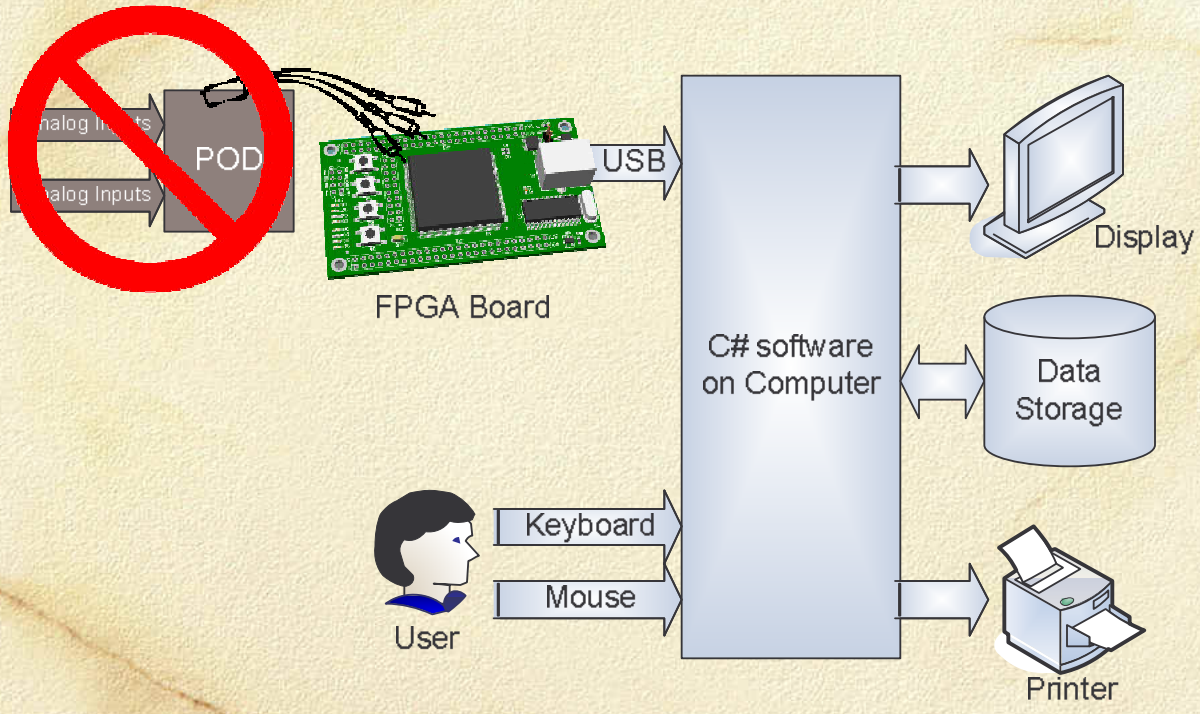
- Get user input from keyboard and mouse
- Send instructions to C# Core



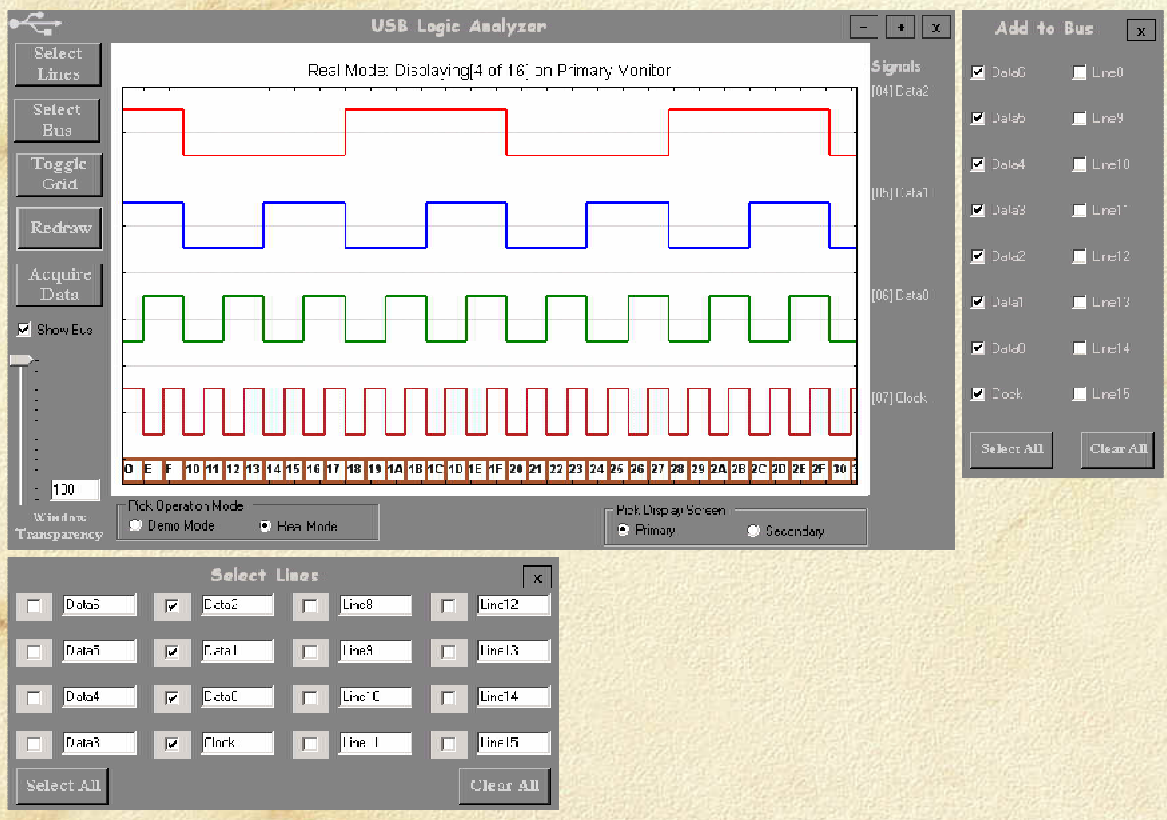
Presentation Outline

- Background Information
- Project Summary
- Project Goals
- Functional Description
- System Block Diagram
- **Results**
- Questions

