The Abdus Salam
International Centre for Theoretical Physics

United Nations
Educational, Scientific and
Cultural Organization

IAEA
International Atomic Energy Agency

2057-13

First Workshop on Open Source and Internet Technology for Scientific Environment: with case studies from Environmental Monitoring

7 - 25 September 2009

Reconfigurable Virtual Instrumentation based on FPGAs

Maria Liz Crespo and Andres Cicuttin

Multidisciplinary Laboratory (ICTP MLAB)
ICTP
Trieste
Italy

Innovative Hardware/Software Platforms

Reconfigurable Virtual Instrumentation based on FPGAs

New opportunities in scientific instrumentation

Maria Liz Crespo – Andres Cicuttin

ICTP - Multidisciplinary Laboratory

I. Brief review of FPGA and trends

II. A case study to illustrate some new related opportunities

*The **R**econfigurable **V**irtual **I**nstrumentation project*

What is an FPGA?

FPGA: **F**ield **P**rogrammable **G**ate **A**rray

Essentially it is

A 2D collection of interconnectable and configurable logic blocks

+

A flexible interface

What is an FPGA today? (I)

Some New Features:

- True dual port RAM (several Mb)
- Clock management units (DLL, PLL)
- Optimized arithmetic blocks (multipliers, DSP)
- Embedded processors (PowerPC)
- Highly specialized In-Out ports (Gbits/sec)

What is an FPGA today? (II)

- More specialized arithmetic units
- Analog blocks (A/D, D/A)
- More optimized for System-on-Chip (SOC) implementation
- Larger number of embedded (hard/soft) processors
- Others: FPAA, FPOA, multi Giga-trans FPGA, hibrid digital analog FPGA (Actel Fusion FPGAs), etc.

What is the main difference between microprocessors and PLDs ?

Examples:

Processors

- MicroControlers
- DSP
- General Purpose

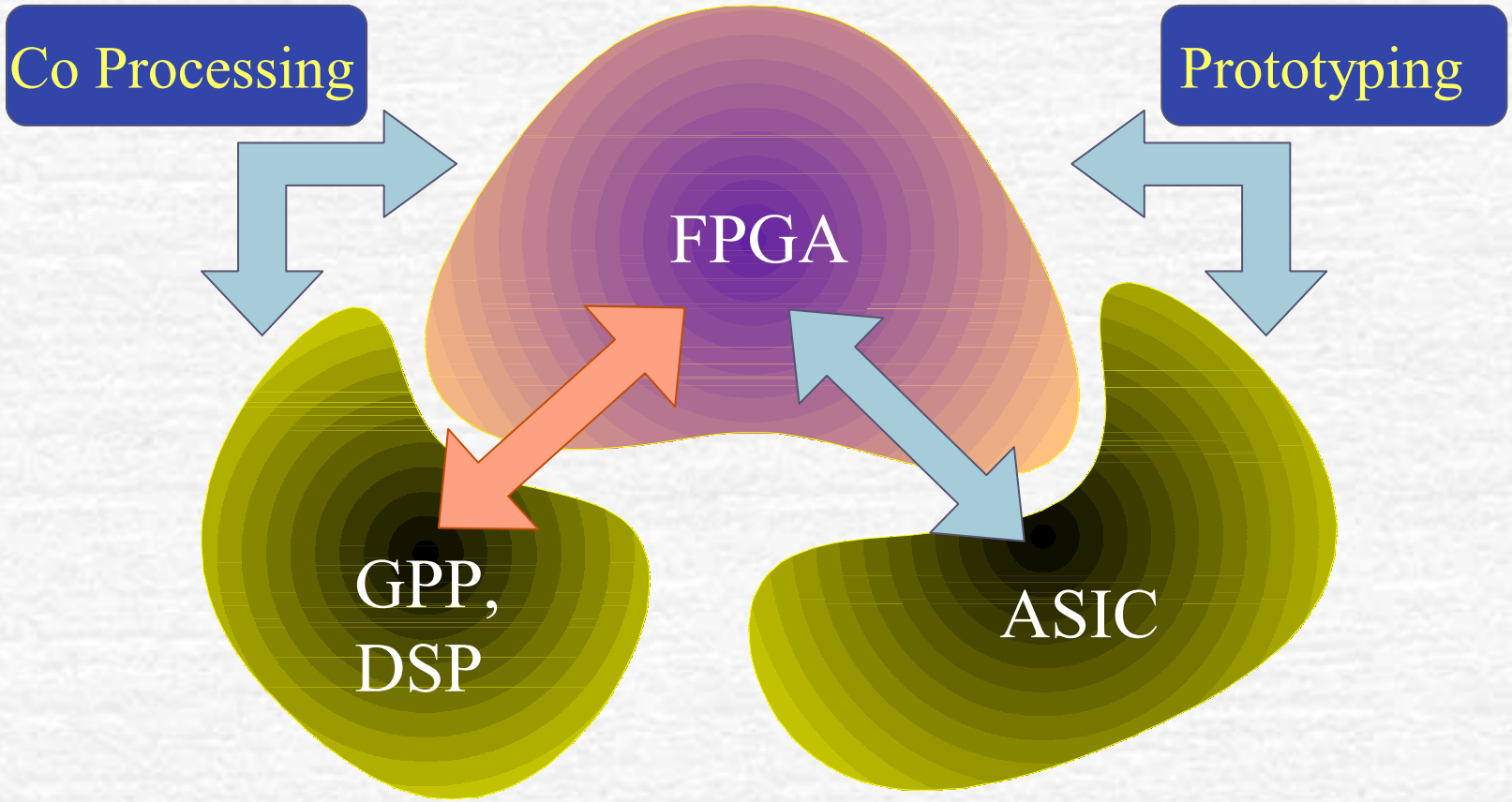
PLD

- Pal, Pdl, etc (simple)
- CPLDs (complex)
- FPGAs (more complex)

Comparison

	Microprocessor	FPGA
Optimized	to execute an instruction set	to implement a wide variety of designs
Configuration (Hardware)	Very little Static	Extremely versatile Partial and dynamic reconfiguration
Programming	High level language Compilers	Depending on design
Operation	Mainly sequential Some parallelism	Essentially parallel
Input Output	Electrically and Logically Fixed	Extremely versatile

FPGA are growing faster than traditional alternatives like ASIC and Microprocessors in terms of performance-cost ratio.

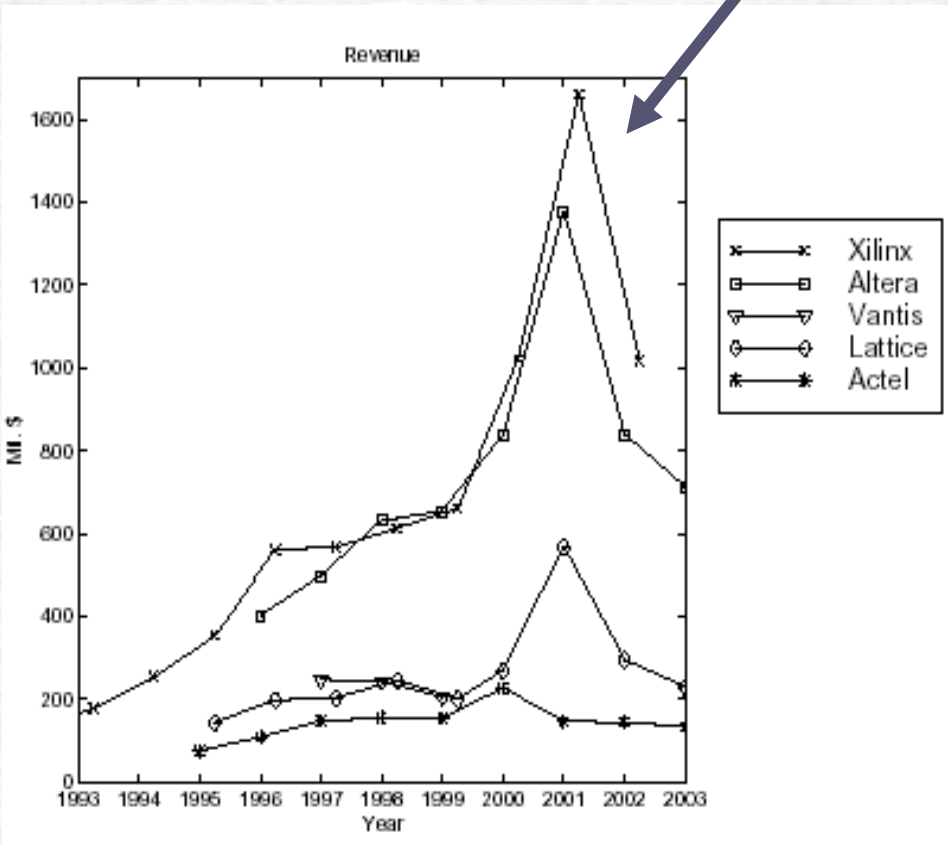


Revenues of the top five PLD vendor (1993-2003)

Several factor have contributed to this

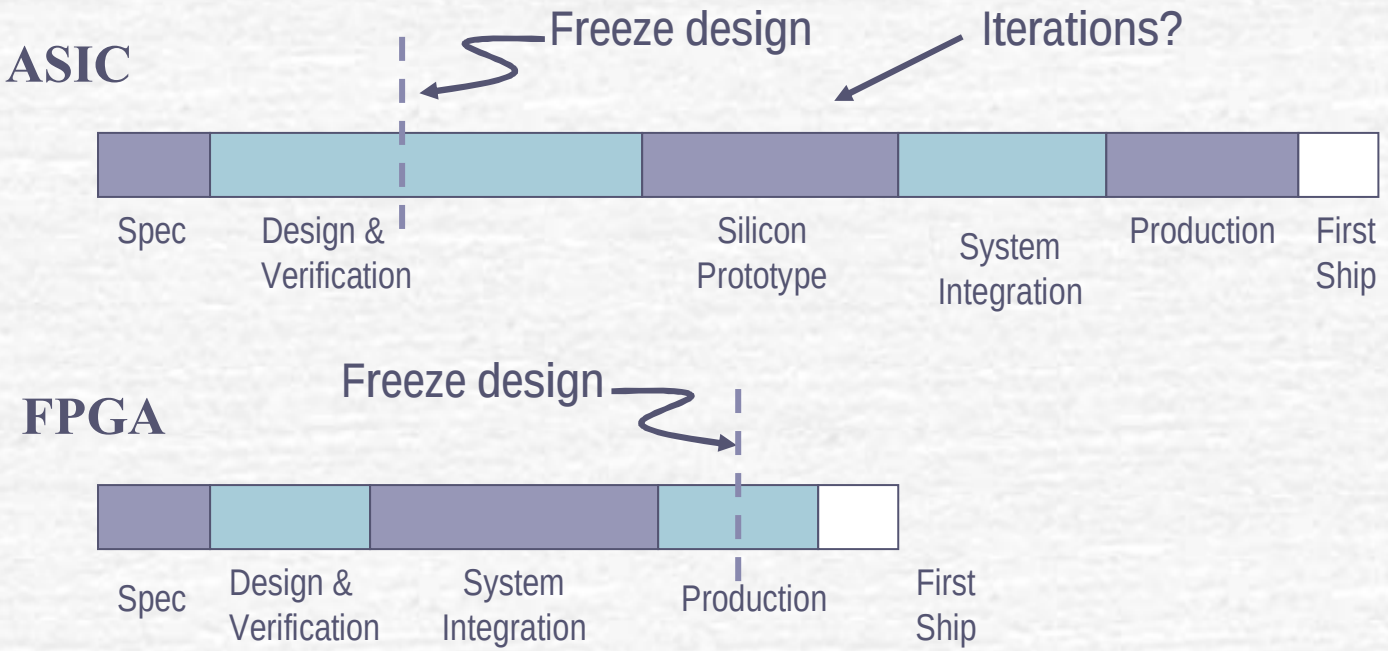
Internet bubble

- Reduction of size and power dissipation
- Higher throughput
- In-circuit reprogramability
- Lower NRE and total costs (FPGA-ASIC crossover growing)
- Almost free CAD tools
- Fast prototyping time



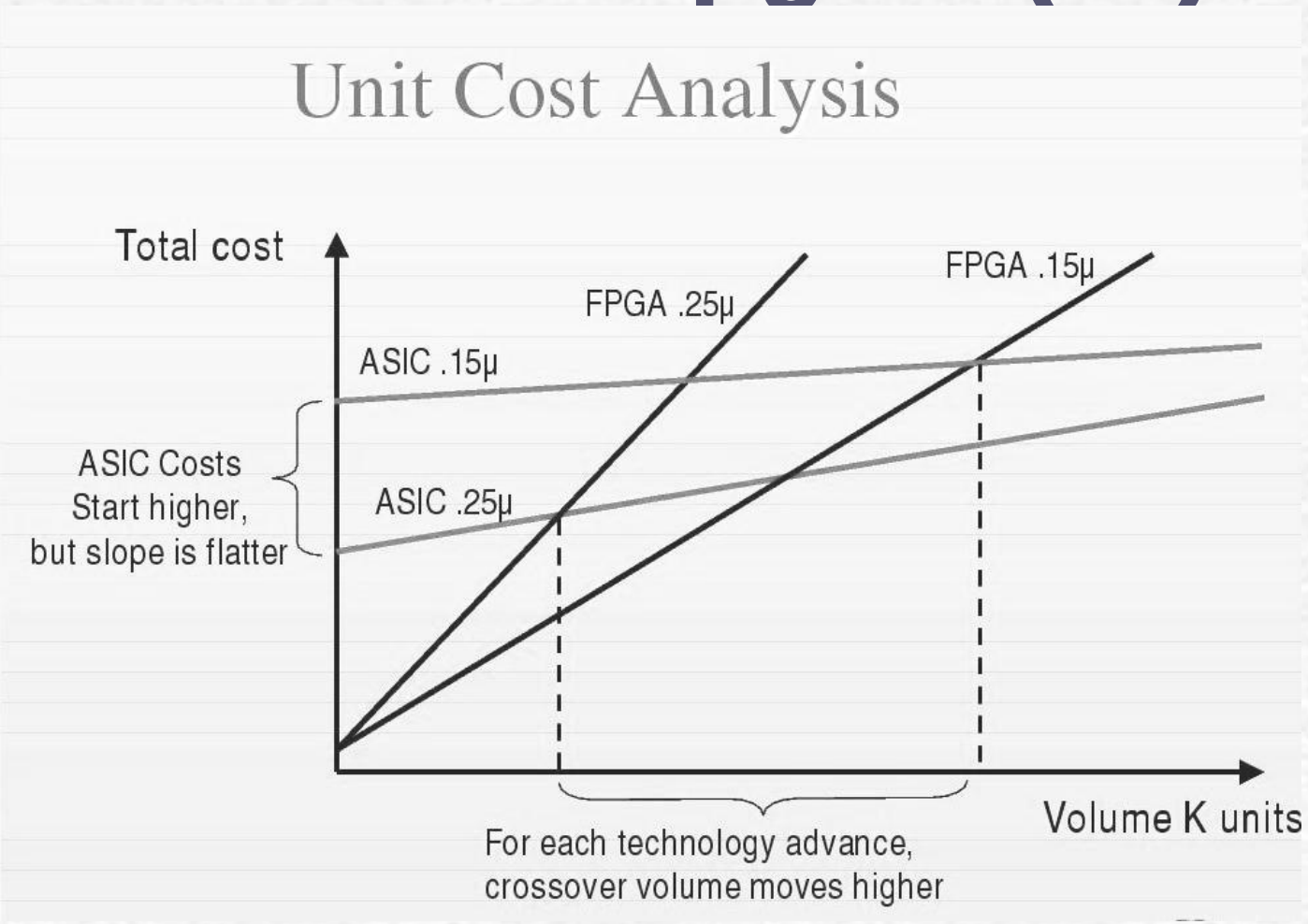
Asics vs. Fpgas (I)

Time to Market: Design Cycle



- ASIC methodology is very unforgiving
- FPGA flexibility allows late design changes

Asics vs. Fpgas (II)

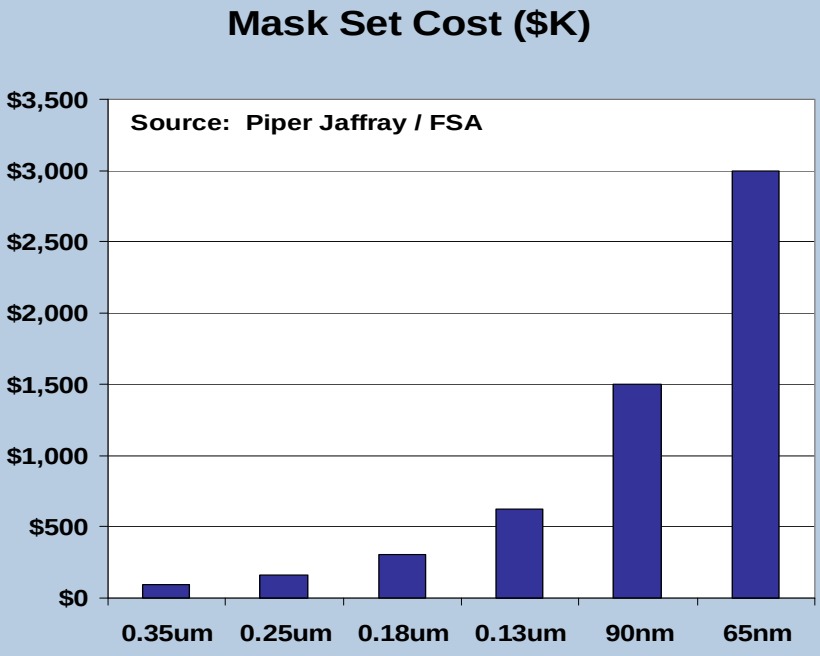


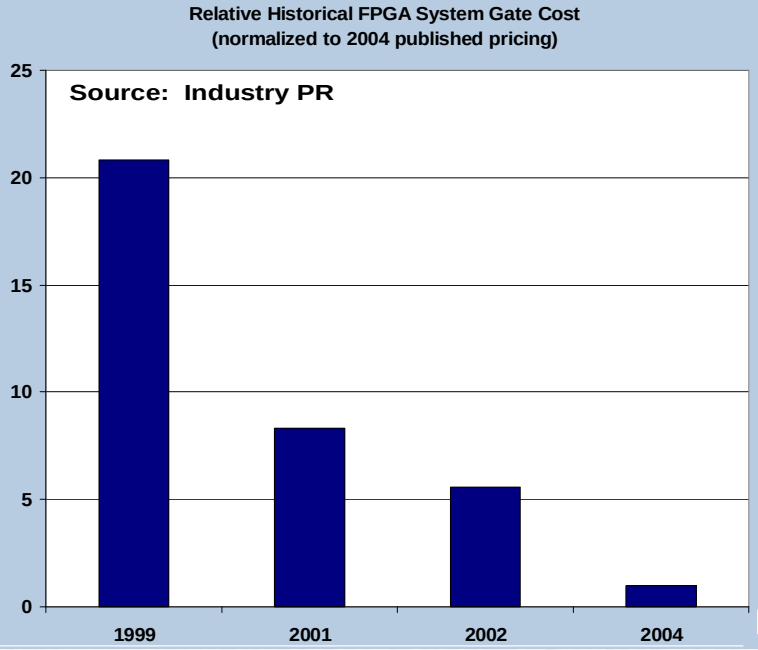
Asics vs. Fpgas (III)