Results


POD
USB
FPGA Board
C# software
on Computer
Display
Data
Storage
Keyboard
Mouse
User
Printer

Graphical User Interface



GUI Demonstration

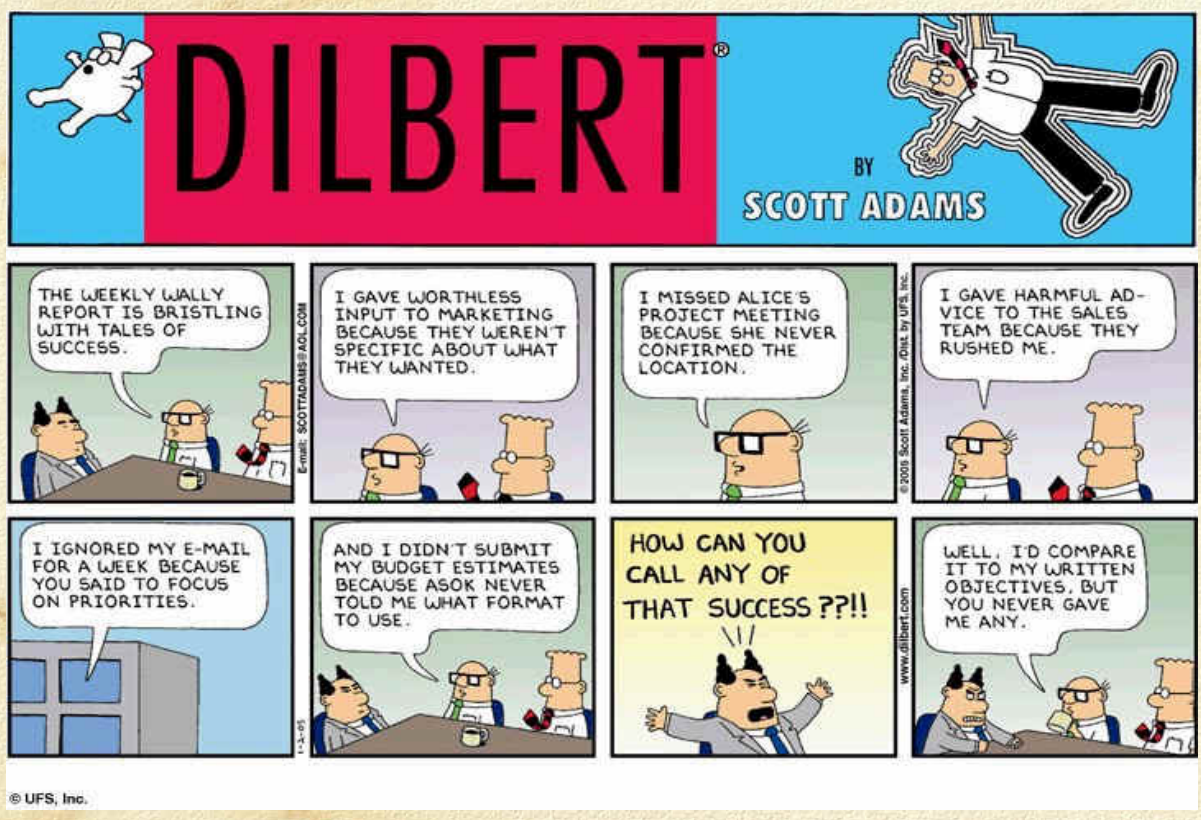
Original Goals Accomplished

- **Software**
  - ☑ Display engine using NPlot graphing library
  - ☑ Plot acquired data as discreet logic signals
  - ☑ Combine data lines into data bus
  - ☑ Efficient data processing
  - ☑ Print/Save data capture

Additional Accomplishments

- **Hardware**
  - ☑ Interface a Spartan3 FPGA board [XEM 3001] over USB
  - ☑ Place data in buffer to transmit over USB
- **Software**
  - ☑ Acquire instantaneous data on user demand
  - ☑ Parse acquired data into discreet logic signals
  - ☑ Load previously saved data acquisition session

Questions?


DILBERT®
BY
SCOTT ADAMS
THE WEEKLY WALLY REPORT IS BRISTLING WITH TALES OF SUCCESS.
I GAVE WORTHLESS INPUT TO MARKETING BECAUSE THEY WEREN'T SPECIFIC ABOUT WHAT THEY WANTED.
I MISSED ALICE'S PROJECT MEETING BECAUSE SHE NEVER CONFIRMED THE LOCATION.
I GAVE HARMFUL ADVICE TO THE SALES TEAM BECAUSE THEY RUSHED ME.
I IGNORED MY E-MAIL FOR A WEEK BECAUSE YOU SAID TO FOCUS ON PRIORITIES.
AND I DIDN'T SUBMIT MY BUDGET ESTIMATES BECAUSE ASOK NEVER TOLD ME WHAT FORMAT TO USE.
HOW CAN YOU CALL ANY OF THAT SUCCESS??!!
WELL, I'D COMPARE IT TO MY WRITTEN OBJECTIVES. BUT YOU NEVER GAVE ME ANY.
E-mail: SCOTTADAMS@AOL.COM
© 2005 Scott Adams, Inc. /Dist. by UFS, Inc.
www.dilbert.com
© UFS, Inc.