FPGA vs ASIC Cost

90nm ASIC: ~$3 of amortized cost to ASP of a 500k unit socket

Process + Architecture + Volume = greatly reduced prices





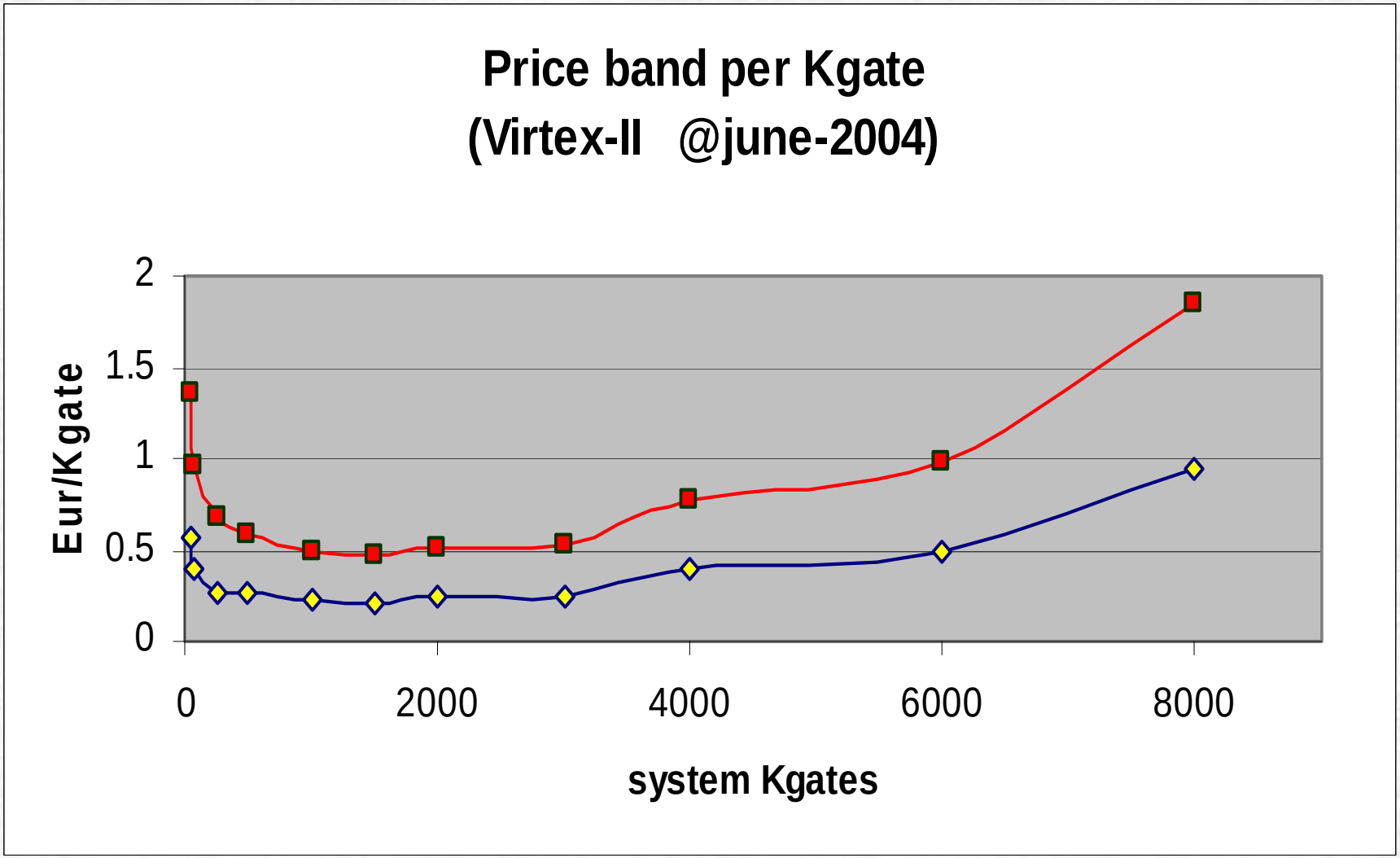
Some numbers

(Xilinx Virtex-II)

Device:	XC2V1000	XC2V8000
• User In-Outs:	432	1200
• Number of slices*:	5000	46000
• 18-bits Multipliers:	40	168
• CLB Array:	40 x 32	112 x 104
• System Memory:	0.7 Mbit	3 Mbits
• Approx. Price**:	~0.4 K$	~10 K$

* Elemental reconfigurable logic unit (LUT + FF)

** Lowest prices start from less than 1USD for some FPGAs


Price band per Kgate
(Virtex-II @june-2004)
Eur/Kgate
2
1.5
1
0.5
0
0
2000
4000
6000
8000
system Kgates

Let us have a closer look at an FPGA

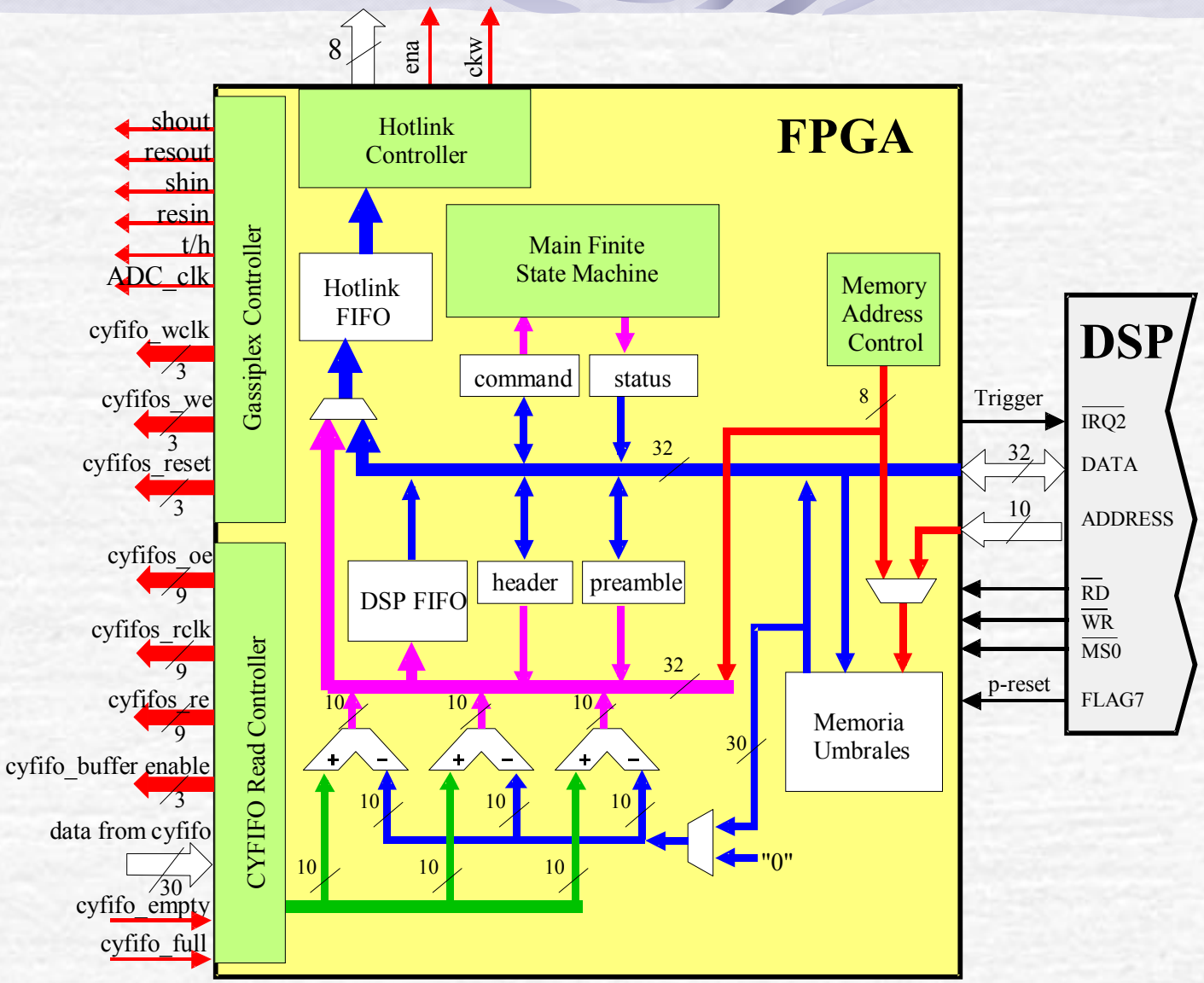
Example:

COMPASS BORA FPGA Design on a Virtex 100

* Among the largest FPGAs in 2000

* Among the smallest FPGAs in 2008

Global Architecture of BORA-FPGA



More Numbers …

(Virtex-II Pro)

Feature/Product	XC 2VP40	XC 2VP50	XC 2VP70	XC 2VPX70	XC 2VP100
Logic Cells	46,632	53,136	74,448	74,448	99,216
BRAM (Kbits)	3,456	4,176	5,904	5,544	7,992
18x18 Multipliers	192	232	328	308	444
Digital Clock Management Blocks	8	8	8	8	12
Config (Mbits)	15.56	19.02	26.1	26.1	33.65
PowerPC Processors	2	2	2	2	2
3.125 Gbps RocketIO Transceivers	12	16	20	0	20
10.3125 Gbps RocketIO X Transceivers	0	0	0	20	0
Max Available User I/O	804	852	996	992	1164

How to deal with such a complexity ?
How to get the most out of it ?

- ✓ Rigorous Top-Down Design Methodology using HDL
- ✓ CAD tools for simulation and synthesis
- ✓ "***Design for reusability"*** strategy
  - Standards, guidelines, interfaces, protocols, etc
- ✓ ***Intellectual Property*** *blocks* (*IP* ***cores***)

What can be achieved ?

- Co processing to speed up DSP and GPP computations
- Real Time systems for high performance data acquisition and process control
- Reconfigurable computing (Quantum Monte Carlo simulations, Pattern recognition, high speed data compression-decompression)
- In general: Radar/Sonar, Telecom, Medical Instr., Sci. Instr., Robotics, Ad hoc high performance instrumentation (*CASIS, new SDD readout*)
- **Reconfigurable Instrumentation** (ICTP RVI System)

Which instruments could be implemented?

Traditional

(General Purpose)

- Arbitrary Waveform Generators
- Multimeters
- Spectrum Analyzers
- Digital Oscilloscopes
- Logic Analyzers
- etc

Which instruments could be implemented?

Non Traditional
(Application specific)

- Image processing
- Data acquisition
- Custom digital filters
- Radar/Sonar
- Reconfigurable computing
- etc

Which instruments could be implemented?

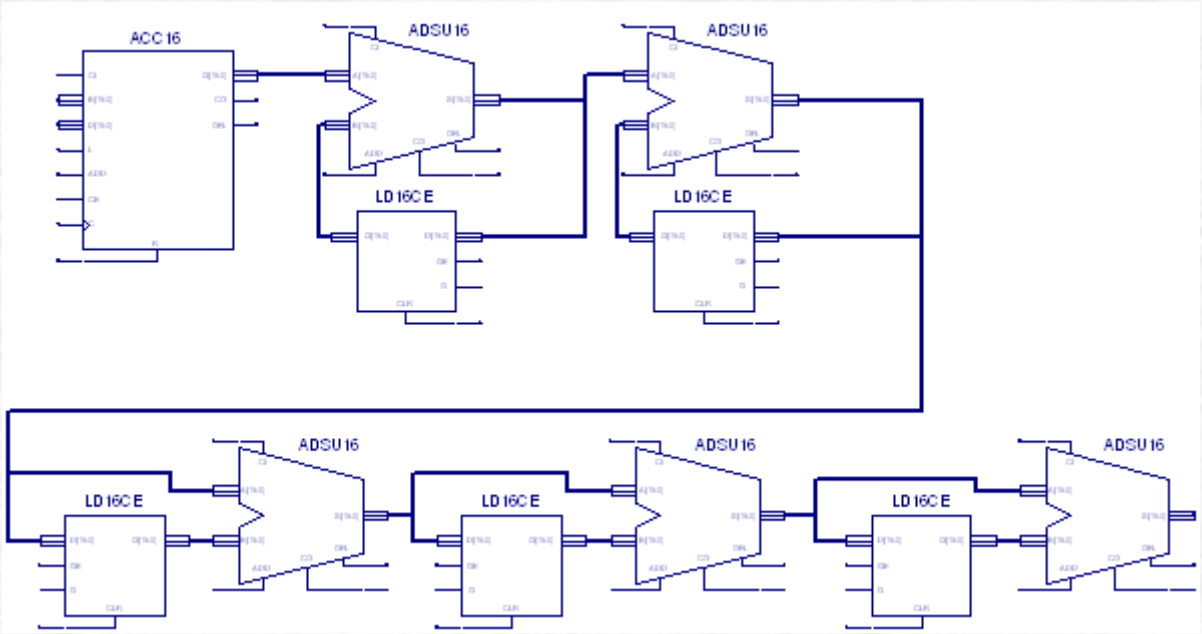
Scientific

- High performance data acquisition and processing
- Real time image processing
- Pattern recognition, neural networks
- Hardware algorithm implementation
- Non Linear Filters, and on line data filtering (HEP)
- High performance specific computing
- Vibrating Sample Magnetometer, Susceptometer, Raman spectroscopy, etc

Example

One Channel CASIS Demodulator: Data Path Architecture

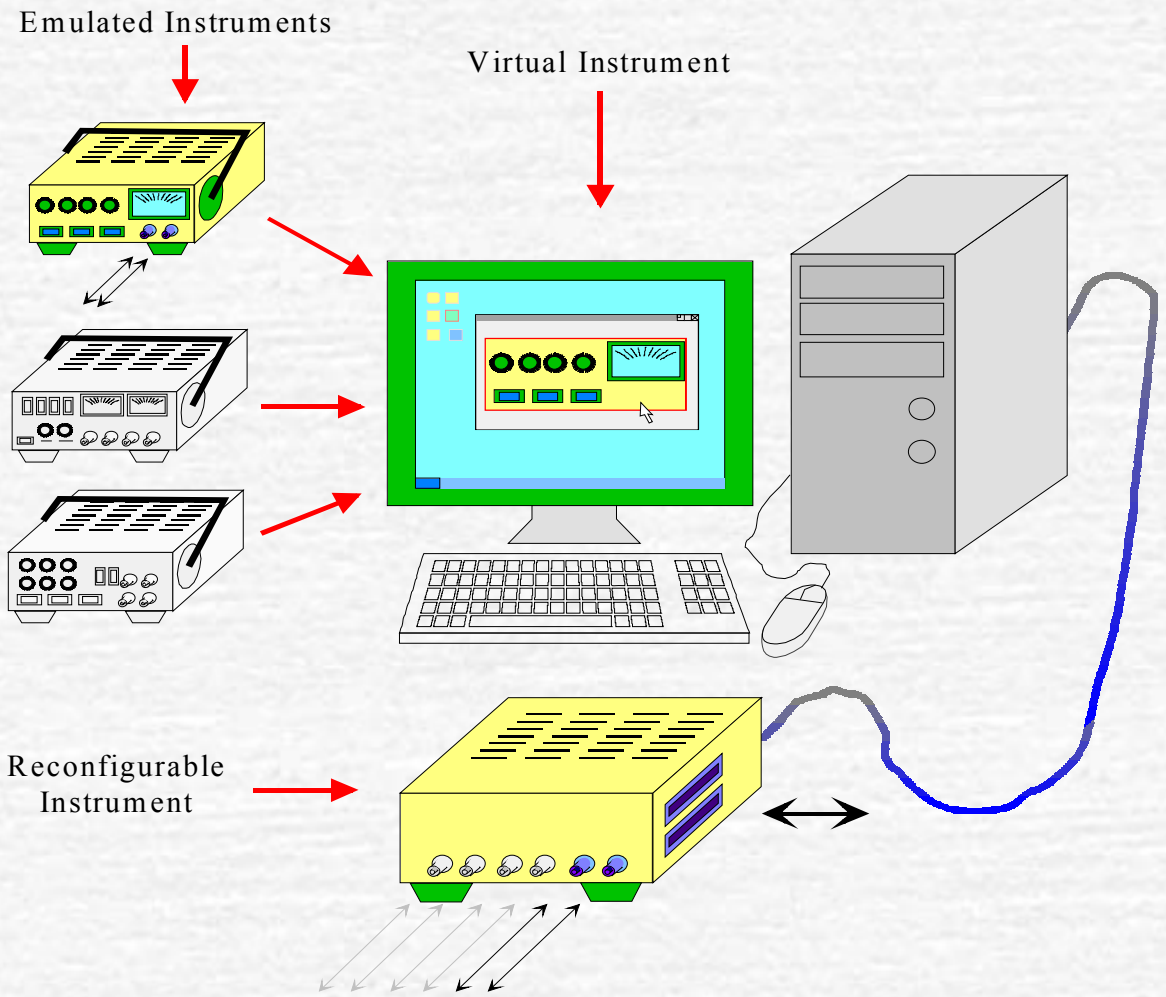
It will be implemented through an RVI system



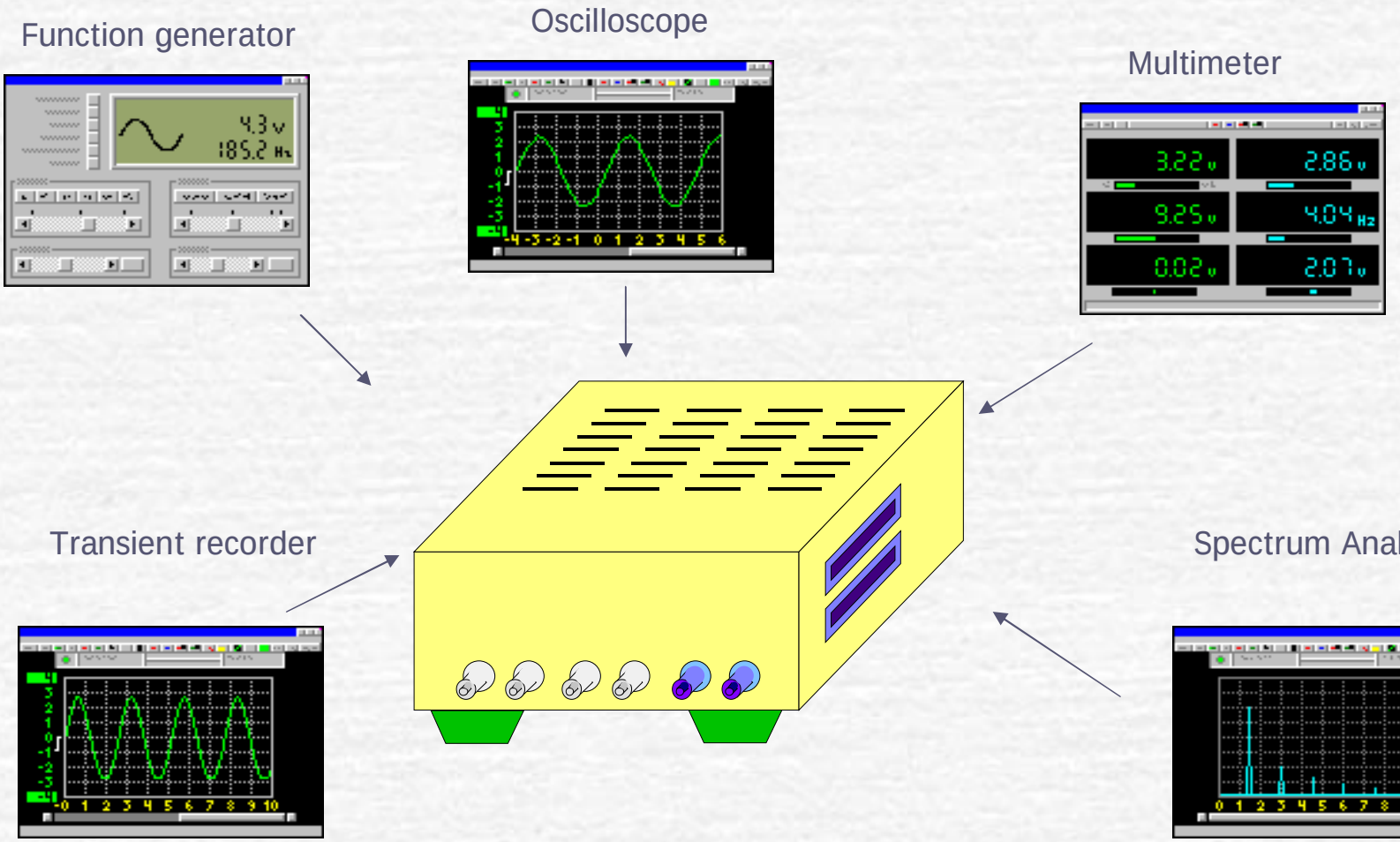
6 fix point op./ch x 16 ch x 10 Mbits/sec.ch= 960 MOPS

Note: all partial results depend on previous partial result. This prevents full activity of microprocessor pipelines diminishing its effective processing throughput.

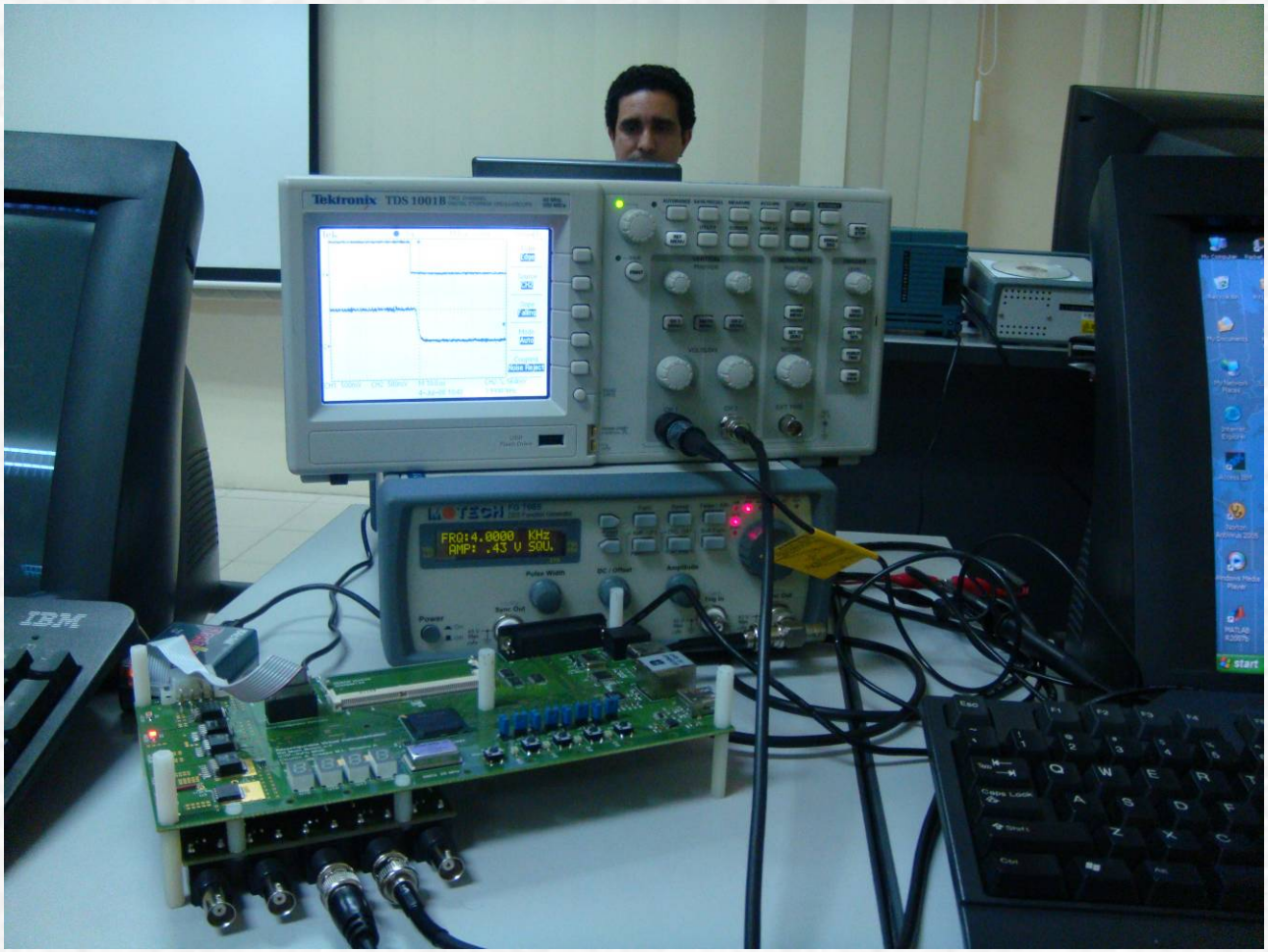
Reconfigurable Virtual Instrumentation



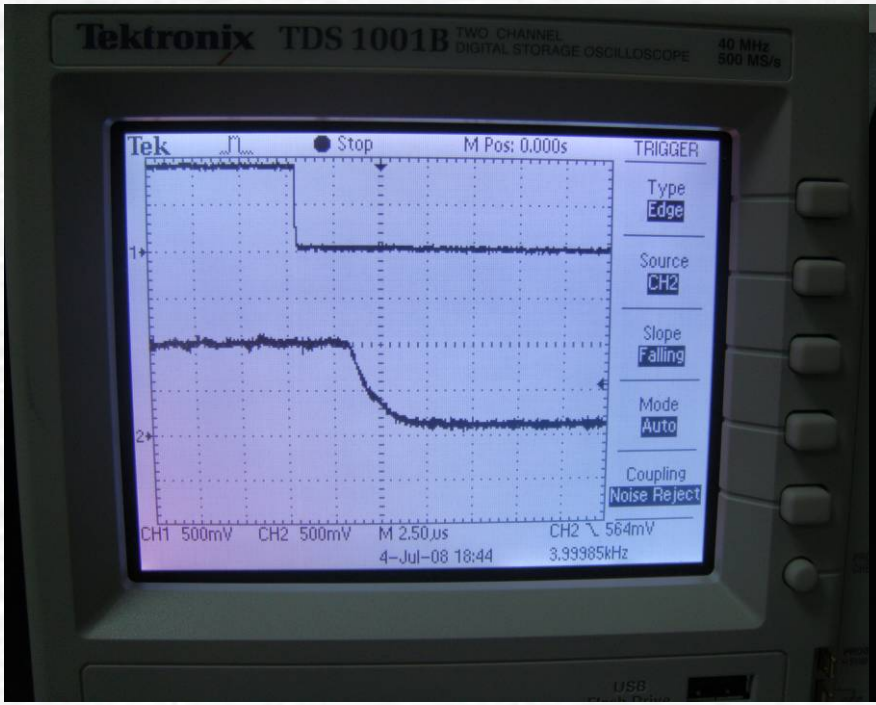
Reconfigurable Virtual Instrumentation

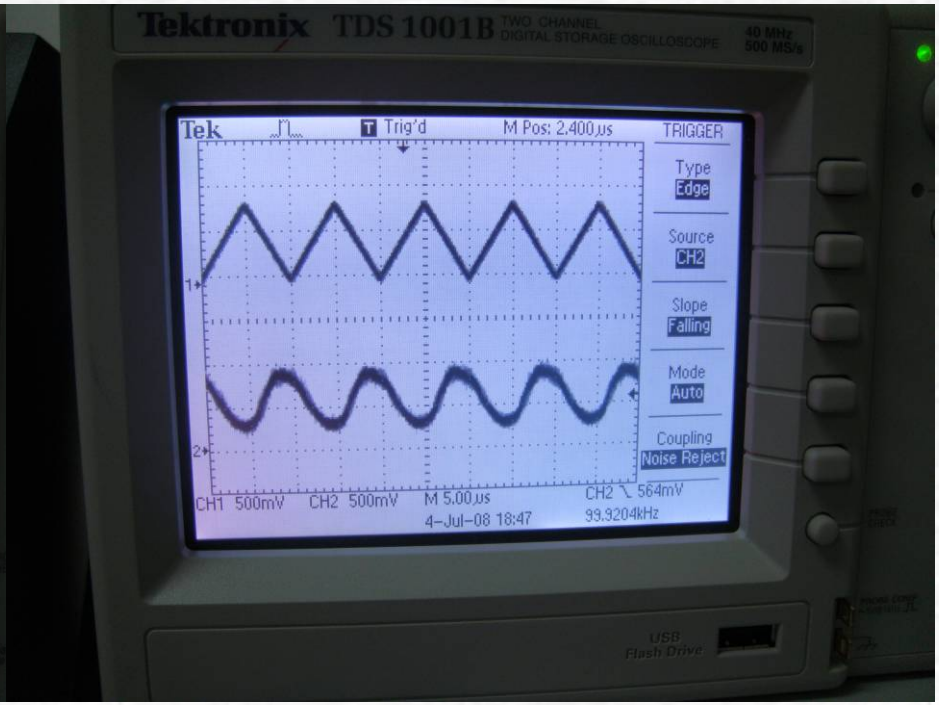


A low pass filter

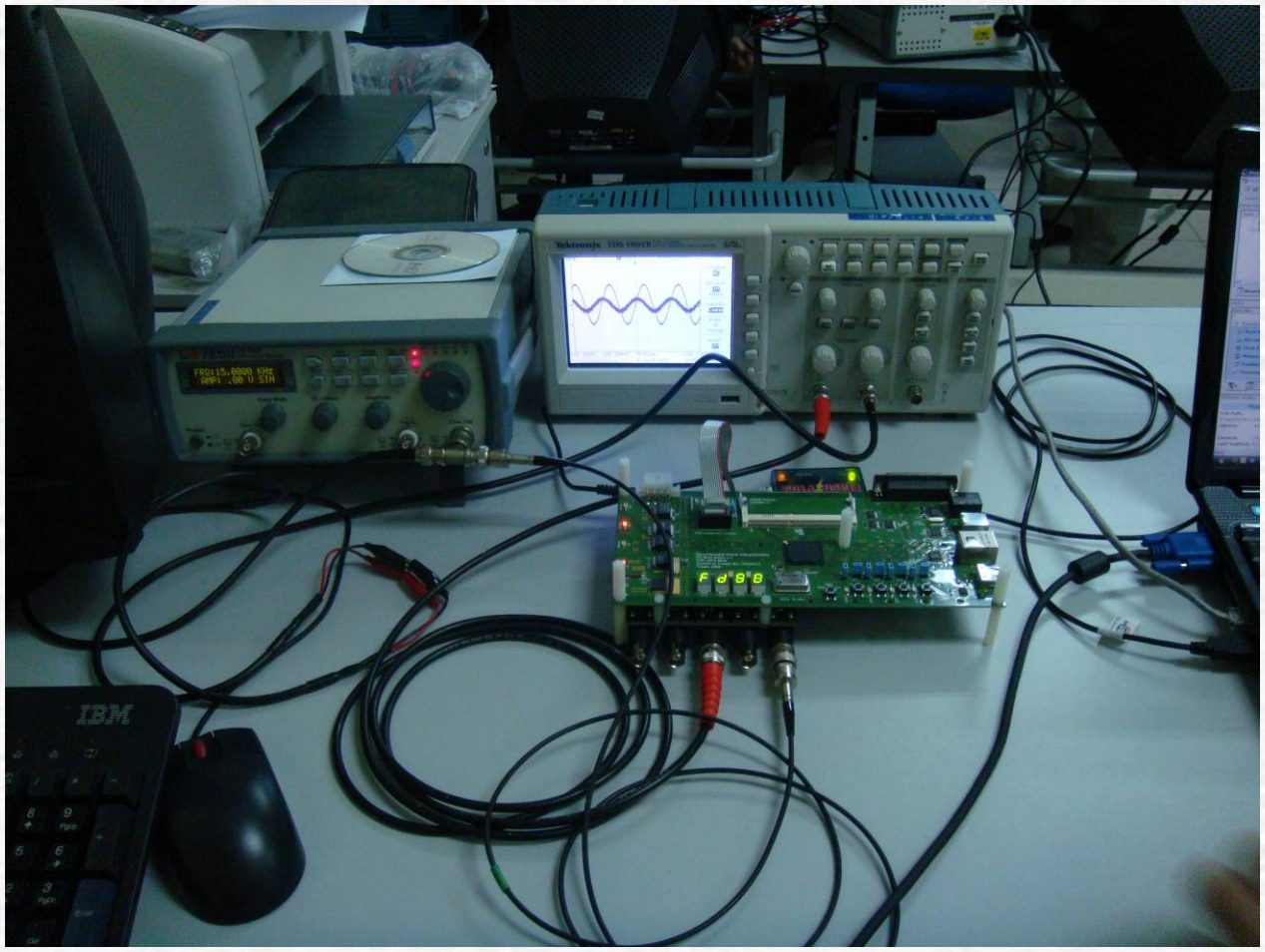


Low pass filter

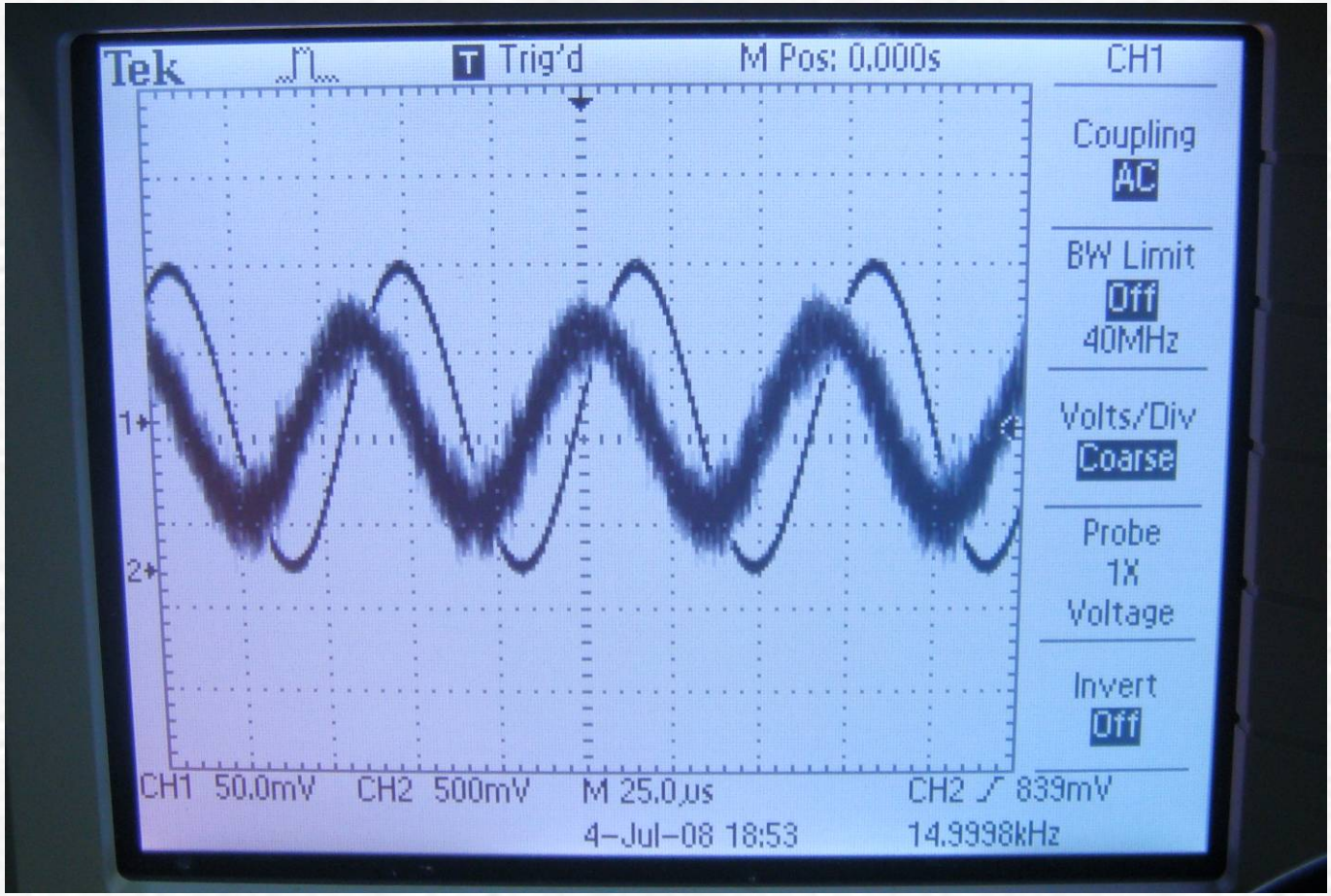




High Pass filter (differentiator)



High Pass filter (differentiator)



Key aspects for an RVI system

Hardware & Software modularity

- Block-based design methodology
- Hierarchical structure

Common standardized global architecture

- Block interfaces definition
- Clear mechanism of blocks interaction

Open Source & Open Cores

- Sharing the design effort and results by a large community of users and contributors with different expertise's and backgrounds (EE, Physicist, Comp. Sci. , DSP experts, etc)

RVI SYSTEM

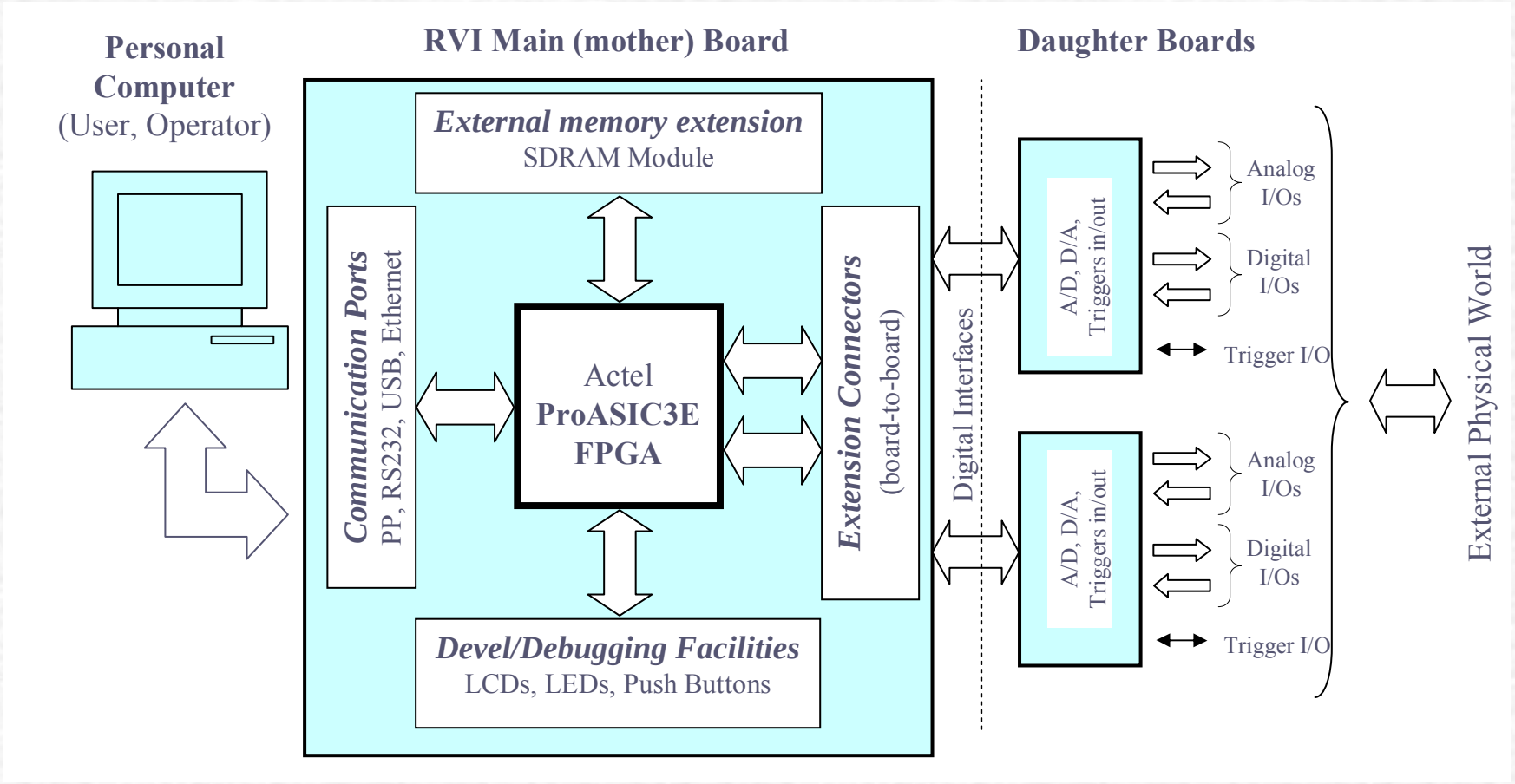
- ***Reconfigurable Instrument*** (the magic box)
  - a versatile hardware device that can be reconfigured into different electronic instruments using a software tool

- ***Virtual Instrumentation***
  - a hardware and software combination that allows the emulation of an instrument through a custom virtual console and a graphical user interface

High-Level RVI System Architecture

- Hardware sub-system
  - RI connected to a PC through a physical connection.
- Software sub-systems:
  - software related to the PC
  - the code corresponding to the FPGA of the RI

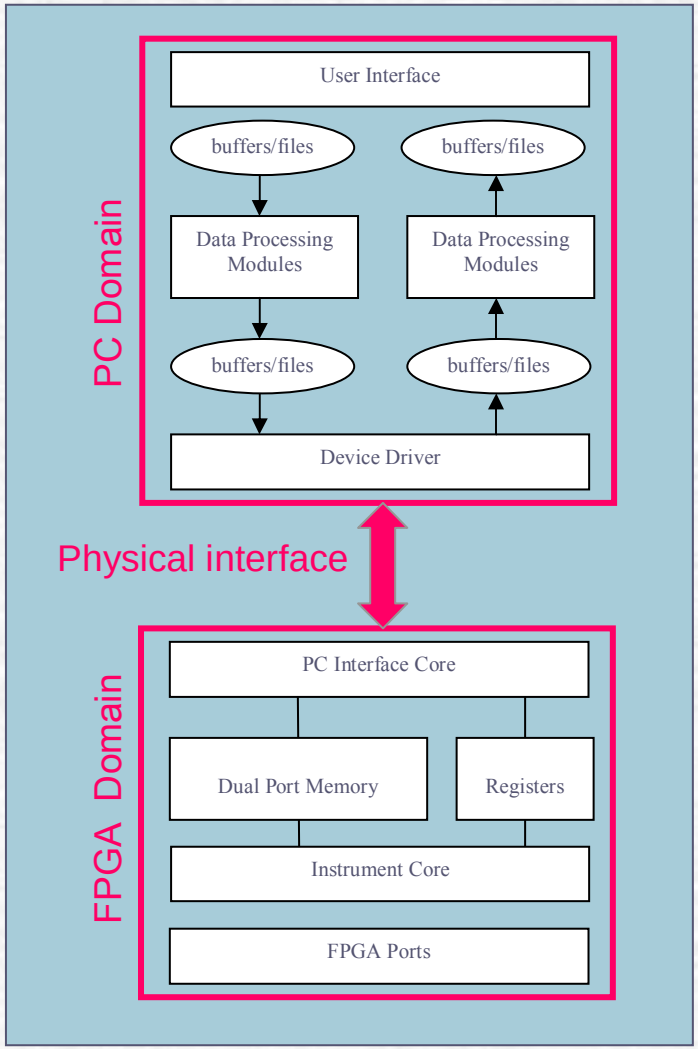
RVI Hardware Sub-System



Reconfigurable Instrument

- **RVI mother board**
  - FPGA device (ACTEL AP3E family)
  - a block of communication ports
  - an extension memory
  - debugging facilities and miscellaneous components
  - two high quality board-to-board connectors with 54 pins directly connected to the FPGA gp-I/O
- **Low Performance Daughter Board**
  - dual channel 10-bits 20 MSPS ADC (AD9201, Analog Devices),
  - dual channel 14-bit 1 MSPS DAC (LTC1654, Linear)
- **High Performance Daughter Board**
  - single channel 14-bits 125 MSPS ADC (LTC2255, Linear)
  - single channel 16-bit 50 MSPS DAC (LTC1668, Linear)

The Global Software Architecture



- ***Computer Software***
  user interface, port management, and offline data elaboration programs and utilities

- ***Synthesizable Hardware Description Code***
  management of the physical connection with the PC, ADC and DAC operations, data generation and acquisition, real-time online data processing, and on-board real time data handling

The Computer Software

- collection of independent modules hierarchically organized
- basically, the CS provides:
  - a generic RVI graphical and textual user interface
  - a library of virtual instruments with custom user interfaces
  - data storage facilities
  - physical communication control (drivers)
- optionally, the CS could also provide:
  - an internet connection for remote instrument control and operation
  - specific data analysis packages and other facilities
  - a friendly interface with a general purpose in-chip logic analyzer for development and debugging.

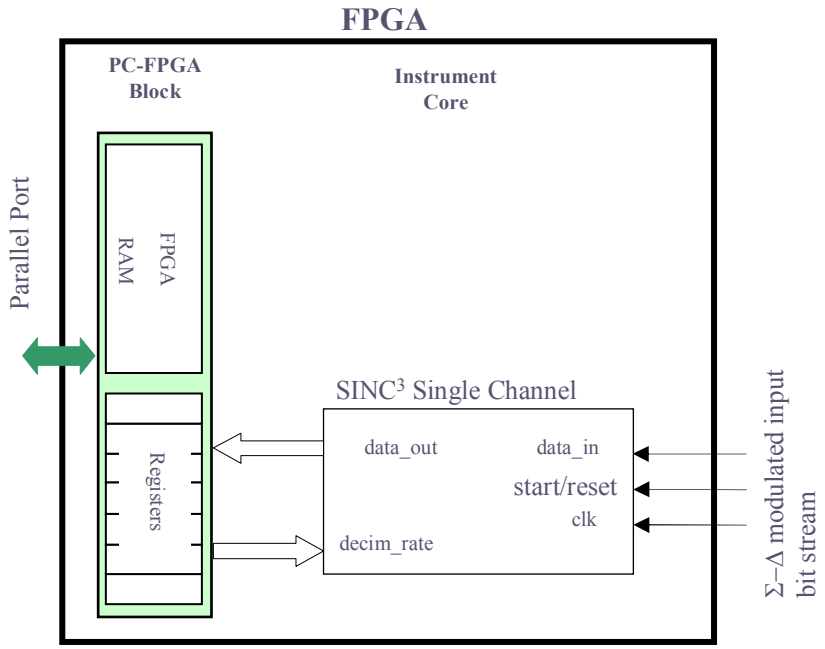
Synthesizable Hardware Description Code (SHDC)

- basically, the SHDC provides:
  - PC-FPGA communication block
  - the instrument core
  - external hardware specific interface block.

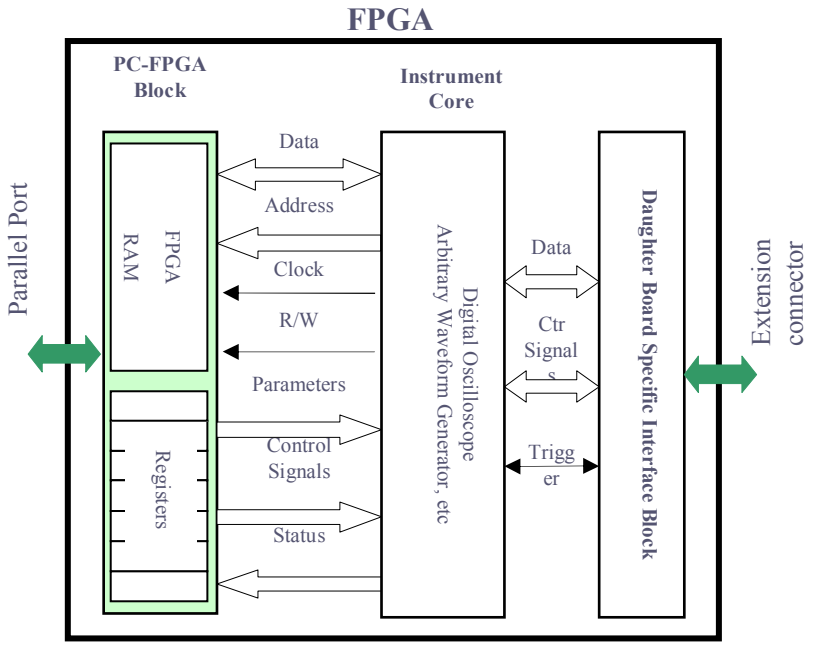
Integrating a reconfigurable instrument core in an RVI system

- The core must comply with
  - the standardized interfaces of the PC-FPGA communication block and the external hardware specific interface block
  - a common mechanism of interaction
- If the three main blocks: PC-FPGA communication block, instrument core, and the external hardware interface respect both previous conditions, then each block can be updated or upgraded independently and can be reused in different contexts.

Architecture for Single Instruments
Implementation Examples

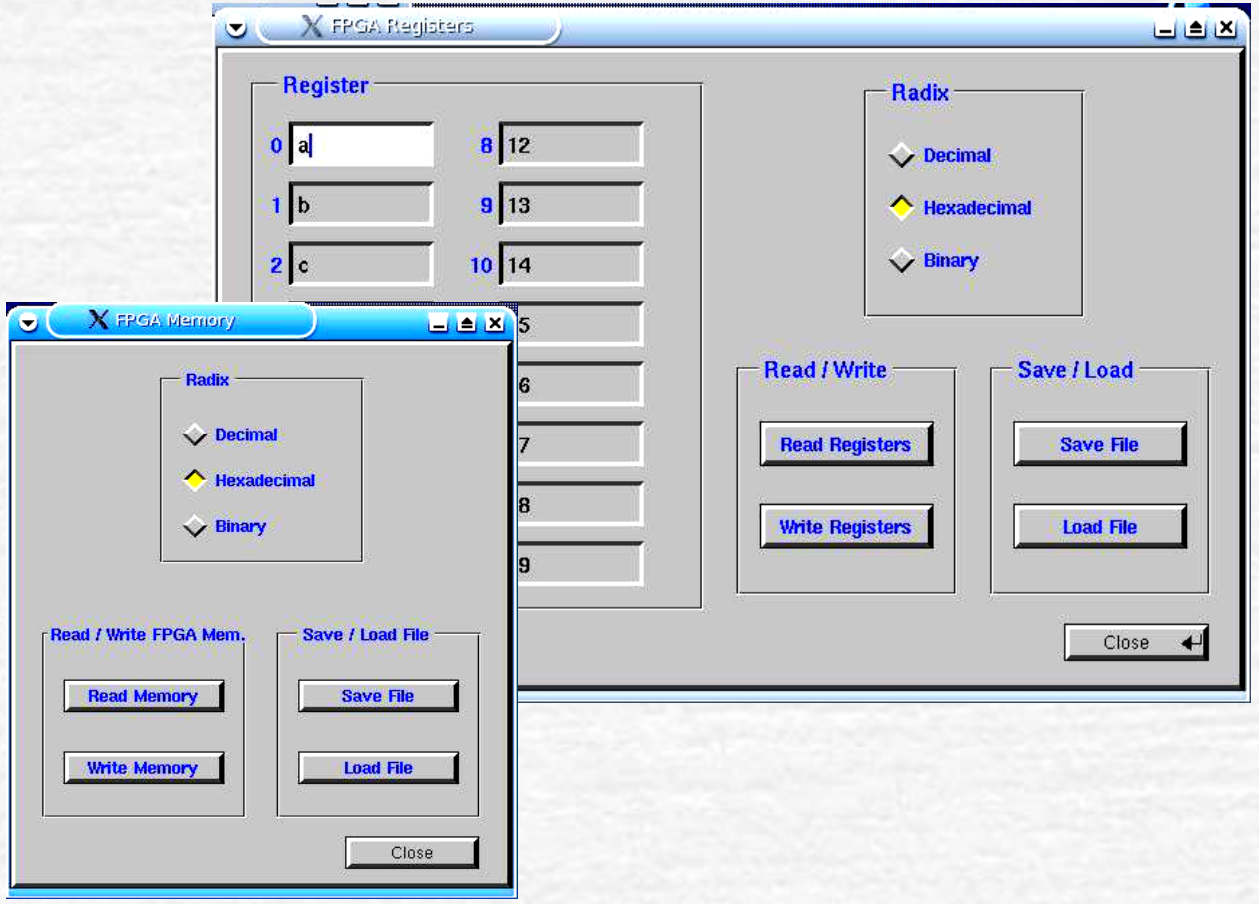


Block diagram of the FPGA architecture for a Reconfigurable Σ–Δ demodulator for Anolog to Digital conversion.

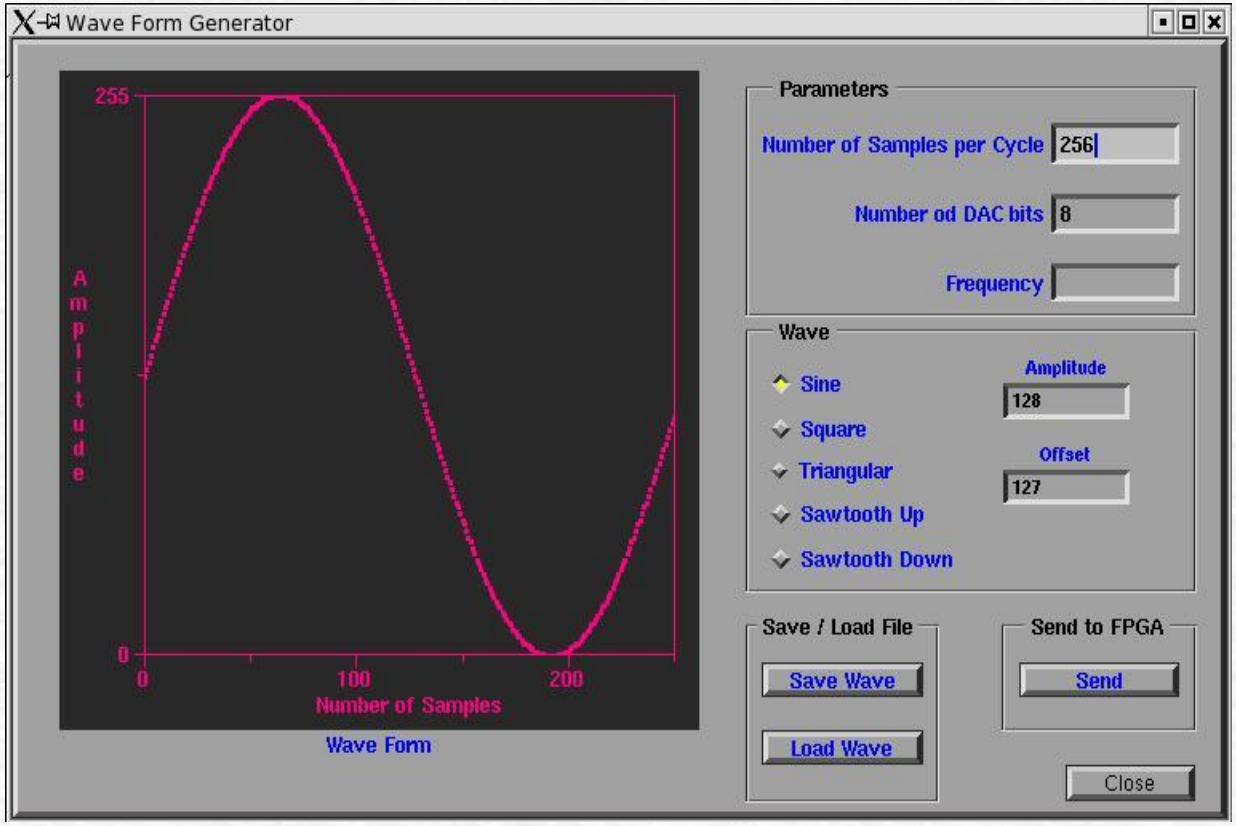


Block diagram of the FPGA architecture for a Reconfigurable Virtual Arbitrary Waveform Generator or Digital oscilloscope.

General Purpose Debugging Interface



GUI for a Wave Form Generator



- DAC (Linear 1668)
  14bit @ 1MSPS

Conclusion (1)

- **FPGA is a key technology for developing countries**
- **FPGA technologies are opening up new opportunities including in the field of scientific instrumentation**

- Excellent hardware cost/performance ratio
- High effort required to develop all the software/hardware chain of new systems
- Wide freely available collection/library of standardized functional blocks (at PC and FPGA levels)

Conclusion (2)

- **A Reconfigurable Virtual Instrumentation system based on FPGA is possible now**

Many areas of applications from basic research to Industry

Emulation of:

- standard general purpose instruments
- sophisticated instrumentation for custom specific applications

Low cost solution for universities and research institutions in developing countries

Conclusion (3)

- A Reconfigurable Instrument could also be seen as a parallel coprocessor of a PC

* Reconfigurable Computing

* Accelerated execution of time consuming or time critical tasks such as

  - **high performance online digital signal processing**
  - **extreme real time hardware control** (easily < 1μs)

Conclusion (4)

- **Open Source & Open Core Approach**

Is Affordable and Accessible; Production, Distribution and Exchange of IP cores can be done through websites

Stimulates Scientific Research and Production of Intellectual property

Encourages South-South and Industry-Academy cooperation

Creates new business opportunities based on free software and double licensing schemes